UNCOVERING NUANCES IN COMPLEX DATA THROUGH FOCUS AND CONTEXT VISUALIZATIONS

JEFFREY M. RZESZOTARSKI
Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
jeff@rzeszotarski.com

CMU-HCII-17-101
May 2017

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

COMMITTEE:
Aniket Kittur, Carnegie Mellon University, Chair
Jodi Forlizzi, Carnegie Mellon University
Jason Hong, Carnegie Mellon University
John Stasko, Georgia Institute of Technology

Keywords: Human-Computer Interaction, Data Visualization, Big Data, Small Data, Medium Data, Exploratory Data Analysis, Sensemaking, Physics-based Visualizations, User Interfaces, Zoomable Interfaces, Unsupervised Machine Learning

ABSTRACT

Across a wide variety of digital devices, users create, consume, and disseminate large quantities of information. While data sometimes look like a spreadsheet or network diagram, more often for everyday users their data look more like an Amazon search page, the line-up for a fantasy football team, or a set of Yelp reviews. However, interpreting these kinds of data remains a difficult task even for experts since they often feature soft or unknown constraints (e.g. "I want some Thai food, but I also want a good bargain") across highly multidimensional data (i.e. rating, reviews, popularity, proximity). Existing technology is largely optimized for users with hard criteria and satisfiable constraints, and consumer systems often use representations better suited for browsing than sensemaking.

In this thesis I explore ways to support soft constraint decision-making and exploratory data analysis by giving users tools that show fine-grained features of the data while at the same time displaying useful contextual information. I describe approaches for representing collaborative content history and working behavior that reveal both individual and group/dataset level features. Using these approaches, I investigate general visualizations that utilize physics to help even inexperienced users find small and large trends in multivariate data. I describe the transition of physics-based visualization from the research space into the commercial space through a startup company, and the insights that emerged both from interviews with experts in a wide variety of industries during commercialization and from a comparative lab study. Taking one core use case from commercialization, consumer search, I develop a prototype, Fractal, which helps users explore and apply constraints to Yelp data at a variety of scales by curating and representing individual-, group-, and dataset-level features. Through a user study and theoretical model I consider how the prototype can best aide users throughout the sensemaking process.

My dissertation further investigates physics-based approaches for represent multivariate data, and explores how the user's exploration process itself can help dynamically to refine the search process and visual representation. I demonstrate that selectively representing points using clusters can extend physics-based visualizations across a variety of data scales, and help users make sense of data at scales that might otherwise overload them. My model provides a framework for stitching together a model of user interest and data features, unsupervised clustering, and visual representations for exploratory data visualization. The implications from commercialization are more broad, giving insight into why research in the visualization space is/isn't adopted by industry, a variety of real-world use cases for multivariate exploratory data analysis, and an index of common data visualization needs in industry.

Acknowledgements

I would like to thank my advisor, co-founder, and collaborator Niki Kittur for his guidance as well as his positive attitude, kindness, and patience in the face of sometimes withering cynicism and procrastination. I would not be at CMU at all if not for Niki making time after a long international flight to chat about sci-fi books rather than research with me during the department's open house.

Commercialization and tech transfer have been at times thrilling, difficult, enlightening, emotionally draining, happy, and frustrating, but I could not have done it at all without the hard work of CMU and IW mentors like Jim Jen, Aaron Tainter, Lenore Blum, David Mawhinney, Kit Needham, and Andrew Moore.

My research has been financially supported by National Science Foundation grants (IIS-1149797, IIS-1217559, IIS-0968484, and IIS-1111124), a Microsoft Research Fellowship, a Carnegie Mellon University Student Innovation Fellowship, and a Siebel Foundation Scholarship.

This dissertation has benefited from the mentorship (and, thankfully, the patience) of my committee members: Jason Hong, Jodi Forlizzi, and John Stasko. The technical portion of this work relied much on the advice of Joseph Chang, Joel Chan, and Nathan Hahn as well as the hard work of a number of staff and research interns including CMU undergraduates Michael James and Westin Lohne who both did as interesting work as I'd expect from any graduate student.

I have been so lucky to have the personal, moral, emotional, and technical support of a great number of mentors inside and outside of my academic life, including (but certainly not limited to) Queenie Kravitz, Justine Cassell, Jeff Bigham, Jim Huggins, Bob Kraut, Michael Bernstein, Polo Chau, Haakon Faste, Laura Dabbish, Jess Hammer, David Musicant, Katie Ryor, Jeff Guerrero, and Merrie Morris.

Though it feels a bit gauche to call out particular friends, this document would not exist at all if not for the consistent support of Beka, Alyson, Tatiana, Dan, Caitlin, Chris, and Mike R. among many others. The judgmental husky stares of Valkyrie Clansman-Liontale and the many past tail wags of Inka Cassell helped too.

Finally, I would most like to thank my parents Jodi and Mark and my brother David, to whom this dissertation is dedicated.

Contents

List of Figures

List of Tables

1 — Introduction

Whether a new product review on Amazon, a social network posting about meeting an old high school fling, minute-by-minute investment data, or a shared document in the cloud, users create, consume, and disseminate an astonishing amount of valuable information on digital devices. Wikipedia recently surpassed 5 million English articles [175], every second 300 hours worth of videos are uploaded to YouTube [56], and nearly 90 million users searched Zillow's store of real estate properties [187]. However, interpreting these rich datasources is often not an easy task. For Amazon products, reviews encode more useful information than numeric scores, but are time consuming to digest. In the case of Wikipedia, a new contributor must navigate a complex web of policies, discussion, and history in order to make a successful contribution [63]. For businesses exploring sales data, an employee may need to inspect many points manually before building a model that captures complex relationships. For real estate shoppers on Zillow, a user may need to manually explore hundreds or thousands of listings to find a house that satisfies their needs.

While there is no shortage of raw data in all of these situations, the process of creating meaning from data is dependent upon identifying useful examples quickly and effectively, whether they come from reviews, rows of data, or content histories. This foraging process can be costly for users [126]. On one hand, users need to explore lots of data to develop a frame that describes the data completely. However, examining many points can be costly because of the need to shift attention from point to point and observe as many entities as possible. On the other hand, users need to also be able to narrow their focus so that they can exploit the data they have found, enriching and refining their understanding based on collected examples. This shift introduces costs in terms of moving from exploration to enrichment (and back again) and in terms of being able to identify important hinge points within the smaller dataset. Unfortunately, digital interfaces often support *either* exploration or enrichment, and are poorly suited to do both. For example, while search interfaces are effective at returning useful queries during exploration, users have a much harder time organizing and managing the results, necessitating systems that help users curate and rediscover the knowledge they develop over the course of searching [41].

In the examples listed earlier, the problem may be even greater than poor affordances for exploring, enriching, and exploiting datapoints. Often in open-ended sensemaking and decision-making tasks [25] users have soft constraints and nebulous goals that must be explicated before data can be properly modeled and exploited. For example, an analyst may want to identify crowd workers that perform well, but they may not yet have a clear definition of what good performance looks like or what particular features in a dataset might best separate them from poor workers. While they could easily explore data and find points, at the start of the sensemaking process they may not be able to accurately identify examples that increase their knowledge [126]. Time spent exploring may help users develop a better understanding of their constraints and goals, but it has a risk of being costly if the means for exploration does not match this need for open-ended exploration and applying/testing soft-constraints.

At present, many existing commercial systems are optimized for hard boundaries and satisfiable constraints. In microtask markets like Amazon Mechanical Turk, gold standard test questions and majority vote answer selection algorithms are commonly used to identify good and bad submissions, but neglect to consider nuance such as how much expertise the workers possess, their working strategies, whether they submitted in good faith, and their similarity to other workers in the pool. In Wikipedia, giving users pure accept/reject feedback can elide details such as the policy or historical background considered when evaluating the edit, other similar edits in the past, and ways to proceed [64]. When analyzing multivariate data, dynamic query sliders and filters impose hard boundaries on data analysis. As data scales up, these methods become insufficient as the task becomes as much about understanding what criteria to employ as it does findings points that meet them.

1.1 SENSEMAKING, FOCUS+CONTEXT, AND PHYSICS

Considering the sensemaking process as a whole, there are several places where reducing cost and improving signal might help users make sense of large scales of data more efficiently and effectively. In the Notional model of sensemaking [126], there are two major processes involved in making sense of data: a foraging loop and sensemaking loop. As envisioned by Pirolli and Card, the foraging loop can be improved by balancing the trade-off of exploring all data and focusing on points, reducing the costs of identifying points as relevant, reducing the attentional load of switching point to point and task to task, and easing the shift from one search into a follow-up search. The sensemaking loop might be aided by improving working memory use (e.g. by augmenting user chunking strategies or offloading some information onto the system), helping users generate alternative hypotheses and contradictory evidence, and reducing confirmation bias by encouraging verification and diagnostics.

In this dissertation I will explore several ways to support the sensemaking of data explorers by developing and evaluating novel data visualization systems. My primary focus is on visualizing multivariate data with either high cardinality or high dimensionality in a manner that supports *data exploration* and handles *both large and small scales of data*. Spence and Tweedie nicely summarize the division between traditional *information retrieval* tasks (i.e. find a point that meets some known requirements) and what they call *information synthesis* tasks which are as much about identifying key breakpoints, important criteria, and data trends as they are about retrieving specific elements of data [153] during data exploration. They point to two existing information retrieval techniques, database querying and dynamic querying, as being ill-suited due to their hard constraints. They note that database queries are: hard to encode, handle errors poorly, often return too many hits, don't guide users towards the next query, hide context, and make it hard to model data. Dynamic queries [150], while much more responsive, still suffer from hiding context with hard constraints and inhibiting "what if" explorations. Spence and Tweedie introduce a new approach in their Attribute Explorer which allows a user to vary min/max bounds stacked across multiple, independent attributes of data visually. For instance, a user might explore how different price bounds affect how close potential houses might be to their workplace. Because it might be hard to attribute a point's inclusion/exclusion across a boolean composition, they also introduce a Link Crystal metaphor that visually represents a Venn diagram of filter matching across attributes.

While the Attribute Explorer system does highlight a potential solution for showing more data context and helping users identify breakpoints and criteria during their exploration, it still employs a hard filtering metaphor. Points either fit within the min/max bounds specified in the interface or they do not. This implicitly hides potentially useful context, such as a slightly too expensive house that otherwise exceptionally matches other criteria. While the focus on boolean filtering does match use cases such as house hunters who have a fixed budget and size needs, scenarios where the user might be willing to satisfice along a large number of variables (such as looking for affordable but nice restaurants that are among the best of their particular genre) or need to explore in detail to identify small trends (such as identifying what a "nice restaurant" is for a particular genre in a particular geographical area) require softer constraints and might demand the user see counter-examples or examples out of their search area in order to gain an accurate understanding of the data in order to fulfill the scenario. This dissertation focuses on this portion of *information synthesis* rather than places where hard constraints, min/max filters, or dynamic queries might be better suited. As outlined later in Chapter 4, there are many real world cases where analysts may miss important details if they apply boolean filters too early in the exploration process. My overall goal across the designs and systems outlined in this document is to explore approaches that expand upon the hard filters outlined by Spence and Tweedie [153] and either afford softer filter boundaries or intentionally reveal points that may lie outside of filter bounds but may be useful for the user to see during their exploration.

In addition to affording tools that better support data exploration, I argue that interactive data exploration systems must carefully balance the trade-off between showing too many datapoints at once (i.e. supporting exploration through coverage and context) with showing too much detail at once (i.e. supporting enrichment and exploration through focus). During exploration, users' needs and tasks change frequently. This makes it crucial for exploratory interfaces to adapt to changing user needs, providing more detail or coverage based on the user's process. Further, interactive interfaces must not introduce extra attentional load as the user switches from filter to filter or task to task. One common approach that the research community has developed in this space is generally called Focus+Context [52].

In Focus+Context visualizations, more pixels are devoted to a focal area while fewer pixels are devoted to other surrounding details (but surrounding details are not completely eliminated) [52]. This provides detail in the area of expected user interest and contextual information in reduced or compressed form. A wide variety of different metaphors for showing focus exist, including fisheye lenses [52], stretched rubber sheets [144], and vanishing point perspective [134]. While very few if any visual compression schemes are perfectly lossless (consider a theoretical Kolmogorov complexity for representing data in discrete pixel elements), the right metaphor can minimize this lossiness by directing more attention to important elements. For instance, in DocumentLens [134] more detail is devoted to the top of contextual documents where summary information might be stored and the perspective view still provides a general picture of the layout of the contextual document's text.

This relates directly to portions of the sensemaking process. In Pirolli and Card's *foraging* loop [126], it is most important for explorers to see a wide variety of examples and be well prepared to compare them. The context, therefore, allows explorers to better situate and compare their focal area. Further, the context helps explorers guide their view towards the next target (imagine dragging a magnifying glass over a map). However, if the context portion of the view does not adequately direct users to other potentially valuable areas in the dataset, it risks increasing the risk that they remain unexplored because the cost of moving the focus onto every point in a focus+context view may be too high. Therefore, there is a risk should the focus be too user-directed. On the other hand, if the focus does not involve user agency readily and solely focuses on data features, then the view may not adequately adapt to users' perceptual shifts as they explore.

The second half of Pirolli and Card's model, the *sensemaking* loop, involves taking all of the foraged evidence and building a structure, or schema, out of the evidence. For example, a data explorer may take foraged evidence from a census dataset that people with lower ages have lower wages compared to higher wages at higher ages and extract a hypothesis that wage tends to increase with age. This higher level abstraction summarizes much of the foraged evidence (but maybe not all) and is readily portable to new, similar data or situations. Focus+Context views are not primarily adapted to this portion of the sensemaking process. While they permit easy identification of useful points, their metaphor for focus and context is rarely aligned with the eventual schema (e.g. a general stretched rubber chart likely will not stretch in a way that naturally matches this eventual inference about wage and age). However, if a focus+context view used a metaphor that was able to align with an expected later schema (or at least match it during a portion of the process), it might help improve this portion of the process.

In a followup paper Furnas considers some of these more broad questions by framing Focus+Context in terms of a Degree of Interest (DOI) function [53]. While in a traditional fisheye view the DOI devotes most pixels non-linearly towards the center area of focus, a DOI function might also reserve some pixels for the periphery of the dataset so that the bounds or range of the data are more obvious. The DOI function instead might not operate purely over 2 visual dimensions, but instead also take into account time by allowing a user to pan and zoom a view while not distorting the detail. Drawing a parallel with distributed activation in neurons, Furnas also constructs an idea of spheres of influence that drag more pixels or detail towards them if their activation level is high enough. This begins to solve the foraging problem mentioned earlier of too much focus preventing an explorer from noticing an important data region. Van Ham and Perer take this even further as they extend the DOI concept into graph visualization [166]. In addition to spreading interest out over nodes in the graph so that the topography of the network is smoothly represented from focus to context, they introduce a User element into the existing a priori and distance portions of Furnas' DOI function. This allows them to incorporate declared user interest into the constant DOI function, for instance weighting towards nodes that match a user's written search query and therefore expanding them into focus. Schaffer et al. [147] take a somewhat orthogonal approach, using a hierarchical clustering pass to provide levels of focus rather than a smooth DOI function as defined by Furnas.

This dissertation builds upon this existing Focus+Context and exploratory data visualization research in a number of ways:

- Introduces a hierarchical clustering pass to the focus+context process (akin to Schaffer [147]) that is designed to both spread information evenly across the screen and provide a scaffold upon which for users to develop initial sensemaking schema

- Develops a DOI function, applied after this initial clustering, that incorporates weighted measures of user interest and a priori data saliency in multivariate data that uses considers both global- and cluster-level features to show more relevant trends between conceptually (or geographically) adjacent points

- Builds a new visual metaphor for representing both cluster- and DOI-mediated point-level multivariate data by outlining clusters and selectively compressing points based on DOI

- Explores techniques for explaining why a DOI function is curating certain content for a user verbally and justifying recommendations and cluster divisions

- Applies physics-based visualization techniques to the focus+context metaphor to disambiguate densely packed focal areas and more fluidly transition between changes in zoom (i.e. cluster level) and user interest (i.e. DOI function).

My initial insight is that hierarchical clustering can prime users during the foraging process with a scaffold upon which they can later construct preliminary schema. Imagine a case where a county government is trying to identify regions in a city most in need of a new food bank. While an analyst could examine the statistics of each address or street in isolation on a map, they probably will instead fall back to commonly-understood abstractions such as blocks, neighborhoods, or boroughs. These ”clusters” of points naturally provide an initial schema upon which the analyst can hang inductions. For instance, they might note that a square of 9 blocks is in a ”food desert” that has few grocers but also suffers from a high poverty rate, even if the display does not directly present those points as belonging to a particular neighborhood or subdivision, and use the metaphor of city blocks as a way to make a generalizable induction. If we can construct similar divisions (whether the same geographic ones when dealing with map data or conceptual boundaries within multivariate features) within a focus+context view to use as a baseline before applying a DOI function, then the user has a head start on the second half of the sensemaking process. By applying the DOI function within these clusters we still compress data so that it is not overwhelming but also maintain an even level of detail spaced within conceptually useful partitions.

My second insight is that physics-based visualization can help to intuitively disambiguate the grouping and un-grouping of points in a multivariate focus+context plot. As points bunch up in a chart, they naturally pack into clusters. If points group, then those clusters shrink down. If points un-group when a user zooms in or the DOI function changes, then the pack expands as the un-grouped points begin to take up space. This matches some of the intuitions behind the tools outlined in the initial exploration into Physics-Based Visualizations in Chapter 3.

1.2 Thesis Statement & Overview

Selectively compressing necessary but potentially overwhelming contextual information by grouping it into fewer visual representations will help to reduce user overload in data exploration tasks. However, compressing features based on users' expressed needs risks allowing users to tunnel into a narrow section of a dataset and miss other, important features. On the other hand, compressing features based solely on a measure of the predicted information gain or utility of data features risks showing the user unhelpful and unresponsive information. Grouping or compressing visual representations itself risks leading users to a mistaken understanding of data by obscuring small- and large-scale structures and relationships in the data. As a result, I consider this central thesis:

> **By dynamically balancing between users' stated (or inferred) interests and data features judged to be independently useful when compressing contextual information visually, users will be better able to make sense of complex and large scale data.** In the initial stages of data exploration, such compressed context will help a user forage for useful examples without tunneling into a narrow portion of the data. Further, applying this compression on smaller, local clusters instead of globally on an entire dataset will help to scaffold users' schematization of the data as they move from foraging into sensemaking and mitigate some of the cognitive and perceptual risks of visually grouping/compressing points.

In this document I initially consider how one can support data exploration through the lens of two constrained use cases: evaluating Mechanical Turk work and making sense of Wikipedia contributions. These use cases reveal the need to compress extraneous information while at the same time balancing between supporting user exploration and making sure to show useful examples or counter-examples. Taking knowledge from those explorations, I develop a general approach for visualizing multivariate data that uses intuitive, physics-based models to reduce the costs of task switching and surface salient data features. This investigation demonstrates several ways that physics-based visualizations improve users' sensemaking and exploration of complex multivariate data, but also shows that physics-based visualizations suffer from a significant drawback in terms of handling large scales of data. Extending this initial exploration through commercialization and deployment, I further refine my understanding of the benefits and costs of physics-based visualizations and begin to model the ways that such tools can have impact on the real world. Commercialization activities suggest that consumer exploratory search is one core potential use case for physics-based visualization, however geographic mapping and handling large scales of data are important elements of many of these search activities.

I evaluate in a lab study how physics-based visualizations more broadly compare to other visualization systems and examine how physics-based visualizations perform when switching between geographic and multivariate plot views. Out of this study I identify two crucial areas for improvement: scalability and surfacing important points. I introduce a new model that uses clustering and modeling of user interest to improve scale and make relevant data features more salient. However, as implied in the thesis statement, clustering and compressing points visually can be risky in terms of misleading users by hiding important points or features in a dataset. To consider this possibility, I compare between several modalities for displaying large scales of data (overlapping, physics-based collision, and the new model for clustering based on data and user interest), showing that my new model for compressing data does not harm users' interpretation of their dataset despite reducing its overwhelming scale. Applying this model, I develop and evaluate an interactive visualization system, Fractal, oriented around a consumer decision-making task, showing that users can maintain a mental model of a dataset across focus/context transitions and that focus/context-like compression can work within physics-based visualization approaches. Finally, I consider possible implications and directions for this line of research.

This thesis makes the following research contributions: findings from real-world testing and user study emerging from a commercialization of physics-based visualizations, results from a comparative empirical study of physics-based visualizations to other commonly used visualization tools showing the inherent trade-offs in using physics-based visualizations, a new model for visualizations that balance between focus and context in multivariate and geographical data which builds upon existing research on Focus+Context and visualization DOI, empirical tests of the benefits and costs of representing data at different levels of detail using this new model, and the development and evaluation of a system based on the aforementioned model designed to support consumer decision-making.

2 — Prior Work

In this chapter I will elaborate on past work in modeling crowdworker behavior and Wikipedia history data. While in this document the summaries will be brief, more details can be found in the original publications cited within the sections.

Crowdworker Behavior

Crowd labor markets such as Amazon Mechanical Turk, TaskRabbit and UpWork enable employers to access large, instant-on pools of workers. However, the large scale of these markets limits their usefulness for complex or subjective tasks in which there is no single right answer. For example, there is no gold standard test question for tagging an image, and voting approaches don't work when workers write poems. Even for normal tasks such as image tagging, gold standard and majority vote analysis methods can ignore nuances such as worker expertise and task completion strategies.

To address this challenge I identify behavioral signals as a powerful new data source for understanding complex and subjective crowdsourcing workflows. My key insight is that the way workers work can often be as informative as workers' end products. Using a technique I call Task Fingerprinting [140], I instrumented web tasks with a Javascript plugin to capture workers' mouse movements, keypresses, scrolls, and timing information. I developed machine learning models that can accurately predict workers' performance on a granular level (i.e. not just binary pass/fail) without looking at their output across a variety of tasks contexts (e.g., image tagging, noun classification, reading comprehension). Importantly, I also showed that the models learned could generalize beyond their original task context, which relaxes a major constraint of having to train per-task models. This approach has applications beyond crowdsourcing to a variety of domains where gauging subjective user performance is important, such as evaluating software interfaces or information retrieval tasks.

Even so, users may not have enough manually labeled traces to apply Task Fingerprinting models, or the task may be poorly suited for a single, quantitative outcome measure of performance. Instead, we can combine behavioral traces with traditional output analysis techniques to bootstrap the process. For example, imagine a worker that has tagged some images. We may not yet have enough labeled data to build a predictive model, but we might notice that the worker has submitted two short tags with very few pauses for deliberation in their behavioral trace. Another worker might also enter the same two tags and similarly rush in their trace. From this I can posit that those two tags are indicators of workers who behave in a slipshod manner. In CrowdScape [138], I created a novel timeline visual metaphor (Figure 2.1) to succinctly portray many workers' activities in aggregate, and paired them with worker end products. This allows a task

Figure 2.1: Two workers completed a reading comprehension test. Rendering their behavior as Crowdscape timelines, we can see two different working strategies. The top worker clicked answers (blue) and rarely scrolled (orange). The bottom worker scrolled back to reference the passage (top of the timeline) before scrolling back down to choose their answers, creating a W pattern. This worker gave more accurate results, despite both workers having similar task completion times.

organizer to identify high- or low-performing outliers, observe working strategies, or spot failure points in the task design. For example, users of CrowdScape were able to identify which workers had employed Google Translate and which translated a passage themselves by looking at commonalities in their submissions as well as a distinct lack of typing in some workers' behavioral traces. The CrowdScape approach has applications in web analytics and personal informatics. CrowdScape also attracted commercial interest from microtask market providers.

Even though CrowdScape shows workers employ a wide variety of strategies to varying success, crowdsourcing workflows often treat workers as interchangeable, consistent computational units. For example, many systems assume that workers will perform consistently during a long workflow. Yet, we know that fatigue and boredom can affect work performance and cause workers with usually good reputations to perform poorly. In collaboration with an interdepartmental team of researchers at Google, I conducted an experiment on the strategic introduction of microbreaks [38] into a crowdsourcing workflow. I found that inserting microbreaks that were aligned with a task (i.e. snappy for quick tasks, curiosity-provoking for detailed ones) improved worker retention. In another experiment on task sequence and delay [101] I found that workflows that deliver tasks which appear similar to previous tasks in the workflow can cause confusion, since workers often remember tasks they completed hours or days past. These studies point towards a tantalizing future goal: if we can develop accurate models of worker's mental state and capabilities, then we might give task organizers useful and nuanced information about their crowd rather than just an ongoing count of success/failures. For instance, if we could recognize that a normally excellent worker is fatigued, then a system could suggest that the organizer consider a paid break. Similarly, if we recognize that a certain worker is employing a very successful strategy for completing a task, then we could inform the organizer so that they can use that in a future iteration of the task design.

Wikipedia History

Crowds of volunteers also communicate and collaborate to build massive scale projects such as open source software, encyclopedias, and discussion forums. However, the very success of these systems generates with it a tremendous amount of historical data that pose a serious barrier to new contributors. For instance, for a newcomer to familiarize themselves with past editors' discussions on the Wikipedia article on Abortion, they must read more than 20 copies of Pride and Prejudice worth of content. As a result, most users simply do not attempt to use historical information to make better contributions. They contribute in the blind. If their work is accepted, they receive little feedback. If their work is removed (reverted) by an experienced editor, they

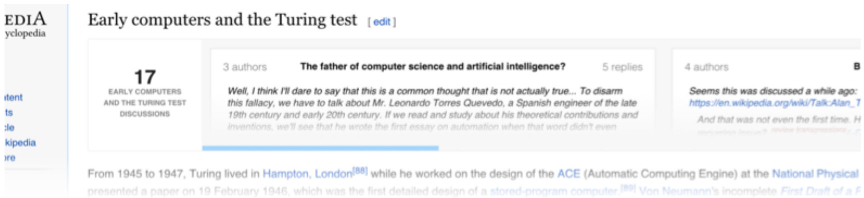


Figure 2.2: Discussion Lens shows related editor discussions for a section of an article on Alan Turing.

also receive little feedback or a discouraging message [64].

My research aims to transform this historical data from a barrier into a beneficial resource for newcomers using machine learning models and visualization. In predicting reverted work [139], I mined Wikipedia's huge store of past failed contributions to help new contributors avoid making similar mistakes. I trained machine learning models on the words editors added and removed to understand the kinds of content editors accepted and rejected. These models were able to accurately explain whether the community would accept a contribution within a wide range of articles. Further, the models keep article context, for instance recognizing that ”murder” may be accepted by the community in an article titled ”regicide” but not in ”abortion”. I am now building on these models to develop systems that offer realtime contextual feedback to editors while they edit. We can give timely feedback informed by the actual practices of the community to new contributors before they are surprised by a negative response.

While identifying controversial terms from previous edits can help newcomers avoid individual words, it does not explain the controversies themselves, which may be needed to help someone effectively contribute. To help newcomers develop such an understanding, I turned to discussion pages that encode a wide variety of contributors' arguments and viewpoints. Discussion pages are not tied to the article content. Instead, users see a list of unstructured topics that are virtually inaccessible, even for the expert Wikipedians that I interviewed in a formative study. My Discussion Lens system under current development uses topic modeling to relate editor discussion topics to individual sections of an article, providing realtime related content in the form of cards of information (Figure 2.2). Much like in CrowdScape, the system surfaces complex behavioral data so that users can recognize patterns as they contribute. Editors using Discussion Lens were better able to summarize and discuss article history in a user test.

What if we were to link Wikipedia's contributions, discussions, editors, policies, and articles into one connected graph? We might direct users to areas of articles ripe for new labor, keep them away from conflict, or give them feedback about how to properly enforce WikiPolicy. There are similar models for helping users make sense of contributions in other user-generated content systems, including the network of commits, comments, requests, and discussion in open source projects and thread participation over topics in discussion forums.

3 — Kinetica: Physics-Based Multivariate Data Visualizations

People are increasingly using mobile devices for everyday computing tasks, augmented by multi-touch interactions which break down the barrier between user and system through interactions that feel natural and match users expectations. I have developed data exploration techniques that use multi-touch and naturalistic interaction metaphors to closely match the sensemaking process users employ to make sense of complex data. By easily pivoting through dimensions, fluidly transitioning between views, and enabling direct touch interactions for data, we can help users more deeply encode data relationships and build better mental models of their information.

In this chapter I will explore related literature on visualization techniques, natural interfaces, sensemaking, and approaches for scaling up visualizations. Using that literature as a base, I will describe my work thus far on physics-based visualizations. This work was previously published in SIGCHI 2014 [141] and has been adapted for this document.

3.1 Exploring and Making Sense of Complex Data

In light of Wong and Bergeron's thorough overview of different visualization techniques [179], I will only highlight several threads of research important to the work described in this dissertation. This section begins by examining different ways of visualizing multivariate data. I finally consider research on how users gain insights from visualizations and make sense of information.

3.1.1 Antecedents

Data processing has been a core demand of computers since their conceptual development. In 1890 the US Census Bureau, worried about projections that the census would take 13 years to manually count, implemented an electro-mechanical punch card system to tabulate and evaluate results [37]. The Bureau later became one of the first 'clients' of the UNIVAC. The Memex was described to collate and process stores of complex documents in a way that more closely matched how Vannevar Bush consumed research materials [24]. As computers developed increasing capabilities to store and render data, researchers have developed new techniques for visualizing data.

Traditional Visualization Techniques

Along with the Xerox Star [152] came a core idea of using direct manipulation [150] to interact with Windows of content containing Iconic representations of objects/features and Menus of options using a Pointing device. This WIMP metaphor has largely dominated data visualization techniques for traditional desktop computers, and is seen across a wide variety of systems. The development of WIMP systems brought about an explosion in approaches for exploring complex datasets.

With the rise of WIMP came a realization that users may not be able to develop structured written queries to demand specific data. Previous query-based terminal interfaces did not help users to browse data fluidly, and could not handle poorly structured data [19]. If a search string returned zero results, it was unclear if the search was poorly formatted or if there were indeed no data matching the criteria. As a result, researchers developed systems that visually presented formerly queriable data to support exploration. Dynamic query techniques provided early gains for browsers. In a dynamic querying system, users could vary search parameters using slider widgets and instantaneously see how their parameters changed the search results [5].

These approaches were immediately employed in interfaces that showed complex, browsable data. Dynamic HomeFinder is one such early system [176], which allows users to perform multi-dimensional filtering operations (number of bedrooms, square footage, etc.) on a real estate dataset while it is visually graphed filtered responses on a map. One crucial component of this visualization were starfield displays, which showed concentrations of datapoints on a 2-dimensional plot. FilmFinder enhanced this basic approach by tightly coupling input and output, allowing users to refine their queries based on the instantaneous feedback they received from a previous query [4]. Further work introduced zooming/panning [84] and point details on demand [6]. Histograms and other displays can further augment dynamic queries for more informed filtering [163].

One additional core primitive for interacting with charted data is the ability to visually select a region of the 2-dimensional space. Becker and Cleveland named this action *brushing*, allowing users to identify points to select, label or delete on a scatterplot [13]. Martin and Ward later expanded brushing to intelligently bridge across dimensions, combine using boolean logic, and perform complex operations [113]. Timebox views apply brushes to match patterns over time in time series data [73]. Brushes have an advantage over dynamic query sliders in that they can innately capture two dimensions at once, and, for visual interfaces, may more closely align to the presentation of the data itself. In an empirical examination comparing querying and brushing, Li and North found that brushes were more effective for trend and relationship tasks while dynamic queries were better for single-dimension filtering tasks [107]. This fits with the general idea that dynamic queries can provide very particular criteria on one dimension, but may not be suited for open sensemaking tasks. Further, brushing and dynamic querying interfaces still do nto match user cognitive models in cases where the constraints are complex. For instance, querying sliders may not allow a user to capture soft constraints when satisficing, and brushes often are hard to combine.

Starfields, scatterplots, and histograms do not much branch into three and higher dimensional representations unless one incorporates color, sizing, or iconic representations. This can add undue cognitive load to users, especially when one considers that users can only differentiate a few gradations of size/color [35]. To truly incorporate high dimensional data, researchers have developed a number of approaches. Stacking scatterplots into an aligned matrix or trellis is one convenient way to display multidimensional data [14, 162, 170]. Indeed, linked displays of different dimensions of data (including aligning quantitative dimensions with graph, text, or

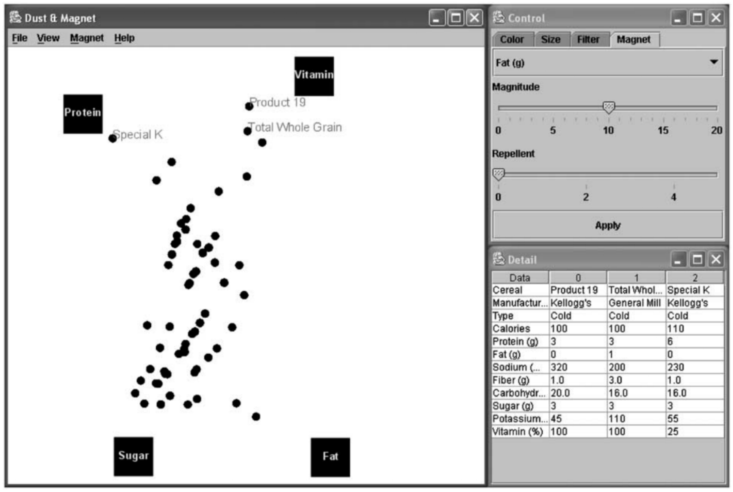


Figure 3.1: Four magnets pull points in Dust & Magnet by Yi et al. [183]. Note the points particularly high in protein and vitamins pulled towards those magnets strongly.

other datatypes) can be effective in helping users make sense of more data [133]. However, these come with a significant risk of increased burden on the user, as increasingly dense visualizations can add to cognitive load [8].

Parallel coordinates views similarly relate multiple dimensions of data with a line chart metaphor. Each point comes a line connecting to different columns corresponding to individual values [76]. To handle brushing and querying, traditional selections might be made over individual dimensions. However, the visualization also affords selecting by line angle, corresponding to a 2-dimensional trend [68]. Because these visualizations might become overly full when there are many data points, researchers have employed soft brushes and gradients to show more detail [120]. To handle categorical data, coordinates can be combined into regions that capture subsets of points [96]. These developments provide a nice example of how a new visualization technique can be expanded with new affordances. A simple, static view of lines gains interaction techniques, scalable metaphors, and adaptations to new datatypes.

One highly relevant inspiration for this document is that of Dust & Magnet, which used a magnet physical metaphor to visualize multivariate data [183]. In order to view trends over multiple dimensions, Dust & Magnet had users place magnetic icons onto a field representing a dimension of data. These magnets would then attract (or repel) points that had specific values. Through arrangements of magnets, users could identify trends shares across dimensions (Figure 3.1). For instance, the pull or one magnet and the pull of another might perfectly cancel, suggesting the two dimensions are correlated. I build on the idea of using physics and interaction to compress the dimensionality of multivariate data.

POST-WIMP VISUALIZATION TECHNIQUES

With the growth of new, multi-touch devices that are powerful and compact, a whole new variety of interaction modalities and device sizes have reached consumers. While for decades one could rely on windows, icons, menus, and pointers as the core metaphor for interacting with software and data, tablets and multi-touch mobile devices have disrupted that stability. Existing information visualization techniques are only now beginning to make use of the multi-touch, direct manipulation capabilities of these new, ubiquitous devices in an effort to explore the design space that these new devices create.

Lee et al. use the terms post-WIMP and post direct manipulation to describe the rich area of research currently filling this gap [102]. One of their core critiques of direct manipulation interfaces is that in order to add functionality, the visualization designer must add more and more manipulatible widgets to the screen. Interactions remain low-level and very granular, such as individual touches or tool actions. In light of rich multi-touch devices, this seems insufficient. Interactions can be complex, indirect, or not rely on traditional WIMP metaphors. Jetter et al. describe this in the ubiquitous computing context in terms of a blend between real world concept and an interface [82]. This blending of concepts into Natural User Interfaces (NUIs) necessitates more complex interactions than a traditional WIMP metaphor might afford.

Such techniques that move towards gestures and naturalistic interactions may have benefits for information literacy and awareness. For exploratory data visualization, post-WIMP approaches have the potential to preserve users growing mental model of an information space as they explore it, and help them interact with more variables more fluidly than traditional interfaces. This might help them understand the structure and distributions of data even without significant training or statistical expertise by leveraging their models of the physical world [81].

Gestures, multi-touch, and physical interactions are key components of NUIs regardless of the particular application. Keefe and Isenberg highlighted some of the major interest areas for natural interaction in data visualization [88]. These include identifying more natural equivalents to traditional visualization interaction techniques, constructing useful toolkits, and exploring new affordances. In understanding how we can construct new relationships between data and visualization, Isenberg and Isenberg break into the relationship between data representation and interaction technique [78]. What are the interaction primitives one uses, and how do we bind them to data visualization operations? Even further, how do we evaluate these primitives? Drucker et al. provide one such example of evaluating interface actions [40]. They use a grounded methodology to study user preference and effectiveness in different visualization prototypes in comparison, findings NUIs to be effective but at times constraining when people already knew what to do.

RoomPlanner uses physical actions like cupping a hand and transparency to make data easy to explore on a DiamondTouch surface [181], and Wilson et al. explore a wide variety of game physics-based interactions [177]. These interactions have the benefit of both ease of use and the ability for users to improvise new ones based on their experiences in the real world. Wobbrock et al. take this idea further, allowing users to build their own gestures for a touch interface with minimal expertise [178]. These unstructured approaches make the user into a stakeholder for the interface. To a visualization designer, these approaches provide an interesting opportunity to allow users to design and operate personalized tools that fit their data and use case with minimal training.

Moving towards more structured interaction techniques, Sadana and Stasko developed a gesture set and interactive exploratory prototype for representing scatterplot data fluidly on tablet devices [142]. NEAT and interactive grids [49, 50] provide a set of pen and touch primitives to align and locate data on a field. This can be critical given the imprecision that can accompany touch interaction. TouchWave displays stacked hierarchical graphs much in the way a WIMP application might, but uses drags and swipes to scale and separate overlaps naturalistically [12]. This helps to reduce the normal overload associated with stacked graphs, as users can efficiently and quickly resolve ambiguities. Schmidt et al. treat graph edges and nodes like beads on strings that can be gathered and stacked [148]. Much like TouchWave, this also helps to reduce load when graphs are densely packed. SketchInsight and SketchStory combine interactive sketching with dynamically constructed traditional charts for didactic presentations [103, 168]. The interaction techniques necessary to visualize data themselves become teaching and illustration tools. Some approaches even employ physical objects. Senseboard [80] places pucks on a grid to organize and manipulate information, while Ullmer et al. [164] use physical tokens to control and visualize database queries. On the other hand, other approaches use physical objects as probes to identify fruitful gestures or interaction patterns [66].

