

# VizRec 2.0: Towards a holistic workflow for visual data exploration

Doris Jung-Lin Lee, Aditya Parameswaran  
{jlee782,adityagp}@illinois.edu  
University of Illinois, Urbana-Champaign

## Abstract

*Abstract Placeholder*

## 1 Introduction

In this paper, we advocate supporting a desiderata consisting of 3‘T’s in the cycle of visual data exploration:

- **Informative:** Section 2 discusses how precise visual query systems provide informative visualizations to accelerate the process of data discovery.
- **Interactive and iterative:** Section 4 Joining the flow, query refinement, dialogue (not a one-shot query), feedback (latter is quite unexplored), and recommendation, expressivity (how easy is it to express what to do via interactions) and diversity of actions that could be performed.
- **Integrated:** Section 5 discusses the challenges and opportunities in moving from intention-driven querying to facilitating more integrated data understanding and awareness during the analysis workflow.

## 2 Precise Visual Querying

Visual analysis often reveal important anomalies or trends in the data[9]. However, it is often challenging to find the appropriate piece of information to realize these insights.

### 2.1 Motivating Example

Astronomers from the The Dark Energy Survey (DES) are interested in finding anomalous time series to discover astrophysical transients (objects whose brightness changes dramatically as a function of time), such as supernova explosions or quasars [3]. When trying to find celestial objects corresponding to supernovae, which have a specific pattern of brightness over time, scientists need to individually inspect the visualizations of each object until they find ones that match the pattern. With more than 400 million objects in their catalog, each having their own set of time series brightness measurement, the process of manually exploring a large number of

---

*Copyright 0000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.*

**Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**

---

visualizations is not only error-prone, but also overwhelming for scientists who do not have extensive knowledge about their dataset.

The astronomy use case highlights a common challenge in exploratory data analysis (EDA). There is often a large space of possible visualizations that could be generated from a given dataset and manual search through this large collection is inefficient. Visualization authoring tools such as Tableau and Excel focusses on presenting one visualization at a time. There is no systematic way to create, compare, and query large collections of visualizations.

## 2.2 Effortless Data Exploration with *zenvisage*

The challenges presented earlier points to a need for tools that enables users to create and search through large collections of visualizations. Therefore, we developed *zenvisage* a visual query system that allowed users to search through large collections of visualizations. *zenvisage* is built on top of a querying language called ZQL, which provides a mechanism for managing collections of visualizations [18]. Contrary prior work on visualization languages for specifying visual encodings of individual visualizations [19, 21], ZQL supports high-level queries over visualization collections, such as composing, sorting, filtering a collection of visualization. ZQL functionals and primitives can be constructed into rich and expressive query semantics, with functionalities including :

- Finding top-k visualizations whose y values are most or least similar from a queried visualization (e.g. Find other cities with sold price over time similar to Manhattan. Varying along CITY while keeping X=TIME, Y=AVG(PRICE) fixed.)
- Comparing across a collection of visualizations by iterating over one or more x, y, z attributes while fixing other attributes (e.g. Find a y attribute that varies with time similarly how average price changes over time)
- Finding a pair of X and Y axes where two specific visualization instances differ the most. (e.g. Finding pairs of attributes where visualizations for the products ‘stapler’ and ‘chair’ differ the most.)

Given a ZQL query, *zenvisage* parses the query into a graph of visual component nodes(containing visualization information, such as X, Y columns) and task nodes (common and user-defined primitives for processing visual components, such as sort-filter). *zenvisage* then performs query optimization to merge together multiple nodes, as well as reducing the processing time required for individual visualization components. Using the optimized query plan, the executor compiles visual nodes into SQL queries for retrieving the visualization data and postprocesses the result via the defined operations.

While ZQL provides powerful mechanism for expressively specifying queries on large collections visualizations, writing ZQL queries can be daunting for novice users. Therefore, we extracted a typical workflow of visualization querying (finding top-k most similar visualization from a collection with fixed X,Y while varying Z) to allow users to formulate ZQL queries through interactions. The user can either directly input ZQL queries through a frontend table input or their frontend interactions is mapped into ZQL queries. The query results are rendered as a ranked list of visualizations in the Results panel in the frontend. *zenvisage* is a full-fledged visual querying system that supports a variety of querying interactions as illustrated in Figure 2.2. In the following section, we will discuss the design process of how we developed this visual query system and the lessons that we have learned for designing future visual data exploration systems.

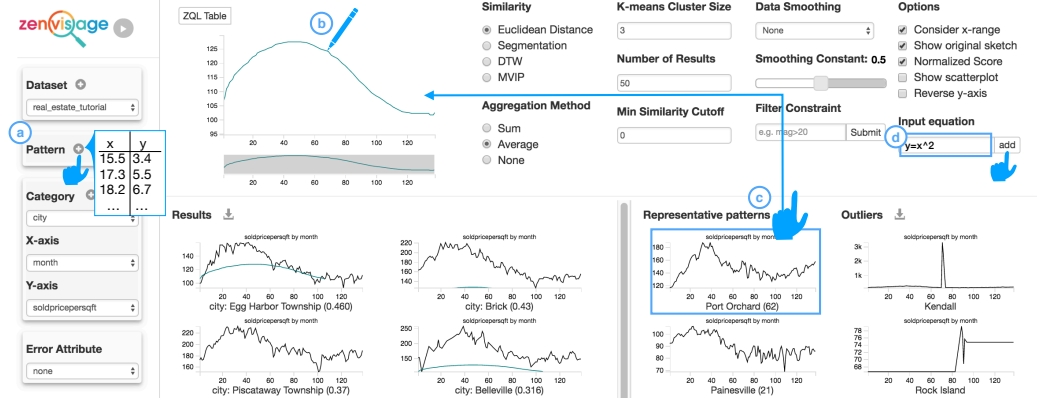


Figure 1: *zenvisage* offers a variety of querying modalities, including: a) uploading a sample pattern from an external dataset as a query, b) sketching a query pattern, c) dragging-and-dropping an existing pattern from the dataset, and d) inputting an equation as