One avenue for intuitive, interesting interactions cited by Jacob et al. is the introduction of physical metaphors [81]. Bumptop shows one such example in a general computing context [3]. Files become physics objects that can be pinned to walls and moved around in an environment. This has the benefit of requiring minimal training, since people already intuit that objects fall and pins stick things to walls. North et al. extend this sort of physics approach into data sorting in comparison with a traditional WIMP system [119]. People readily adapted to physical models, outperforming those using a mouse. Force-based graph layouts use attraction and repulsion, which people naturally intuit, to make graph exploration easier [51]. Even text can be subject to physics-based transformations, making it more evocative and matching descriptive features [151]. Sticky tools make touch manipulations of digital objects behave more like a finger's slight stickiness and therefore become easier to handle [65]. Visual sedimentation metaphors use the buildup of material to analogize the aggregation of time-series data in an easily comprehensible way [74]. Even physical objects can become links into data visualizations, such as the exploration probe objects in ArtVis [42] and movable physical rods of EMERGE's bar chart [158].

Klemmer, Hartmann, and Takayama suggest a more general explanation for the benefits of tying natural metaphors to digital artifacts [94]. The tangibility of an artifact, or how naturally it maps actions to physical reality can greatly influence the performance of a system. While closer mappings to reality may come with additional constraints and limitations, they also provide benefits for training and understanding, especially for inexpert users. This might extend beyond just NUIs. The description of post-WIMP possibilities that Jacobs et al. provide and the design considerations of Lee et al. [102] still might work within traditional mouse-and-keyboard computers. While they may not be quite as tangible as if one were touching the screen, the interactions and feedback may nonetheless remain physically grounded and beneficial.

3.1.2 Insights & Evaluation

No matter the interaction modality or particular form of visualization, the ultimate goal of information visualizations is to direct users to some new insight they did not yet possess. This was one of the core motivations of Card et al. as they collected articles to build a grand vision for visualization [26], and continues to be the outcome measure inherent in designing new visualization techniques or interactions (though in general outcome measures for information visualization systems vary substantially, as a survey conducted by Lam et al. demonstrates [99]). Yet, the act of gaining insight can be ill-defined and hard to encapsulate experimentally.

A number of researchers have investigated this question in systematic ways. North offers a general guide to evaluating visualizations for insights, suggesting that benchmarking tests (such as time-limited task performance) are insufficient [118]. Instead, he recommends focusing on insights using qualitative examination of longer sessions of participants using a tool, with an emphasis on tasks that are environmentally valid and relevant to a domain (as illustrated in a qualitative inquiry by Faisal et al. [47]). Carpendale also compares methodologies, suggesting both quantitative performance and qualitative insight measures have use [28]. To obtain more ecological validity, Plaisant et al. suggest that contests and public-facing challenges can help insight evaluation scale [128]. Because evaluations of visualization tools for insights are quite specific to the individual system, there is no simple proscriptive resource for constructing an experimental methodology.

One central aspect of the data exploration process is sensemaking, or the process of constructing meaning from experiences and knowledge [137, 172]. For exploring unfamiliar datasets, the sensemaking process supposes an interative process between querying for data, examining the results, and then refining the exploration parameters. Qu and Furnas observationally examined the data exploration process, finding that exploration is much more about building a model of the data as opposed to identifying and remembering specific instances [131]. A meta-analysis divided this data sensemaking process into four steps: Provide Overview, Adjust, Detect Pattern, and Match Mental Model [182]. This captures the core sensemaking concept that users first examine all of the data, adjust their view through filtering/exploration, find patterns, and then align their findings with their mental representation of the data or scenario. Another investigation by Perer and Shneiderman used a tool that displayed political relationship data, finding that users employed visual search and exploration to do initial passes on the data, but as they became more expert began to understand how and when to employ statistical analyses techniques [124].

Pirolli and Card describe the sensemaking process in terms of a sort of foraging, where users initially cast wide to find early fruitful exemplars, and gradually narrow their focus as they learn more about the environment [126]. This perspective provides a great opportunity for visualization environments that deal with complex or massive-scale data. Rather than force users to examine everything, the tool can help users learn from small steps to start, and then grow in complexity and scale (informed by the users' interactions and newly developed knowledge). In one such example using text data, Endert et al. found that using traces of users' initial explorations of a text corpus helped to better dynamically weight models as users continued exploring, adapting to each user's individual impression and use of the tool [44]. The Apolo system similarly uses small organizing interactions as users begin to make sense of a graph of citations in order to better construct a high level organization of the information [29].

However, there are serious limits to the sensemaking process. A human brain can only store a set number of items in working memory, and this can limit the number of hypotheses an analyst can consider [26]. As the size of datasets scales up, this can pose an increasingly hard challenge. If there are too many points or too many interesting outliers, an analyst might become overwhelmed. Even further, limited attentional resources can be consumed trying to manage a visually complex display of massive amounts of information, resulting in overload [174]. Even motivation can pose an issue, as studies of organizational sensemaking suggest that an expectation of efficacy and capacity for performing the task can also affect the insights made [112]. These limitations point to a necessary tradeoff: As datasets grow larger and larger, it becomes critically important for visualization technology to manage overload and help analysts scale their sensemaking process.

3.2 Naturalistic Visualizations

The research reviewed in the previous chapter points to several opportunities for improving visualization technology in mobile multi-touch devices. Reviewed literature on sensemaking suggests that users iteratively develop an understanding of a dataset by taking a series of small insight-generating steps as they explore. They develop hypotheses, test them through experimentation and observation, and finally redirect to continue exploring. One crucial challenge is helping users to develop a consistent model as they explore the data. Operations that dramatically change the visual appearance of the data (for instance if a filter action were to remove points and instantly relocate others as axis scales adjust) risk disrupting users' gestalt perception of the scene, affecting recall and requiring more attention [69]. Further, users take many small steps when analyzing data, but interfaces rarely reflect the traces of their past actions. Users may have difficulty remembering what particular steps lead to their current viewport on the data. As data scales up, another challenge is to help users model information that may exceed their working memory limits. For instance, a user might attempt to track a relationship that spans three or four dimensions, but due to interface restrictions could have to compare between three or four different charts to find it. The load of such a task might be difficult to manage even for a trained expert.

Given that sensemaking performance is influenced by motivation and feelings of efficacy [112], it is important that interfaces encourage exploration and digging deep into data. Complex interfaces may require long periods of training and raise intimidating walls for newcomers [157]. Ongoing work into NUIs provides an interesting hook for not only delivering easy and motivating interfaces, but also solving the problems of enhancing recall and modeling mentioned in the previous paragraph. Users are already well trained to interact with objects in the real world. By designing exploratory visualization tools that use physics and naturalism to match users' real life experiences, fluidly move between analysis steps (as real objects do), and utilize touch modalities that meld a user closely with the interface actions [185], we can improve exploratory performance.

In this section I will describe a general theoretical framework towards exploratory data visualizations that use naturalism and physics to improve user efficacy and satisfaction. I will then describe the design process of and eventual final result for a prototype data exploration tool that evaluates several concepts from the framework.

Forces - Act on points to move, repel, attract, or accelerate
Layouts - Place points into meaningful locations
Mutations - Change points' internal data, combine points
Barriers - Prevent points from occupying a region of space
Filters - Selectively include or exclude based on criteria
Queries - Change the display appearance/behavior of points

	Forces	Layouts	Mutations	Barriers	Filters	Queries & Overlays
Forces	***Magnets*** Points are pulled to a finger					
Layouts	***Force plots*** Points are pulled to their place in a chart	***Fixed charts*** Points snap to their place on a chart				
Mutations	***Group by vector*** Group points that are moving similarly	***Group by layout*** Group points by their place in a chart	***Groups*** Put points into abstract groups			
Barriers	***Jelly walls*** Points need an extra push to pass a wall	***Layout walls*** As points pass, put them into a layout	***Collision to data*** Points that hit a wall are assigned values	***Walls*** Block points from a region of space		
Filters	***Query magnet*** Attract points that meet specific criteria	***Detail view*** See point details in a layout popup	***Group by filter*** Group points based on specific criteria	***Sieves*** Walls only permit certain points	***Show/hide*** Display points by criteria	
Queries & Overlays	***Query fields*** Visualize the pull of different forces	***Area details*** See details for points that meet criteria	***Group zoom*** Zoom in and manipulate points	***Lens*** Highlight certain points in a region	***Selective overlay*** Highlight/select points by criteria	***Overlays*** Color/scale points by data dimensions

Figure 3.2: My physics-based generative framework that combines different affordances to generate new physics-based visualization interactions.

3.2.1 Framework

I define a naturalistic or physics-based visualization affordance as an interaction or visual affordance that resembles a real world physical phenomenon. For instance, Dust & Magnet [183] uses a naturalistic magnet affordance for organizing data in its data analysis process. Much like real grains of iron would be attracted to a bar magnet, data are attracted with varying force to Dust & Magnet's iconic representations. It is important to note the distinction between *naturalism* and *realism*. The Dust & Magnet magnets do not resemble magnets pictorially, and their magnetic effect does not decay with a lifelike inverse square. The visualization affordance resembles the real world when it proves an effective metaphor, but does not necessarily mimic the real world. While this is largely up to the interaction/visualization designer, my general design philosophy towards naturalistic visualizations has been to use pure mimicry strategically and sparingly.

Physics-based visualization affordances make use of the inherent expertise users have based on their experiences in the everyday world in order to help them develop an understanding of data. These techniques are different from traditional visualization approaches, and, in light of users desire at times for more familiar controls, may work well in concert. However, because they are different from traditional approaches, it is not always easy or intuitive to create new interactions. In considering and developing naturalistic affordances, I have developed a theoretical framework outlining several different modes of interaction with data entities. It is composed of the following primitives:

- Data are represented as physical points that have associated physical properties that correspond to their values in different dimensions
- Data occupy a sandbox that contains them and allows for interaction. Interactions with the sandbox change the physical arrangement of the data, and leave traces of their activities.
- The user can employ forces to act on physical points either independently or as a result of their unique data. For instance, a magnet may repulse points with low values in a particular dimension.
- The user may use layout tools to force points into strict, meaningful locations, breaking with the physics metaphor when necessary (such as when allowing points to pass over or under others to avoid being trapped).
- The user can mutate points, for instance combining multiple points into one group so as to observe more points at once or see larger trends.
- The user may place barriers that block or selectively block points based on criteria.
- The user can employ filters to selectively include or exclude points to help avoid overload or choose only a small subset of interest.
- The user may use queries and overlays to change the appearance or behavior of points on the screen.

I imagine these affordances could work in a variety of contexts and situations. They could be used in two dimensions with which may be easier to interact or in three where there is a richer space of interactions but with a potential tradeoff in clarity. Likewise, they can function using a keyboard, mouse or multi-touch, though multi-touch offers the greatest opportunity for naturalism and closeness to the user [185]

Furthermore, these concepts can be combined to generate a much richer set of potential physics-based affordances. Mixing different primitives together provokes new ways to augment data visualizations with physics. Figure 3.2 provides some examples of different tools that can come out of a combinatorial brainstorming on these primitives. For instance, combining a barrier that blocks points with a selective filter could generate a permeable barrier that allows some points through but shows the data left over after filtering. Combining forces with a scatterplot layout could create a force-based plot that pulls points to their proper location, and might work in concert with other barriers and forces.

3.3 Design Process

The first iteration of Kinetica, then called TouchViz, was extremely rudimentary (Figure 3.3). Constructed as a CMU course project and inspired by Dust & Magnet [183], it introduced several of the ideas that would later become core tools. Initially I encoded data into circles colliding in a sandbox, implemented a touch-responsive magnet tool much like Dust & Magnet [183], and added gravity based on device orientation. This prototype by itself was evocative. By tilting the device so gravity took hold and pulling points with a magnet such that the forces balanced each other, data readily sorted itself and separated, highlighting outliers. By adding a bar that filtered

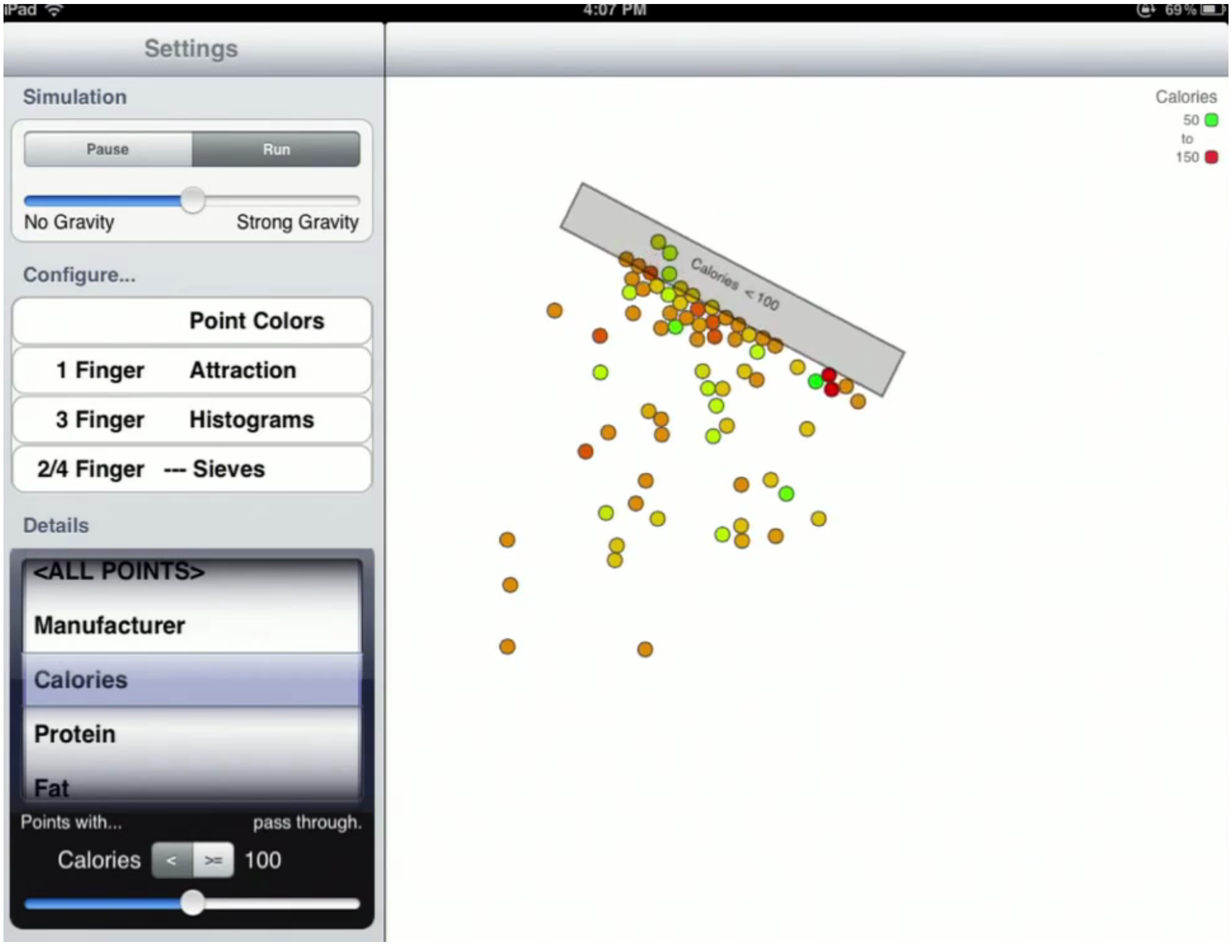


Figure 3.3: Original TouchViz proof of concept as demonstrated in an online video [27].

points that passed through it as well as force-based plots, the tool began to encode common data visualization primitives like charting, filtering, and color-coding points. However, this prototype was extremely limited. Gestures were heavily overloaded, requiring up to four fingers. Users could not easily create more than one tool or chart on the screen without significant overlap and gesture target confusion. Further, the sandbox itself was restricted by the need for a large instructions/configuration bar along one side.

The process that followed would eventually be refined into a general pattern that Aniket Kittur and I have termed ”data-driven prototyping”. There is an interesting body of work examining the different ways that practitioners and researchers can both explore the space of possible instantiations of an idea and examine the benefits and tradeoffs in a particular iteration of an idea. Often this process is referred to as prototyping. One axis on which these approaches vary is in terms of its fidelity or closeness to a developed product. Low fidelity artifacts such as sketches [57] force the researcher to focus on ideas rather than implementation details, helping to prove out early concepts and explore a design space. Higher fidelity artifacts such as working, coded prototype tools are easier to subject to evaluations such as user testing, but run the risk of fixating on a particular part of the design space and missing the bigger picture. Paper prototyping [149] occupies an interesting middle ground, reducing the cost of implementation but still focusing narrowly on one instantiation of an idea. Another axis on which prototyping varies is in terms of how close the artifact is to evaluation. The participatory design process, where researchers work with individuals or organizations (often who are stakeholders or domain experts in the problem area) to construct narratives, brainstorm, mock up, game, or model a problem [89], does not lend itself to quantitative evaluation in the way that a fully formed working prototype would but provides a more room to express different ideas.

Lim, Stolterman, and Tenenberg argue that evaluation of prototypes is a risky business that can often cut off interesting design possibilities by quantifying or specifying features too early in the design process [108]. Indeed, a more holistic way to consider the prototyping and design process is to focus on the work done before any particular product or artifact is produced. Design research focuses on the generation of knowledge necessary to inform the production of a complex artifact [188]. The research through design approach seeks to leverage design thinking in order to integrate knowledge from a variety of sources to produce the ”right” thing to solve an underconstrained problem (i.e. no clear specification to match). This thinking has a strong relationship to the visualization design process. Novel visualization techniques and interactions rarely have a defined product specification or a clear and obvious pathway to solving a complex issue like representing data relationships over time or projecting many dimensions of data onto a 2-d screen. While they may not be so-called ”wicked” problems [20], their open-ended nature makes prototyping and development a complicated task. Oftentimes identifying the ”right” way to represent data can be as challenging as figuring out the details once a representation is identified.

In ”data-driven prototyping”, we integrate datasets and anticipated analysis results into the early design process as a way to ground our design thinking. Any sketch, prototype, or brainstorming session ought to be matched with real datasets that reflect the issues we hope to solve and our ideal realized outcome (whether it be a trend exposed through the visualization, particular aspects of the data emphasized, or even a feeling such as an ”aha” moment on part of the user). While not explicitly evaluative in that the ideal outcome is often shaped by the design process rather than fixed as a pass/fail outcome measure, this approach provides a filtering mechanism that helps to identify the ”right” ideas earlier in the process. In effect, the data guides the design process, augmenting the designer’s evidence-driven intuition of what end product ought to be. As new ideas are developed, so should new data sources be integrated. While one sketch might work for several kinds of data, a new dataset may break assumptions the prototyper did not realize they had made. While in practice much of our process involved low- to high-fidelity technical prototypes guided by this data-driven thinking rather than sketching or participatory design processes, this is not the only way to incorporate datasets into the prototyping process.

Moving from the initial TouchViz prototype, I incorporated several datasets into the design process, including data on the nutritional values of cereals, a small and large dataset of Titanic shipwreck survivors/victims, a small listing of car models and specifications, and city census data. One clear objective coming from playing with the dataset of cereal brands is that one often tests different data attributes rapidly, investigating different trends and identifying breaking factors. Having already shown the working prototype to users, I already knew that 1-4 finger gestures

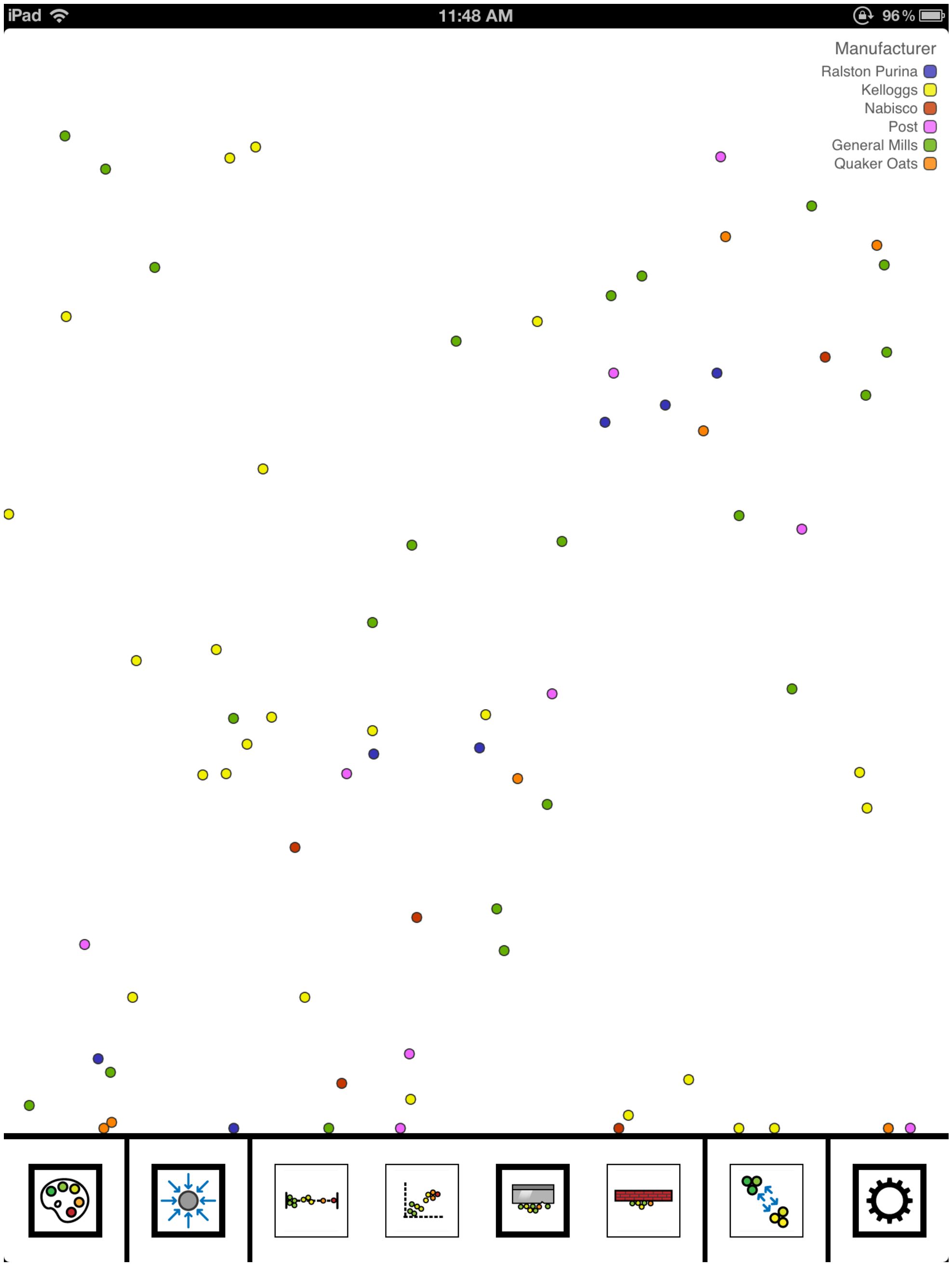


Figure 3.4: Palette-based iteration of TouchViz created shortly after the original prototype showing a dataset of breakfast cereals. Data were evenly scaled and began spread randomly across the screen. Tilting the device introduced a gravitational force, as if the data were marbles on a tilting table top. Users could tap the palette at the bottom to create different tools with two-finger gestures. For instance, they could press the scatterplot button and then draw the corner bounds of a plot, magnetically pulling points into the chart. Gravity, while interesting in terms of making filtering a very *physical* kind of activity, ultimately confused users if they were using charts (which was often the case). The palette interaction became clunky when users always wanted to use certain tools, such as starting the exploration with a scatterplot or being able to filter quickly with a barrier without multiple taps.

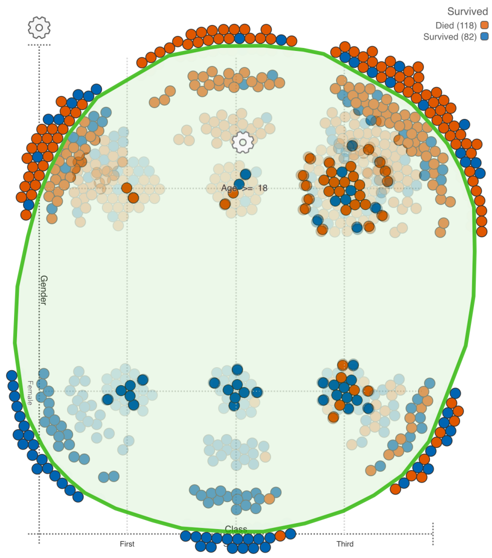


Figure 3.5: As an area filter is laid down, points that are excluded from the area are ejected into emergent clusters corresponding to their original ones as a result of the physics-based forces.

were overloaded and cumbersome for this rapid exploration process. I first worked to reduce the gesture set down to 1-2 fingers and implemented a palette of interface actions (Figure 3.4). The Titanic dataset exposed a particular flaw, in that when large amounts of data were represented in a physics-based simulation, points that were filtered using the sifting tool would bunch up and not filter properly (much like corn starch can clog a kitchen sieve). This was one of the first signals that physics-based models ought not to simulate the real world but behave naturalistically, imitating the real world but also breaking with realistic conventions when necessary. As a result, I developed a system inspired by layers in photo manipulation tools that allowed some points of data to pass over or under other points, avoiding clogs.

The Titanic dataset also revealed another insight about the two-finger sifting tool: cutting across the screen so that filtered points were on one side and unfiltered on the other was a poor way to filter large or complex datasets. The crowd of points did not take advantage of the physics-based charts and instead consumed valuable space. As a result, I began to develop tools that operated on polygonal areas rather than lines of force. The razor-blade like sifter became drawing tool that allowed the user to specify an area that would allow or deny certain points. This produced an unexpected emergent property: the force-based charts would orient the excluded points in meaningful clusters around the filtered area (Figure 3.5).

Another key observation focused on the ways we colored and labeled points. In the initial TouchViz prototypes very little information was given about data range, category, or value on the main sandbox interface. Users had to double tap points to see any of the encoded features. As we moved to more complex datasets, we were forced to integrate more detailed feedback into the scatterplot axes about what was being plotted and develop more effective color schemes (especially with regards to colorblindness) that made trends more obvious. Notice the clear distinction between categories and the categorical labels within the plot in Figure 3.5.

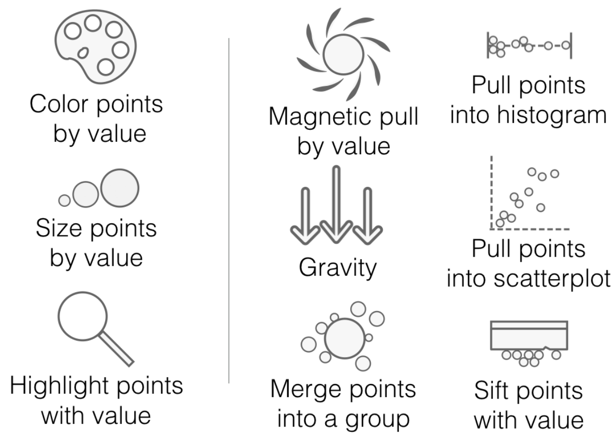


Figure 3.6: Final set of affordances for the Kinetica prototype.

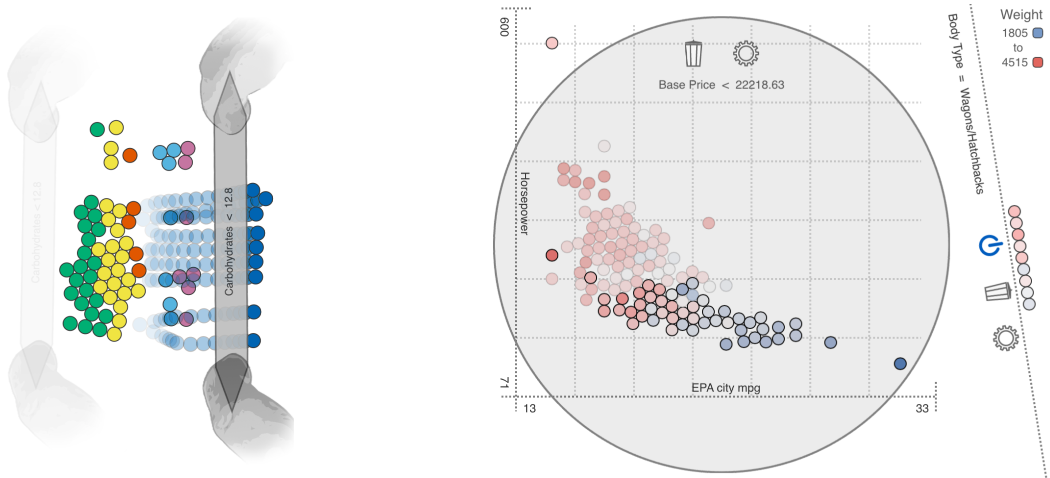
As I began to test the prototype with new users, we developed two primary types of tools: manipulative tools and interrogative tools. Manipulative tools alter points locations, or move them around the sandbox. Interrogative tools change the appearance of a point or its interactivity. A mix of these tools can be layered to explore multiple dimensions at once, and the tools leave traces on the sandbox field so users can see what is affecting points. Because it was often the case that users had points exactly where they wanted them to be on the sandbox but still wanted to filter, I integrated more interrogative tools such as lenses that highlight points based on criteria. The final feature set is illustrated in Table 3.6 I eventually hit a point where I had developed enough fidelity in the prototype that quantitative evaluations would provide more feedback than informal user studies and tests with new datasets. As a result, I optimized the performance of the tool and constructed a final version.

3.4 KINETICA

I implemented several physics-based affordances in a prototype system, Kinetica[1], to formally explore the benefits of physics-based affordances and multi-touch interactions for data visualization and to understand the technical challenges of implementing such techniques. There are more details in my publication on the prototype [141], but I will summarize the major development process here. For my proof of concept, I used a tablet computer because it occupied the physical space of the user, could be twisted and turned, and was responsive to touch. I developed for iPad in this prototype due to the convenience and general availability of the device at the time. Such a system could now be implemented easily on any number of Android- or Windows-based tablets as well as full size laptop or desktop computers.

I implemented Kinetica using Objective C, and make use of an open source physics library called

[1]The system moved from ”TouchViz” to ”Kinetica” as a result of a strange coincidence. At the 2013 SIGCHI conference where I first presented physics-based visualization work under the TouchViz name, Steven Drucker published a quantitative analysis of touch interactions for visualization [40] also called ”TouchViz”. After joking about it a bit with Steven, I opted to rename the project to avoid any confusion. Prior to TouchViz, the project was named ”PhysViz” for approximately two days before common sense won out.



(a) A user drags a two finger semi-permeable filter across some data. Points that meet its criteria collide and are pulled with it to the right, passing over unfiltered points.

(b) A user explores car model data using Kinetica. She cares about mileage and power, so she placed the points into a scatterplot. She doesnt like wagons, so she used a wall to filter. Finally, she added a lens to highlight cheaper vehicles.

Figure 3.7: Kinetica tool examples.

Chipmunk Physics [106] to handle forces and collisions. This library makes use of multiple CPU cores to improve efficiency. There are two major areas in the rendering pipeline where the implementation can bottleneck. The first occurs if there are many points overlapping that must be resolved by the collision solver. This process can take well over $O(n^2)$ steps per frame until the overlap is resolved. The second occurs as a result of bottlenecks in the iOS rendering pipeline. Without writing an independent OpenGL shader, the existing rendering classes cannot handle thousands of objects projecting onscreen.

The major visualization techniques and interactions in the final Kinetica system include:

- Sandbox — Data will be placed onto a plain, white surface that has a slight amount of friction and damping.
- Point Entities — Each row of data becomes a tangible ”point” that responds to touches and collides with other datapoints.
- Color & Size — Adjust point colors and size based on categorical or nominalized numeric values.
- Histograms — Pull points to a one-dimensional histogram. Points will bunch up in areas of high density.
- Scatterplots — Pull points to a two-dimensional scatterplot. Points will bunch up in areas of high density.
- Walls — Sieve points based on filter criteria, allowing certain points to pass and blocking others. Drawn walls exclude certain points from a region of space.
- Lenses — Highlight/lowlight points based on filter criteria. Points are not affected.

- Groups — Subsume multiple points into a single group point. As this is a subclass of datapoints, they behave as if they were a single datapoint that had the average value of their contained points.

For each of the tools developed, I initially assigned unique gestures and numbers of fingers. For example, to create a barrier, the user would put down four fingers along the contour, while creating a line histogram required only two. This proved cumbersome and confusing. Instead, I adopted gestures that used either one or two fingers. Two fingers define two control points allowing for a histogram between fingers, a bounding box for scatterplots, a barrier between fingers, a lens spanning fingers, or a group that floats between fingers (to avoid occlusion) that selectively consumes points. On the other hand, one finger gives us one control point, which allows for drawing actions. This permits freehand drawn histograms, drawn areas that permit/deny points, drawn lenses, and selecting specific points to form a group.

In testing, I observed several interesting combinations of tools. A scatterplot that bunched points into categorical clusters could be enhanced with a histogram that pulled points into sorted order. Each cluster still felt a pull towards its group, but also self-assembled into sorted order within the cluster thanks to the influence of the histogram. Similarly, drawing an area or barrier that rejected a subset of points while a scatterplot was present meant that even the filtered points still felt a pull to their proper locations, albeit stopped by the barrier. This meant that they still took the form of clusters against the barrier. In practice through techniques like this users could effectively layer 5 dimensions of data within the iPad screen sandbox.

3.5 Evaluation

To evaluate Kinetica, I invited participants to use the software for 45 minutes in a lab study. As a comparison case, I also invited another group of participants to follow the same study protocol using Excel rather than the application. I considered incorporating exploratory visualization suites such as Tableau, Many Eyes, or infogr.am, however I was concerned that training would consume too much time in comparison to Kinetica, especially considering novice computer users. Participants first received 5 minutes of training, then continued training by working through a small dataset answering 5 basic statistical questions of increasing difficulty. I then asked participants to identify an exemplar among a dataset of car models, and explore a dataset of Titanic survivors to identify trends. I recruited 31 participants from a campus participant pool. For more methodological details, please refer to the paper [141].

In the Titanic open exploration task, Excel participants made on average 5.1 findings, and Kinetica participants made on average 5.5 findings (the difference is not significant). The Kinetica participants findings generally encoded more dimensions per finding than Excel participants (M:1.74 vs. M:1.38; repeated measures $F(1,164)=17.67$, $p < 0.001$). Two coders classified each of the findings made by participants into five types: point findings that discussed a particular row of data (youngest passenger), statistical findings that gave summary statistics (mean age), descriptions that capture general trends (less than half of the people lived), comparisons between categories or groups (more third class passengers died), and relationships between dimensions (the older you were, the more likely you were in a high class cabin). Coders were blind to condition and had high inter-rater reliability (N=186, kappa=0.96).

Figure 3.8: Number of participant findings by coding category and tool. Notice that Excel participants (salmon) were better able to make statistical and point claims thanks to easy access to raw data, but that Kinetica participants (teal) were better able to identify relationships and compare between dimensions thanks to affordances for rapidly exploring multiple dimensions.

Interestingly, as illustrated in Figure 3.8 Kinetica users made far more descriptive, comparative, and relationship findings than their Excel peers whom almost always made point or statistical findings (c2(5) = 31.3, $p < 0.001$). Thus while they do not make as many quantitative observations, they demonstrate a greater awareness of distribution and multidimensional trends. This suggests that physics-based affordances may be helping users build a more holistic understanding of data.

Participants also had some issues using Kinetica. One participant complained that it was hard to get exact numbers. Kinetica did not have an undo action implemented, and some participants complained that they wanted to go back, but had already moved data around since then. This raises an interesting question of what undo operations mean in this space. Kinetica also used on-screen controls rather than gestures. This meant that they sometimes got in the way, annoying users. Of all the tools, participants voiced the magnet as least useful, primarily because its one-dimensional force was subsumed by histogram charts whose behavior was clearer.

3.6 Discussion & Open Questions

Considering the findings from the Kinetica evaluation more broadly, physics-based visualizations seem to help users in a number of ways. Participants in the study:

> ...were better able to describe distribution
>
> ...were better able to identify outliers
>
> ...found more relationships and comparisons between dimensions

The results suggest that these improvements might be due to the improved fluidity of transitions in the visualization, the reduced cost of testing different dimensions of data and exploring hypotheses, and the relative lack of training necessary to use Kinetica's different analysis tools. However, the results also highlight a few significant barriers. Kinetica became confusing for participants when lots of physics-based forces acted on points. It was hard to explain the different effects. Similarly, if there were many points on the screen then the points risked being pushed out of their proper places on charts, misrepresenting the data. Further, the performance of the tool suffered due to the inefficiency of simulating thousands of points at a fluid frame rate.

In the following chapter I will look more into ways that Kinetica might help users through commercial explorations, however the issue of scale will remain somewhat unsatisfyingly unresolved. The latter half of this document will more closely explore this issue.

4 — DataSquid: Commercializing Physics-based Visualizations

Shortly after presenting work on Kinetica at the SIGCHI conference, I met with a variety of stakeholders at CMU to discuss how to build upon the technology. Because it was already targeted towards users unfamiliar with advanced data visualization tools, it accessed a set of potential customers that many existing data visualization solutions could not. At the urging of Project Olympus, a Carnegie Mellon University tech transfer incubator, Aniket Kittur and I applied to become members of the AlphaLab Start-up Accelerator in Pittsburgh. After being accepted into the program we worked to commercialize Kinetica technology. This gave us a firsthand view into the fraught process of translating research findings into marketable products. Whereas oftentimes a technology start-up is formed to solve a particular unanswered need, filling a niche in a crowded field or carving out an entirely new area and then building systems and infrastructure to meet needs, start-ups from research programs might enter the process with systems and infrastructure and instead must identify target areas that match the technology. This introduces an unsteady tension between abandoning or adapting technological gains from research in order to better fit a market and working to find a market that minimizes the need for adaptation.

As HCI has grown as a field, commercialization has increasingly become a subject of interest for evaluating research impact. In this chapter I will build on existing work within and outside of HCI which examines the ways that commercialization can feed back into the research community. Extrapolating from notes, conversations, and communications with a wide variety of clients with data visualization or analysis needs, I will use case studies to illustrate some of the different factors that may influence the commercial acceptance and effectiveness of new data visualization tools. I will consider some of the successful and unsuccessful paths taken during commercialization, providing guidelines for future commercialization efforts within HCI. Finally, I will describe the particular development path for Kinetica's physics-based visualization technologies which was guided both by continuing research during this time and by our commercialization efforts.

4.1 Studying Commercialization Successes and Failures

The commercialization of technology and results from research has long been of interest within the computer science community. Many foundational computing innovations originated in the corporate sector rather than the university sector. Xerox PARC [72], a primary source for innovations in programming [91], ubiquitous computing [173], user interfaces and devices [79], and desktop computing [161] in the late 20th century, balanced commercial need with speculative research budgets. While many research projects proved enormously successful, commercialization efforts did not always succeed. In the case of the Xerox Star, a cutting edge desktop computer that had clear advantages over other workstations of the era, shortsighted project management decisions lead to a product misaligned with users' needs sold like it were a copy machine and not a computer [117]. However, Chesbrough and Rosenbloom in a survey of Xerox spinoff companies observe that many were indeed successful because they had a clear plan for commercialization [30]. In some cases, corporate, public, and university resources can be pooled to create valuable new technologies, such as the Internet's emergence from a huge variety of publicly and privately funded sources [115], but with increasingly complex organizational structure come increasing organizational costs. In other cases, successful commercializations of research come well past the initial introduction of the technology and are introduced by second- or third-movers. Apple's application of Xerox Alto and Star concepts in its successful desktop computer lines [117] and it's use of multi-touch interactions first developed in the 1980s [105] illustrate this effective approach.

Commercialization is now common within universities, especially among STEM fields. Surveys performed by Lam suggest that the major motivator driving commercialization within university faculty is not purely extrinsic, but rather more often an intrinsic desire for impact and merit [99]. Many top ranking universities have technology transfer offices that go beyond filing patents and into promoting student and faculty entrepreneurship with the aim to successfully capitalize knowledge [46]. Within Carnegie Mellon University there exists the Center for Technology Transfer and Enterprise Creation, the Project Olympus accelerator program, the Swartz Center for Entrepreneurship, and a variety of grants for entrepreneurs. In a survey comparing tech transfer public policy in Sweden and the United States, Goldfarb and Henrekson identify such university support and a relative lack of regulations as key factors in US tech transfer success [55]. Some evidence suggests that commercializations do not reduce academic output for researchers, (in fact they may increase it temporarily) but that commercial success also has little correlation to later successful research publications [21]. Few long-term studies of impact exist. However, even with a substantial support structure, university technology transfers do not always guarantee positive commercialization outcomes. A meta-analysis of university analysis literature suggests that, while tech transfer is effective, it is diminished by a focus on organizational rather than individual/founder needs, a lack of resources once the first hurdles of commercialization have passed, a relatively poor understanding of the true impacts of commercialization on research careers, and a lackluster ability to identify research on frontiers that may lead to more successful commercializations [125]. The tech transfer process may also exacerbate gender disparities by further magnifying the male gender bias in academic leadership (which most often go on to become commercial founders) [135].

Within computer science and human-computer interaction there has been increasing debate over the success of commercialization efforts. Several panels have been held discussing the challenges of commercializing HCI work [33, 77]. Reflecting one side of the debate in an opinion piece outlining his experiences in research and working in early tech companies, Don Norman argues that design research is useless in truly revolutionary commercial efforts, and that the needs that designers would identify are more likely to naturally emerge as a result of the technology [117]. Technology ought to release as soon as possible so its disruptiveness can be realized. This ”Field of Dreams” approach towards commercialization presupposes that all technology will be

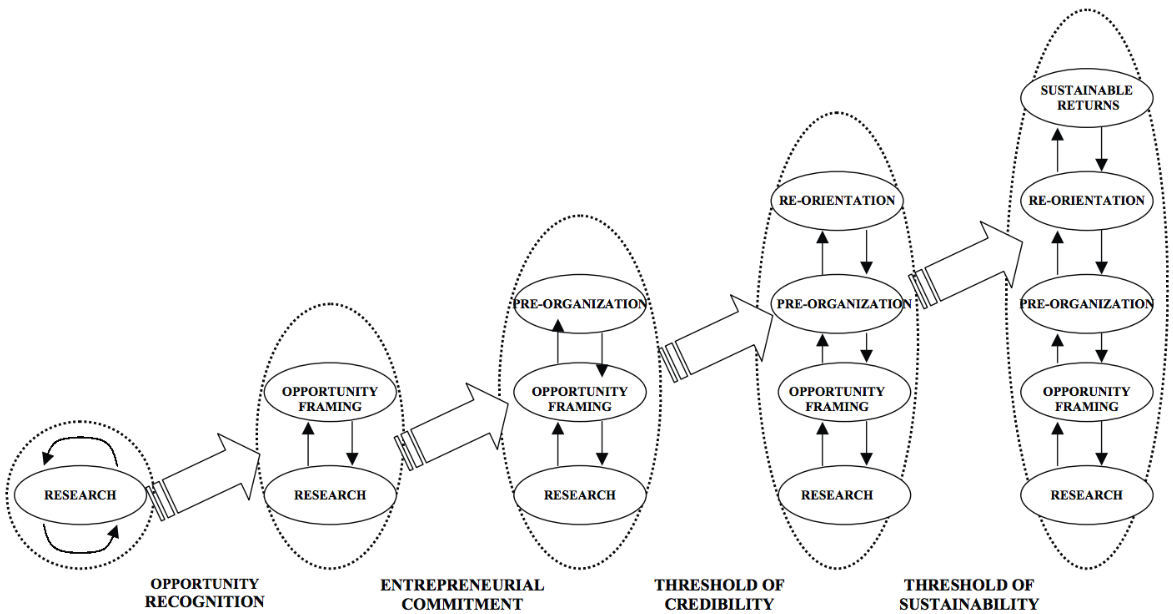


Figure 4.1: Four phases of research commercialization from Vohora, Wright, and Lockett [167].

revolutionary and that stakeholders will be able to not only translate the technology into their own context, but also see how newly discovered needs can be met by the next best thing. For the research community writ large this may not feel like an unusual possibility given the nature of peer review, homophily, and shared domain expertise within academia which serve to make impact feel more obvious and make applying ideas to foreign concepts an assumed practice. Traditional start-up praxis is firm on this issue, demanding a clean story for market and answering specific needs before aggressive testing/validation can even begin [17, 71]. Few customers will be able to translate product features into personal benefits. Clarysse and Moray illustrate this process through a case study, demonstrating that visionary researchers may lack the ability to translate research into marketable solutions (though visionaries on a team are often necessary) [34]. Vohora, Wright, and Lockett use yet more case studies to illustrate the iterative process of commercialization, suggesting that needs-finding and adaptation are critical throughout the start-up life cycle (see Figure 4.1) [167].

Orthogonal to the mercantile aspects of commercialization, a growing body of work within HCI is beginning to suggest that deploying artifacts commercially can be quite valuable for researchers. It might function as validation for research assumptions [95], a way to consider a new and unproven prototype [121], a probe for the generation of new ideas [109], or a way to access a different population for design practice [32]. Indeed, Chilana, Ko, and Wobbrock's description of their path to commercializing LemonAide as AnswerDash with venture capital funding highlights the ways in which their understanding of their research was augmented by commercialization [32]. Investors' focus on knowing customers and the particular needs of stakeholders forced them to refine their problem statement and narrowly focus their prototype. Customer demands forced them to consider software development issues such as security which might otherwise have been ignored in the research program but nonetheless proved interesting challenges. The entire process gave them a deeper understanding of the trade-offs embedded in the technology and design space of possible solutions to a wide range of customer issues. While these findings may not directly translate into research contributions, they nonetheless advance the knowledge of the field by improving one's understanding of what is useful and impactful in a given area and explicating many of the false starts and fruitful paths that research in the area might take.

4.2 Case Studies

Kinetica began its commercial life as a part of the AlphaLab Accelerator in Pittsburgh, Pennsylvania. Mirroring the experiences of Chilana, Ko, and Wobbrock [32] with LemonAide, the first core challenge we faced as co-founders was in translating core research contributions of physics-based visualizations into a value proposition for a specific kind of customer. This is often described as building "product-market fit" [17] because of the demand for each feature of the product to link to particular needs or "pain points" that a specific, well-defined customer suffered. Customers could be individuals, businesses that may use the product in a customer-facing capacity, or businesses that would use a product internally. In any of these cases, it is rare that a very general and unfocused solution will be able to capture a customer's interest. Consider the ways in which Facebook's early alignment with users' needs to administer social networks and communicate via broadcast allowed it to supplant much more fully featured, general social networking tools of the time because users' needs were so well answered by the limited feature set.

In the case of Kinetica, by then renamed to DataSquid[1], the research program focused on exploring physics-based visualizations as a general interaction and visualization technique rather than a tool to solve a particular kind of user's problems. Much like in the case of LemonAide, early investor mentors exclaimed the need to focus on a small target that could be easily communicated both to funders and customers. The initial value proposition of "allow novices to explore data more effectively than ever before, democratizing data analysis throughout an organization" did not convey particular markets where such a product would be especially effective and the actual payoff that might result. With this in mind, we began an intense process of investigating potential markets for DataSquid by interviewing stakeholders, analyzing potential competitors, and identifying ways in which markets shared the same kind of problem so that we could identify a clear niche for the technology.

Because DataSquid's explorations of product-market fit reached many different data visualization stakeholders across a wide variety of disciplines, I amassed a large body of notes, communications, prototypes, sales pitches, and brainstorms. I created a number of small prototypes, akin to technology probes [75], based upon the core DataSquid code base that explored potential features in concert with clients. Table 4.1 outlines the needs and faced by different stakeholders and their associated challenges. Considering these documents and prototypes, I have extracted several case studies that reflect different scenarios commonly shared among stakeholders DataSquid encountered. After this section I will discuss the technical improvements made in parallel to the needs finding process in light of some of the lessons from case studies.

[1] an unfortunate result of Sony holding a broad software trademark for Kinetica, a 2001 video game

Target	No. Points	Kinds of Data	Objective	Visualization Needs	Challenges
Education Benchmarker	100-1,000	Tabular, relational, geographic	How do we help customers find valuable insights in our data product?	Evaluate point vs. competitors Identify outliers Quarterly self-evaluation	Data pipeline from secure repository Sharing findings with non-technical staff Platform rigidity (Java web client) Customers intimidated by technology
Government Benchmarker	100-10,000	Tabular	How do we convey our analyses to a variety of government stakeholders?	Present analysis results Identify outliers Group points by similarities	Data needs cleaning and reformulating Customers intimidated by technology Platform rigidity (No native apps)
Financial Data Provider	100-1M	Tabular, relational, time series	How do we verify our models work and keep up to date with minute-by-minute changes?	Verify model results Observing live streaming data Identify outliers	Data pipeline from secure repository Entrenched procedures and policies Platform rigidity (offsite SQL and Windows)
Wealth Manager	100-10,000	Tabular, time series	How do we convey the value of our analysis services quickly and simply?	Present analysis results Identify outliers Justify findings with data	Novice users Need to provide a reliable data source Function at different scales of data smoothly
Independent Medical Practice	10-1,000	Tabular, relational	How do I identify outlier patients who cost/save my practice money?	Diagnose known issues using data Find unexpected relationships Present analysis results	Data pipeline from secure repository Data needs cleaning and reformulating Novice users
Defense Contractor	100-100M	Tabular, relational, time-series, graph, geographic	How do we represent complex relationships within our supply chain?	Generate ideas for new models Group points by similarities Diagnose known issues using data Present analysis results	Data pipeline from secure repository Entrenched procedures and policies Function at different scales of data smoothly Data needs cleaning and reformulating
Automotive Tech Provider	100-100M	Tabular, relational, time-series, graph, geographic	How do we identify places where our product does not work well?	Generate ideas for new models Diagnose known issues using data Observing live streaming data Identify outliers	Data pipeline from secure repository Platform rigidity (Windows) Internationally distributed teams Function at different scales of data smoothly
Consumer-facing Web	100-10,000	Tabular, relational, geographic	How do we make our web product feel more valuable to users (and make them want to come back)?	Group points by similarities Identify outliers Present analysis results	Platform rigidity (established web infrastructure) Customers intimidated by technology Data pipeline from secure repository Function at different scales of data smoothly
Product-hunting Consumer	10-10,000	Tabular, geographic	How do we quickly and easily find the right thing from a big pile of them?	Identify outliers Group points by similarities Find unexpected relationships	Novice users Platform rigidity (must be web & mobile) Need to provide a reliable data source

Table 4.1: Overview of DataSquid customer investigations and general findings.

4.2.1 Case Study 1 - Portfolio Evaluation

Sara is wealth manager at a local Pittsburgh firm. It is her job to watch client portfolios on a daily basis, making sure that the distribution of investments in the funds she uses for her clients is satisfactory. The stock market changes rapidly, and Sara needs to be on top of any trends before they become problems for her clients. Several times a week she meets with different clients in order to explain to them how she has managed their money thus far, and to set their expectations in terms of growth. This is a very difficult task for her, as markets are hard to explain and she needs to earn her clients' trust. After all, they work with her because she can give them an edge in returns. The state of the art in her field are large prospectus binders containing roughly 100 pages of information. She would like to use interactive data visualization tools to better walk clients through their portfolios.

Sara's data is relatively straightforward, but it is quite heterogeneous. Each of her funds vary from including dozens of companies into thousands across a wide variety of sectors and industries. For each company, she has a wide array of metrics she uses to evaluate quality. She examines price changes on day, week, and quarterly time scales, comparing them to yearly min/max. She considers debt, profits, capitalization, and growth. She estimates future performance using her impressions on individual companies and sectors, comparing and contrasting her choices. One core process she uses is "what-if" analysis, where she considers if she were to invest more heavily in a sector or fund, and computes how that might change her projections.

While these tasks seem numerically bound, they are much more about sensemaking. Sara uses her intuition, examining all of the factors and judging using her knowledge gained from years of experience. A huge part of her expertise is in comparing individual entities to their peers. For instance, she will inspect high and low performers in each sector, trying to understand whether they are indicative of a trend in the sector, or just outliers. Her focus jumps rapidly between the individual point level and aggregate trends. When necessary, she brings in comparison data, or benchmarks, to see whether the fund outperforms them.

Physics-based visualization technology well matched these needs by allowing Sara to quickly pivot between multiple dimensions of data, tracking and comparing changes as the points animate. Using the DataSquid interface (Figure 4.2), Sara was well able to view the diversity of her funds by charting the them versus sector. She noted that her benchmark and two funds seemed heavily invested in Finance. Switching to performance and highlighting the fund that interests her, she sees Health-care performed particularly well. She wants to understand what drove growth in Health-care, so she filters down to those points. Finally, she isolates the top performers, finding that she in fact knows those 6 companies well based on the investment news she follows. She could comfortably walk a client through these steps, providing a narrative along the way that shows her skill and experience.

Sara's exploration worked well in this limited case. The funds were small, and she was only pivoting between performance and a few other measures. When Sara wants to compare funds where each contains several thousand different investments, the existing DataSquid approach is insufficient. Indeed, her iPad screen won't even allow that many circles in that area of space. Instead, she needs an ability to collect points together at a high level, and then query/zoom/focus

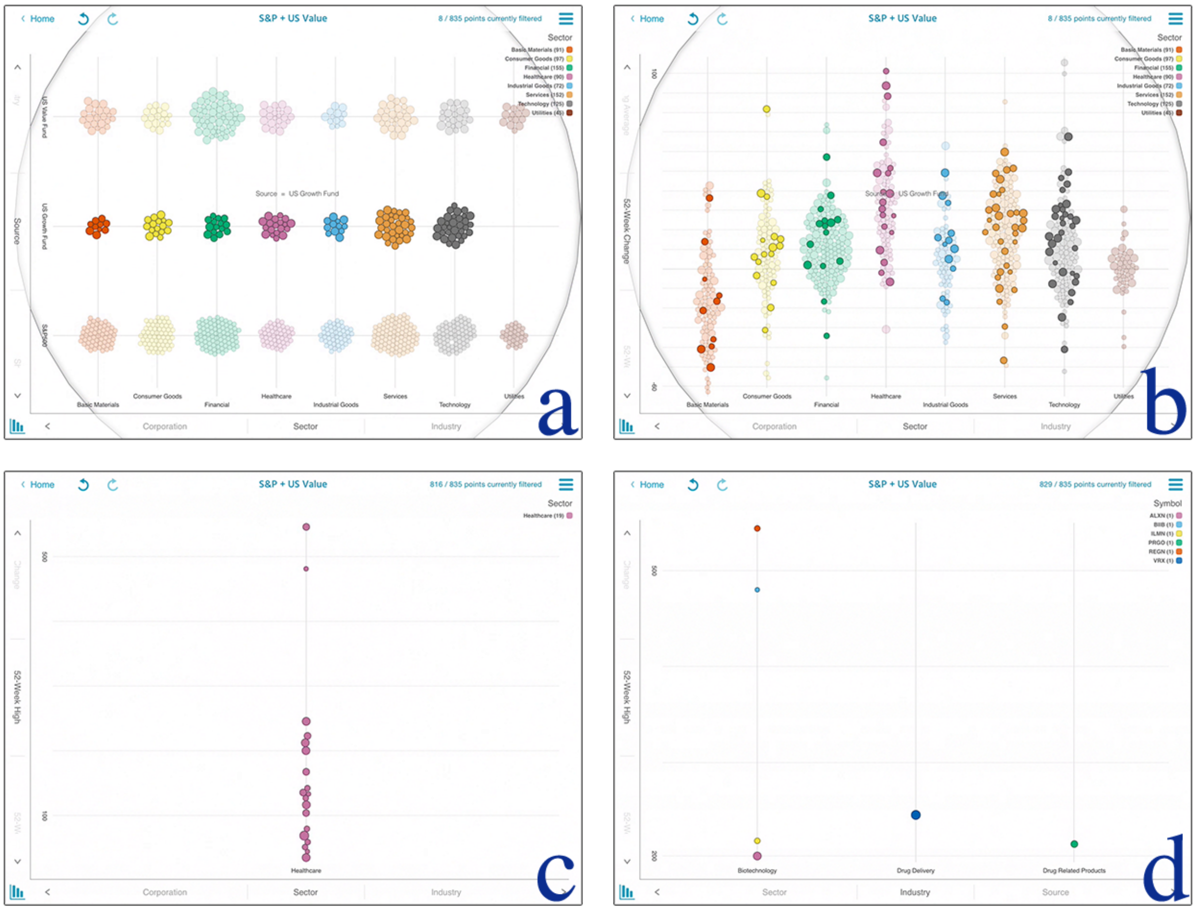


Figure 4.2: Sara explores a Vanguard fund (highlighted) in comparison to two benchmarks (lowlighted). In (a) she observes the distribution of funds across sections, switching to view yearly performance (b). Noticing that Health-care (purple) is outperforming her benchmark sets, she selects those points and isolates them (c). Finally, she wants to look at the top 6 performers in Health-care in more detail, so she focuses on them and tries out different measures on her plot to learn more (d).

as her area of interest begins to focus. This points to a general technical need for physics-based visualizations to scale fluidly when interrogating large datasets.

Ultimately, uptake of physics-based visualization for wealth management was poor for a number of reasons. While wealth management clients were interested in mobile devices for the portability and marketability, they also conducted a lot of business using desktop computers and email. A mobile application would be insufficient. Further, data analysts like Sara often performed the same analyses over and over again. Every meeting might start by walking through the previous year's growth in a portfolio using a small subset of attributes. While a fully featured sandbox interface that allowed users to explore any dimension had many affordances for different kinds of data exploration, that many features can prove intimidating and overwhelming in the face of smaller, rote tasks. This suggests a need to carefully strip down exploratory data visualization tools so that their affordances closely match anticipated user needs but do not go too far beyond them. In the commercial space we saw evidence of this issue in clients' resistance to larger solutions such as Tableau because of perceived high training and integration costs. There may exist a niche for both high-cost but high-featured tools like Tableau as well as low-cost but concordantly lower-featured or constrained tools like DataSquid.

PORTFOLIO DATA PROVIDERS

Through DataSquid's conversations on wealth management we also were able to connect with firms that provide data to finance analysts and wealth managers. These organizations have on one hand quite similar needs to wealth managers: they need to show the value of their data products efficiently to an audience that may not always be familiar with machine learning models or statistical analysis. On the other hand, such data providers also have a need to validate, diagnose, and develop models internally. Similar to Sara's use case, physics-based visualizations can allow such a data provider to quickly walk through a number of different aspects of the data, demonstrating their value efficiently and visually to users. However, data providers suffer even more acutely from issues of scale (as indeed scale is a major selling point), having on the order of hundreds of thousands of data points potentially repeated minute-by-minute during trading hours. Even if data quality can be demonstrated visually by DataSquid's interactive visualization, their clients may not trust seeing only a small subset of data. Organizationally, there also existed an interesting divide within data providers. Technical staff developed and processed models regularly, but sales, marketing, and management staff were not integrated into the data generation pipeline. In fact, those professionals may not have deep insight into how the company's product is constructed. This points to another key area where exploratory visualization tools might provide real commercial value: exploratory visualization tools that are novice-friendly might allow more staff in an organization to make sense of and contribute to making sense of company data. A marketer might have key insights into the kinds of data features that sell well, but be unable to evaluate how well the product achieves those goals without tools that make the data more comprehensible.

Working with financial data providers, I developed a prototype based upon a common issue: representing trends over time fluidly without requiring a dozen frame-by-frame snapshots in a report. I imported time series data into DataSquid and initially made points move to their new

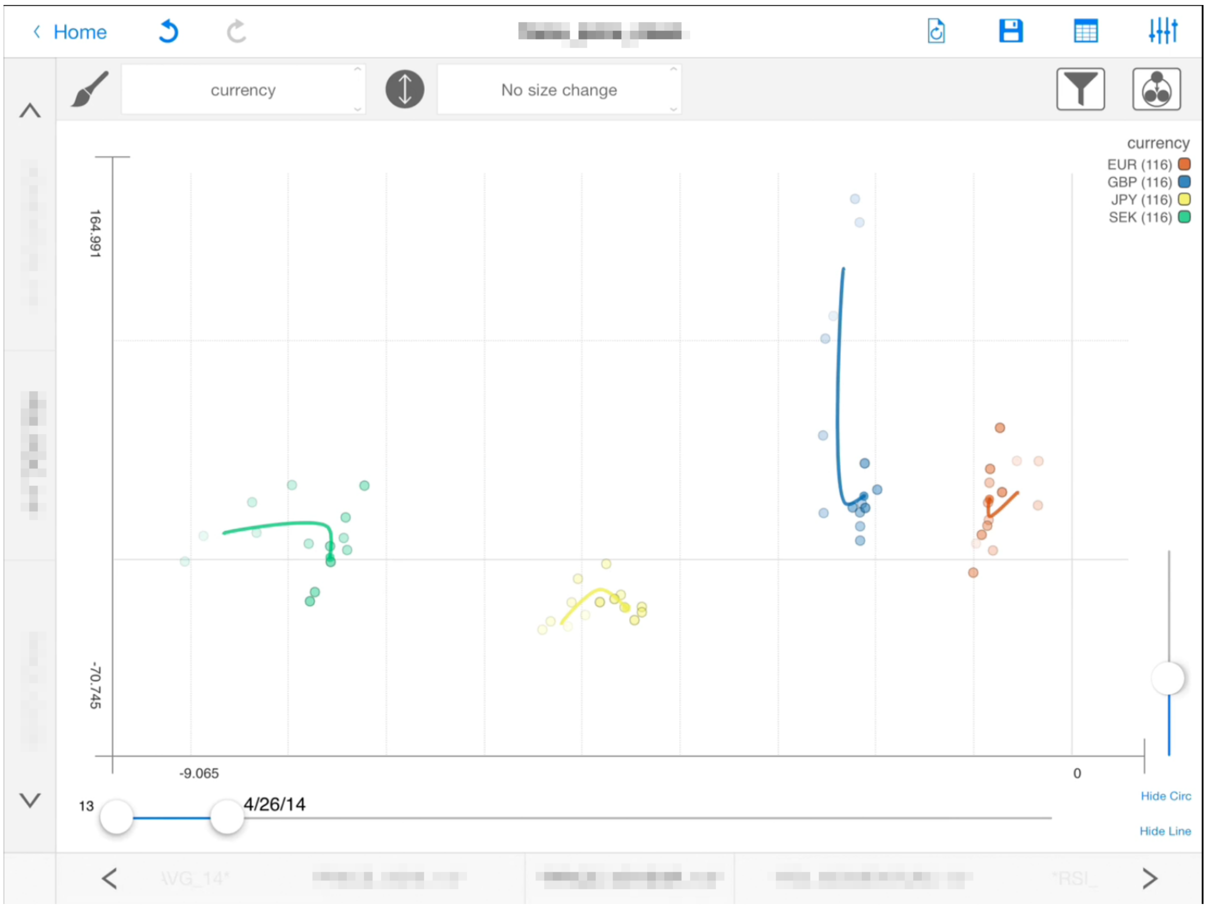


Figure 4.3: Here a user is showing a small window of time in a dataset on foreign currency values. By moving the slider, the user can expose new points and the trend lines and other physics-based tools will adapt accordingly. Because analysts in this use case voiced wanting to see ”trajectory” rather than individual values, the trend line helps them to estimate change over time without the overload of examining dozens of points at once.

values as the user changed the time step represented on screen. For instance, if one point rose in value along the y-axis of the chart, it would feel a new magnetic pull to that place, moving slowly to its new value unless another physics-based tool like a filter wall held it up based on other changing values. However, animation of this nature became quite confusing when only dozens of points were onscreen. Users simply could not track motion of that many targets. As a result I replicated points of data for each point in time, filtering them down so that users only saw their selected time range. I used the collision layering system (designed to keep points from bunching up as they were being filtered by allowing points to selectively pass through other points) to separate each ”day” into its own grouping of points. By automatically employing DataSquid’s lens highlighting system, a user could employ all existing physics-based tools and then integrate them over time by manipulating a slider (bottom of Figure 4.3).

However, separating each time into a separate layer produced jarring jumps between time steps, as the same point of data may have ”bunched up” in an entirely different region of the screen each time step. There is no guarantee that the collision solver will behave perfectly deterministically when the initial point locations overlap heavily (floating point error dominates the first few collision resolution steps). As a result, I introduced a sliding window technique that allowed datapoints from several instances of time to collide together within one sandbox environment, and pre-computed where future or past datapoints would begin to appear should the user continue to manipulate to slider to see more dates. They collided as well, but were hidden until the user chose to show that range of times. This approach worked, smoothly showing trends. After viewing the prototype, data providers realized that the main data feature they cared about was the general trajectory of the points on the map over time (i.e. are the rising in value or falling in value over time) rather than their initial assumption of wanting to explore individual values. Incorporating trend lines composed of bézier curves approximating the position of points at each time step revealed these trajectories. Because points gain and lose visibility as the user moves the time slider, the lines grow and move like a snake as new approximate positions are added and removed, allowing the user to estimate second derivative features, or the trajectory of the trajectory. However, while animation and interaction readily showed trends the visualization does not necessarily lend itself to static views. Resulting from these discussions it is clear that a subset of users is in need of time series visualizations that readily let them identify first- and second-derivative changes in their data, but the task is complicated by the need to represent trends accurately even in the face of thousands or millions of data points.

4.2.2 Case Study 2 - Manufacturing Logistics

Teri is a manager in a large multinational manufacturing company. She controls a division of the company responsible for systems that diagnose and deliver repair parts for automobiles. Her main responsibilities are to identify ways that the company can better diagnose vehicles and to make sure that the company manufactures and delivers the right parts to the right repair stations. This is an over-constrained problem. Her company sells thousands of different parts for several hundred different automobiles in dozens of countries. The software she manages is complex, with millions of potential repair codes and instructions.

Recently, Teri lead an initiative to try and improve BMW repair at her company. There were too

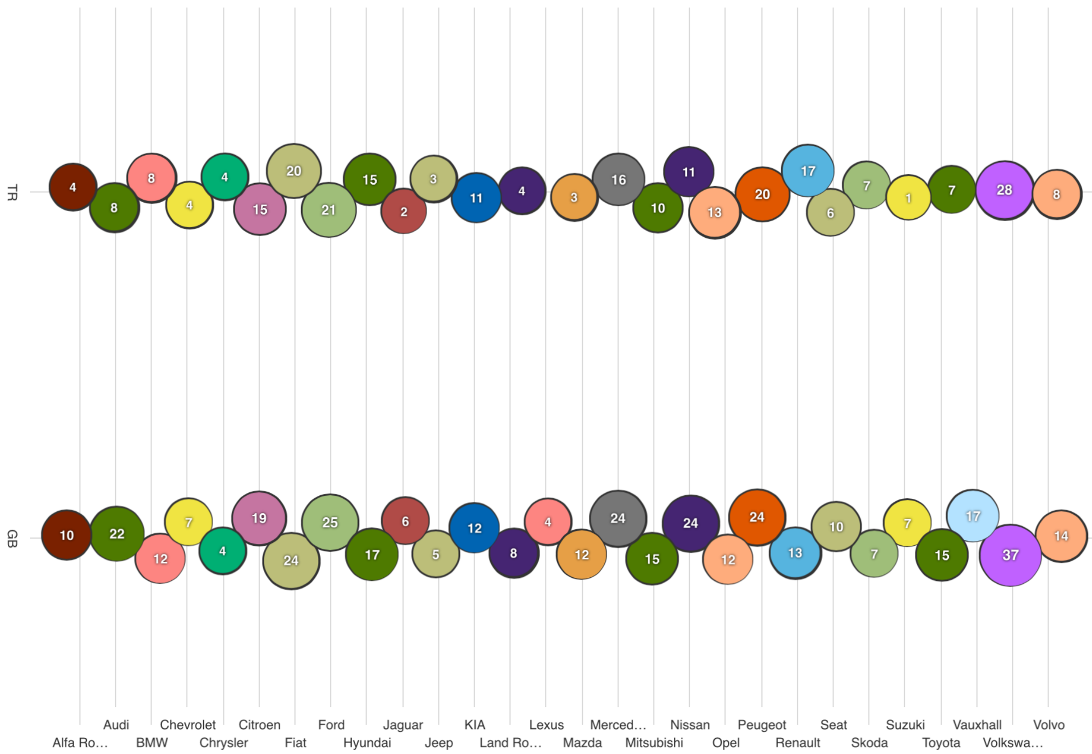


Figure 4.4: Example groups of models diagnosed versus country Teri uses in her analysis. Without a querying ability, it is hard to Teri to slice and explore these different groups of models.

many reports of BMWs being misdiagnosed in certain countries. Eventually, she realized that certain languages were missing sets of diagnostic instructions in their translations by checking to how different diagnostic codes were used across the world. This lead to a shortfall in the number of engine parts going to those countries as well, because the diagnostic codes were never given in order to suggest a part be ordered. This complicated chain of mistakes was hard to explain to her VPs, and was exacerbated by her need to constantly ask her data team to perform certain queries over their analytics databases.

Teri wanted to see if there were any other diagnostic issues in other models of cars similar to the one she just noticed. Using DataSquid on an example dataset, she was able to see stark differences in the models of Ford sold in Canada versus the United States. From her expertise, she knew they were just different names for the same models, so she switched files to view one of diagnostic codes by model and country. By visually inspecting different diagnostic codes, she could see some gaps. However, the database stores well over a million different entries, and DataSquid could only show a thousand at any given time. She has a good idea of ways she might connect and group the data that her company stores, but doesn't know SQL and can't aggregate the data herself.

FINDING DATA

Assuming one had different datasets aggregated from Teri's data stores, physics-based visualizations would readily help her diagnose by triangulating on key data attributes that relate to the models she is comparing and drawing attention to outlier points as shown in the Kinetica study. However, Teri does not arrive with a dataset. In our investigations of different potential customers, this was an especially common theme. Some managers we spoke to mentioned that they would have to wait on their data team to respond before they did any analysis at all. Others weren't aware of where sales data came from and how it was used within their organization, other than that it appeared in their inbox once a month in summary form. Another group knew what sorts of data they might use, but was daunted by the process of moving a file to a mobile device or uploading it to the web.

This challenge is even more critical if the data come from a relational database. Whereas a tabular dataset fits relatively easy into the core Kinetica mapping of one point of data to one physical object, relational databases encode more information and do not always easily map to a plot-based visualization. For instance, in Teri's case, she may need to aggregate a store of individual diagnostic codes with a store of diagnosis events with a store of customer details. This goes well beyond the grouping and aggregating aspects found in plotting tools like DataSquid, and it even poses a challenge in commercial data visualization software because configuring queries and generating results remains very complex. For our customer investigations with organizations like Teri's, we opted to prototype systems by simulating a successful querying engine. For instance, if the customer in a design session wanted to group by car models, we would load another pre-generated file that reflected the GROUP BY command and present it. This wizard of oz style approach allowed us to more rapidly iterate on ideas given the challenge of building a visual relational database querying engine, but it mandated that we generate a large set of possible queries first. This task was not always tractable, especially when working with analogous scenarios in health-care (i.e. which patients are diagnosed with which diseases more often, and what are we under-diagnosing) where HIPAA and data security become issues.

There exists an interesting body of research on constructing and visualizing query results, including systems that use tables and fillable fields to query and represent data [156], interfaces organized around flowcharting and diagramming [61, 62], simple languages for routing databases to visualization elements [145, 146], and tools that use intermediaries like recommendation systems to construct desirable queries for the user [180]. The key question is where to integrate such techniques in the data exploration process. Should users construct queries and then view/explore their results, or should querying be an inherent part of the interface. Perhaps, once Teri finds a suspicious car model, she has the option to GROUP BY, COUNT, or SELECT from a list of connected elements. If Teri could query her databases in a fluid way and preview different slices of results, than she could more quickly zero in on interesting trends. Given naturalistic interfaces' ability to be used with minimal training, an ability to link relational database content would further empower novice users to interact with complex data.

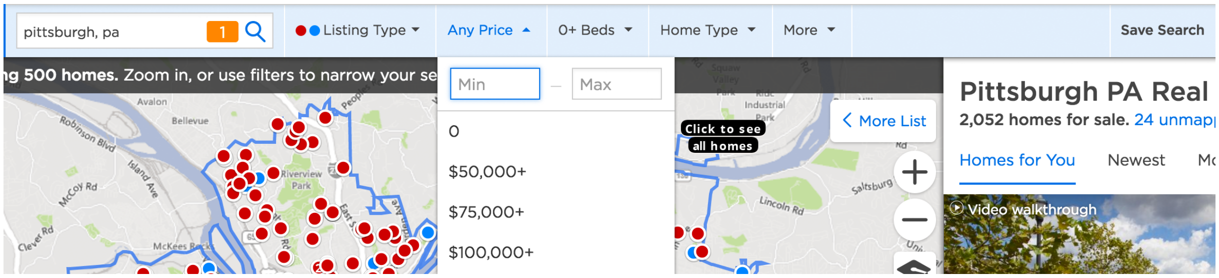


Figure 4.5: Example affordances in a consumer product exploration tool. Notice faceted browsing elements and a map for charting point locations.

4.2.3 Case Study 3 - Real Estate

Ray is looking to buy a house in Pittsburgh, but is having a hard time finding the perfect one. He's an optimizer after all, and he won't be able to pick an place until he feels like he has seen all of the options and made the best decision he can. Ideally, he wants a place with 3 bedrooms for his partner and future children, and at least 2.5 baths. His budget is roughly $300,000 for a nice neighborhood location. However, he is willing to shoot for more bedrooms or a cheaper place in a less nice part of town, or get a bit less if the area is good. If the place has a pool, he is willing to go for a bit more expense. In general, he could probably list out a page of different "if it has <blank>, then I guess I could give a bit on <blank>," sentences.

He tried using existing tools, but dynamic query sliders weren't sufficient for his soft constraints because he wasn't completely sure how his budget and choosing broad filters gave too many results for him to easily sort through. Trying to sort through a set of narrowly defined filters made it hard for him to compare across them, even if individually they produced a more tractable number of results. His attempt at using online realty websites like Zillow were frustrating, as at most he could look at paginated results or a map where he had to individually check each option for pricing details. His main goal is to get a sense of the realty market before zeroing on several candidates, but he doesn't want to learn a complex tool like Tableau to do it.

Ray was able to chart out real estate along a wide variety of dimensions using DataSquid. Initially he charted houses by price versus square footage. He could see the general distribution, noting that there was a decent alignment between cost and size of house. He was after a moderately priced, moderately sized house, so he drew a circle around them. He chose to focus on those points, hiding those that didn't meet his criteria. Using the scatterplot axis selectors, he switched rapidly between different dimensions, including bedroom count, bathroom count, price if it were a rental, lot size, and neighborhood. He felt he was able to find some interesting candidates and understand what he was looking for better, but he wanted to be able to compare his finds in Pittsburgh to other cities. Ray would like to be able to hop between different levels of detail, exploring Chicago or Cleveland realty while still being able to see a bit of his Pittsburgh points for comparison. This points to a need for context in addition to an ability to focus in on different regions of a dataset.

MAPPING POINTS

For consumer web tools like those of Zillow, AirBnB, Yelp, Google Places, VRBO, Trulia and many others, datapoints have a geographical component. Ray cares just as much about the house's neighborhood as he does other characteristics like price and square footage. At the start of customer exploration, DataSquid did not possess any mapping functionality. However, it quickly became apparent as we exploring consumer-facing scenarios that mapping was a critical feature. As a result, I quickly prototyped and tested mapping tools within the physics-based visualization using the Apple Maps API (Figure 5.1 in the following chapter). One core design question dominated this implementation: should the data guide the map display (i.e. the map is fixed to the current data on the screen), or should the map guide data display (i.e. should the map be pannable, and panning filters and moves data). Given DataSquid's touch interactions for drawing upon the map and specifying points, I chose to implement the former since allowing the user to pan the map would necessitate a method of choosing either to manipulate the map or the data. Points are pulled magnetically to their correct point within the map, and should the user focus in on particular points or filter others off of the screen, the map adjusts its bounds accordingly, zooming in or out. While this may not match typical web interactions for exploring map data, it builds an interesting relationship between the map and data exploration actions. Rather than acting as a separate controllable entity, the map reacts to and brings the focus to the data points on the screen. Since DataSquid employs a consistent physics-based sandbox, it is seamless for users to move from the map view to the plot (since points are just pulled towards a new place) and all existing physics-based tools function as expected.

In the initial prototyping phase, this consumer use case by far gathered the most interest from potential customers and funders. Test users expressed that its ability to help users quickly develop and test criteria sped up the decision-making process, and that the collision aspect of the physics-based simulation was an improvement over occlusion common in web map interfaces. Test users also expressed excitement over the ability to take whatever they have found in the tool and then use the tool as a presentation device for sharing their findings. However, several core issues remain with this approach. While collision helps against occlusion, it suffers if too many datapoints occupy a small area (as would be the case in a city like Tokyo or New York). As most people we talked to did not have the skills necessary to write scraping scripts to gather product data to visualize, the process of finding and installing data, as in Case Study 2, remains too complex for novice users.

4.3 TECHNICAL DEVELOPMENT

I continued development of the Kinetica tool following its initial publication and into its commercialization, building out features that appeared lacking over the course of customer discussions, user interviews, and experimental prototypes. As I was now receiving new datasets from customers with associated problems, the process of prototyping using data became much more grounded. Instead of exploring the design space, my focus became refining the feature set of the Kinetica tool as it transitioned into DataSquid. No users expressed a need for the histogram tool, and every one used a scatterplot. In the first iteration, the scatterplot became an inherent part of the sandbox environment so points always had a position to go. Following that initial change, I implemented more tools for showing exact details, filtering for hiding points completely, and undo/redo functionality. Undo did indeed prove particularly troublesome in the physics environment, as some tools permanently mutate the environment. For instance, after adding a permeable wall, some points will have be relocated. I experimented with storing complete environment states in the undo stack, so that the environment would be exactly as it was, but this proved inadequate. Upon undoing, users would see the interface snap back to a position confusingly given the fluidity they had grown to expect. Instead, the undo/redo stack simply stores interactions, and unwinds them as best as possible without influencing the points. For instance, undoing a permeable wall means that the points now have the ability to return to their previous position.

Using knowledge gained by creating and testing prototypes of new, speculative features, I incorporated mapping into DataSquid, using the Apple Maps API to project latitude and longitude onto a virtual scatterplot, which, as in the data view, magnetically pulled points to their proper places. The transition from data to mapping and back again naturally emerged out of the physics model, with points smoothly transitioning from state to state. As users increasingly showed interest in presentation and sharing of findings, I incorporated a more fully featured save state function that allowed users to share their dataset and associated physics environment. By including the complete undo/redo stack in the saved program state users could begin to create rudimentary presentations. While more an appropriate of existing technology, it proved useful both for presentations of the product and consumers practicing data walkthroughs using the tool. Table 4.2 outlines some of the evolutions made to the Kinetica tool due to commercial feedback, and Figure 4.6 shows the interface's current state.

4.3.1 Web Prototyping

Commercializing DataSquid allowed us the ability to hire skilled developers onto the team in order to build out a future version of the tool. As many of our customer interviews revealed, mobile native clients were not well suited to reach businesses as customers. Many businesses made use of antiquated desktop computers, and in some cases company policies actually prohibited the use of mobile devices within the company network. When the Kinetica project began as TouchViz, Javascript engines in web browsers generally had low performance. By 2016 that had changed. Transitioning DataSquid into an HTML5+Javascript application was achievable.

With the help of an experienced web developer on the team, we were able to create an initial prototype of many of the DataSquid tools within a web browser. Unfortunately, this process was neither simple nor straightforward. Moving interactions from touch (where users might felt connected to the device and tools) to a mouse where there were fewer control points proved immensely challenging. Tools like lenses and walls now required the use of lasso-like selector in order to define a region of interest, and window/screen size became a consideration given the design had to be responsive to various device sizes. Ultimately this translation process was unsuccessful because a design made for native, touch interactions on a device held in the hands could not ported without losing a significant portion of the interaction model that was shown to be successful in our Kinetica study. In Chapter 6 I will revisit this open question through the development of a new web-based prototype inspired by but not duplicating DataSquid tools and interactions.

Feature	Rationale
Scatterplot configuration	Since one core use is rapidly switching between axes to compare dimensions, making the scatterplot easy to configure is important.
Eliminating tool palette	Rather than requiring users to choose a tool and then draw, every interaction now begins with drawing a region and then selecting an action (like grouping or adding a lens).
Style configuration	Users often restyled points, and a tabular view for picking color, size, and axes makes it easier when there are many dimensions.
Axis stretching	Many beta users' data was skewed. Stretching axes helps users identify trends even in areas of high density in the chart.
Hide/focus	Users often wanted to isolate interesting outliers, or exclude a region as if they were brushing.
Time-series data	Allowing users to pan over time is an easy addition, since the physics model adapts to changing values among the points.
Auto-grouping	Inspired by pivot tables, auto-grouping allows users to quickly aggregate and split points along numeric/categorical lines.
Screenshot/sharing	One common use of Kinetica among beta users was for presenting data to others.
Overlappable points	Another way to manage high density areas on the chart is to break with physics a bit and allow point entities to overlap slightly. Opacity shows density gradients.
Interactive tutorial	Face-to-face training does not scale when beta-testing, so an interactive tutorial framework was developed for onboarding new users and participants

Table 4.2: Features added to Kinetica as a result of the commercial inquiry and beta process.

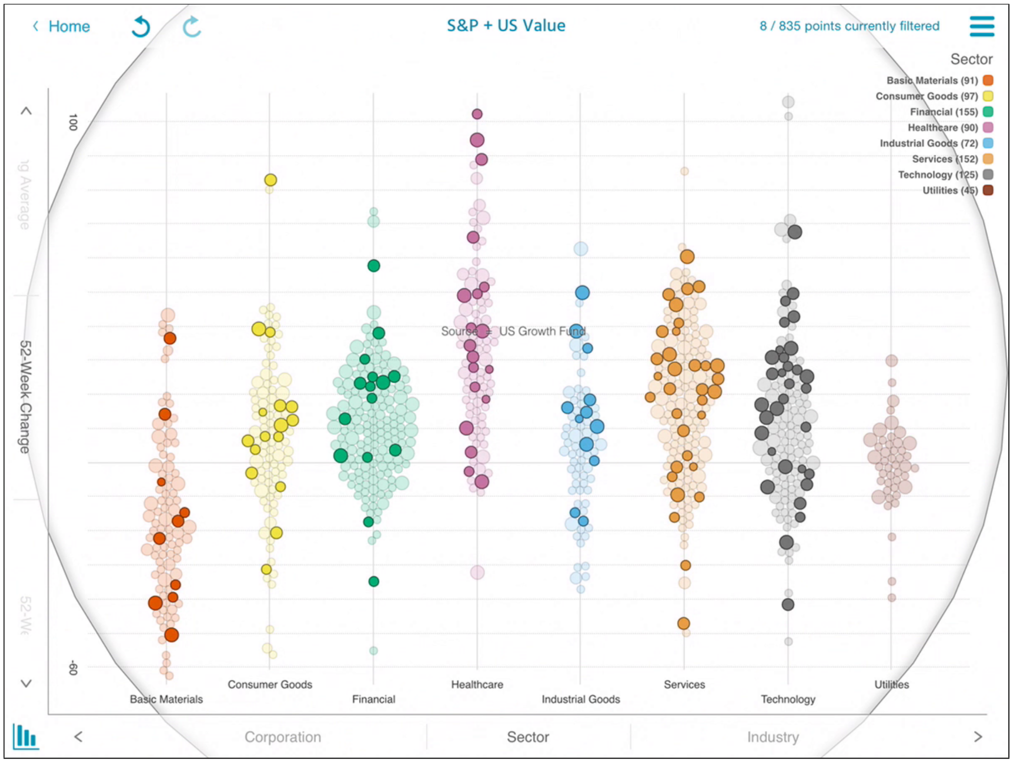


Figure 4.6: The current DataSquid interface. Note the removal of a palette and scatterplot configuration bars. Touchable items are colored blue, and users create/manipulate data by drawing on the data area with one or two fingers.

4.4 Open Questions

Commercialization revealed a number of interesting properties in the Kinetica/DataSquid prototype. Primarily, the needs-finding process helped to pare down the number of features that DataSquid incorporated and to reveal some use cases that were especially well matched for physics-based visualization. In particular, one core need coming out of conversations on both logistics analysis and real estate search was the accommodation of geographic data. This introduces an entirely new set of complications for physics-based visualizations.

While the bare-bones probe of geographic data simply involved setting latitude and longitude as numeric attributes on its default multivariate plot, this was unhelpful for users as it lacks any geographic context and misrepresents distances as a result of the geographic projection. Thankfully, a number of mapping data APIs exist that can provide tiles of map information containing roads, labels, and geographic landmarks. However, this still presents a significant challenge as the map is an entirely different representation from the multivariate plot view that is the default in DataSquid. This presents a number of interesting questions which will be explored in the following chapter:

- How well do physics-based interactions integrate with mapping points geographically? Are they intuitive?
- Can users make sense of the transition between maps and multivariate charts? Are they

able to make use of both numeric and geographic data in the same exploration?

- How well do physics-based visualizations help with open-ended decision-making tasks? How do they compare to other data analysis software?

5 — Evaluating Physics-Based Maps & Consumer Search

As discussed in the previous chapter, mapping became a core component of DataSquid, however it largely remained un-evaluated during commercialization. In order to examine whether users would successfully be able to transition between physics-based plots and physics-based maps, as well as to integrate knowledge from both views into their mental model during an exploration task, I designed a study focusing on data that make use of both multivariate data features and geographic features. Considering the kinds of data one might choose that involve both numeric and geographic components, consumer search provides an interesting candidate because consumer decision-making use cases proved especially compelling on the commercial side of DataSquid/Kinetica.

While there has been an explosion of web-based tools for finding restaurants, houses, vacation homes, products, movies, art, and others, many share similar design elements which do not necessarily help users who do not enter the site with clearly defined search criteria. Affordances such as faceted browsing and dynamic querying [4] allow users who know their needs well the ability to quickly focus on relevant targets. I identify the following design elements shared between tools as particularly relevant to exploratory product search:

- There exist many points of data, often clustered into dense groups and numerous enough to make individual inspection difficult or impossible.
  - (e.g. houses, restaurants, cameras)

- Points of data have many different quantitative and qualitative attributes differentiating them from other points.
  - (e.g. price, user rating, reviews)

- Points of data often have geographical locations, allowing them to be charted on a map.
  - (e.g. hotel addresses)

- In exploration, distribution within an attribute and outliers help form a baseline for searches.
  - (e.g. average price of a house, unusually well reviewed restaurants)

- In exploration, users often must consider relationships between 2 or more attributes.
  - (e.g. balancing meal cost and rating, average price of small homes in a neighborhood)

- Explorers often must compare one point to another efficiently, identifying attributes that differentiate them from one another.
  - (e.g. comparing one's set of top items to find several unique candidates)

- Explorers sometimes need to convey their search or findings to others.
  - (e.g. justifying a restaurant choice to one's picky friend group)

Product search shares these design elements with many other data sensemaking tasks. In my study on Kinetica, participants who were examining a dataset of Titanic shipwreck passengers similarly sorted through many points trying to identify relationships and outliers while making sense of the overall dataset. The results of the Kinetica study suggest that physics-based visualizations can provide some concrete benefit to product search. Kinetica participants were shown to be better able to make claims about relationships between factors and summarize what made points unique. This is analogous to making sense of and balancing product attributes in a consumer search. Similarly, Kinetica participants were better able to identify outliers in a dataset thanks to the physics-based animations.

With respect to these potential benefits, in this chapter I describe the results of a preliminary study designed to evaluate the effectiveness of physics-based visualizations as compared to other data analysis tools (for the purpose of consumer search). While one could directly compare DataSquid to an existing product search web site, for this study I have opted to compare DataSquid to two tools commonly used by consumers to analyze data: Excel and Tableau. Excel maps to tabular data analysis tools and Tableau maps to more complex dashboard/visualization builders. While Spotfire and Qlik might provide slightly reduced training and easier initial use compared to Tableau, I intentionally chose Tableau as an extreme case of an expert visualization builder. To reduce some of the potential load, I made sure that an expert Tableau user prepared a dashboard and associated views for participants so that less training was required. By treating this as a data exploration task rather than a product-finding task, I sacrifice a portion of ecological validity for greater external validity. Product search websites are often very tailored to a particular data exploration task, while DataSquid and its physics-based visualization prototypes are not so specified. Should one show that a particular web tool for finding restaurants outperformed in a consumer search task, it is difficult to attribute that performance to the tool's close alignment to the task or to particular design characteristics in the tool. While my findings may not allow one to directly point to certain web-based products as superior systems, they do permit greater generalization about how different aspects of data visualization techniques are beneficial for search/exploration tasks (as mapped to the design elements listed above) and further refine the findings of my earlier Kinetica study.

5.1 CONSUMER SEARCH

Consumer search has long been studied by marketing and communications researchers. Before the advent of Internet shopping, consumer search largely depended on brand identity, word of mouth, print/video advertising, and brief time in stores (potentially comparison shopping) [18]. In the 1990s and 2000s researchers began examining the impact of Internet shopping on traditional consumer search behaviors. In light of the changes that the Internet has experienced over the past decades, it is worth remembering that recent consumer search tools take advantage of technical improvements such as interactive HTML5 elements that were not achievable a decade earlier. In the year 2000, Internet shoppers proved less affected by brand identity [171]. Instead, shopper behavior online was mediated by shoppers' expertise, with experts spending more time searching and giving up less easily [171]. Interestingly, expertise in using the Internet as a resource proved more impactful on self-reported search time than objective experience with the product itself [93]. While search breadth (number of sites) was influenced by experience, using the Internet to research products did not ultimately save time for users in 2003 [93]. Search time is a critical factor in one's later satisfaction with the purchase. Whether people spent low or high amounts of effort on the search, perceived time pressure due to a search taking too long or not having enough time to complete a search reduced a buyer's eventual satisfaction [18]. This suggests that making consumer search more efficient may produce better outcomes for the searcher.

In a seminal 1983 study, Punj and Staelin showed that both inexperienced and experienced shoppers cut their searches short as compared to shoppers with average levels of self-reported experience [130]. In a sort of Dunning-Kruger-like effect [97], inexperienced shoppers were unaware of the aspects of the product search that they were missing, while experienced shoppers cut short their search as a result of overconfidence before they saw an adequate number of options. This suggests that tools that push users to explore more options regardless of experience may produce better outcomes. Indeed, systems developed to explore users to more trade-offs and comparison points between examples improved users' confidence in their search process [129]. The presence of certain kinds of information also helps to improve users' confidence. Surfacing community written reviews and signals of others' informed purchases had a significant impact on search results [31].

Many of these features translate from a pure consumer search problem into the perspective of a data exploration problem. Search time is analogous to time spent identifying points that meet particular criteria. Satisfaction in a search result can be directly measured if the data analyst has some stake in the result, and also can be informed by the analyst's confidence and likelihood to use the tool or dataset again. By examining the kinds of options analysts identify as meeting their criteria, one can estimate whether the analyst explored enough data in order to find the optimal point or cut their search short out of hubris, frustration, or inexperience.

5.2 Study Design

For this study I utilized the mapping prototype of DataSquid as illustrated in Figure 5.1 featuring the technical and design improvements outlined in sections 4.2 and 4.3. The DataSquid tool reflects all of the tools and interactions used in the Kinetica study with the exception of histograms and the ability to place a scatterplot (the tool places one over the entire screen for users). It remains a native iOS application. While in the Kinetica system users chose tools from a palette and then used one or two fingers to draw them onto the screen, in DataSquid a user draws a region first and then chooses what type of action to assign to the region (for instance, choosing to make it a wall that excludes certain points from a region of screen space). Maps are provided by the native Apple Maps API with geocoding embedded into the dataset rather than done on demand.

For the consumer search task, I chose to sample real estate listings from Zillow.com. Real estate poses an interesting challenge for consumers because it is both an expensive, high-risk purchase and subject to many soft constraints (e.g. one might be interested in another bedroom, but be open to one fewer if the place is cheaper). The ramifications for making a mistake can be high,

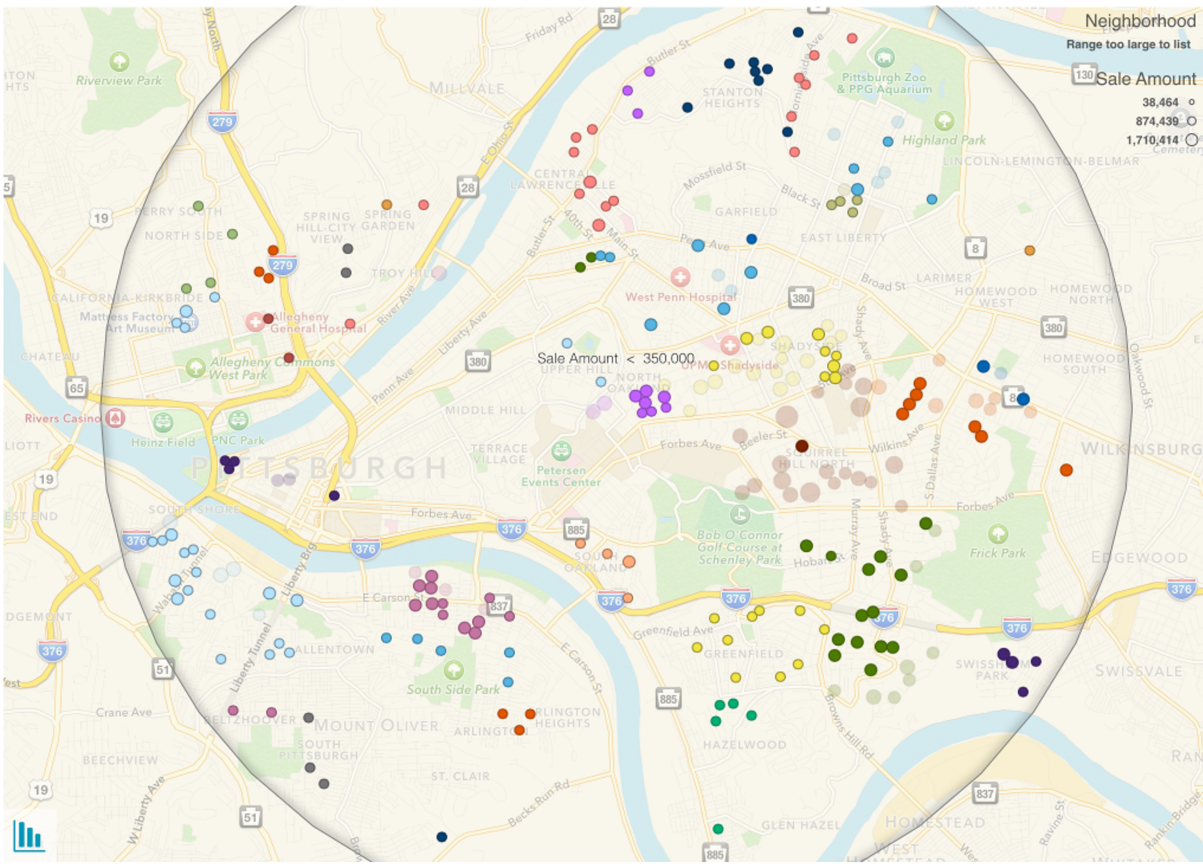


Figure 5.1: DataSquid charts a set of Pittsburgh houses on a map. The user has added a lens that is highlighting homes within their budget.

Sale Price	Sale per sq.ft.	Rent Price	Rent per sq.ft.	Bedrooms	Bathrooms
Total Rooms	Furnished Sq.Ft.	Lot Sq.Ft.	Year Built	Latitude	Longitude

Table 5.1: Features gathered for each of the comparative study points.

and often searchers develop an understanding of the market as they explore rather than entering with a clearly delineated picture of the exact home they want. Spence and Tweedie provide an excellent, detailed explanation for why home buying proves to be an interesting data visualization challenge [153]. To develop the dataset I generated a list of 9 neighborhoods in the East End area of Pittsburgh. I chose the East End for two reasons: the study population as a whole will have the highest likelihood of even domain expertise for this area; and the study search area will plausibly resemble that of a young adult looking to settle down in a Pittsburgh urban neighborhood. Domain expertise is an interesting trade off for this study design. On one hand the study could focus on a city for which no participant would likely have any exposure. This would likely force participants to first make sense of the geography of the city, identifying downtown areas and attempting to encode different neighborhoods to memory. On the other hand, using Pittsburgh neighborhoods with Pittsburgh area participants (and choose high familiarity neighborhoods) means that participants will likely already have a mental schema for the neighborhoods. Many home buyers also have a degree of familiarity with the area they are investigating.

I sampled all 123 real estate listings contained within these 9 Pittsburgh neighborhoods, covering approximately 12 square kilometers of urban terrain. For each listing I scraped 10 data attributes from the Zillow listing as described in Table 5.1. I explicitly did not gather textual information such as real estate agent descriptions so as to focus on multivariate visualization. In Chapter 6 I will investigate the impact of review and textual information on consumer search visualizations. Excel participants received the data in raw form with a separate Pivot Table sheet pre-loaded. Tableau participants received a workspace with the dataset already loaded and several visualizations ready to explore that had been previous prepared by an expert Tableau user. DataSquid participants received a read-only save state showing a default starting view of the points charted by Sale Price X Neighborhood and colored by Neighborhood.

I sampled participants using a local psychological participant pool for hour long sessions in an on-campus lab reserved for user study research. All participants were required to be of at least 18 years of age, but no previous expertise with real estate or data visualization/analysis software was required. In each session I brought 4 participants into an office-like environment with either a desktop computer or a tablet and a desktop computer depending on their experimental condition. Participants were identified by ID number randomly assigned on entry by a collaborator. They received directions and completed all tasks through a computer-based survey, though the survey protocol was also printed as a reference (and for participant scratch paper). Conditions were assigned randomly using a separate table of pre-randomized participant ID to condition pairs. After completing a short onboarding survey participants received 10 minutes of training and 5 minutes of practice exploration time. Excel and Tableau participants watched training videos and DataSquid participants used an interactive walkthrough. During the practice period participants were asked to explore the data and were given sample questions to investigate such as "Which house has the biggest lot?" and "Which neighborhood has the largest homes?" These sample questions did not directly relate to measured tasks later in the protocol.

After participants had time to practice using the tool, they received the Task 1 scenario: "Alice is looking for houses with at least 3 bedrooms and at least 2 bathrooms. She wants a home that is moderately priced for Pittsburgh. She is willing to buy a more expensive place if it gives her good value for her money, or a cheaper one if it is a great deal," and were asked to give Alice two candidate homes and explanations for why they were good matches for her needs. This question considers three different attributes at once, two with hard constraints and one with a soft constraint. As I wanted to measure the time it took participants to explore and answer, I asked participants to move on to another question when they were finished. They received another scenario until the end of the time period that was not later evaluated. Participants had a maximum of 15 minutes to complete Task 1, and all participants were able to finish within that time period. In Task 2 (20 minute duration), participants were given the scenario: "Alice is looking for houses with at least 3 bedrooms, at least 2 bathrooms, more furnished square feet, good value (lower sale price per sq.ft.), and one that is in an expensive area. She wants a home that is moderately priced for Pittsburgh, but she is willing to buy a more expensive place if it gives her good value for her money, or a cheaper one if it is a great deal." Participants were asked to provide 3 suggestions with written explanations for why they chose those homes. This scenario contained two hard constraints and then 4 criteria to optimize. "Near expensive houses" introduces a geographic element that can also be answered by numerical means since Excel does not provide a map view. When complete participants were asked to move on, completing a post-survey asking about their experiences, confidence in their answers, and likes/dislikes.

5.3 Results

In total, 60 people from the Pittsburgh area participated in the study, allowing for 20 participants per condition. Of those 60, 3 (one in each condition) did not successfully complete the protocol. 34 participants identified as female and 23 identified as male. The youngest participant was 19 years of age, the oldest 61 years, and the mean age was 25.47 years old. As one might expect from a university participant pool, 45 participants reported they were full time students. 26 reported having a college degree, 11 reported having a post-graduate degree, and 20 claimed they either were currently students or had taken college courses.

2 participants reported having used Tableau prior to the study. All participants reported experience with Excel, with 30 reporting weekly use. 27 participants claimed they had previously used a touch interface tablet device prior to the study. Of the 57 qualified participants, 15 reported that their occupation involved data entry, analysis, or graphing. For each of the prior experience measures I checked to determine whether their were significant main or interaction effects with my dependent measures in my analyses. None were detectable. 49 participants reported having rented a house or apartment in Pittsburgh (not including college dormitories), 5 claimed experience in buying a house or condo in Pittsburgh, and 22 said they had previously used a web tool like Zillow, Trulia, or PadMapper. These past experience measures also did not demonstrate any main or interaction effects with dependent measures in my analyses.

In Task 1, all participants in the study chose two suggestions that met Alice's criteria. Exam-

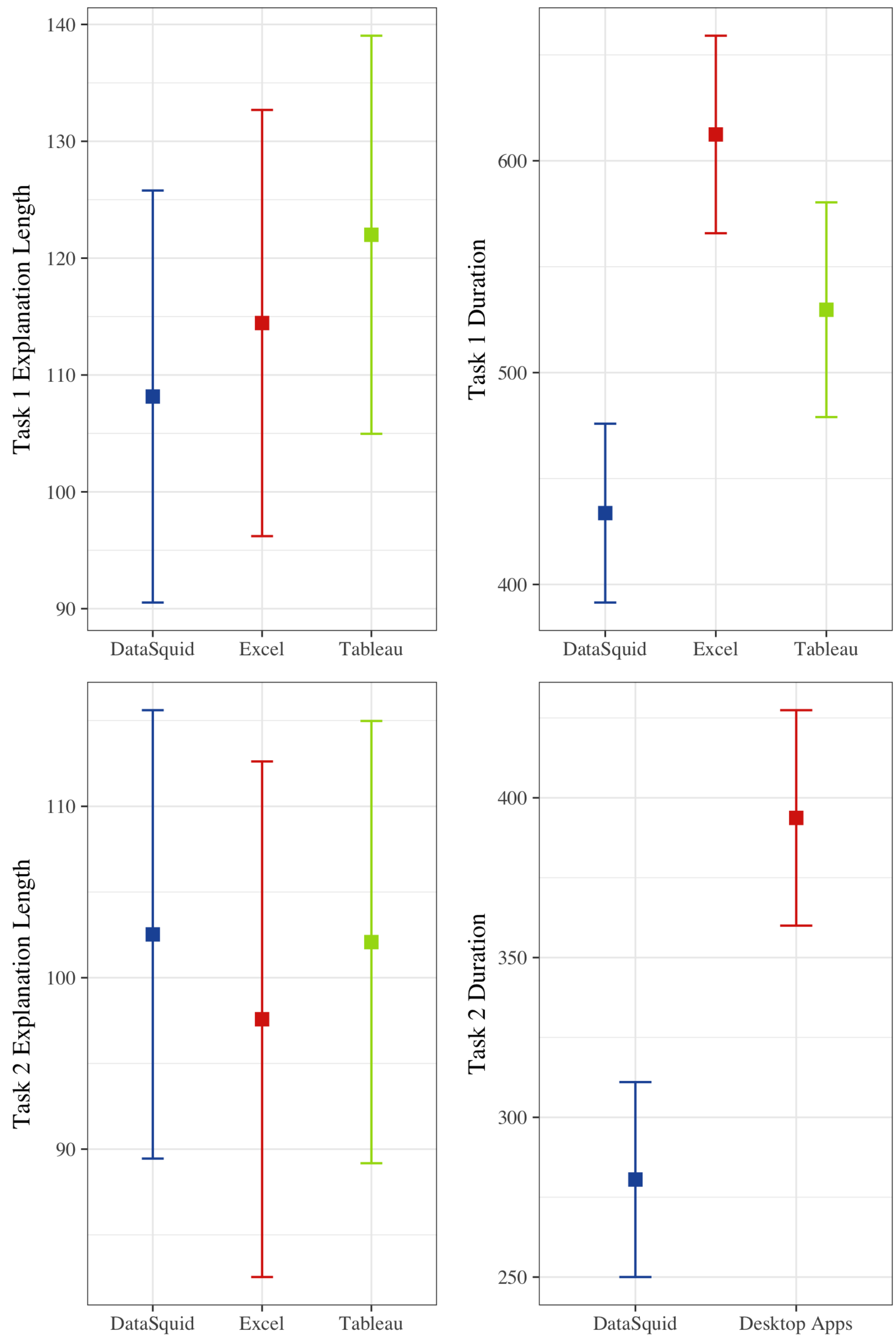


Figure 5.2: Task 1 and 2 - Explanation Length and Time Spent.

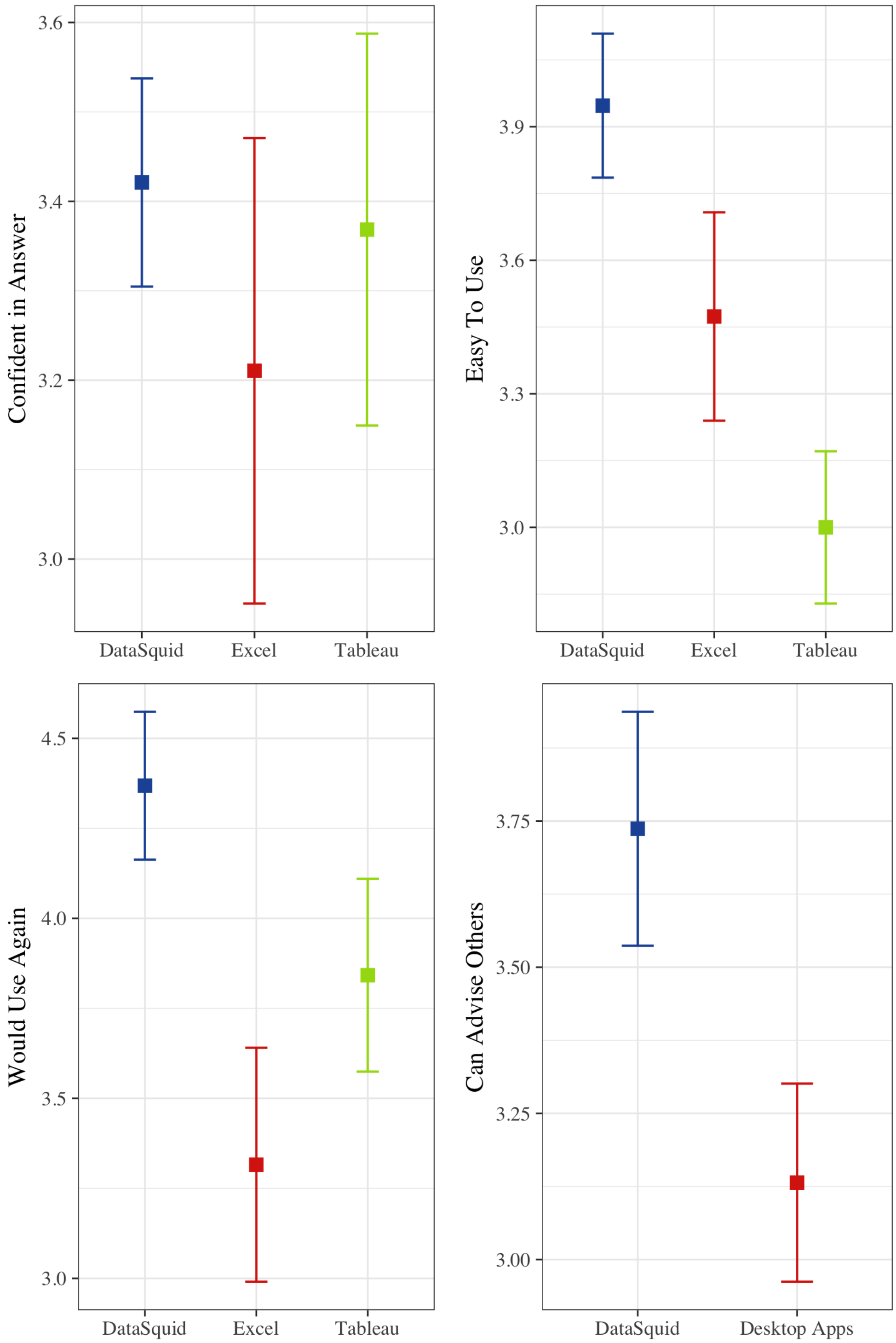


Figure 5.3: Post-survey findings

ining the price of the homes participants suggested as normalized to a range of 0-1 (so moderately priced homes occupy 0.5), participants in all conditions chose very moderately priced places (M:0.51, SD:0.12; F(1,55)=0.038, $p = 0.846$). When examining the descriptions participants provided there was no measurable difference in length between conditions (F(2,54)=0.155, $p = 0.857$), however participants took increasingly less time to complete the task from Excel to Tableau to DataSquid (Figure 5.2, F(2,54)=3.68, $p = 0.032*$).

All participants were able to complete Task 2 within the allotted time. I had three independent raters examine the dataset of houses to compose ranked lists of candidates that met the scenario requirements well. Using their 3 lists, I generated a master list of "correct" house suggestions composed of places that appeared in the top 20 ranks of all three expert lists. Rating participants' three home suggestions based on whether they appeared in the master list or not (0 or 1), a marginally significant trend suggests DataSquid users produced slightly better suggestions, however in all three conditions the suggestions were not exceptionally high quality (F(2,54)=2.668, $p = 0.079$). Additional qualitative examination may bear out more nuanced differences in suggestion quality. As in Task 1, participants wrote explanations that were not measurably different in length (F(2,54)=0.04, $p = 0.96$), however DataSquid participants completed the task faster than either Excel or Tableau (binning Excel and Tableau together as they had nearly identical distributions; Figure 5.2; F(1,55)=4.658, $p = 0.035*$). It seems that in each condition participants did not suffer from the expert/novice search effects that Punj and Staelin describe [130] since they took adequate time to supply quality results.

The Task 1 and Task 2 findings suggest an interesting overall effect. While participants ultimately provided similar suggestions and explanations, DataSquid users took less time to provide those results. Considering Klein and Ford's research on the impact of search time on customer satisfaction [93], DataSquid ought to have produced a more satisfying search experience for users than either of the other tools. Examining participants' post-survey responses, all participants left the study with indistinguishable levels of confidence in the answers they provided (F(2,54)=0.279, $p = 0.758$). However, participants reported that DataSquid was the easiest to use and Tableau the hardest (F(2,54)=6.109, $p = 0.004*$). Inverting those results slightly, participants were most interested in using DataSquid again, but reported more interest in trying out Tableau one more (F(2,54)=3.787, $p = 0.029*$). When asked how prepared they felt to advice a friend on Pittsburgh housing, DataSquid outperformed either desktop application (binning Excel and Tableau together as they had nearly identical distributions; F(1,55)=4.724, $p = 0.034*$). Participants' ease in using DataSquid aligns with the reduced task times.

5.4 Discussion & Open Questions

This study is encouraging in that it brought about the same motivational aspects of physics-based visualizations found in the original Kinetica study. Linking to consumer search literature, this means that such approaches might deliver a more satisfying and useful experience for users (as also evidenced in the post survey responses). Unlike in the original Kinetica study, the comparative study focused much more on individual points than overall trends. It remains an open question how physics-based visualizations may help users in the initial sensemaking phase of house-hunting or product finding (though the Kinetica study findings suggest that relationships ought to be more salient)

Perhaps unsurprisingly given its complexity, participants had a hard time coming to grips with Tableau. One participant wrote, ”it seems it'll take a lot of practice and to optimally use this software,” and another wrote ”I like all of the options that it gives, however I know that I am not fully using its capabilities. For example, I don't know how to integrate the filters well, which is why I had difficulty when called upon for multiple criterion.” Perhaps awareness that participants were not using the full feature set made them more interested in trying it again. Despite difficulties in working the software, participants were still able to produce adequate results. While a vast majority of participants reported using Excel regularly, few were terribly excited by it. One reported, ”It was hard to get it to do exactly what you want,” while another expressed excitement over Pivot Tables, saying, ”pivot table was a nice tool.” DataSquid users expressed a mix of opinions. While many responded with praise for interactivity, learnability, or usability, others reported issues with interactions (”It was really easy to accidentally hit somewhere on the screen with your finger.”), animations (”When I accidentally moved one of the points, I had to wait for it to go back to its original position.”, and summarizing data (”I couldn't figure out how to summarize or show houses once I added filters.”).

Thinking about consumer search more broadly, this comparative study brings up a number of interesting general questions (some of which will be explored later in Chapter 7):

- Given that DataSquid was designed as a general-use data visualization tool but most web-based consumer search tools are tailored to the product and dataset, how would a new tool based on the design principles of DataSquid but tailored more for consumer search perform?
- While participants found it useful, DataSquid's double-tap-for-details interaction is cumbersome and its modal popovers introduce delays in exploration. What better ways are there to surface point level details without overwhelming users?
- How can we better support comparison between points, a commonly cited desire among participants?
- Of the features in DataSquid, participants most liked observing point details, applying lenses, and putting points onto a map. How would these translate to a web-based tool to have greater potential impact?

While the dataset in the study contained 127 points, many other consumer-facing data sources will have far more to explore. There may be 127 restaurants in only a small corner of Manhattan. How should one represent data at scales where datapoints may not even be able to map to a single pixel, and how can this be done in a way that makes users feel confident in their results? Are there ways to handle scale that still allow for the benefits of physics-based visualization techniques? Moving forward in this dissertation, I turn towards Focus+Context techniques as one potential solution for handling scale.

6 — Point Salience Models: Multivariate Focus & Context

There is an inherent tradeoff in data visualization between faithfully displaying all data attributes and condensing information so that it better matches cognitive or computational capabilities. There may be too much data to reasonably represent or the data may be so dense that it is hard to render. A viewer may not be able to make sense of a display that includes millions of individual datapoints or densely packed datapoints because of perception limits (not enough cone cells in one's eye to perceive all points), cognitive limits (limited working memory and attentional capacity), or kinesiological limits (unable to interact with interface on a fine enough level to work point-by-point). Computationally speaking, large amounts of data pose a challenge in terms of efficiency ($O(n^2)$ algorithms begin to break down for large Ns), storage (trading memory for speed begins to become impossible for consumer devices), and hardware (not enough pixels on a monitor or cycles in a CPU). A dataset can reach human or system limitations surprisingly sooner than anticipated. A densely packed graph of 30 nodes may pose intense sensemaking challenges for users despite the small N, and some processing tasks (such as satisfiability, traveling salesman, or other NP problems) can be computationally intractable at even small data sizes. When applying physics-based visualization techniques in Kinetica, data scale also posed a new problem in that points could needless bunch up and confuse users at high densities (Figure 6.1.

There are a number of ways that information can be condensed so that its scale is more manageable. Unsupervised clustering, grouping based on proximity or occlusion, summaries, and query building all can help to reduce the cardinality of a large scale dataset. Motif simplification, generating composite measures of multiple data features, and summarizing data features can help to reduce high dimensionality. Focus+context displays apply a mix of cardinality- or dimensionality-reducing techniques to the periphery of a dataset so that there is more room for information in the focal point without losing sight of the data's broader context. Semantic zoom systems apply different techniques at different levels of zoom, packing or unpacking points based on the location of a viewport.

In this dissertation I focus on techniques that help to reduce the cardinality of large scale datasets by clustering and compressing the data within a pan-and-zoom sandbox (and which might be translated to other data visualizations). Compression can be a risky business, as reducing data scale is rarely a lossless process. Inevitably outliers, trends, and other data features will be incorporated and hidden within clusters or summaries. Introducing a schema for compression also adds cognitive load as now a user must make sense both of the data and the way it is compressed down. Compression may introduce an element of nondeterminism into the analysis process, since clusters or summaries may not be consistently represented across minor data changes, as well as a degree of uncertainty since summary and descriptive elements risk leaving valuable information out (especially if it is an outlier that does not conform to anticipated results) [7]. However, compression also helps to dramatically reduce data scale without switching to different visualization primitives at different levels of data exploration. While a compressed point may aggregate more than one row of data, it can be displayed and interrogated much like single point. Kinetica demonstrated the value of maintaining consistency in data presentation. The key question in this chapter, then is how to properly balance between complex compression schema that faithfully represent data but may be hard to users to chunk into understandable elements and simpler schemas that risk losing a bit of information in exchange for easier sensemaking.

I point to one particular way to manage this balance between complex-but-faithful and simple-but-incomplete representations of compressed data. Focus+Context views encode a general approach for devoting resources (screen pixels, bits of information, or computational capacity) to places where extra detail may be necessary for users to complete a task or to make sense of data at the expense of detail elsewhere [52]. These techniques are effective but risk pushing the user to tunnel to a particular conclusion or answer. If the focal point is purely user guided, then the user may continue narrowing their focus until they completely elide another valuable part of the dataset. On the other hand, if the focal point is purely dictated by data features, the user has no agency in the exploration process and cannot easily apply their own cognition towards the analysis process. Focus+Context views have been generalized under Degree of Interest (DOI) functions [53]. DOI functions broadly describe Focus+Context views in terms of what makes a point interesting or not. This allows visualization designers to capture new behaviors like distributed spheres of interest [53], particular user interests in data [166], and higher order structures like clusters [147].

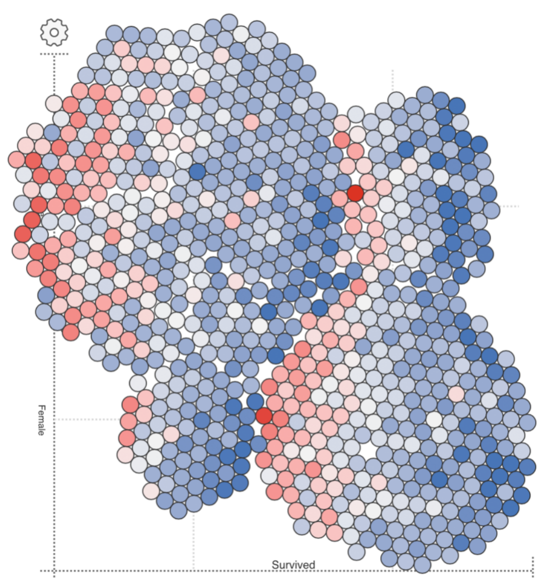


Figure 6.1: Physics-based visualizations break down when too many points are onscreen. Charts and walls can handle a fixed amount of points before overlap makes them too hard to interpret.

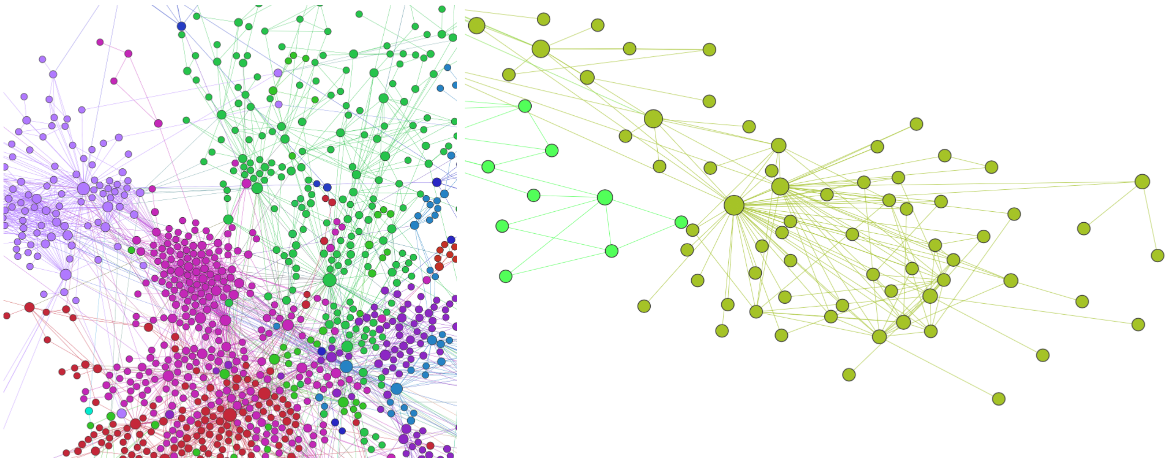


Figure 6.2: Even when examining a small portion of a graph of Twitter hashtag co-occurence, the density of the dataset can make it difficult to identify graph features such as cliques and stars even after applying a layout algorithm.

I propose to construct Point Salience models that balance both user- and data-level features when curating focal and contextual information. For instance, if the user has demonstrated specific interest in a dimension of data either by specifying it in an interface or through their analysis behavior, a Point Salience model will weight points that are outliers or otherwise meaningful along that dimension higher than others. However, if another point has unique data features that make it exceptional overall but perhaps remain uninteresting within the user's narrow interest area, the Point Salience model ought to surface it as well either to act as a counterexample or a new avenue of exploration. Russell et al.'s sensemaking model [137] suggests that people exploring a dataset need to gather examples that match their hypothesis or mental model of the data to refine it while at the same time also gathering counter-examples that may disrupt or disprove their hypothesis. The user level features in my model then capture this reinforcing effect while data level features help to guarantee that users do not see only confirmatory results

Figure 6.3: Van den Elzen and Van Wijk [165] apply motifs to a dense visualization in order to make the large trends more salient and understandable, but at the potential cost of hiding smaller trends.

and get a stream of valuable oppositional or orthogonal features. One key aspect of this approach is also that Point Salience models should not explicitly hide points. Since one is estimating interestingness through user weights imprecisely, the risk of users missing a critical point is high. Instead, I propose that the model primarily used as a way to manage how data are grouped or compressed and how aspects of data are prioritized in summaries and displays. The model presents several challenging design questions. What is the right balance of data vs. user interest? How should users encode their interest (or should it be inferred)? How does one surface the model weighting to the user so that they understand why points were shown or compressed?

Additionally, models built upon an entire dataset may mute out local features in exchange for extreme examples. For instance, a Point Salience model for a participant trying to identify high diversity workplaces across the country may fixate on extreme examples of large workplaces that demonstrate generally high levels of diversity. However, there may also exist places that, while not extremely diverse, are remarkably different than other places in their community. These may be more valuable examples to the analyst as they could demonstrate strategies for countering entrenched or otherwise prevailing issues in a community, versus a workplace that just happens to be large and diverse in a generally accepting community. While, of course, one could build a quantitative data feature that captures this distinction, a person may not arrive at the dataset with that kind of knowledge. If their exploratory interface directs them immediately to a particular kind of example, they may never realize they ought to look for that distinction. In applying Point Salience models to datasets, I propose weighting data on a local rather than global level by integrating a schema that captures ”locality” in a dataset. If the data are geographic, regions of geography may be the most natural way to capture locality. If the data are numeric, they may be key orientating features that cleanly split different categories of data (for instance species in the classic Iris dataset) inferred algorithmically or integrated with the help of domain knowledge. Ideally, models of locality also ought to function at different scales of data. For instance, if a user initially explores country-wide data but then wants to move into census blocks, a system ought to break apart the original localities into representations that better fit the new scale.

In addition to capturing smaller scale features within the data, considering local features also brings about another beneficial property in the sensemaking process. Because users must take their findings and integrate them into more generalizable schema [126], providing some initial hints as to ways that users might schematize their data ought to help ease this process. By surfacing the *clusters generated to capture local features* as well as the local features themselves, one might naturally provide an initial schema upon which the analyst can hang inductions. For instance, the clusters used to capture local trends in the participant's diversity search might align with neighborhoods or city blocks which can trigger preexisting conceptual units that are familiar to the participant. Instead of being forced to categorize and aggregate their point-level findings completely, the clusters might provide hooks for initial groupings to start the integration process more smoothly.

In particular, Point Salience models build upon existing Focus+Context and sensemaking research by:

- Introducing a hierarchical clustering pass to the focus+context process (akin to Schaffer [147]) that is designed to both spread information evenly across the screen and provide a

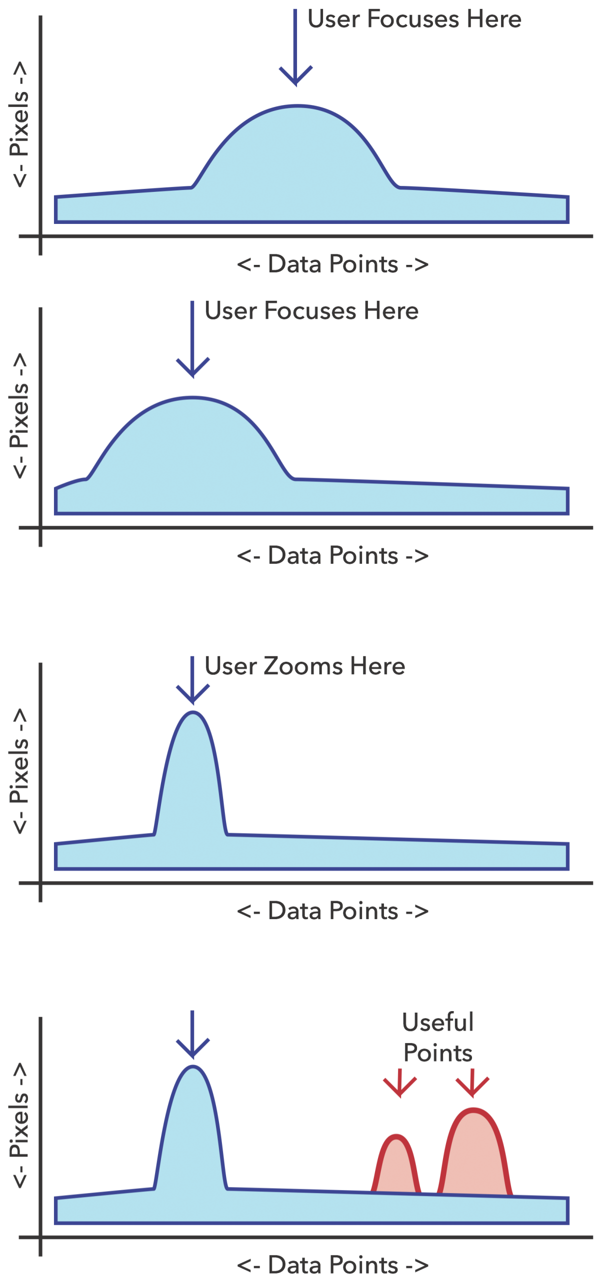


Figure 6.4: Visual representation of a focus+context process. On the X axis is a spread of data, and on the Y axis are the number of pixels devoted to each data point. As the user moves their viewport, more pixels are devoted to a different region of the screen. As they zoom, pixels focus into a smaller segment of points. However, perhaps there are data features that are a priori of interest. If they are not given enough pixels (red bubbles), the user may completely miss them in their exploration process.

scaffold upon which for users to develop initial sensemaking schema

- Developing a DOI function, applied after this initial clustering, that incorporates weighted measures of user interest and a priori data saliency in multivariate data that uses considers both global- and cluster-level features to show more relevant trends between conceptually (or geographically) adjacent points
- Building a new visual metaphor for representing both cluster-level and DOI-mediated point-level multivariate data by outlining clusters and selectively compressing points based on DOI
- Applying physics-based visualization techniques to the focus+context metaphor to disambiguate densely packed focal areas and more fluidly transition between changes in zoom (i.e. cluster level) and user interest (i.e. DOI function).

In this section I will first outline related research on representing large scales of data. Afterwards, I will formally describe the implementation of the aforementioned technical contributions. Applying Focus+Context views to large datasets can be risky because compression inevitably may hide useful data features, even if the algorithm is tailored to try to present them to the best of its ability. As a result, users may be left with an incorrect or biased view of the data, especially as more and more data are combined and compressed. In the final section of this chapter I will examine these risks, focusing on the following guiding questions:

- Does mixing focus and context cause users to have an unrepresentative view of data as they explore?
- Do users accurately interpret groups of compressed points as they explore a dataset?
- Does hiding points confuse users as they explore, and can they track how changes in model alter their view of data?

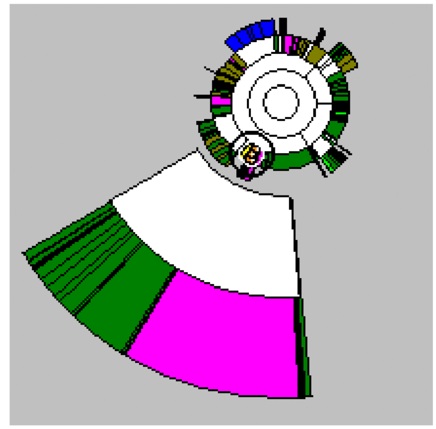
6.1 Scaling Up Visualizations

There are a number of approaches researchers have used to confront problems of data scale in visualization systems. At its core, the challenge is that data can grow to a scale where even mapping data to single pixels, using opacity, or applying jittering are not effective. Moreover, a user may have a hard time understanding data well before the display is saturated. In this section I will consider first two major approaches for presenting large amounts of data effectively: using Focus+Context techniques to limit the load on a user and introducing intelligent querying systems to help users better winnow and slice. Afterwards I will survey existing visualization systems that make efforts to gracefully handle large scales of data, identifying key unifying and differentiating features.

Focus+Context

In human vision, the eye has a small focal area with high acuity and a much larger peripheral area with low acuity. This trade-off allows the eye to focus on important areas while retaining information about their surroundings to identify future points of interest. Similarly, one can present data in a way that focuses on a particular detail (or allows the user to choose a focal

(a) Penfield's somatosensory homunculus devotes more size and detail to areas of high nerve density on a human body.



(b) A focus+context view of a file hierarchy in Sunburst [154]. The focused element is aligned with its context in the main radial view.

Figure 6.5: Example focus+context views.

point) and presents the contextual information in a reduced or less salient way. This allows a user to avoid being overwhelmed with detail, while still understanding the global context of the dataset. When dealing with overwhelmingly massive amounts of data, such detail-limiting approaches are especially important. In an quantitative evaluation by Baudisch et al., focus and context display provided users a significant time savings over traditional overview and detail displays [9]. While I will explore a selection of systems in the following subsection, I would point readers to Cockburn et al. [36] for a more lengthy review of approaches.

Early Focus+Context systems have acted much like a lens by applying a degree of interest (DOI) function on top of a traditional visualization. In an early example, Furnas applied a fisheye DOI function to visualize graphs [52]. DocumentLens simply magnifies a small textual section of a large corpus while gradually shrinking the rest, faceting the DOI function for flat textual data [134]. The hyperbolic browser more discretely uses a fisheye lens analogy to present hierarchical graphs [100]. By clicking around the graph, the user can refocus on different areas of interest. Sarkar and Brown further standardized this approach, describing different lensing techniques (i.e. Euclidean vs. Manhattan geometries) and general approaches for graph visualization [143]. There is also room for other physical metaphors. Sarkar et al. used the idea of stretching rubber to selectively expand interesting regions of a visualization, while shrinking others [144]. In the Sunburst system for exploring file structures, the authors experiment with different ways to show focus and context in a radial diagram [154]. Interestingly, the authors show success in both traditionally expanding the radial segments into space as the hierarchy is traversed, and also by inverting the visualization so that the overview shrinks into the center of the radius. This points to the need to explore different ways of showing focus (that may even break with a strict DOI function) based on the particular visualization and data being used.

In a follow-up 15 years after his original work, Furnas set out several new goals for displaying focus and context [53]. He proposes a more holistic model for focus, past simple DOI functions. To properly match how humans explore data, it is important to consider the salience of individual elements. Even if a particular piece of data isn't in the central focus area, its important as an outlier or high degree node in a graph might suggest that it should still be quite visually salient. This aligns with a quantitative evaluation by Pirolli et al. which suggests that users shift to focal areas in short hops following information scent [127]. Furnas emphasizes that navigation and deeply encoding focal points into memory are necessary goals as visualizations scale. We can see this sort of approach emphasized in more recent Focus+Context displays. DOITrees uses traditional fisheye views, but then uses user input to manage how interesting or disinteresting parts of the visualization appear, curating the focal area [70]. Magic Volume Lenses identify salient features in 3-dimensional volumetric data, emphasizing their size when projecting onto the screen [169], while ClearView only shows certain focal features, using opacity to exclude confusing other features [98].

On the other hand, another way to achieve focus and context is to support interactive panning and zooming into visualized data. Pad++ is one such example, allowing users an infinitely zoomable surface to encode information [15]. Users can see an overview simply by zooming out, but expose more and more detail as they zoom. Unnecessary information is simply elided as it becomes to small to render. A physical analogue to this is a system developed by Yost et al. that uses very high resolution displays [184]. In this case, the user must approach the display so that they have the visual acuity necessary to view small labels and details. However, one problem for such zoomable environments is maintaining context while navigating. It is easy to lose perspective on one's zoom level and location in the space with few navigational cues, a situation known as ”desert fog” [85]. In effect, context may be lost when focusing. Baudisch and Rosenholtz propose an interesting solution: use the edges of the display to show traces and halos of peripheral entities on the space [11].

In general fisheye views seem to improve user performance but users rarely prefer them [10], even potentially on small devices [59]. However, when compared on equal footing, fisheye views were preferred for a small screens because of potential desert fog navigation problems when zooming [22]. The small screen space of mobile devices also poses problems for interactions. Touch interactions are overloaded, and specifying areas of interest for fisheye views may be nonintuitive. Little work has been done on alternative ways to navigate an environment beyond the now standard two-finger pinch/pan gesture (though 3-space [2] and flex [23] gestures have been explored). There remains much investigation into ways to blend fisheye views, zoomable interfaces, and overview+detail in mobile and multitouch environments.

QUERYING

Another way to explore large amounts of data in detail is to effectively query for different data features. Often large amounts of data are stored in database structures that make constructing queries an effect mode for exploring datasets. However, this can be challenging partly because complex queries may be computationally intense, but more so because constructing and displaying queries on data can be un-intuitive and hard to visualize. A body of research has explored ways to handle large scale data through querying interfaces.

There are several simple forms of queries that can reduce data complexity. Subsampling data randomly or in a structured fashion can give users a smaller, representative sample in the best case that will be easier to examine [16]. Filtering data using dynamic query sliders can cut off portions of data to prevent overload [4]. Another approach is to model or cluster the data, showing only trends rather than individual data features [114]. Finally, based on the current display mode, data can be placed into bins so that single entities in the visualization correspond to groups of points [43]. Binning can be particularly effective, as it maintains outliers while also cutting through very dense regions of data [111]. However, binning is also quite computationally expensive, as in the ideal case data need to be binned into an N-dimensional hypercube corresponding to the number of features in the data. As a user explores, the hypercube must be post-processed to collapse bins into the dimensions current displayed/manipulated. imMens, Profiler, and Nanocube are examples of systems that use a variety of optimizations to quickly and effectively evaluate binned data [87, 110, 111]. imMens goes one step further, allowing users to query over the bins using brushing.

In the database realm, there are a number of systems designed to support constructing and visualizing complex queries. Tableau uses a visual drag-and-drop system to construct queries on single of multiple data tables and visualize the results [157], while VizDeck lets users mix and match small traditional visualizations into a dashboard [90]. For very large datasets, visual results may take hours or days to receive. sampleAction runs queries on smaller amounts of data, providing feedback and confidence bounds before contributing computational resources for long periods [48]. This idea of progressively refining results ([136]) can also extend into visual appearance, for instance producing charts that devote computational resources towards adding detail only where the user needs it (or the device can reasonably display it) [155].

Assembling queries can often pose a challenge. SeedDB uses post-processing to match query results to the proper visualization, such as line or scatter plots [123]. The Wrangler system uses plain English phrases to decompose common database actions [86]. DataPlay instead uses a visual language and flowchart to support iterative query development [1]. This challenge becomes even greater when using a gestural interface. Without a keyboard or pointer, building complex queries requires careful interaction design. Nandi et al. have done initial explorations into this area [116]. They developed a palette of gestures for database operations, such as joining tables by dragging them together with multitouch and aggregating by pulling labels to tables. Jiang and Nandi also implemented a means to snap gestural controls to values obtained during query execution, such as proper increments in a nested WHERE statement [83]. The challenge moving forward in this space is delivering satisfying, natural feedback as queries are generated. How does the interface respond once a user successfully combines/reorganizes/filters their data? Is there a way to portray the procedure of a JOIN or GROUP BY visually so that users can track data as it mutates during a query? Further, when data scales grow very large, is there a way to show users how much they are increasing or reducing the size of the data they manipulate?

6.2 OVERVIEW

One challenge when schematizing or compressing a large number of points is choosing where and how to group, combine, summarize, or eliminate points so that the amount of data falls within computational or cognitive capabilities. Focusing on global features across an entire dataset runs the risk of smearing out valuable information shared between points on a local or less intense

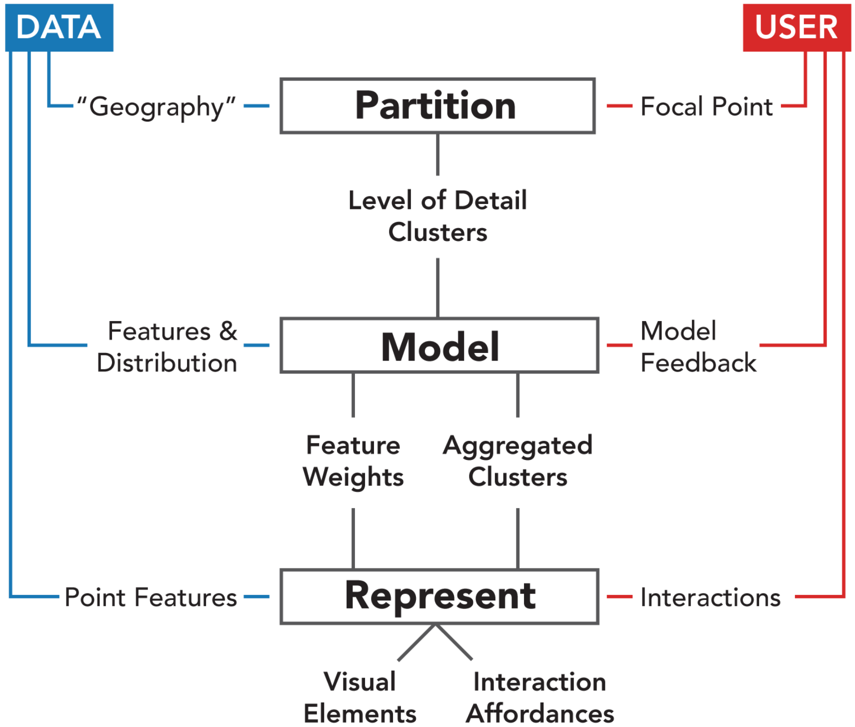


Figure 6.6: Overview of data and user interactions used in the Fractal interactive visualization

level, but examining only small-scale details risks ignoring relationships across the entire dataset. One's task, then, is to balance the trade-off of showing more information so that nuances in the data are maintained while still compressing or schematizing the data. Additionally, the cost for a user in understanding the compression technique, especially if it changes as they explore the data, must be balanced.

In the following sections I will outline the technical details of the Point Salience models, shown in schematic view in Figure 6.6. The high level goal of this model is to support both the *foraging* and *sensemaking* loops in Pirolli and Card's sensemaking model while at the same time reducing complexity when data are too numerous or too dense to properly represent on a screen. The model can be divided into three layers:

- Partition (§6.3) - Build a hierarchical clustering model along two *conceptual dimensions* of the dataset, ignoring user interest or a priori data feature importance.
  - These dimensions should ideally align with anticipated ways a user might schematize or model their data. For instance, if the data have a geographic element, latitude and longitude are natural choices because they leverage users' preexisting understanding of block, neighborhood, city, region, and country. Lacking geographic data, other conceptual dimensions like computer monitor screen size vs. resolution or breakfast cereal calories vs. protein may instead be useful ways to subdivide.
  - For a given user viewport onto the data and having built a hierarchical clustering

model, traverse down the hierarchy in a breadth-first manner, breaking apart the hierarchical clusters one by one into increasingly small partitions of the dataset. After each break, check to see whether the amount of screen space vs. number of onscreen partitions has hit a desirable ratio (i.e. a number matching user working memory or a reasonable percentage of screen space per partition). This iterative deepening means that partitions have a comparable amount of *screen size* to each other due to the way hierarchical clustering applies a distance function, but does not necessarily mean that partitions have comparable *cardinality*.

  – While Schaffer et al. [147] use hierarchical clustering to decide which view elements to expose, they do not use a mixed model as described here and also incorporate a more complete focus+context DOI function at this stage. The goal here in my model is not to fully achieve a focus+context visualization. Instead, it is designed to make sure that local-level features and detail are allocated evenly across the screen into units that might later scaffold the *sensemaking* loop where users attempt to gather, categorize, and generalize their individual discoveries [126].

- Model (§6.4) - Define a Degree of Interest function that takes into account weighted a priori data interest (points that have unusual values along important dimensions are weighted higher) and weighted user interest at the given moment (points that one expects to be of interest to the user) through a geometric mean function aggregated over one or more rows of data since a representation may in fact be several points collapsed together.

  – Compute pairwise distances within each partition. The distance function balances Euclidean/orthodromic distance along the hierarchical clustering's *conceptual dimensions* with the output of the DOI function. If the partition is too dense (i.e. too many visual elements per pixel area in the partition's bounds), then begin combining the closest elements until an ideal density is achieved. Note that because the DOI function is taken into account, points that are both especially interesting and different from each other but very close together may remain separated while other, further points are bundled. Since partitions may have uneven densities of points (their only rough guarantee is that they take up similar amounts of screen space), some may experience more pointwise combining operations than others.

  – This aligns with Furnas' [53] and Van Ham & Perer's [166] use of User factors in a DOI function. However, instead of incorporating distance into the DOI function and applying it globally, my model applies DOI as a component of distance nested within the partitions themselves. This guarantees that partitions do not overflow with too much detail, but maintains the evenly-spread local features from the partitioning process. This step primarily aids the *foraging* portion of Pirolli and Card's sensemaking model by compressing examples judged to be less useful and making interesting examples more obvious and interpretable.

- Represent (§6.5) - Display the partitions in schematic form so that they frame the individual partitions of the screen. Display the aggregated datapoints within each partition, marking elements that contain more than one point (as a result of the pairwise combining process) so that the user knows they are conceptually different from individual point representations. Use the DOI model to color, scale, and highlight/lowlight points that might be relevant to the user at a particular moment.

– Now with few enough points onscreen, a Kinetica/DataSquid physics engine can use collision to manage the occlusion of points and smooth data exploration. As a user zooms or the user component of the DOI changes, the physics model can help to tween and animate the changes in how points are grouped. Because the hierarchical clustering powering the partitioning process does not take into account the DOI function, it acts as a consistent frame upon which the points transform during exploration.

For the purposes of interactive data visualization in a pannable and zoomable environment on a web browser, the area of the viewport is a useful comparison point for judging whether data is too dense or too sparse. If there are too few pixels reserved for each data point, then more reduction might be needed, whereas if there are very many pixels for each data point then there is spare room for more data representations. While cognitive factors also influence the decision of too much or too little data onscreen, this objective function balancing pixels per datapoint represented can be tweaked to stay within sensible limits. For instance, if initial testing suggests that users inspect one eighth of the screen at any given time, then the points in that region should scale within perceptual and working memory limits. Because these factors depend very heavily on the kind of data representation and how users chunk the data representations, I do not define an ideal factor of pixels per data point. Instead, I urge iterative user testing in order to establish bounds for the level of detail objective function during the visualization design process. This process was employed throughout the development of a prototype tool described in the following chapter.

6.3 MAINTAINING CONSISTENT LEVEL OF DETAIL

One way to maintain a consistent level of detail across a visualization is to compute a global pixel per data ratio and then combine or break apart visual representations until a desired limit is reached. For instance, if a user can reasonably perceive one piece of data for every 10 pixels onscreen, then data might be compressed until there are few enough elements to hit that goal. However, this process risks unexpected output for users since depending on the viewport size any given set of points may be aggregated or split. Figure 6.8 illustrates this disruptive effect as in one view a southern outlier point is visible but in the zoomed in view it is has been hidden away. The user's model between zoom steps is highly inconsistent and confusion results. In order to better capture local features, especially in cases where data charted on a map, it is helpful to have a model that will balance the need to collapse points across the screen with users' need to maintain a consistent mental model across combine/split steps. In this section I propose introducing an intermediate representation for the data that smooths transitions as points are combined, split, or schematized by grouping them into a small set of partitions that combine/split during browsing. Instead of aggregating individual points on a global basis, each partition can aggregate points in parallel based on local criteria until a local point per partition area goal is achieved. The

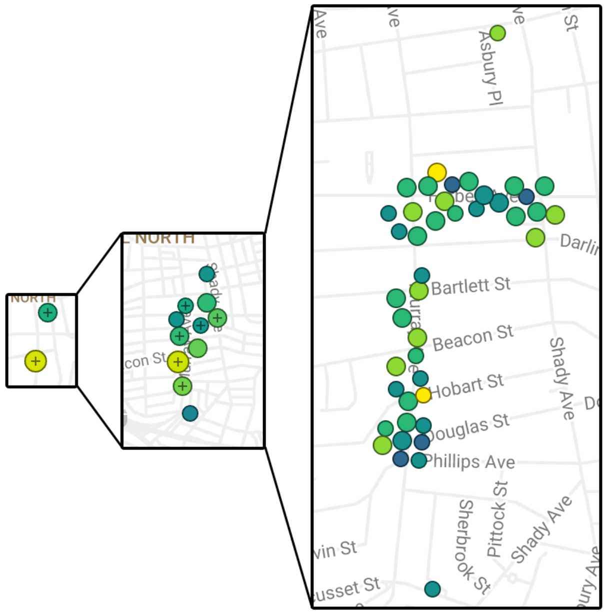


Figure 6.7: A user zooms in on a set of points charted on a map. Initially there is not enough screen real estate to show all points, so they are grouped (+ sign). As the user zooms the points are split until every point can be represented.

Figure 6.8: A search for AirBnB rentals in Florence reveals a dense area of potential candidates. An algorithm attempts to highlight meaningful examples and hide less relevant ones, but occlusion still results. As the user zooms from the left image to the right, the representation becomes inconsistent, and the outlier south of the river disappears.

intuition here is that the features that make a point "interesting" in one partition may be entirely different in another, and operating on a per-partition basis rather than a global basis will better highlight local differences and explain why they matter. These partitions also use an objective function, combining and splitting based on how much screen area they encompass. For consumer search tasks that involve geographical components, geography is a natural candidate for this partitioning representation since people already tend to partition geography into units such as cities, neighborhoods, blocks, and buildings.

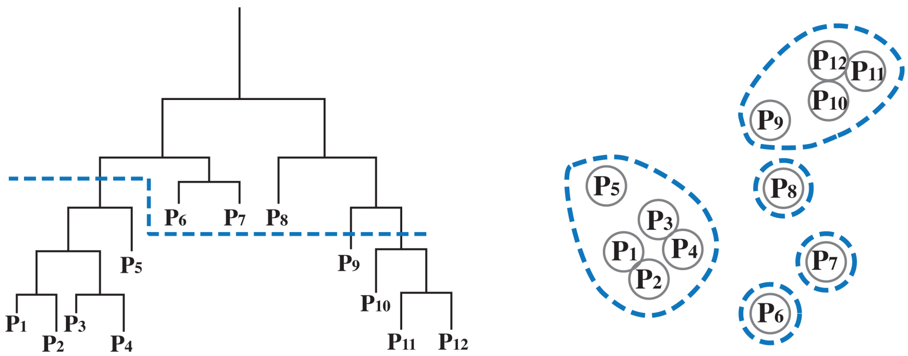


Figure 6.9: A simple hierarchy (left) for a set of points (right). The system has broken apart hierarchical clusters using iterative deepening until a sufficient level of detail has been achieved (blue dotted line), resulting in a set of two groups of points and three independent points.

6.3.1 Partitioning

Consider a set of points charted onto a zoomable and pannable map. At the highest level all of these points can be placed into one group encompassing all data. This is analogous to the global case where every point is compared with every other point, reducing complexity by aggregating points until the scale/density is sufficient. As the user zooms in on the map, points are not spread evenly. Instead, they concentrate in busy parts of the city or in quiet neighborhoods. As they zoom even further, they break into separate commercial blocks (see Figure 6.7). These conceptual divisions in geography are shared by most people in urban societies, though mediated by experience and socialization [122].

I employ hierarchical clustering to achieve a similar result to these conceptual geographical units. A hierarchical clustering model compares points in a dataset using a distance metric (recently researchers have employed hierarchical clustering to spatially aggregate large datasets to provide coarse-grained visual representations [186]). I use the orthodromic distance between latitude and longitude components of points (which is slightly different than using Euclidean distance as latitude and longitude map to a spherical space rather than a flat one). The model does not employ any other data at this time, as the map/geography itself matches the cognitive model many users start with. Introducing other features at this stage of clustering might mean that the resulting clusters diverse significantly from natural boundaries in exchange for other data similarities (which would prove disruptive against a map of points). Starting with the points that are closest in distance, the hierarchical clustering algorithm builds a dendrogram of point

```
generate a heap that sorts first by minimum node depth and then by maximum
  objective-function(node);
populate the heap with the root node of the hierarchy;
while heap contains nodes and not achieved desired ratio of partition area vs. screen area do
    extract the highest priority node from the heap;
    for each of the node's children do
        if the child is a leaf node then
            add it to the final list of partitions;
        else
            add it to the heap;
        end
    end
end
add the remaining heap contents into the final list of partition;
```

Figure 6.10: Pseudocode for splitting hierarchical model to show an ideal level of detail in a given viewport/zoom. (Note that nodes that are offscreen are still processed but not factored into the area computation.

relationships (Figure 6.9). The node at the top of the tree corresponds to a partition containing all of the points, and groups of points are gradually broken apart as one moves down the tree.

The hierarchical clustering model encodes geographic relationships between points at varying levels of detail. One can traverse down the tree breadth-first in order to create more partitions that have similar areas (but not necessarily similar cardinalities since the hierarchy splits by distance and is not necessarily balanced). In practice, this process can produce meaningful partitions at varying levels of zoom. To better split partitions, one can employ an objective function for nodes on the hierarchy that characterizes how desirable it would be to split them up (i.e. if a particular partition has many points that match a user's interest, it may be better to split than one that is wholly uninteresting). The pseudocode for splitting apart hierarchical clusters to achieve a particular level of detail when given the root node of a hierarchy is outlined in Figure 6.10

In practice this nesting maps reasonably well to natural geographic features in the dataset (in the future I hope to investigate just how well these clustering models map to users' mental models of city topography). Figure 6.11 illustrates how the hierarchy maps to geography, showing how successive passes of the splitting algorithm will increase the number of partitions and decrease their area. One can either terminate the algorithm by achieving a set percentage of screen area covered in partitions or by setting a specific target number of onscreen partitions. Incorporating a objective function into the splitting rather than breaking by iterative deepening might help to prioritize information gain for the user, breaking much farther down certain parts of the tree than others if they have high expected values.

6.3.2 Level of Detail Within Partitions

The hierarchical model provides a method for identifying a set of regions onscreen that share features that users often use for chunking (in this case, geography). Each of these regions contains

Figure 6.11: Grey outlines show the boundaries of nodes in the geographic hierarchical clustering model. Moving from top to bottom, the view zooms in on downtown Pittsburgh, focusing in on successively smaller groups of points. Notice how the nesting behavior of the dendrogram provides a wide variety of partition sizes.

```
compute target number of visual elements for given partition (representation area vs. partition
  area);
while partition contains more than the target number of elements do
    identify the pair of data points (A, B) in partition for which distance(A,B) is the minimum;
    create a new aggregate point, C;
    if A is an aggregation of points then
        add A's contents to C;
    else
        add A to C;
    end
    if B is an aggregation of points then
        add B's contents to C;
    else
        add B to C;
    end
    add C to the partition and remove A and B from the partition;
end
```

Figure 6.12: Pseudocode for aggregating elements within a partition until a desired level of detail is reached. The algorithm greedily combines data points in a pairwise fashion using geographic and model distances as outlined in this section.

any number of points as the splitting algorithm focuses on the total area of partitions onscreen versus viewport area. A relatively empty region that spans a wide area is treated the same as a dense partition occupying the same amount of screen real estate. Initially this sounds paradoxical, but the insight here is that the partitions are designed to help users track regions of the dataset as they explore, zooming and panning. It is more important that the regions appear comparable and that one is not extraordinarily more salient than another (i.e. there should not be four regions in a city downtown and another region capturing everywhere else, leading users to potentially make false conclusions about the contents of each).

Partitions may contain thousands of points crammed into a small area. There must be, therefore, a technique for maintaining an adequate level of within-partition detail. Analogous to the generation of partitions for a given viewport level, in order to keep the level of details within partitions even I also employ a balance of data points versus partition area. If there are too many data points to display successfully within cognitive and computational limits, then they must be aggregated and compressed together. This poses a challenge since points cannot be aggregated in any way. One could certainly apply the hierarchical model, bundling points together in a bottom-up fashion from leaf nodes until there are few enough points. However, this approach neglects to consider the data features of the points. While it was important for partitions to remain grounded in a common frame of reference, points within a partition ought to be aggregated in a way that makes user- and data-level features more salient (the focus+context aspect of the visualization). To achieve this I propose incorporating the Point Salience model.

The general pseudocode for the combination algorithm is outlined in Figure 6.12. The key operator in this algorithm is the distance function that characterizes how similar a single data point, s is to another (or a group of others), t. I define distance as:

$$d(s,t) = \min_{\forall p \in t} \left(\alpha \, d_{geo}(s,p) + \beta \sum_{a \in M} d_{model}(s_a, p_a) \right)$$

where M is the set of the Point Salience features, α is a scalar factor biasing for or against geographic closeness, and β is a scalar factor biasing for or against model closeness. Distance, therefore, is a balance between geographic and model closeness for points. Points that are very close together or very similar in data characteristics are more likely to be combined, however the model can also incorporate negative weights which serve to increase distance if items are similar (e.g. a cases where one wants to be sure every instance of a particular category is present and separated rather than bundled together).

Applying the splitting and combining algorithms together, this produces a set of partitions and visual elements (corresponding to data rows) that have an ideal level of detail for a given viewport. As the user focuses in on a smaller number of points the algorithm has more headroom to devote detail to individual points. As the user focuses in on a large number of points, the algorithm necessarily compresses points together, starting with ones that the Point Salience model suggests are least salient.

6.4 Generating and Applying Point Salience Models

The aggregation of points presupposed the existence of a model that captured both data similarity and important factors for the user exploring data. In this section I will describe one way to formulate such a model. The core aim behind this model is to reasonably capture what might make a point *salient* for a data explorer. There are likely any number of formulations which capture different aspects of saliency which I will consider in more detail in the concluding chapter.

At its core the Point Salience model distills into this formula:

$$weight(p) = \alpha \left[\prod_{a \in M_{data}} w_{data,a}(p)^{\gamma_a} \right]^{\frac{1}{\sum_{a \in M} \gamma_a}} \times \beta \left[\prod_{a \in M_{user}} w_{user,a}(p)^{\delta_a} \right]^{\frac{1}{\sum_{a \in M} \delta_a}}$$

where $a \in M$ captures attributes that are currently a part of the Point Salience model, γ_a and δ_a weight particular user-level and data-level attributes (for instance if the user has specified more interest in one attribute than another or interaction logs suggest more interest), and α and β bias the model towards data or model attributes. The formula combines the weighted geometric mean of both data and user characteristics so that no one attribute swamps the overall weight should data ranges not be identical. Within both models a weight value is computed for a point with respect to a particular attribute. This weight might be calculated at a global level for elements such as coloring points which affect the whole screen while it might also be calculated on a per-partition basis to capture local features such as the best exemplar in a region. To compute model distances between points, an analogous formula is used with Euclidean or Levenshtein distances substituted for w_{data} and w_{user}.

6.4.1 Modeling Useful Data Features

Before the user has done much exploration or expressed any interest using the interface, features within the data can already suggest some points are more salient than others. For instance, points may lie far outside of the distribution for a particular attribute, suggesting they are unique outliers worth noticing. For categorical or ordinal features, points may have uncommon values (e.g. the only restaurant of a certain genre in a city). In practice I have found that for numeric features measuring how many standard deviations away a point is from the mean provides enough signal to generate a weight. For categorical variables outliers can be identified by inverse document frequency, meaning that members of categories with few elements are weighted higher than those with many elements.

If we were to treat every single attribute as equal, then there is a high risk that an attribute that is not particularly valuable may have an overly large effect on the eventual model weight. As a result, we also computationally develop per-attribute weights, γ_a. There are any number of ways to identify attributes that effectively split a dataset or convey the most information gain [60]. For the purposes of this initial exploration, I estimate the weight of a numeric attribute by prioritizing attributes that have little correlation with any other attributes and then prioritizing attributes with more normal distributions. For categorical variables I prioritize attributes that have a wider distribution of membership counts.

If we are examining points on a partition-by-partition level, there are more possibilities for weighting individual data attributes. I compute the difference between means within and outside of the partition for numeric attributes, prioritizing attributes that are markedly different from the rest of the data. For individual categorical values, a variant on tf-idf that examines frequency within-partition versus in the entire dataset roughly estimates how salient each value might be. By aggregating these scores one can approximate how much a particular category differentiates a partition from other parts of the dataset.

6.4.2 Modeling User Interest

For the purposes of this initial exploration in Point Salience modeling, the user model entirely depends upon priorities explicitly provided by the user. This potentially has the highest signal for relevance to the user as they must directly interact with the interface and specify their interest. A user might state that they are looking for higher values in a particular numeric attribute, or that they are especially interested in a particular value within a categorical variable. They might also provide negative criteria, reducing the weight of points that have certain characteristics. Per-attribute δ_a weights can be directly exposed to the user as part of entering a new preference in the form of an estimation of how much they care about a criteria or how important it is.

My prior work in Task Fingerprinting suggests that simple interaction monitoring can give insight into users' work processes and focal points during a task [138]. One might better establish user weights by monitoring user interactions in addition to allowing the user to specify specific priorities. For instance, if the user chooses to search for places that meet a particular criterion, then the user level model ought to weight points meeting that criterion higher than other points. Similarly, if the user has been checking summary details for a certain subset of points, the user model might weight points that have similar reviews (by N-gram similarity) or similar data features. There also exists an interesting possibility for users to express different interests on a partition-by-partition level, though this likely will impose a high amount of complexity in interactions with the system.

6.5 Combining Models and Level of Detail

Figure 6.13 shows an example of the Point Salience model working in concert with the level of detail algorithms on a visualization that uses Kinetica/DataSquid colliding points to show estimated distribution. Because the user model has been programmed with an interest in blue points within the ”color” category, it increases the pairing distance between points that both have blue values so that they are rarely packaged together. The user model adds a general prioritization for the ”color” category, and since red values are more unique they are boosted. This means that grey points are most aggressively bundled (because they are the closest and least salient), then red points, and finally, if all else fails and there is still not enough room, blue points. As the user zooms (top to bottom images) there is more room to expand points. Red points expand first, followed by less salient grey points if the user continues to zoom.

This combination of salience with aggregation surfaces the blue points in the example readily, letting the user quickly estimate how many blue points are in any given region. Because the algorithm is aggressively managing the level of detail, the collision model from Kinetica rarely overflows a region and moves points well beyond their correct position. The Point Salience model also provides more affordances for visualization. When points are aggregated together, the model prioritizes which point is on ”top” of the group for signaling color and size.

6.6 Experimental Validation

Having developed a technique for representing data at various scales and for prioritizing displaying the most salient points, there are several questions that are in need of validation:

- Does mixing focus and context cause users to have an unrepresentative view of data as they explore?
- Do users accurately interpret groups of compressed points as they explore a dataset?
- Does hiding points/examples confuse users as they explore, and can they track how changes in model alter their view of data?

To examine these questions, I will compare Point Salience aggregation, Kinetica style collision alone, or allowing all points to display and overlap (i.e. no level of detail management at all) within a zoomable mapping interface. This ought to provide a baseline for future visualization development in this area by characterizing where each technique performs well and poorly.

6.6.1 Study Design

In order to investigate the previously described research questions, I designed a comparative study to conduct over Amazon Mechanical Turk using a configurable HTML5/Javascript web tool. The tool (adapted from Fractal, outlined in more detail in the following chapter) had the ability to model and aggregate points following the algorithms outlined in this section and then

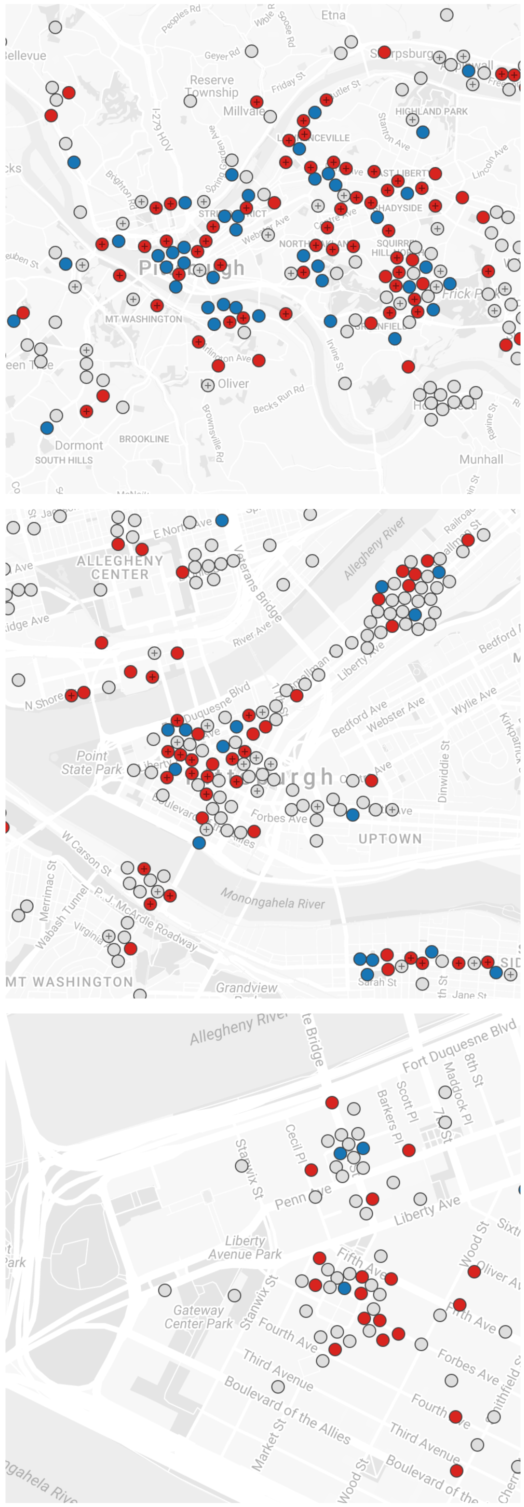


Figure 6.13: A user zooms in on a synthetic dataset. The user model weights heavily towards showing blue points (and the "color" category in general), and the data model recognizes that red values are less common than grey ones. As a result all the blue points are always visible, red points are prioritized but bundled if there is not enough screen space (+ signs denote groups), and grey points unpack last.

map those results back into a circular icons on a pannable and zoomable map. A user could mouse over points to see additional details. The aggregation can be turned off resulting in every point appearing on the map regardless of crowding. Kinetica-like collision functionality can be enabled to allow points to bunch up, otherwise all points will occupy their exact position on the map regardless of occlusion.

For this study I designed a synthetic dataset using positional data of places in Pittsburgh scraped from Yelp mapped to a numeric and a categorical feature. The numeric feature ranged from 0 to 100 and was randomly assigned to the 734 points in the dataset. The categorical ”color” feature (as illustrated in the aggregation+collision condition in Figure 6.13) had three randomly assigned category options: blue (5% of points), red (20% of points), and grey (75% of points). Participants completed several tasks split between the numeric and categorical features. For the numeric feature visualization the Point Salience model was seeded in with an interest in the numeric feature, prioritizing higher value points. For the categorical feature visualization the Point Salience model was seeded with an interest in blue points within the ”color” category and a general preference for the ”color” category.

Participants were split into three conditions using a between subjects study design. They either completed tasks using a visualization that showed all points and allowed occlusion (”Overlap” condition), a visualization that showed all points but enabled Kinetica-style collision (”Collision” condition), or a visualization that enabled Kinetica-style collision but reduced overload by activating the Point Salience aggregation model (”Aggregate” condition). Regardless of condition, 5 rectangles were drawn onto the map and used as zones of interest for the study tasks. Participants were told explicitly not to take an extreme amount of time counting each and every point. They instead were urged to spend at most 10 minutes on the task, using estimates as their accuracy would not factor into their compensation. Figure 6.14 illustrates the three conditions showing the numeric visualization scenario.

Participants were asked the following questions in each condition:

- Tool: **Numeric Visualization**
- Q1: Please estimate how many points of data (circles) are in the dataset.
- Q2: What point on the screen has the highest value (i.e. is the most yellow)?
- Q3: For each of the rectangles on the screen, estimate how many points of data are in each rectangle.
- Q4: Please rank the rectangles by which have the fewest/most points of data with a value greater than 90 (yellow circles).
- Q5: For the following rectangles, describe the shape of the data:
- Tool: **Categorical Visualization**
- Q6: Please estimate how many blue circles are in the dataset.

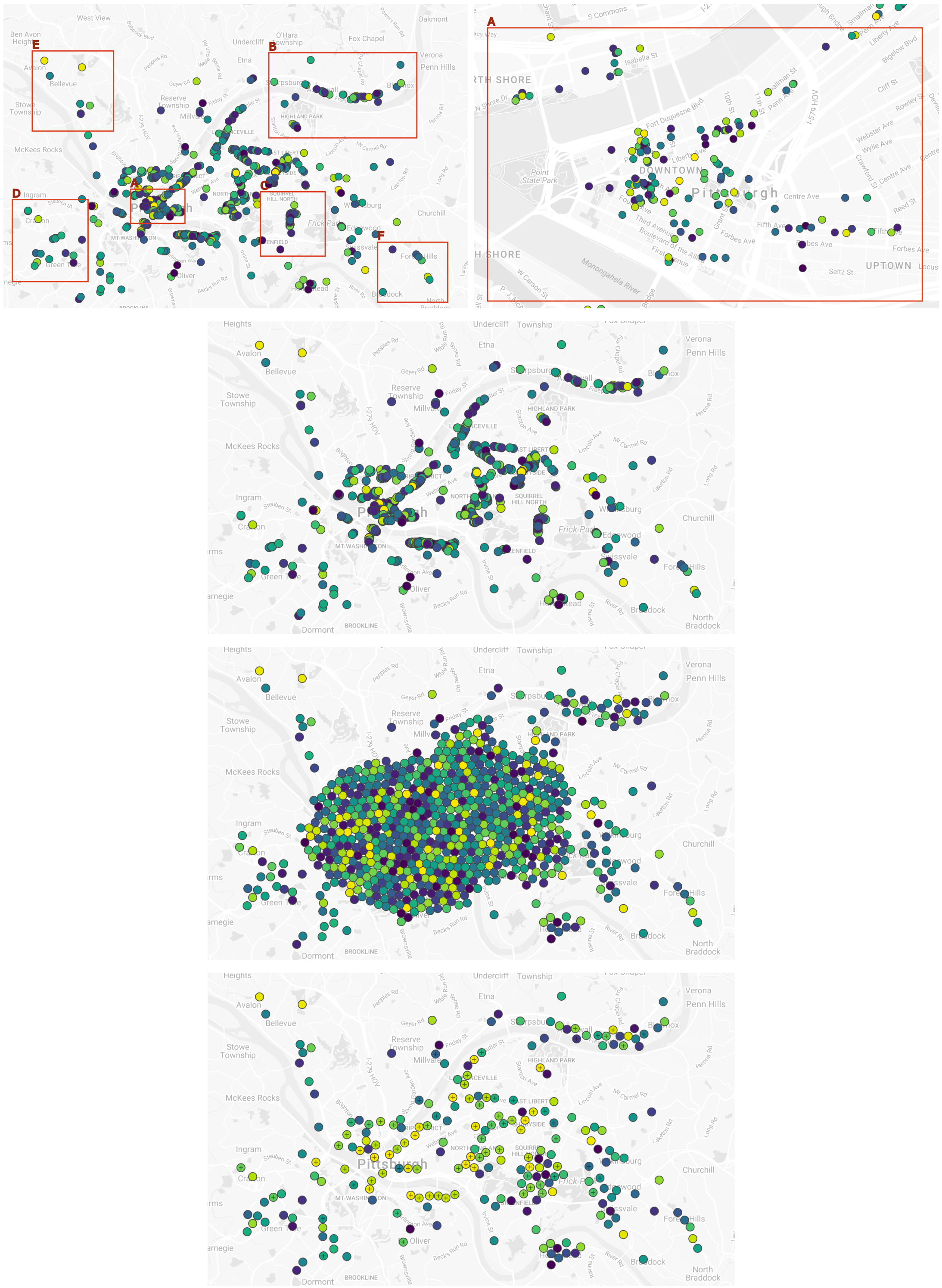


Figure 6.14: Rectangle regions for participants to examine (top) and the three Point Salience validation conditions: Overlap, Collide, and Aggregate (bottom). As the user zoomed into the map Overlap occlusion reduced, Collide points had more room and did not bunch up as much, and Aggregate points split to reveal more detail

- Q7: Please estimate how many blue circles are in each rectangle.
- Q8: Please rank the rectangles by how many **red** circles they contain.

6.6.2 RESULTS

I recruited 30 unique participants from Amazon Mechanical Turk to complete the study, mandating that the participants were at least 18 years of age, lived in the United States, and had a decent track record on the site. Preliminary tests indicated the protocol took between 10 and 15 minutes to complete, so compensation was fixed at $1.60 (8$ per hour * 12min). Upon requesting a task participants were assigned randomly to one of the 3 conditions and received a qualification that prevented them from completing any other between-subjects studies. Hauser & Schwartz outline the potential benefits and risks of using crowdworkers [67], suggesting that they are largely comparable to university study participant pools in performance.

There was no observable difference between participants' estimation of the number of points on the screen in Q1 (F(2,27)=0.976, $p = 0.390$). This is especially encouraging for the aggregation condition as there are far fewer points shown onscreen than in the other two conditions which show all points to users. Likewise, there was no observable distance between average rectangle estimate accuracies in Q3 (F(2,27)=0.411, $p = 0.667$, Figure 6.15b) Yet, when asked to identify the circle with the highest overall value in the dataset in Q2, aggregate users outperformed collision (2nd) and overlap (3rd) (F(2,27)=4.551, $p = 0.020*$, Figure 6.15a [lower is better]). This is likely because the Point Saliency model was prioritizing high value circles, guaranteeing that the high value point appeared onscreen. However, this also meant that in fact the top 20 points were onscreen in the aggregate condition (so there were in effect at least 19 potential confusors present). The level of detail function, limiting the number of onscreen elements versus the other two conditions, likely is at play here in improving participant accuracy.

When asked to estimate the number of blue points onscreen in the categorical dataset in Q6, aggregate condition participants performed significantly better (F(2,27)=4.014 $p = 0.0298*$, Figure 6.15c). Collide participants performed nearly as well since the heavy occlusion in the Overlap condition made it harder to spot all blue targets. Recall that the Point Saliency model for categorical data was programmed to prioritize splitting blue points, so aggregate participants had a high likelihood of spotting all of the blue targets. However, when asked to estimate the number of blue points within each rectangle there was no observable difference between the conditions (F(2,27)=0.547 $p = 0.585$, Figure 6.15d). Because participants could pan and zoom within the interface, it is likely that they were able to zoom in enough to avoid occlusion and bunching up, leveling out performance across the conditions.

For both the numeric and the quantitative ranking conditions (Q4 and Q8), I computed a Spearman distance for each participants' rankings. Smaller distances correspond to fewer misranks, and missing a rank by several places is penalized more heavily than being 1 off. Combining conditions that had collision (aggregate & collision), they marginally and significantly outperformed the overlap condition (Q4: F(2,27)=4.058 $p = 0.0537$, Q8: F(2,27)=5.063, $p = 0.0325*$, Figure 6.15ef). Because participants could zoom the overload from the collision points bunching up was slightly mitigated. This meant that the occlusion from the overlap condition ultimately became the dominant hindrance on participants' performance.

6.7 DISCUSSION

Overall these findings are quite encouraging and offer a bit more nuance into the intersection of physics and aggregation. Users in the Point Saliency aggregation case were readily able to estimate amounts of points and identify single values, performing at least comparably to the other two conditions. Their view seems to remain representative despite much fewer points onscreen. The "+" sign design element for representing aggregations of points seems to be effective. The show-everything-but-overflow Collision condition performed well in situations where a user had to estimate total numbers of points, but, as expected, overloaded users when they had to find specific items. The overlap condition often occluded useful points, explaining its poor performance in the categorical questions where some blue items were hard to spot, but otherwise did give participants an accurate view of the general layout of the data.

Because some of the tasks involved panning and zooming, I also have reason to believe transformations as a result of partitions breaking and points re-aggregating did not disrupt users as they explored. Practically speaking, the aggregate condition was the most performant of the 3 conditions as the algorithms to collect and curate the points on zooms and viewport changes were much faster than the performance hit of rendering every point on screen. While these results are encouraging, they are not situated within an actual data exploration task. In the next section I will describe the development and testing of a tool, Fractal, which gives more insight into how aggregation affects the sensemaking process.

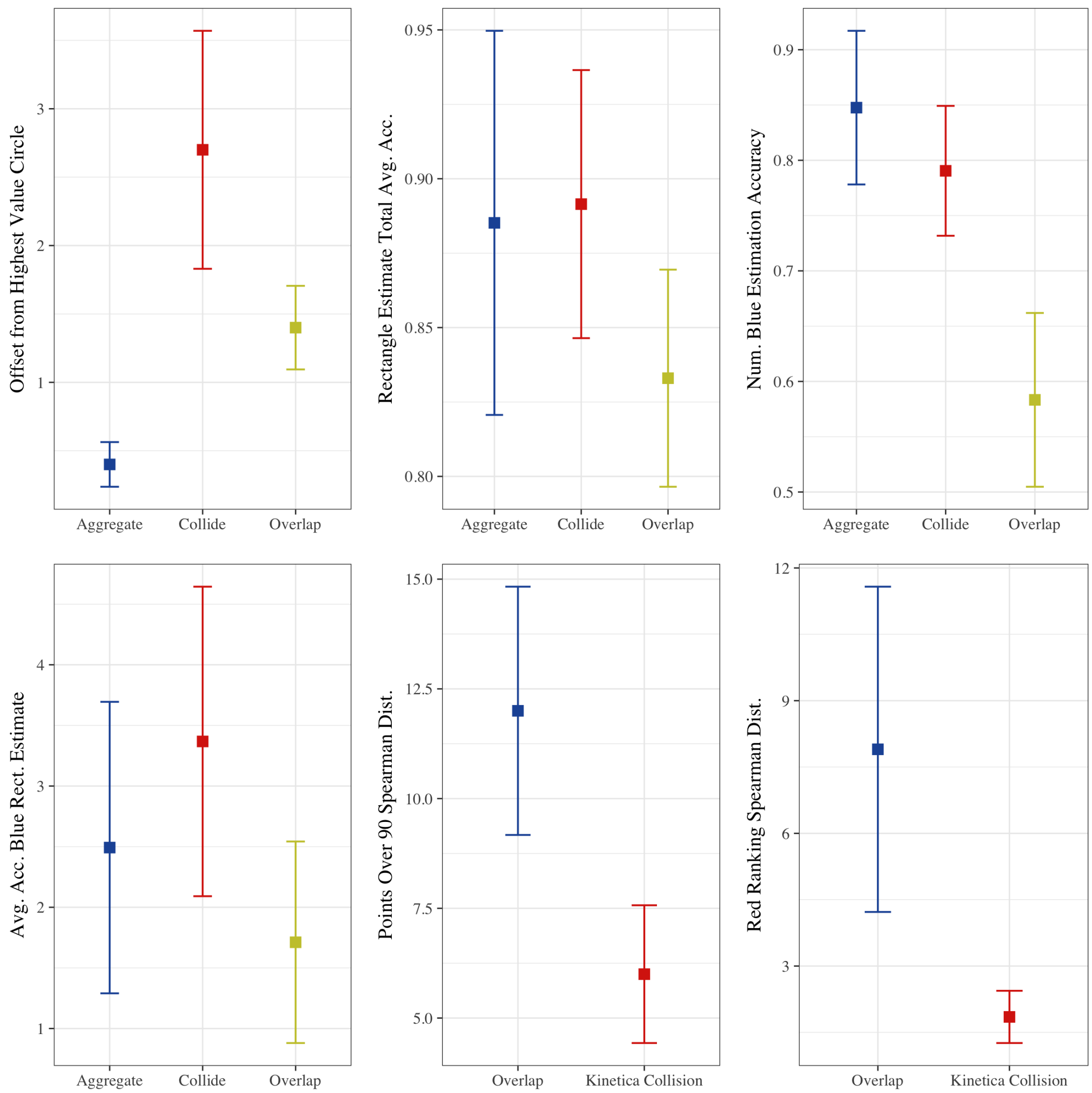


Figure 6.15: Results from the synthetic study comparing aggregation using Point Saliency models, Kinetica style collision alone, and overlapping points.

7 — Fractal: Applying Point Salience to Consumer Search

In the previous chapter I developed and evaluated a general technique for maintaining a consistent level of data elements per pixel on a screen while at the same time prioritizing salient points that have interesting data features or, based on user interactions, seem relevant to the data analyst. Returning again to the area of consumer search, the Point Saliency techniques seem particularly well suited to these sorts of tasks. In consumer search users first need to make sense of the dataset and identify what factors they really care about by examining a wide variety of attributes and options (the sensemaking process [137]), and then need to rapidly and effectively find points that meet their particular criteria [18] which may still be ill defined or represent a balance of competing costs (e.g. price versus quality). Because the Point Saliency models might help users bootstrap into the process by revealing useful data features and then transition into highlighting points that satisfy criteria by incorporating user features, the technique has the potential to provide meaningful improvements for searchers.

Moving forward, there are a number of questions situated within the sensemaking and exploration process to consider:

- How do Point Saliency models play out in a real data exploration situation?
- Does hiding examples break users mental models and confuse them while they explore points?
- How does aggregation/focus affect physics-based visualization techniques?

In this chapter I will outline the development and user evaluation of a web-based tool, Fractal, for restaurant consumer search using Yelp restaurant review data and a Point Saliency model. Restaurant search makes for an interesting use case (not only because it is a very common form of consumer search among conference-going peer reviewers!) because there is a degree of high dimensionality to the data (ratings, reviews, popularity, genres, price) that must be balanced, a degree of high cardinality (there are potentially many restaurants in even the small Midtown Manhattan area) that require managing level of detail so users aren't overwhelmed, a mix of numeric and categorical features (numeric rating vs. particular genres), and grounding onto the real world through the geographic coordinates. Additionally, current web interfaces for interacting with this sort of data are ill suited to open exploration (Figure 7.1). Only a dozen or so points are mapped at any one time, occlusion still occurs, and users must draw a mapping between text elements on the left and map pins on the right themselves using number labels.

Reflecting on the way that Point Saliency models break and reform data, they exhibit a similar sort of self-repetition with mathematical fractals (Figure 7.2). When zooming into smaller portions of a space, the Point Saliency model repeatedly applies the same algorithm to a dataset with an increasingly smaller range, much like a fractal iterates through a formula with complex numbers. The detail at large and small scales repeats as clusters form and break apart in order to maintain an adequate level of detail. This self-similarity is one potential benefit of Point Saliency models, since it might help users better maintain their mental model between interface transitions like panning or zooming on a map. In the following sections I will describe the technical implementation details for Fractal and then conduct a short user study comparing Fractal's performance to browsing restaurants on Yelp.

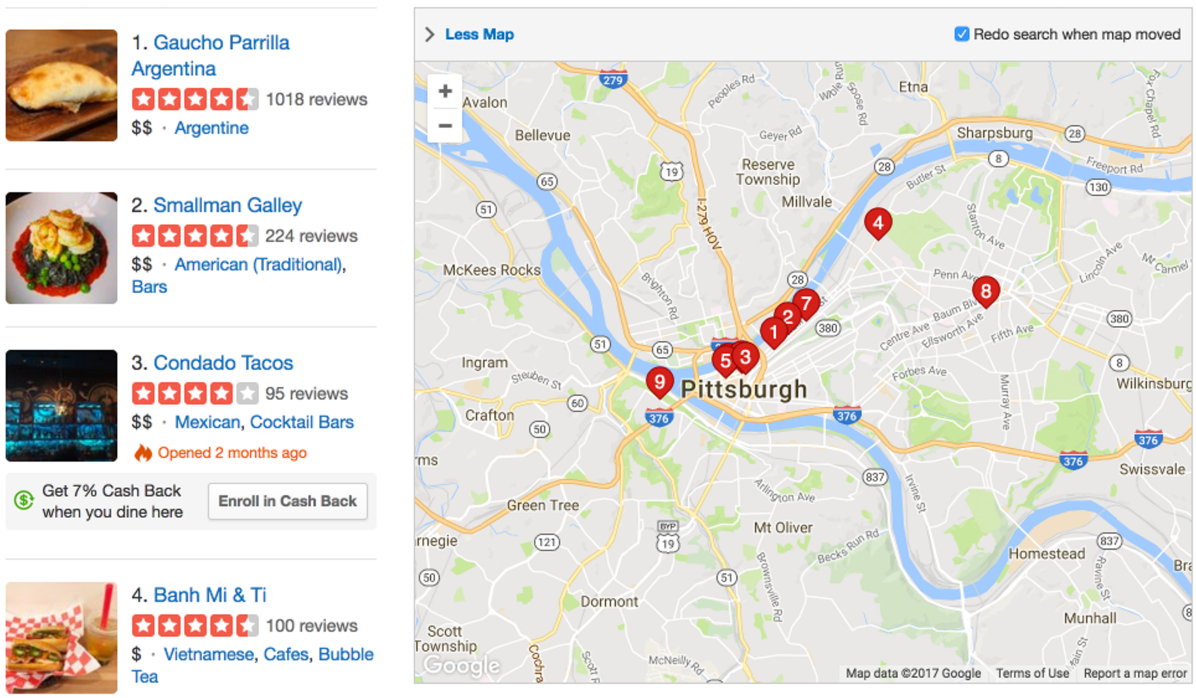


Figure 7.1: A Yelp search for restaurants in the Pittsburgh metro area. Notice that only a dozen or so points are mapped at any one time, occlusion still occurs, and users must draw a mapping between text elements on the left and map pins on the right themselves using number labels.

7.1 Technical Details

I implemented Fractal in pure client-side HTML and Javascript. While in the early days of Kinetica this may not have been feasible, Javascript engines have now become performant enough to allow for realtime hierarchical clustering, model generation, and large rending loops. As a prototype operating on numbers of restaurant datapoints on the order of 500-5,000, client side processing is adequate. However, larger or more complex datasets may necessitate additional server side programming as well in order to parallelize model generation and data operations. Fractal uses SVG for rendering data elements; a mix of d3.js, jQuery, and native Javascript for its core rendering and interaction loops; *hcluster* for its level of detail hierarchical clustering model; Google libraries for Haversine distance calculations and geographic projections; the Google Maps API for map tiles, the Yelp API for data aggregation (though stored currently as a static file), and adapts the force-directed model of d3.js to simulate the colliding behavior of Kinetica/DataSquid. The Point Saliency model is implemented as described in the previous

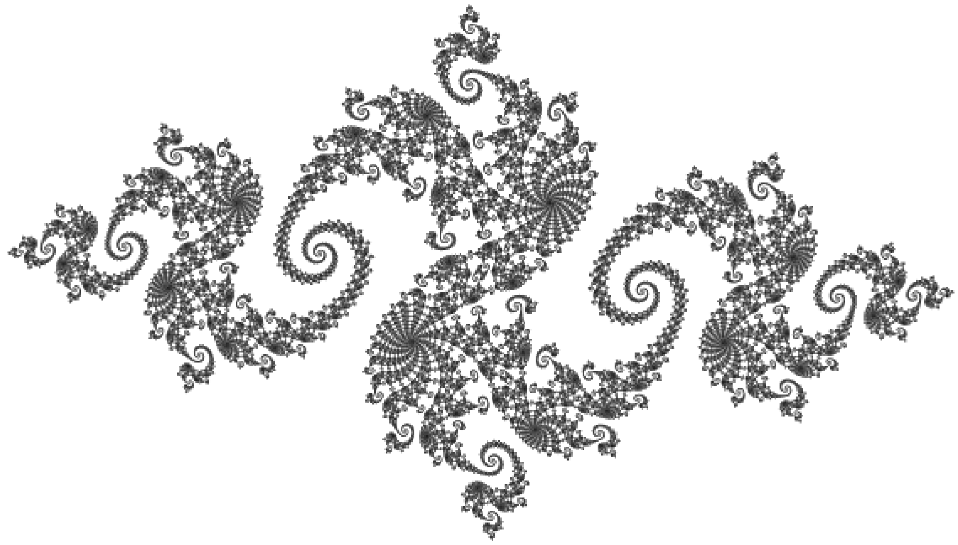


Figure 7.2: A Julia fractal exhibiting self-repeating detail.

chapter. Fractal also borrows Kinetica/DataSquid's stacking lens metaphor and algorithms for coloring and sizing datapoints. Though it has its roots in the Kinetica mobile app, in its state during evaluation it does not have full mobile support. Small screens and touch interaction impose additional constraints beyond this initial exploration, though I hope to investigate these issues in the near future.

7.1.1 MAIN VIEW

The focal point in Fractal is a large map view using the Google Maps API to generate tiles and project individual restaurant datapoints onto an SVG canvas. Each point becomes a bubble that is scaled and colored according to weights generated by the Saliency model. In order to prevent occlusion I iterate through a Kinetica/DataSquid physics simulation whenever circles are aggregated or split (generally during a zoom/unzoom operation on the map since it processes even offscreen points). While I have tested employing animations in the same way that Kinetica animates colliding points, in practice the animation combined with map panning and zooming becomes a bit confusing. Since the level of detail model is already trying to minimize the amount of overlap within a given cluster by bounding the number of visible elements, the risk of points dramatically and instantly changing location from one aggregation step to another is relatively low. Instead the physics-based simulation occurs without a rendering loop and only the final result once points have settled is pushed to the canvas. No participant in the evaluation noted points moving in unusual or confusing ways. A plus sign within circles indicates that a particular element contains aggregated points (i.e. it is a stack/group of points rather than a singleton). When points are aggregated the Saliency model determines how best to color/size them and their geographic position is determined as the average of the position of all included points.

On the left side of the screen the user can choose different filter criteria in a hide-able drawer that

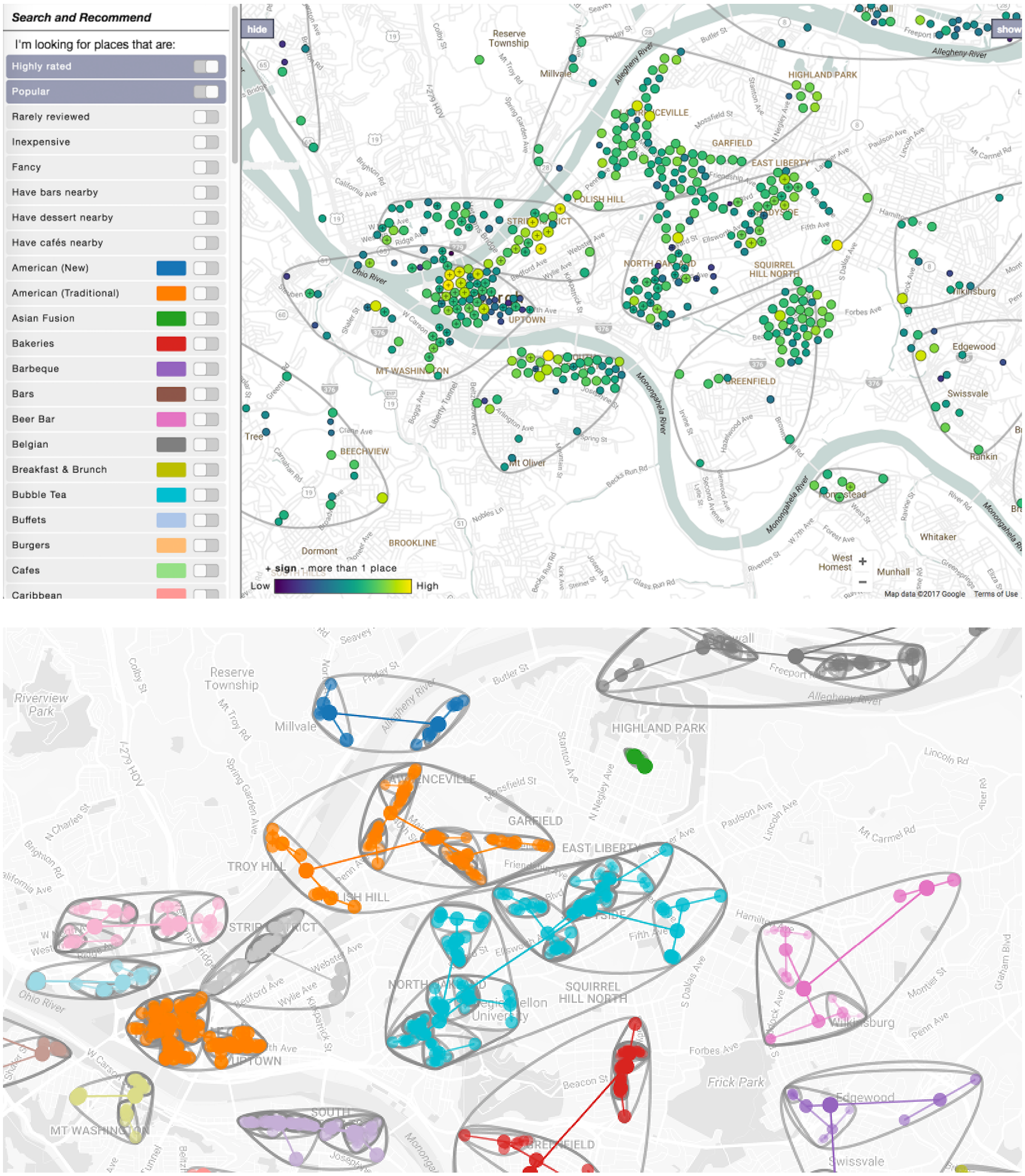


Figure 7.3: Top: The Fractal interface. Here a user has chosen to highlight points that have high ratings and popularity. Points that have been aggregated are marked with a + sign, and yellow points indicate higher rating+popularity weights coming out of the Point Saliency model.
Bottom: A debug view showing the hierarchical clustering model for a given zoom level in Fractal identifying how points might be broken down if users zoom in on an area further.

by default is opened. On the right side of the screen a hidden drawer contains a list of restaurant entries much like the left side of Figure 7.1. Colors are chosen dynamically from a large list of valid colors (ordered to minimize confusion between colors) for filters as the user scrolls through the list. When a user mouses over a filter or applies one, that color is assigned to the points and a lens highlighting applicable points is mapped onto the screen (see Figure 7.6).

Because mousing over an individual circle can sometimes be challenging, I implemented Grossman and Balakrishnan's Bubble Cursor [58] which prioritizes selecting the closest point to the curser within a set range. As a user mouses over points, they see additional details about the individual restaurant (or group of restaurants aggregated together) including name, rating, popularity, and curated blurbs described later in this section (see Figure 7.4). The bubble cursor responds if particular points are highlighted by prioritizing their selection over lowlighted points.

7.1.2 AUTOMATIC CURATION

Fractal makes one key extension to the Point Saliency model described in the previous chapter. In addition to generating weights and distances used in rendering points, the model also outputs values into an engine that attempts to provide useful summaries for points. Figure 7.4 illustrates one such summary appearing for a restaurant on mouse-over. Because the user has specified that they want popular places, the summarization engine is more likely to mention popularity if the point has a remotely interesting value for that attribute. The model also notices that the point has an extremely high ranking in a large category, American(new), and adds that to the short summary as well. Other factors that might appear in summaries include rare categories, unusually low values, local features such as the highest rated place in a cluster/neighborhood, and fallback features in case model weights are generally low such as simple summaries of a point's place in an attribute distribution (e.g. "moderately popular restaurant").

Summaries are an easy way to provide the user with another tool for chunking/modeling points of data in addition to their name and geographic location. They also give the Point Saliency model the chance to surface more highly weighted model features to the user. Given that consumer search often depends strongly on user motivation [18], surprising or interesting summaries might also help keep the user's attention. Summaries are also another vehicle for giving users insight into how the Point Saliency model is curating what they see. Given user resistance to management by hidden algorithms [104, 132], providing more insight might improve users' comfort with the system.

The engine works via a mapping that takes a model attribute and associated weight/value as input and outputs a verbal summary. For instance, if the model were to emit (category, [American(new),rank=3]), the summary engine might output "Top 10 restaurant in the American(new) category." In general, the chosen summary is a concatenation of the top 2 weighted user model features, 1 top data and 1 top user feature if few user features are specific, or 2 top data features if the user model is undefined. In the Fractal tool these mappings are encoded as a set of rules generated by ideating in person with Yelp users over potentially interesting findings and by examining the results of short survey task posted to Mechanical Turk asking participants to write 10 word blurbs about a Yelp restaurant. There is an exciting possibility to allow users to create, curate, and share their own set of rules for the model the highlight, perhaps even distributing

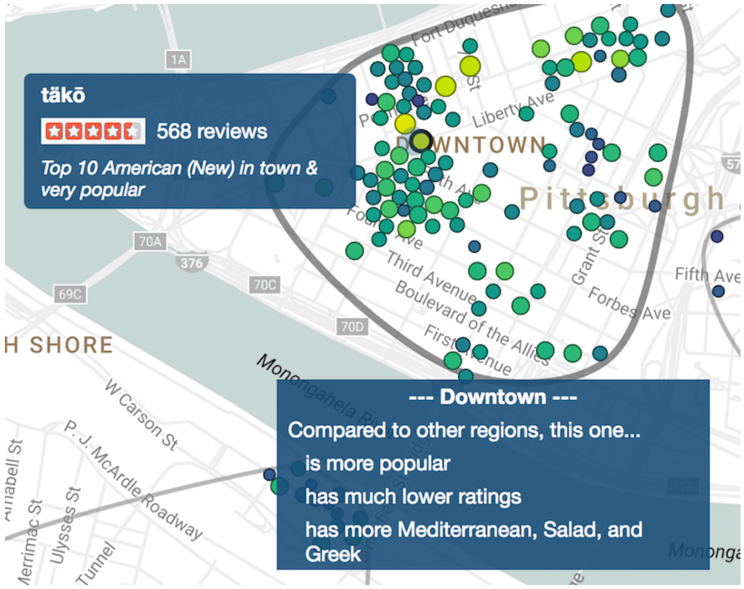


Figure 7.4: A user has moused over a particular point, exposing additional details about the restaurant. Notice at the bottom curated details for this particular cluster comparing it on a whole-cluster level to other clusters on the map.

them across the entire Fractal platform userbase should they prove useful.

The summary engine also uses cluster-level weights from the model in order to explain what might make a cluster unique. Using the correlational and tf-idf components of the model, one-line summaries about differentiating categories or numeric attributes are composed using a mapping table similar to that of the single point summarizer, ordered by the model's estimate of feature importance. If a cluster has an unusually large amount of popular places or contains all of the Thai restaurants in the entire city, the cluster summarizer surfaces these features. As with single point summaries, these cluster-level summaries provide more hooks for users to chunk the geography of points into conceptual units. If the majority of the points in a cluster share a geographic division such as neighborhood or street, the summarizer also surfaces them as a way to help the user reference clusters during their search.

7.1.3 Selection and Comparison

As the user selects individual circles, cards are dynamically composed (see [39]) for the restaurant or group of restaurants corresponding to the selected element. If the summary engine can produce a valid summary, that is also added to the card. Given some selected circles may be aggregations of points, the Point Saliency model chooses the highest weighted point as the card contents (making it more likely that users see results that matter to them even if the model is compressing data to save screen space). These small cards allow the users to rapidly examine and compare points.

Eventually a user may want to store some points for later use or keep them on screen so they can directly compare the places. By clicking a point the user can generate a movable window containing the restaurant card. An expansion button is added to the card, allowing the user to expose more details such as reviews, highlights, and images (Figure 7.5). There is no upper

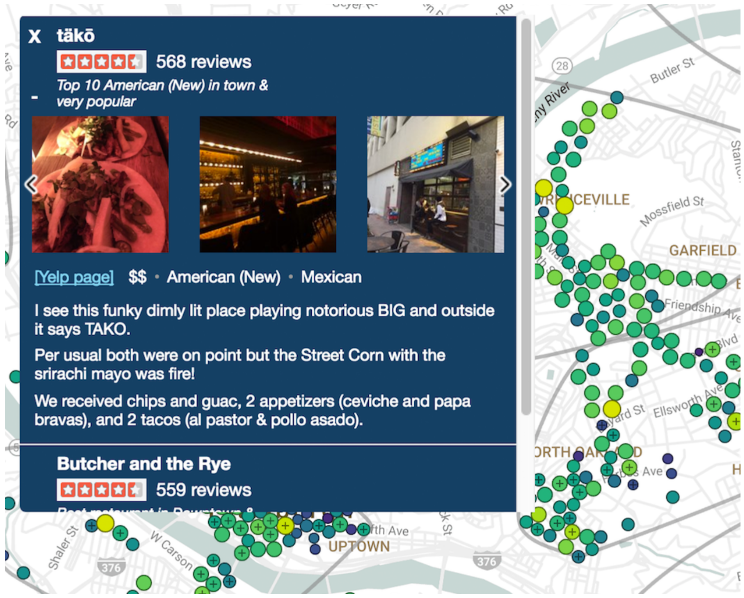


Figure 7.5: A user has clicked on a particular group of restaurants. They have formed a sticky panel that can be expanded to show more details or images and moved around the screen for later use in comparisons or note-taking.

bound on the number of windows visible, and clicking on an aggregation of points creates a scroll-able list for all places contained within that element.

7.1.4 Filtering

Fractal has no method for hiding particular points. This was an intentional design decision in order to compare the Point Saliency model approach of weighting/prioritizing points based on user criteria with the traditional model consumer search sites like Yelp employ of hard filters that hide points (often through faceted browsing panels). Instead, a user employs the panel on the left side of the interface to specify what kinds of points interest them. For numeric features these tools solely influence the weights generated by the model, adjusting how points are colored, sized, and summarized. For features like specific genres or geographic proximity measures, the system applies a Kinetica/DataSquid lens which is a slightly harder filtering mechanism, explicitly highlighting and lowlighting points that meet criteria. The interface has an affordance where mousing over genre attributes temporarily highlights and lowlights so users can estimate distribution, similar to the filter brushing in Kinetica/DataSquid.

Recall that the Point Saliency model is taking into account both data- and user-level features. In Fractal I have opted to set the scale factor for data features at 40% of the total model weight and the user feature scale factor at 60% of the model weight. This means that user entries in the left panel do take precedence, but there is still a chance that the model will surface unexpected but meaningful data characteristics.

7.2 User Study

In order to better understand how the Point Saliency model and interactive elements of Fractal might help users make sense of consumer search results, I conducted a user study where partici-

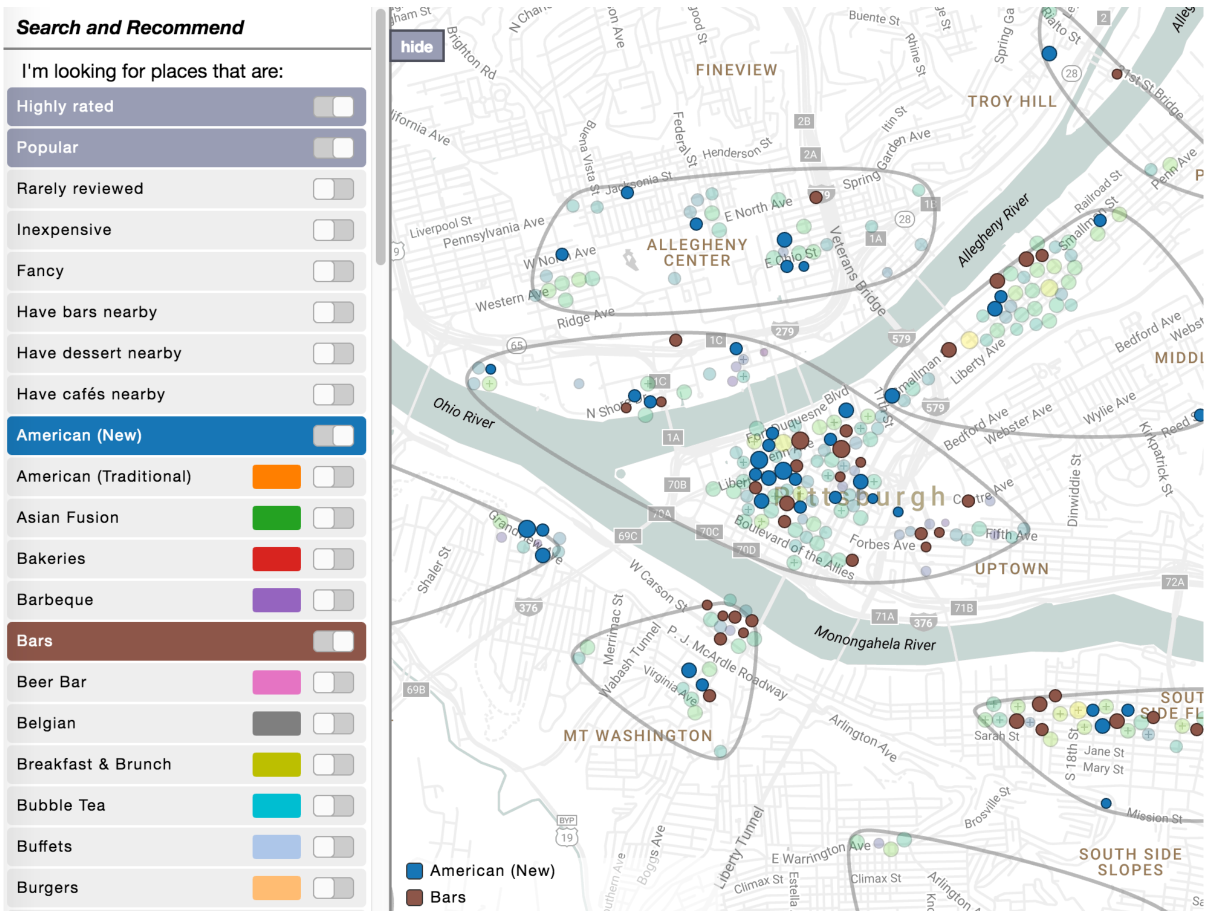


Figure 7.6: Here the user has overlaid two filters highlighting American(new) and Bar genre places. The Point Salience model has increased the distance between points that match those criteria, forcing the level of detail function to prioritize keeping points with those genres as un-aggregated singletons.

pants received a scenario describing a person's needs and then explored data in order to answer it. Rather than use Pittsburgh residents as in the comparative consumer evaluation study, I opted to recruit from Mechanical Turk so that I had no guarantee of domain expertise in the Pittsburgh food scene and captured a wide range of user interests and backgrounds. As both Yelp and Fractal are delivered via web browser, the crowdsourced web-based protocol looks very similar to that of a lab-based study of the same tools. To push participants to satisfice, they were urged to spend at most 10 minutes searching on Yelp or Fractal before providing their answer to the scenario and completing a post-survey. All participants were trained using an instructional passage at the beginning of the task (no participant later mentioned feeling unprepared to use either tool) rather than through a detailed training both due to limitations of the crowdsourcing platform and a desire to verify that Fractal did not impose an unexpectedly large expertise burden. The total task time was limited in the Mechanical Turk interface. Participants were compensated $2.50 for a completed, valid submission ($8 per hour wage * 15-20 minute expected maximum duration).

Participants were randomly assigned to either the Yelp or the Fractal condition, and then were randomly assigned one of three potential scenarios. I used a between-subjects study methodology, applying Mechanical Turk qualifications automatically to avoid spillover from one condition into the next. The three potential scenarios are as follows:

- Imagine that you are traveling to Pittsburgh for a convention. You've never been to the city before and are going to meet up with some other convention guests for dinner and drinks while you're there. It's your job to pick the place and convince everyone else that it's a good choice. The restaurant needs to be near the convention center, and it ought to be near a good bar for after dinner drinks. Your group doesn't have strong preferences on the kind of food to eat, but you'd like a New American place or Japanese place. On the other hand, you're up for any restaurant if it looks especially good and is popular. Please write 3-5 sentences explaining your choice, how you went about finding it, and convincing the rest of the group that you've made a good call. The convention center has been marked with a red square on the screen.
  - This scenario asks users to consider geographic proximity to a landmark when exploring data (landmarks were marked with an indicator icon in Fractal and a specified Yelp search). The task contains a soft constraint about cuisine and a hard constraint of proximity to a Bar genre place. It offers suggestions on genre, but leaves participants room to investigate and explain any option they feel might work.

- Imagine that you are planning a date in Pittsburgh. You and your date are up for driving anywhere in the city limits, but you're both foodies so it better be somewhere good. You plan to start at a cafe and then walk from there to your restaurant. You don't want to decide on a place until you meet up, so you're exploring potential neighborhoods in which to meet. Ideally the neighborhood should be popular and hip. You don't care too much about cuisine, but you'd like there to be a bunch of different kinds of food so you have lots of choices. Please write 3-5 sentences suggesting a neighborhood to your date and giving a few diverse recommendations for them to consider.
  - This scenario asks for proximity and gives several attributes to balance, but pushes users to think about neighborhoods rather than individual restaurant points.

- Imagine that you are taking your partner and young daughter out for lunch after a morning at the zoo. Your daughter is surprisingly not picky for her age, though your partner wants to eat at only the highest rated places. Your daughter especially likes Asian food and your partner is up for trying something new. It shouldn't be somewhere too expensive. You know your daughter is going to want ice cream afterwards, so the place ought to be very close to a dessert shop. Please write 3-5 sentences giving three restaurant options to your partner and describing why you chose them and what differentiates them from each other.
  - Compared to the convention scenario, the zoo scenario has fewer specific constraints and is more open ended.

Afterwards, I gave users a short post-survey inquiring about their confidence in their answers, how much data they have seen, and their satisfaction with the tool that they used.

7.2.1 RESULTS

I recruited 30 participants over Amazon Mechanical Turk. 15 participants were assigned to either Fractal or Yelp and 5 participants were assigned per scenario, giving a total N of 30 participants. All participants were able to complete the survey methodology, and no participants exhibited obvious gaming behavior.

Examining the answers that participants provided for scenarios (see Figure 7.7, in a two way analysis of variance there was no observable difference in the number of suggestions participants provided between tools (F(1,24)=0.022 $p = 0.883$), but there was a main effect for task as some tasks asked for more than others (F(2,24)=6.067 $p = 0.0074*$) (no interaction effect was observed [F(2,24)=0.156 $p = 0.857$]). In a two way analysis of variance there was a significant difference in the number of data attributes that participants mentioned in their answers between tools (F(1,24)=15.7 $p < 0.001***$), and a significant difference between tasks (F(2,24)=4.04 $p = 0.031*$) (no interaction effect was observed [F(2,24)=0.565 $p = 0.576$]). Participants who used Fractal might have mentioned more attributes because the filter panel on the left of the screen primed them to consider specific data characteristics, while Yelp users did not conceptualize page elements in those terms.

Participants expressed no observable difference in their level of confidence (1-5 Likert scale) for their answers (Tool: F(1,24)=0.305 $p = 0.586$; Scenario: F(2,24)=1.322 $p = 0.285$; Interaction: F(2,24)=0.102 $p = 0.904$), but Yelp users thought it significantly more likely that they missed another, better option after their search (1-5 Likert scale; Tool: F(1,24)=5.042 $p = 0.034*$; Scenario: F(2,24)=0.5 $p = 0.613$; Interaction: F(2,24)=2.167 $p = 0.136$). Perhaps the map view and increased exposure to every data point made Fractal users feel more confident that they had explored the entire space of restaurant options. This is reaffirmed when participants were asked to estimate what percentage of Pittsburgh restaurants they thought they had investigated over their task (0-25, 25-50, 50-75,or 75-100). Fractal users thought they missed significantly fewer restaurants, and a significant main effect by Scenario illustrated that users in the convention scenario felt they explored a smaller section of the city (Tool: F(1,24)=17.515 $p < 0.001***$; Scenario: F(2,24)=4.606 $p = 0.0203*$; Interaction: F(2,24)=0.970 $p = 0.394$). Fractal users also thought they missed significantly fewer Pittsburgh neighborhoods, and a similar main effect was observed for the convention scenario (Tool: F(1,24)=11.655 $p = 0.002**$; Scenario:

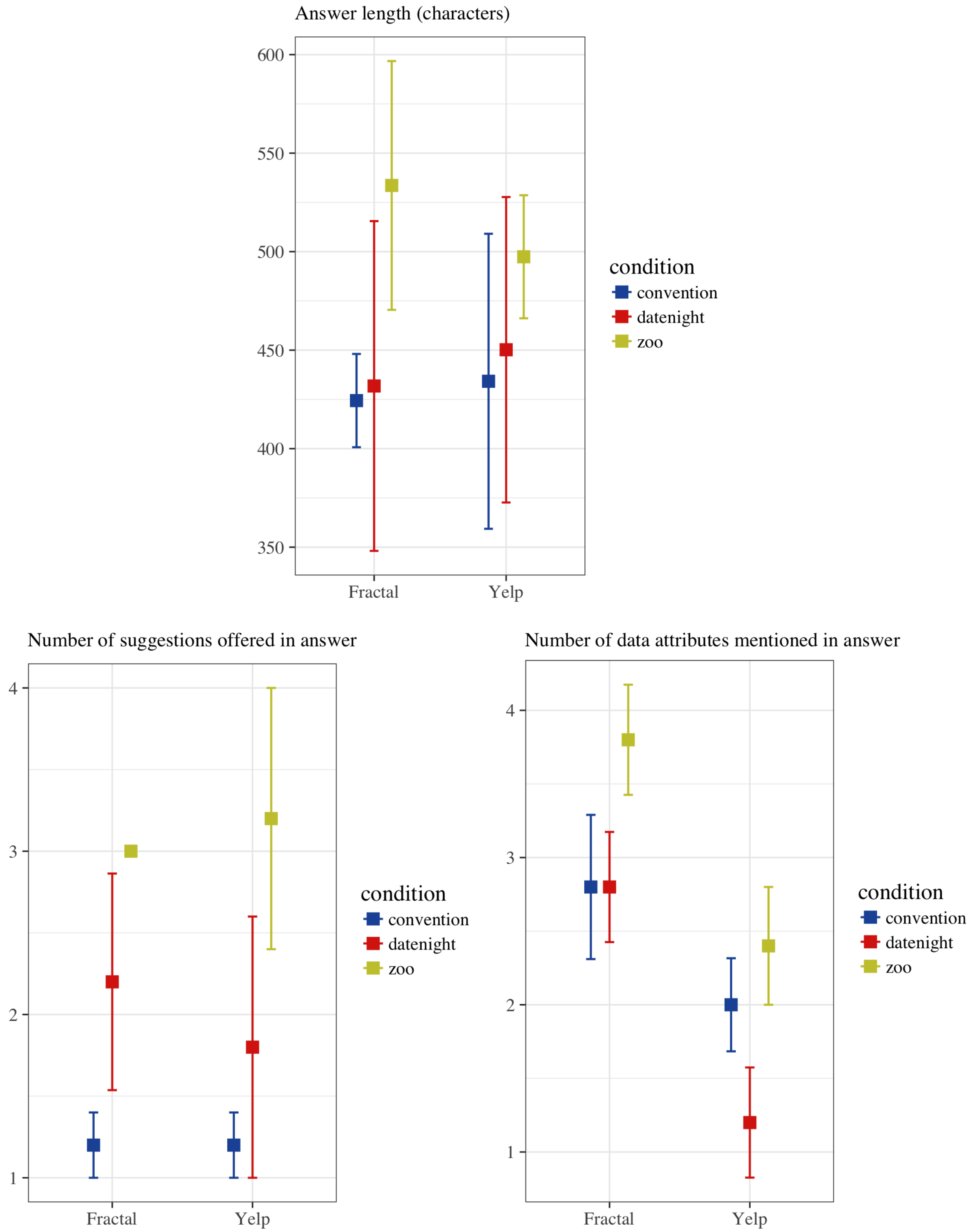


Figure 7.7: Results from the Fractal evaluation scenario answers.

F(2,24)=6.690 $p = 0.005**$; Interaction: F(2,24)=1.31 $p = 0.288$).

Fractal users also reported feeling satisfaction over covering much of the data in their qualitative responses. One user wrote ”I like[d] how easy it was to find what I was looking for, it was very informative. I feel like I saw most of the restaurants.” They reported appreciation of the filtering options, claiming, ”The filtering options helped cut down on restaurants and was really fast to find a few restaurants fitting my criteria,” and ”The option to filter costs definitely. It also helped to see what kinds of foods they serve before clicking on the result.” On the other hand, Yelp users reported enjoying images and review contents, stating, ”[I liked] the ratings and images. It is good to look at the pictures and imagine my party there. I liked the ambiance and so I chose that restaurant.”

Fractal users were not such fans of the number of filter options provided, writing, ”[I disliked] the fact that it seemed to include a lot of options that weren't what you asked for,” and ”[I disliked] the volume of info!” One Fractal user reported performance issues (their user agent string identifies them as an Internet Explorer user). Yelp users, meanwhile, reported issues comparing restaurants (”There is no way to see restaurants side by side or add them to a compare list or a save for later list to choose from.”) and difficulty aggregating across multiple genres (”[I disliked] the inability to combine the search results for ice cream places and asian bistros.”) Interestingly, one user reported intense dislike over feeling that Yelp was manipulating the kinds of results and reviews they were seeing, writing, ”I felt that Yelp was filtering the restaurants for me and although they were good choices, I felt that Yelp was biased. I didn't know if that was the closest best restaurant I could have chosen.” Fractal users did not complain about model curation, though it is unclear to what extent they were aware of what was determining their view state (as opposed to popular culture discussion over Yelp content curation).

7.3 Discussion & Open Questions

Overall Fractal users reported feeling that they have covered more of the space of restaurants and grounded their answers in more data attributes. This suggests that Fractal users indeed were able to maintain their mental model despite changes in the Point Saliency model and across pan/zoom transitions. Many participant explanations that cite neighborhoods and an understanding of regions suggests that the partitioning scheme may in fact be aiding users in building an initial schema for their data. Participants were interested in the tool set and liked the ability to both filter and prioritize without losing information on the screen. However, even with a level of detail function controlling how much was on screen users still reported experiencing overload. It is unclear whether this is a result of the aggregation function needing more aggressive tuning to eliminate dense, potentially confusing regions of points or more lax tuning to represent more points without the extra interaction step of interrogating point groups.

In the future it would be valuable to explore this question of overload in more detail. Do domain experts still feel overload, and do they also benefit from the tool? How can we characterize clusters within Fractal, and do they align with other, crowdsourced descriptions of city neighborhoods [160]? How would introducing more of Kinetica/DataSquid's physics-based tools influence the exploration process (especially swapping between maps and plots)?

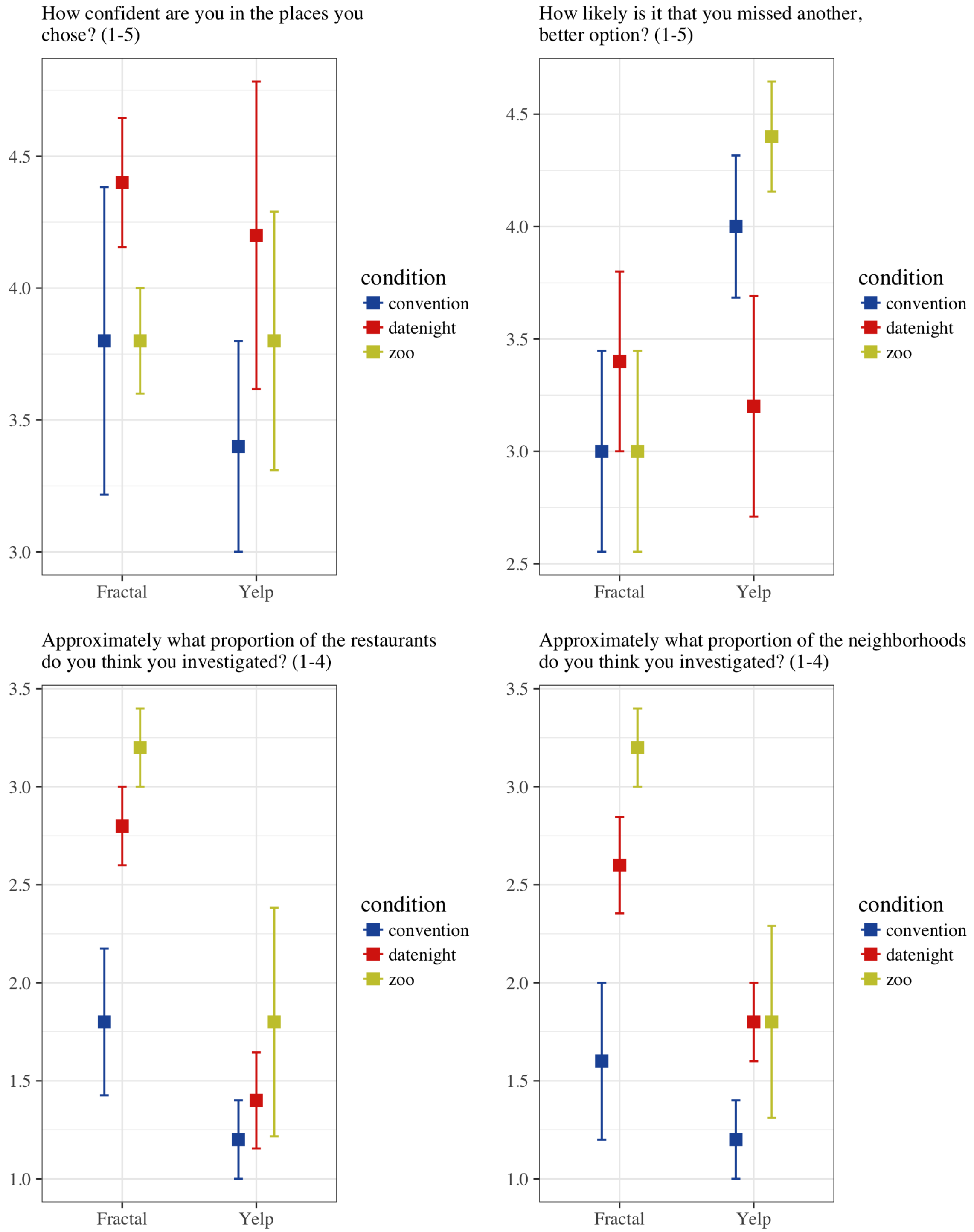


Figure 7.8: Results from the Fractal evaluation scenario answers.

8 — Contributions and Conclusion

Returning now, to the initial thesis introduced at the start of this document:

> **By dynamically balancing between users' stated (or inferred) interests and data features judged to be independently useful when compressing contextual information visually, users will be better able to make sense of complex and large scale data.** In the initial stages of data exploration, such compressed context will help a user forage for useful examples without tunneling into a narrow portion of the data. Further, applying this compression on smaller, local clusters instead of globally on an entire dataset will help to scaffold users' schematization of the data as they move from foraging into sensemaking and mitigate some of the cognitive and perceptual risks of visually grouping/compressing points.

I find proof for this claim in a number of findings in the previous two chapters. In my initial examination into the effectiveness of Point Saliency aggregation, there is evidence that the aggregation approach helped users quickly and readily identify individual examples. They were better able to pick out high valued points or points belonging to a rare category, especially if the user portion of the model took into account interest in those features. This suggests that in a real world case users would indeed be able to forage for interesting breakpoints and examples. This is reaffirmed in the Fractal user study, demonstrating that participants were able to quickly and effectively identify candidates to solve particular data exploration scenarios. The most telling dependent measures relate to participants' belief in how much data they have explored. Fractal participants reported a higher degree of belief that they found the best possible option and a much higher degree of coverage of the entire dataset as compared to the Yelp participants. Given that their answers were comparable if not better with the same level of confidence, this implies that Fractal users were more efficient *foragers* across the entire dataset.

There is less direct evidence for partitions' scaffolding of participants mental schema during the sensemaking process. Participants' use of neighborhoods, regionality, and a variety of data attributes might imply that they had a more complete mental model of the space of data, but the study did not produce (or measure) direct evidence of this fact. More work must be done to explore how visualizations might scaffold the post-foraging portions of Pirolli and Card's sensemaking model. One potential study might investigate how those with domain expertise and those without domain expertise make use of the partitions (as domain experts, presumably, already have a similar mental framework at the start of the task). Another might examine whether the scaffolding portions of the exploration task remain with users days or weeks after the task is complete, providing the compelling possibility that such visualizations might not only help a user complete and exploration but also gain transportable knowledge about a region or data space that can be applied to other, similar explorations.

Revisiting the contributions listed in the introduction, I can now outline the particular elements of this dissertation which ground and reinforce these claims:

- Introduce a hierarchical clustering pass to the focus+context process (akin to Schaffer [147]) that is designed to both spread information evenly across the screen and provide a scaffold upon which for users to develop initial sensemaking schema
  - Outlined algorithmically in Chapter 6 and evaluated with synthetic data in a study which suggests the evenly spread detail provided by the partitions indeed helped participants identify useful data spread across a geographic dataset
  - Given a visual metaphor and evaluated within a grounded, consumer search use case in Chapter 7. The study provides suggestive evidence that the clusters helped users better chunk and compare information across the dataset, but much more research remains to properly explore this effect.
  - This approach is transferrable across a number of visualization domains, even in cases where focus+context approaches are not directly applied. For instance, it may provide improvements over existing quad-tree based bundling algorithms used in mapping applications to combine overlapping annotations by aligning the bundling process more closely to conceptual relationships between the points rather than arbitrary, order-dependent bundles produced by quad-tree construction.
- Develop a DOI function, applied after this initial clustering, that incorporates weighted measures of user interest and a priori data saliency in multivariate data that uses considers both global- and cluster-level features to show more relevant trends between conceptually (or geographically) adjacent points.
  - Outlined algorithmically in Chapter 6 and evaluated with synthetic data in a study which suggests that DOI-based aggregation within partitions is effective at reducing overload while at the same time avoiding hiding detail such that a user develops a mistaken or incomplete understanding of the dataset
  - Given a visual metaphor and evaluated within a grounded, consumer search user case in Chapter 7. The study provides evidence that the aggregation does not lead to users feeling that they missed out on certain points, as instead lends more confidence and coverage to the process.
  - This DOI function is transferrable across any domain where user interest must be balanced with a priori data features. Combined with the top-down partitioning process, this approach allows detail to be spread more evenly across a visualization than global fisheye-style focus+context systems allow.
- Build a new visual metaphor for representing both cluster- and DOI-mediated point-level multivariate data by outlining clusters and selectively compressing points based on DOI

  - Outlined in Chapters 6 and 7, my design process suggests one potential way to represent aggregated multivariate data against a scaffold of data partitions. While much design thinking went into this particular representation of partitions (using soft outlines) and aggregations (using DataSquid-style points and subtle "+" marks rather than scaling elements), there are any number of ways to represent such data. It is my hope that this document will help others to explore this design space more fully rather than to assert that Fractal's representation is the best possible way to present focus+context multivariate data.

- Explore techniques for explaining why a DOI function is curating certain content for a user verbally and justifying recommendations and cluster divisions.

  - Outlined in Chapter 7 and explored later in this Chapter. I have introduced several approaches for summarizing point-level and partition-level data features in natural language. The Fractal user study suggests that these summary approaches did indeed help users make sense of data points in addition to raw data features.

  - Much more research remains to be done in terms of identifying which features are best for summary (which may be different from other DOI-determined features), how best to surface such summary information, and how it might influence the latter half of the sensemaking process by providing yet more scaffolding for schematization.

- Apply physics-based visualization techniques to the focus+context metaphor to disambiguate densely packed focal areas and more fluidly transition between changes in zoom (i.e. cluster level) and user interest (i.e. DOI function).

  - After demonstrating that physics-based techniques translate smoothly into map-based views in Chapter 5, integrated into Point Salience models as a way to reduce occlusion and smooth transitions in zoom and DOI changes. Experimental evidence in Chapter 6 and 7 suggests that physics-based techniques do indeed work within focus+context views.

  - Much more research remains in terms of translating other portions of the Kinetica/DataSquid toolset into the focus+context realm, and exploring how they might interact with a dynamic DOI function.

The initial Kinetica study demonstrated a few different ways that physics-based visualizations might help users. On one hand the touch-based interaction modality of Kinetica might lend a fluidity and inviting quality that helped novices better acclimate to the tool and the data. There is some evidence of this in the form of quick training times in both the Kinetica and DataSquid consumer studies and user qualitative responses. On the other hand, the fluid animations, naturalistic behavior of points, and transitions between interactions that allow users to map changes in data to their interactions likely also have a factor. We see some evidence of this in users' increased relationship and comparison findings in the Kinetica study and improved task time and self-reported qualitative findings in the DataSquid consumer study. Through commercialization of Kinetica as DataSquid I uncovered a variety of interesting use cases for novice-friendly interactive multivariate data visualization tools. Consumer search tasks such as choosing the best product or picking a vacation rental proved most compelling across test users. However, the consumer study's constrained, less than 200 point dataset highlights the need to handle data scale more gracefully.

Both of these studies raise the question of how to best manage differing scales of data more fluidly in physics-based visualizations (even the DataSquid prototype after several commercial iterations suffered at large scales). DataSquid's design at its core pushes for users to see contextual information at all times in the form of points bunching up, tools that avoid explicitly hiding points, interactions that afford quick exploration of data attributes (e.g. configuring filters immediately shows ramifications through brushing), and fluid transitions between states for all points charted. However, this comes at a significant cost. The maximalist approach of showing most or all points and placing them into an interactive physics environment significantly limits the amount of data that can be rendered performantly and can be interpreted by a user. If there are so many points on the screen that the physics model is forced to shift them out far of their preferred position on the chart substantially, then the user risks making a false conclusion about their data. If there are even more points (Figure 6.1), then there may be no sensible visualization at all emerging from the physics. Computationally collision and physics may only have efficiencies of $O(n^2)$ and $O(n^3)$ per frame when lots of activity is happening onscreen, which can be prohibitive if device capabilities are limited.

There are a number of potential ways to make scale more manageable coming out of prior research. Focus and context displays devote more screen space to more important areas of data. Dynamic querying lets users show and hide data rapidly to focus on smaller trends. Querying engines let users specify and display specific portions of data interactively using language or interactive tools. Motif simplification schemes convert complex or large scale portions of data into archetypal representations that are easier to understand. Others use clustering to add higher order motif-like summaries such as tag clouds for groups of textual data or visual summary statistics for multivariate data. Algorithms can reduce or aggregate multiple dimensions of data into fewer, more manageable dimensions. Some systems introduce approximate answers for large scale datasets while the rest of the data are being slowly processed.

In this document I proposed a hybrid of several of these approaches to help reduce scale in multivariate data while retaining some of the benefits found in physics-based visualizations. The Fractal system applies an objective function which balances more detail onscreen for user focus and less detail onscreen for user context. Additionally, the objective function also weights data features the user may not yet be interested in. This Point Salience mathematical model optimizes for both meaningfulness to user and usefulness of data, redefining the criteria for focal area and context as the user explores. This interactive approach helped users make better use of a large set of dimensions in a user study. Additionally, the Fractal system applies this focus and context approach to DataSquid's physics-based multivariate visualization techniques. Now, instead of solitary physics-based points colliding and responding, points may instead be grouped together if they are ruled as less interesting by the objective function. Since Fractal users can interact with data at different scales of detail through zooming and panning (another way to free up screen real estate to unpack points near the focal area), Fractal makes use of a higher level geographic abstraction to help ground users and show local, small-scale trends. This hierarchical geographic model delivers unsupervised clusters of consistent size, balanced for coverage of geography versus available screen real estate, which form and break apart sensibly as the user changes display scales. Applying this tool to a consumer search task, I identified several key ways it helped reduce complexity for new users and improve search times.

8.1 General Design Pattern

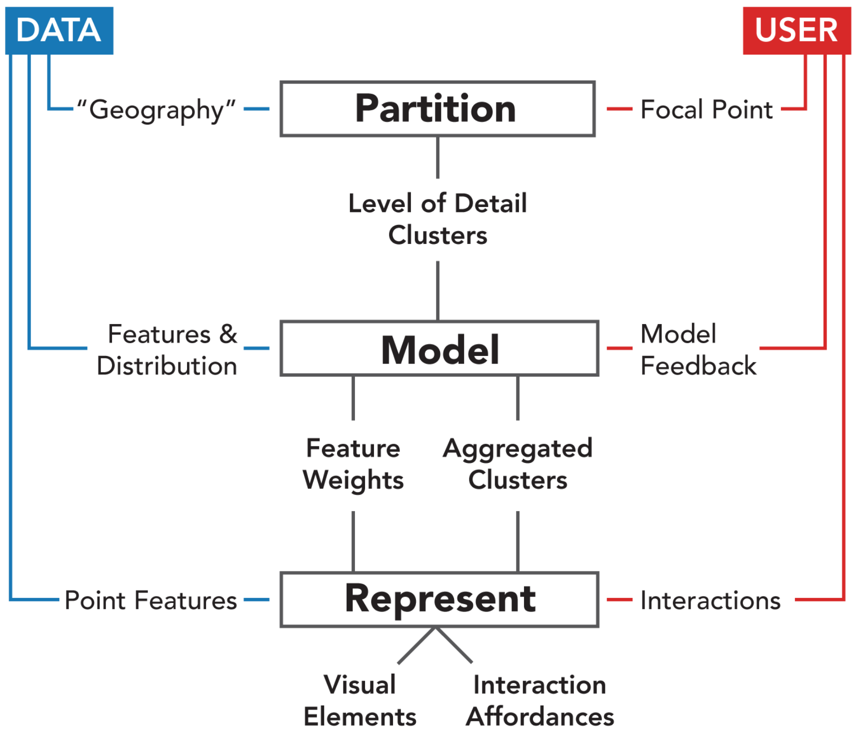


Figure 8.1: Overview of data and user interactions used in the Fractal interactive visualization

Moving towards a more general pattern, the Fractal prototype made use of three independent, interchangeable layers in order to deliver physics-based interactive visualizations to users (Figure 8.1). The topmost layer, partitioning, examines where the screen viewport and "geographical" features in the data to provide clusters or partitions of data that deliver the best granularity for users. In the case of a geographical or mappable dataset, the clusters might map to cities, neighborhoods, or blocks. In the case of a product dataset, the clusters might map to particular product segments such as consumer handheld cameras with certain price/sensor characteristics, pro-sumer models, or professional ones, or subcategories within those segments. In a dataset of personnel the clusters might instead map to divisions of a company. While the specific algorithm powering the generation of meaningful clusters at a proper scale may change, the system input/output is identical and the ultimate goal of providing a higher structure upon which for users to build their mental model remains the same. This begs the question of what sorts of partitioning approaches work within Fractal's design pattern? Would Fractal-for-NLP make use of LDA topic models to help break down the space of potential entities? Is there an analogous clustering approach for graph data? Even further, in Fractal the assumption was that the partitioner should evenly spread clusters across the dataset, balancing screen size versus cluster area. What if, instead, the partitioner used a fisheye magnification function or an entirely new objective function to split then space nonlinearly? Because the ultimate goal is to scaffold users' initial sensemaking of the data rather than optimize a particular cluster metric, it remains unclear what best practices exist for this level.

The partitioner feeds these clusters, in addition to info about user interest and data features, into a model that attempts to determine what parts of the data are/should be most meaningful for the user exploring the dataset. This model has a serious demand for bootstrapping, as the entire user half of the model may begin an exploration unpopulated. On the other hand, in the future perhaps crowdsourcing, induction from similar tasks, and collaborative generation could work to fill the user model initially. The model layer's ultimate goal is to guide the visualization towards meaningful bits and help to compress away similar elements or unhelpful noise. Because the visual compression of data is lossy, it is important that the model not overestimate the usefulness of some points without a high degree of certainty. While in Fractal the model layer was occupied by the Point Salience mathematical model that attempts to weight salient features more highly based on distribution and expressed user interest, other models might be substituted in their place. For instance, one could train a supervised learning model or recommender system that identifies important features based on past successful exploration logs. One also might instead use a more formal mathematical optimization to explicitly construct the aggregated clusters of points rather than iteratively and greedily combine them together. It remains unclear how accurate the model actually needs to be for users to see success. In my static evaluation and integration with Fractal the model feedback was imperfect but ultimately useful. Moving forward, it is important to understand how the costs of model false positives and false negatives differ, and other ways in which one might curate the data and clusters coming from the partitioner.

The representation layer takes clusters, raw data values, and weights from the model, representing them onscreen in an interactive fashion. In the case of Fractal this took the form of a web-based tool that used a pan/zoom map and a subset of Kinetica/DataSquid techniques to represent the data. At its core the representation must gracefully handle the aggregated data from the model, showing users where points may be bundled and where they may be intentionally split. The model must smoothly transition between detail and zoom states so that the user's mental model, as scaffolded by the partitioner, does not become disrupted as more or less information can be shown onscreen. Fractal's representer took the additional step of generating useful blurbs and summaries from model weights in order to give even more scaffolding of information. I've intentionally left this layer's definition vague because the space of potential representations is extraordinarily large. Fractal's visual layer may not work for a corpus of text, but it might translate well enough into time series data or graph data. Because the representation is the source and monitor for all interactions with the user (the top two layers are hidden if they do their job well), it must be designed well. In practice I can only point to iterative design and good design practices (some of which I hope I have achieved in Fractal's development) as a way to make a good outcome more likely. In the future it is key to better understand the space of potential representations and interactions across a variety of data.

8.2 Summaries, Suggestions, and Algorithmic Curation

Fractal made an explicit effort to offer summaries, one-line blurbs, and explanations for why certain items were prioritized over others. While these were unpolished compared to that of a conversational agent or natural language data visualization tool [54], they did offer users another hook for understanding their data. On one hand surfacing more information about a curation algorithm can help users feel more agency [45], but it is risky. One participant in the Fractal study expressed intense concern over Yelp's ability to choose which places they saw in their interface. A study of user perceptions of the Facebook News Feed algorithm suggests that many users experience concern that they are missing content or that seeing a single "miss" by the recommender belies a systemic problem that is violating their expectations on a daily basis [132]. Giving too much insight into the algorithm, though, can invite more speculation that leads to dissatisfaction [92].

It is unclear how much each component of Fractal might have evoked these sorts of concerns for users. No user recorded concerns about algorithms controlling the compression or presentation of information. In a preliminary study asking users to rate the quality of place- and cluster-level summaries, users generally rated them as decent but not perfect, and in preliminary ratings I could not observe any difference with human written blurbs (though this needs further study to be at all conclusive). It is quite possible that users, as Eslami et al. suggest, weren't aware of the extent to which algorithms controlled their viewport. Would users respond differently if the filter bar in Fractal were staged more as if the user was in dialogue with an agent controlling their view rather than stating preferences? Are there other design features one can include to better surface how the Point Salience model is aggregating information in different ways?

Computer-generated summaries bring up a broader question: "Can we accurately model what users need to know about a place/neighborhood from multivariate data features and a text corpus and then generate a useful blurb?" Studies of neighborhood guides suggest that statistical views are evocative for users in understanding neighborhoods but may not usefully capture the nuance of an area [159]. For instance, while the Fractal model might point to the restaurant Banh Mi and Ti's popularity and high ratings (and a textual model might even identify its most popular dish), a local could provide valuable comparisons to Lucy's, an unlisted seasonal banh mi stand, that is its greatest competition in the city. This sort of domain expertise is not necessarily encoded in the dataset. Crowdsourcing and collaborative content generation might help solve this problem, but then curating and displaying a large body of responses might grow to become an even greater challenge. In the future a design inquiry into the kinds of information that are most meaningful for different kinds of users and the trade-offs in its presentation may help guide these efforts.

8.3 Future of Physics-Based Visualization

One of the core findings from commercializing physics-based visualization technology was that the physics-based tools were not effective in all (or even many) particular use cases. Some prototypes such as the time series visualization introduced new metaphors that only fit a small set of use cases. The core elements of physics-based visualizations, fluidity and collision (which lends bunching up and distribution), seemed consistently valuable. It is for this reason that those two features were incorporated forward into the Fractal prototype. In one iteration of

Fractal the points were free to animatedly move at all times, sliding into new positions as users zoomed. Even this proved too complicated and time-consuming. While sieves that selectively allow points into regions were easily achievable within Fractal's back-end physics framework, I did not implement them because they do not translate well as data scale changes (zooming in on a filtered region quickly eliminates the valuable bunching of filtered-out points). In the quantitative evaluation of Point Salience compression schemes, collision helped users identify amount and single points more easily (though bundling points together resulted in further improvements). Even with many of the animations turned off in the final iteration of Fractal, the collision model gracefully handled the bundling/unbundling that resulted from changes in the user model (for instance, revealing more of a particular genre of interested), helping participants to identify the results of their searches (see Figure 6.13).

I see two courses of research moving forward. Continuing to pare down physics-based visualizations to their core elements and scaling them up to more and more users will help to refine our understanding of the particular benefits they provide beyond the studies outlined in this document. While this does not help to expand the design space of visualizations, it helps to establish best practices and transferable knowledge into other visualization tools. On the other hand, much of the initial framework for physics-based visualizations (Table 3.6) introduced in the Kinetica studies remains unexplored. I propose working to expand this set by carrying physics-based visualizations into new kinds of data. While a bit of physics is already used in graph visualization in the form of force-based layout algorithms, this domain is one good candidate for exploration. Continued commercial activities also help to further this goal by revealing new problems to solve for users, kinds of data, and use cases.

CONCLUSION

A wide variety of problems everyday users face involve data, yet many existing tools are not well matched to users' needs in under-constrained, ill defined, or poorly understand situations. Consumer search is one core example of this, as users rarely enter a search with a perfect set of criteria for which to evaluate options and a perfect understanding of the space of all options. Practically, tasks such as consumer search function more like sensemaking tasks than information retrieval tasks. Exploratory data visualization provides an excellent opportunity to help users make sense of these kinds of data. However, many data visualization tools are also ill-suited for novice or inexperienced users.

In this document I outline a general approach that uses physics to help novice users make sense of complex trends and data distribution within multivariate datasets. The key insight in physics-based visualizations is that one can map data exploration actions to physical metaphors that users will intuitively understand. Investigating these approaches as a pure data exploration task, through a commercialization, and in a consumer search context, I've demonstrated that they do provide users an advantage over traditional visualizations, but that they may have acute limitations in terms of application and scale. I introduce a new approach for managing focus and context in physics-based multivariate data visualizations, and establish its effectiveness in isolation as a visualization technique. Finally, I explore its utility in a user study grounded in real world problems using a working, web-based implementation.

While I do not propose a definite solution for consumer search in this document or identify a perfect best-practice for the area, I identify several key areas of need within data visualization coming out of DataSquid's commercialization and envision one possible solution to some of these needs. I hope that in the future we will be able to answer some of these needs with the help of physics-based tools, models of data- and user-interest, and scaffolding sensemaking with level of detail clustering.

Bibliography

[1] Abouzied, A., Hellerstein, J., and Silberschatz, A. Dataplay: interactive tweaking and example-driven correction of graphical database queries. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 207–218. ACM, 2012. 6.1

[2] Adhikarla, V.K., et al. Freehand interaction with large-scale 3d map data. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2014*, pp. 1–4. IEEE, 2014. 6.1

[3] Agarawala, A. and Balakrishnan, R. Keepin'it real: pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 1283–1292. ACM, 2006. 3.1.1

[4] Ahlberg, C. and Shneiderman, B. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 313–317. ACM, 1994. 3.1.1, 5, 6.1

[5] Ahlberg, C., Williamson, C., and Shneiderman, B. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 619–626. ACM, 1992. 3.1.1

[6] Ahlberg, C. and Wistrand, E. Ivee: An information visualization and exploration environment. In *Information Visualization, 1995. Proceedings.*, pp. 66–73. IEEE, 1995. 3.1.1

[7] Amar, R.A. and Stasko, J.T. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, 2005. 6

[8] Anderson, E.W., et al. A user study of visualization effectiveness using eeg and cognitive load. In *Computer Graphics Forum*, vol. 30, pp. 791–800. Wiley Online Library, 2011. 3.1.1

[9] Baudisch, P., Good, N., Bellotti, V., and Schraedley, P. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 259–266. ACM, 2002. 6.1

[10] Baudisch, P., Lee, B., and Hanna, L. Fishnet, a fisheye web browser with search term popouts: a comparative evaluation with overview and linear view. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 133–140. ACM, 2004. 6.1

[11] Baudisch, P. and Rosenholtz, R. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 481–488. ACM, 2003. 6.1

[12] Baur, D., Lee, B., and Carpendale, S. Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, pp. 255–264. ACM, 2012. 3.1.1

[13] Becker, R.A. and Cleveland, W.S. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987. 3.1.1

[14] Becker, R.A., Cleveland, W.S., and Shyu, M.J. The visual design and control of trellis display. *Journal of computational and Graphical Statistics*, 5(2):123–155, 1996. 3.1.1

[15] Bederson, B.B. and Hollan, J.D. Pad++: a zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pp. 17–26. ACM, 1994. 6.1

[16] Bertini, E. and Santucci, G. Give chance a chance: modeling density to enhance scatter plot quality through random data sampling. *Information Visualization*, 5(2):95–110, 2006. 6.1

[17] Blank, S., Blank, S., and Dorf, B. *The Startup Owner's Manual: The Step-by-step Guide for Building a Great Company*. v. 1. K&S Ranch, Incorporated, 2012. URL https://books.google.com/books?id=MT-TtgAACAAJ. 4.1, 4.2

[18] Bloch, P.H., Sherrell, D.L., and Ridgway, N.M. Consumer search: An extended framework. *Journal of consumer research*, 13(1):119–126, 1986. 5.1, 7, 7.1.2

[19] Borgman, C.L. Why are online catalogs hard to use? lessons learned from information-retrieval studies. *Journal of the American society for information science*, 37(6):387–400, 1986. 3.1.1

[20] Buchanan, R. Wicked problems in design thinking. *Design issues*, 8(2):5–21, 1992. 3.3

[21] Buenstorf, G. Is commercialization good or bad for science? individual-level evidence from the max planck society. *Research Policy*, 38(2):281–292, 2009. 4.1

[22] Büring, T., Gerken, J., and Reiterer, H. User interaction with scatterplots on small screens-a comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS*, 12(5), 2006. 6.1

[23] Burstyn, J., Banerjee, A., and Vertegaal, R. Flexview: an evaluation of depth navigation on deformable mobile devices. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*, pp. 193–200. ACM, 2013. 6.1

[24] Bush, V. and Think, A.W.M. As we may think. *The atlantic monthly*, 176(1):101–108, 1945. 3.1.1

[25] Caplin, A., Dean, M., and Martin, D. Search and satisficing. *The American Economic Review*, pp. 2899–2922, 2011. 1

[26] Card, S.K., Mackinlay, J.D., and Shneiderman, B. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999. 3.1.2

[27] Carnegie mellon university. developing ipad apps course - student project demos. URL https://youtu.be/rczB_cOWsiQ. (document), 3.3

[28] Carpendale, S. Evaluating information visualizations. In *Information Visualization*, pp. 19–45. Springer, 2008. 3.1.2

[29] Chau, D.H., Kittur, A., Hong, J.I., and Faloutsos, C. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 167–176. ACM, 2011. 3.1.2

[30] Chesbrough, H. and Rosenbloom, R.S. The role of the business model in capturing value from innovation: evidence from xerox corporation's technology spin-off companies. *Industrial and corporate change*, 11(3):529–555, 2002. 4.1

[31] Chevalier, J.A. and Mayzlin, D. The effect of word of mouth on sales: Online book reviews. *Journal of marketing research*, 43(3):345–354, 2006. 5.1

[32] Chilana, P.K., Ko, A.J., and Wobbrock, J. From user-centered to adoption-centered design: a case study of an hci research innovation becoming a product. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1749–1758. ACM, 2015. 4.1, 4.2

[33] Chilana, P.K., et al. Technology transfer of hci research innovations: Challenges and opportunities. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15, pp. 823–828. ACM, New York, NY, USA, 2015. URL http://doi.acm.org/10.1145/2702613.2724724. 4.1

[34] Clarysse, B. and Moray, N. A process study of entrepreneurial team formation: the case of a research-based spin-off. *Journal of Business Venturing*, 19(1):55–79, 2004. 4.1

[35] Cleveland, W.S. and McGill, R. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984. 3.1.1

[36] Cockburn, A., Karlson, A., and Bederson, B.B. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)*, 41(1):2, 2008. 6.1

[37] Collen, M.F. Origins of medical informatics. *Western Journal of Medicine*, 145(6):778, 1986. 3.1.1

[38] Dai, P., Rzeszotarski, J.M., Paritosh, P., and Chi, E.H. And now for something completely different: Improving crowdsourcing workflows with micro-diversions. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pp. 628–638. ACM, 2015. 2

[39] Dontcheva, M., Drucker, S.M., Salesin, D., and Cohen, M.F. Relations, cards, and search templates: user-guided web data integration and layout. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 61–70. ACM, 2007. 7.1.3

[40] Drucker, S.M., et al. Touchviz: a case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2301–2310. ACM, 2013. 3.1.1, 1

[41] Dumais, S., et al. Stuff i've seen: a system for personal information retrieval and re-use. In *ACM SIGIR Forum*, vol. 49, pp. 28–35. ACM, 2016. 1

[42] Dumas, B., Moerman, B., Trullemans, S., and Signer, B. Artvis: combining advanced visualisation and tangible interaction for the exploration, analysis and browsing of digital artwork collections. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pp. 65–72. ACM, 2014. 3.1.1

[43] Elmqvist, N. and Fekete, J.D. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *Visualization and Computer Graphics, IEEE Transactions on*, 16(3):439–454, 2010. 6.1

[44] Endert, A., Fiaux, P., and North, C. Semantic interaction for sensemaking: inferring analytical reasoning for model steering. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2879–2888, 2012. 3.1.2

[45] Eslami, M., et al. First i like it, then i hide it: Folk theories of social feeds. In *Proceedings of the 2016 cHI conference on human factors in computing systems*, pp. 2371–2382. ACM, 2016. 8.2

[46] Etzkowitz, H. The norms of entrepreneurial science: cognitive effects of the new university–industry linkages. *Research policy*, 27(8):823–833, 1998. 4.1

[47] Faisal, S., Craft, B., Cairns, P., and Blandford, A. Internalization, qualitative methods, and evaluation. In *Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, p. 5. ACM, 2008. 3.1.2

[48] Fisher, D., Popov, I., Drucker, S., et al. Trust me, i'm partially right: incremental visualization lets analysts explore large datasets faster. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1673–1682. ACM, 2012. 6.1

[49] Frisch, M., Kleinau, S., Langner, R., and Dachselt, R. Grids & guides: multi-touch layout and alignment tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1615–1618. ACM, 2011. 3.1.1

[50] Frisch, M., Langner, R., and Dachselt, R. Neat: a set of flexible tools and gestures for layout tasks on interactive displays. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 1–10. ACM, 2011. 3.1.1

[51] Fruchterman, T.M. and Reingold, E.M. Graph drawing by force-directed placement. *Softw., Pract. Exper.*, 21(11):1129–1164, 1991. 3.1.1

[52] Furnas, G.W. The fisheye view: A new look at structured files, 1981. 1.1, 6, 6.1

[53] Furnas, G.W. A fisheye follow-up: further reflections on focus+ context. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 999–1008. ACM, 2006. 1.1, 6, 6.1, (document)

[54] Gao, T., et al. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 489–500. ACM, 2015. 8.2

[55] Goldfarb, B. and Henrekson, M. Bottom-up versus top-down policies towards the commercialization of university intellectual property. *Research policy*, 32(4):639–658, 2003. 4.1

[56] Google. Youtube statistics. https://www.youtube.com/yt/press/statistics.html. Accessed: 2016-01-01. 1

[57] Greenberg, S. and Buxton, B. Usability evaluation considered harmful (some of the time). In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 111–120. ACM, 2008. 3.3

[58] Grossman, T. and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 281–290. ACM, 2005. 7.1.1

[59] Gutwin, C. and Fedak, C. Interacting with big interfaces on small screens: a comparison of fisheye, zoom, and panning techniques. In *Proceedings of Graphics Interface 2004*, pp. 145–152. Canadian Human-Computer Communications Society, 2004. 6.1

[60] Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003. 6.4.1

[61] Haag, F., Lohmann, S., Bold, S., and Ertl, T. Visual sparql querying based on extended filter/flow graphs. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pp. 305–312. ACM, 2014. 4.2.2

[62] Haag, F., Lohmann, S., Siek, S., and Ertl, T. Visual querying of linked data with queryvowl. In *SumPre-HSWI@ ESWC*. 2015. 4.2.2

[63] Halfaker, A., Geiger, R.S., Morgan, J.T., and Riedl, J. The rise and decline of an open collaboration system: How wikipedias reaction to popularity is causing its decline. *American Behavioral Scientist*, p. 0002764212469365, 2012. 1

[64] Halfaker, A., Kittur, A., and Riedl, J. Don't bite the newbies: how reverts affect the quantity and quality of wikipedia work. In *Proceedings of the 7th international symposium on wikis and open collaboration*, pp. 163–172. ACM, 2011. 1, 2

[65] Hancock, M., Ten Cate, T., and Carpendale, S. Sticky tools: full 6dof force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 133–140. ACM, 2009. 3.1.1

[66] Harrison, C., Xiao, R., Schwarz, J., and Hudson, S.E. Touchtools: leveraging familiarity and skill with physical tools to augment touch interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2913–2916. ACM, 2014. 3.1.1

[67] Hauser, D.J. and Schwarz, N. Attentive turkers: Mturk participants perform better on online attention checks than do subject pool participants. *Behavior research methods*, 48(1):400–407, 2016. 6.6.2

[68] Hauser, H., Ledermann, F., and Doleisch, H. Angular brushing of extended parallel coordinates. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pp. 127–130. IEEE, 2002. 3.1.1

[69] Healey, C.G. and Enns, J.T. Attention and visual memory in visualization and computer graphics. *Visualization and Computer Graphics, IEEE Transactions on*, 18(7):1170–1188, 2012. 3.2

[70] Heer, J. and Card, S.K. Doitrees revisited: scalable, space-constrained visualization of hierarchical data. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 421–424. ACM, 2004. 6.1

[71] Henderson, A. The innovation pipeline: Design collaborations between research and development. *interactions*, 12(1):24–29, Jan. 2005. URL http://doi.acm.org/10.1145/1041280.1041295. 4.1

[72] Hiltzik, M.A. et al. *Dealers of lightning: Xerox PARC and the dawn of the computer age*. HarperCollins Publishers, 1999. 4.1

[73] Hochheiser, H. and Shneiderman, B. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004. 3.1.1

[74] Huron, S., Vuillemot, R., and Fekete, J.D. Visual sedimentation. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2446–2455, 2013. 3.1.1

[75] Hutchinson, H., et al. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 17–24. ACM, 2003. 4.2

[76] Inselberg, A. and Dimsdale, B. Parallel coordinates. In *Human-Machine Interactive Systems*, pp. 199–233. Springer, 1991. 3.1.1

[77] Isaacs, E.A., et al. Technology transfer: So much research, so few good products. In *Conference Companion on Human Factors in Computing Systems*, CHI '96, pp. 155–156. ACM, New York, NY, USA, 1996. URL http://doi.acm.org/10.1145/257089.257227. 4.1

[78] Isenberg, P., et al. Data visualization on interactive surfaces: A research agenda. *IEEE Computer Graphics and Applications*, 33(2):16–24, 2013. 3.1.1

[79] Ishii, H. and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pp. 234–241. ACM, 1997. 4.1

[80] Jacob, R.J., Ishii, H., Pangaro, G., and Patten, J. A tangible interface for organizing information using a grid. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 339–346. ACM, 2002. 3.1.1

[81] Jacob, R.J., et al. Reality-based interaction: a framework for post-wimp interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 201–210. ACM, 2008. 3.1.1

[82] Jetter, H.C., Reiterer, H., and Geyer, F. Blended interaction: understanding natural human–computer interaction in post-wimp interactive spaces. *Personal and Ubiquitous Computing*, 18(5):1139–1158, 2014. 3.1.1

[83] Jiang, L. and Nandi, A. Snaptoquery: providing interactive feedback during exploratory query specification. *Proceedings of the VLDB Endowment*, 8(11):1250–1261, 2015. 6.1

[84] Jog, N.K. and Shneiderman, B. Starfield visualization with interactive smooth zooming. In *Visual Database Systems 3*, pp. 3–14. Springer, 1995. 3.1.1

[85] Jul, S. and Furnas, G.W. Critical zones in desert fog: aids to multiscale navigation. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pp. 97–106. ACM, 1998. 6.1

[86] Kandel, S., Paepcke, A., Hellerstein, J., and Heer, J. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3363–3372. ACM, 2011. 6.1

[87] Kandel, S., et al. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 547–554. ACM, 2012. 6.1

[88] Keefe, D.F. and Isenberg, T. Reimagining the scientific visualization interaction paradigm. *Computer*, pp. 51–57, 2013. 3.1.1

[89] Kensing, F. and Blomberg, J. Participatory design: Issues and concerns. *Computer Supported Cooperative Work (CSCW)*, 7(3-4):167–185, 1998. 3.3

[90] Key, A., Howe, B., Perry, D., and Aragon, C. Vizdeck: self-organizing dashboards for visual analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 681–684. ACM, 2012. 6.1

[91] Kiczales, G., et al. Aspect-oriented programming. *ECOOP'97Object-oriented programming*, pp. 220–242, 1997. 4.1

[92] Kizilcec, R.F. How much information?: Effects of transparency on trust in an algorithmic interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2390–2395. ACM, 2016. 8.2

[93] Klein, L.R. and Ford, G.T. Consumer search for information in the digital age: An empirical study of prepurchase search for automobiles. *Journal of interactive Marketing*, 17(3):29–49, 2003. 5.1, 5.3

[94] Klemmer, S.R., Hartmann, B., and Takayama, L. How bodies matter: five themes for interaction design. In *Proceedings of the 6th conference on Designing Interactive systems*, pp. 140–149. ACM, 2006. 3.1.1

[95] Kolko, J. Running an entrepreneurial pilot to identify value. *interactions*, 21(4):22–23, 2014. 4.1

[96] Kosara, R., Bendix, F., and Hauser, H. Parallel sets: Interactive exploration and visual analysis of categorical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(4):558–568, 2006. 3.1.1

[97] Kruger, J. and Dunning, D. Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of personality and social psychology*, 77(6):1121, 1999. 5.1

[98] Krüger, J., Schneider, J., and Westermann, R. Clearview: An interactive context preserving hotspot visualization technique. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):941–948, 2006. 6.1

[99] Lam, H., et al. Empirical studies in information visualization: Seven scenarios. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1520–1536, 2012. 3.1.2, 4.1

[100] Lamping, J., Rao, R., and Pirolli, P. A focus+ context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 401–408. ACM Press/Addison-Wesley Publishing Co., 1995. 6.1

[101] Lasecki, W.S., Rzeszotarski, J.M., Marcus, A., and Bigham, J.P. The effects of sequence and delay on crowd work. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1375–1378. ACM, 2015. 2

[102] Lee, B., Isenberg, P., Riche, N.H., and Carpendale, S. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2689–2698, 2012. 3.1.1

[103] Lee, B., Kazi, R.H., and Smith, G. Sketchstory: Telling more engaging stories with data through freeform sketching. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2416–2425, 2013. 3.1.1

[104] Lee, M.K., Kusbit, D., Metsky, E., and Dabbish, L. Working with machines: The impact of algorithmic and data-driven management on human workers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1603–1612. ACM, 2015. 7.1.2

[105] Lee, S., Buxton, W., and Smith, K. A multi-touch three dimensional touch-sensitive tablet. In *Acm Sigchi Bulletin*, vol. 16, pp. 21–25. ACM, 1985. 4.1

[106] Lembcke, S. Chipmunk physics. https://github.com/slembcke/Chipmunk2D. Accessed: 2016-01-01. 3.4

[107] Li, Q. and North, C. Empirical comparison of dynamic query sliders and brushing histograms. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pp. 147–153. IEEE, 2003. 3.1.1

[108] Lim, Y.K., Stolterman, E., and Tenenberg, J. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 15(2):7, 2008. 3.3

[109] Lindley, J., Coulton, P., and Sturdee, M. Implications for adoption. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 265–277. ACM, 2017. 4.1

[110] Lins, L., Klosowski, J.T., and Scheidegger, C. Nanocubes for real-time exploration of spatiotemporal datasets. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2456–2465, 2013. 6.1

[111] Liu, Z., Jiang, B., and Heer, J. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, vol. 32, pp. 421–430. Wiley Online Library, 2013. 6.1

[112] Maitlis, S. and Sonenshein, S. Sensemaking in crisis and change: Inspiration and insights from weick (1988). *Journal of Management Studies*, 47(3):551–580, 2010. 3.1.2, 3.2

[113] Martin, A.R. and Ward, M.O. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th Conference on Visualization'95*, p. 271. IEEE Computer Society, 1995. 3.1.1

[114] Michaels, G.S., et al. Cluster analysis and data visualization of large-scale gene expression data. In *Pacific symposium on biocomputing*, vol. 3, pp. 42–53. 1998. 6.1

[115] Mowery, D.C. and Simcoe, T. Is the internet a us invention?an economic and technological history of computer networking. *Research Policy*, 31(8):1369–1387, 2002. 4.1

[116] Nandi, A., Jiang, L., and Mandel, M. Gestural query specification. *Proceedings of the VLDB Endowment*, 7(4):289–300, 2013. 6.1

[117] Norman, D.A. Technology first, needs last: the research-product gulf. *interactions*, 17(2):38–42, 2010. 4.1, 4.1

[118] North, C. Toward measuring visualization insight. *Computer Graphics and Applications, IEEE*, 26(3):6–9, 2006. 3.1.2

[119] North, C., et al. Understanding multi-touch manipulation for surface computing. In *Human-Computer Interaction–INTERACT 2009*, pp. 236–249. Springer, 2009. 3.1.1

[120] Novotny, M. and Hauser, H. Outlier-preserving focus+ context visualization in parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):893–900, 2006. 3.1.1

[121] Odom, W., et al. From research prototype to research product. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 2549–2561. ACM, 2016. 4.1

[122] Orleans, P. Differential cognition of urban residents: effects of social scale on mapping. *Image and environment*, pp. 115–130, 1973. 6.3.1

[123] Parameswaran, A., Polyzotis, N., and Garcia-Molina, H. Seedb: Visualizing database queries efficiently. *Proceedings of the VLDB Endowment*, 7(4):325–328, 2013. 6.1

[124] Perer, A. and Shneiderman, B. Integrating statistics and visualization: case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 265–274. ACM, 2008. 3.1.2

[125] Perkmann, M., et al. Academic engagement and commercialisation: A review of the literature on university–industry relations. *Research policy*, 42(2):423–442, 2013. 4.1

[126] Pirolli, P. and Card, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, vol. 5, pp. 2–4. 2005. 1, 1.1, 3.1.2, 6, (document)

[127] Pirolli, P., Card, S.K., and Van Der Wege, M.M. Visual information foraging in a focus+ context visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 506–513. ACM, 2001. 6.1

[128] Plaisant, C., Fekete, J.D., and Grinstein, G. Promoting insight-based evaluation of visualizations: From contest to benchmark repository. *Visualization and Computer Graphics, IEEE Transactions on*, 14(1):120–134, 2008. 3.1.2

[129] Pu, P. and Chen, L. Integrating tradeoff support in product search tools for e-commerce sites. In *Proceedings of the 6th ACM conference on Electronic commerce*, pp. 269–278. ACM, 2005. 5.1

[130] Punj, G.N. and Staelin, R. A model of consumer information search behavior for new automobiles. *Journal of consumer research*, 9(4):366–380, 1983. 5.1, 5.3

[131] Qu, Y. and Furnas, G.W. Model-driven formative evaluation of exploratory search: A study under a sensemaking framework. *Information Processing & Management*, 44(2):534–555, 2008. 3.1.2

[132] Rader, E. and Gray, R. Understanding user beliefs about algorithmic curation in the facebook news feed. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 173–182. ACM, 2015. 7.1.2, 8.2

[133] Roberts, J.C. State of the art: Coordinated & multiple views in exploratory visualization. In *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on*, pp. 61–71. IEEE, 2007. 3.1.1

[134] Robertson, G.G. and Mackinlay, J.D. The document lens. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pp. 101–108. ACM, 1993. 1.1, 6.1

[135] Rosa, P. and Dawson, A. Gender and the commercialization of university science: aca-

demic founders of spinout companies. *Entrepreneurship and Regional Development*, 18(4):341–366, 2006. 4.1

[136] Rosenbaum, R. and Schumann, H. Progressive refinement: more than a means to overcome limited bandwidth. In *IS&T/SPIE Electronic Imaging*, pp. 72430I–72430I. International Society for Optics and Photonics, 2009. 6.1

[137] Russell, D.M., Stefik, M.J., Pirolli, P., and Card, S.K. The cost structure of sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pp. 269–276. ACM, 1993. 3.1.2, 6, 7

[138] Rzeszotarski, J. and Kittur, A. Crowdscape: interactively visualizing user behavior and output. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 55–62. ACM, 2012. 2, 6.4.2

[139] Rzeszotarski, J. and Kittur, A. Learning from history: predicting reverted work at the word level in wikipedia. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp. 437–440. ACM, 2012. 2

[140] Rzeszotarski, J.M. and Kittur, A. Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 13–22. ACM, 2011. 2

[141] Rzeszotarski, J.M. and Kittur, A. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 897–906. ACM, 2014. 3, 3.4, 3.5

[142] Sadana, R. and Stasko, J. Designing and implementing an interactive scatterplot visualization for a tablet computer. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pp. 265–272. ACM, 2014. 3.1.1

[143] Sarkar, M. and Brown, M.H. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 83–91. ACM, 1992. 6.1

[144] Sarkar, M., Snibbe, S.S., Tversky, O.J., and Reiss, S.P. Stretching the rubber sheet: a metaphor for viewing large layouts on small screens. In *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pp. 81–91. ACM, 1993. 1.1, 6.1

[145] Satyanarayan, A. and Heer, J. Lyra: An interactive visualization design environment. In *Computer Graphics Forum*, vol. 33, pp. 351–360. Wiley Online Library, 2014. 4.2.2

[146] Satyanarayan, A., Russell, R., Hoffswell, J., and Heer, J. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE transactions on visualization and computer graphics*, 22:659–668, 2016. 4.2.2

[147] Schaffer, D., et al. Navigating hierarchically clustered networks through fisheye and full-zoom methods. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(2):162–188, 1996. 1.1, 6, 6, (document)

[148] Schmidt, S., Nacenta, M.A., Dachselt, R., and Carpendale, S. A set of multi-touch graph interaction techniques. In *ACM International Conference on Interactive Tabletops and Surfaces*, pp. 113–116. ACM, 2010. 3.1.1

[149] Sefelin, R., Tscheligi, M., and Giller, V. Paper prototyping-what is it good for?: a comparison of paper-and computer-based low-fidelity prototyping. In *CHI'03 extended abstracts on Human factors in computing systems*, pp. 778–779. ACM, 2003. 3.3

[150] Shneiderman, B. direct manipulation: a step beyond programming languages. *Sparks of innovation in human-computer interaction*, 17:1993, 1993. 1.1, 3.1.1

[151] Small, D. *Expressive typography: high quality dynamic and responsive typography in the electronic environment*. Ph.D. thesis, Massachusetts Institute of Technology, 1990. 3.1.1

[152] Smith, D.C., et al. Designing the star user interface. In *Human-computer interaction*, pp. 653–661. Morgan Kaufmann Publishers Inc., 1987. 3.1.1

[153] Spence, R. and Tweedie, L. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 11(2):137–146, 1998. 1.1, 5.2

[154] Stasko, J. and Zhang, E. Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pp. 57–65. IEEE, 2000. 6.5b, 6.1

[155] Stolper, C.D., Perer, A., and Gotz, D. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):1653–1662, 2014. 6.1

[156] Stolte, C., Tang, D., and Hanrahan, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):52–65, 2002. 4.2.2

[157] Tableau. Tableau software. http://www.tableau.com/. Accessed: 2016-01-01. 3.2, 6.1

[158] Taher, F., et al. Exploring interactions with physically dynamic bar charts. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3237–3246. ACM, 2015. 3.1.1

[159] Tasse, D., Chou, J.T., and Hong, J.I. Generating neighborhood guides from social media. In *Tenth International AAAI Conference on Web and Social Media*. 2016. 8.2

[160] Tasse, D. and Hong, J.I. Using user-generated content to understand cities. In *Seeing Cities Through Big Data*, pp. 49–64. Springer, 2017. 7.3

[161] Thacker, C.P., MacCreight, E., and Lampson, B.W. *Alto: A personal computer*. Xerox, Palo Alto Research Center Palo Alto, 1979. 4.1

[162] Tufte, E.R. and Graves-Morris, P. *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 1983. 3.1.1

[163] Tweedie, L., Spence, R., Dawkes, H., and Su, H. Externalising abstract mathematical models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 406–ff. ACM, 1996. 3.1.1

[164] Ullmer, B., Ishii, H., and Jacob, R.J. Tangible query interfaces: Physically constrained tokens for manipulating database queries. In *Proc. of INTERACT*, vol. 3, pp. 279–286. 2003. 3.1.1

[165] Van den Elzen, S. and Van Wijk, J.J. Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2310–2319, 2014. (document), 6.3

[166] Van Ham, F. and Perer, A. search, show context, expand on demand: supporting large graph exploration with degree-of-interest. *IEEE Transactions on Visualization and Computer Graphics*, 15(6), 2009. 1.1, 6, (document)

[167] Vohora, A., Wright, M., and Lockett, A. Critical junctures in the development of university high-tech spinout companies. *Research policy*, 33(1):147–175, 2004. (document), 4.1

[168] Walny, J., et al. Understanding pen and touch interaction for data exploration on interactive whiteboards. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2779–2788, 2012. 3.1.1

[169] Wang, L., Zhao, Y., Mueller, K., and Kaufman, A. The magic volume lens: An interactive focus+ context technique for volume rendering. In *Visualization, 2005. VIS 05. IEEE*, pp. 367–374. IEEE, 2005. 6.1

[170] Ward, M.O. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the Conference on Visualization'94*, pp. 326–333. IEEE Computer Society Press, 1994. 3.1.1

[171] Ward, M.R. and Lee, M.J. Internet shopping, consumer search and product branding. *Journal of product & brand management*, 9(1):6–20, 2000. 5.1

[172] Weick, K.E. *Sensemaking in organizations*, vol. 3. Sage, 1995. 3.1.2

[173] Weiser, M., Gold, R., and Brown, J. The origins of ubiquitous computing research at parc in the late 1980s. *IBM SYSTEMS JOURNAL*, 38(4):694, 1999. 4.1

[174] Wickens, C.D. Processing resources and attention. *Multiple-task performance*, pp. 3–34, 1991. 3.1.2

[175] Wikimedia. Wikipedia: Statistics. https://en.wikipedia.org/wiki/Wikipedia:Statistics. Accessed: 2016-01-01. 1

[176] Williamson, C. and Shneiderman, B. The dynamic homefinder: Evaluating dynamic queries in a real-estate information exploration system. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 338–346. ACM, 1992. 3.1.1

[177] Wilson, A.D., et al. Bringing physics to the surface. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pp. 67–76. ACM, 2008. 3.1.1

[178] Wobbrock, J.O., Morris, M.R., and Wilson, A.D. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1083–1092. ACM, 2009. 3.1.1

[179] Wong, P.C. and Bergeron, R.D. 30 years of multidimensional multivariate visualization. In *Scientific Visualization*, pp. 3–33. 1994. 3.1

[180] Wongsuphasawat, K., et al. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics*, 22(1):649–658, 2016. 4.2.2

[181] Wu, M. and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pp. 193–202. ACM, 2003. 3.1.1

[182] Yi, J.S., Kang, Y.a., Stasko, J.T., and Jacko, J.A. Understanding and characterizing insights: how do people gain insights using information visualization? In *Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization*, p. 4. ACM, 2008. 3.1.2

[183] Yi, J.S., Melton, R., Stasko, J., and Jacko, J.A. Dust & magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005. (document), 3.1, 3.1.1, 3.2.1, 3.3

[184] Yost, B., Haciahmetoglu, Y., and North, C. Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 101–110. ACM, 2007. 6.1

[185] Yun, M.H., Han, S.H., Hong, S.W., and Kim, J. Incorporating user satisfaction into the look-and-feel of mobile phone design. *Ergonomics*, 46(13-14):1423–1440, 2003. 3.2, 3.2.1

[186] Zhang, J., et al. Topogroups: Context-preserving visual illustration of multi-scale spatial aggregates. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 2940–2951. ACM, 2017. 6.3.1

[187] Zillow. Zillow, inc. reports record third quarter 2014 results. http://investors.zillowgroup.com/releasedetail.cfm?releaseid=880857. Accessed: 2016-01-01. 1

[188] Zimmerman, J., Forlizzi, J., and Evenson, S. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 493–502. ACM, 2007. 3.3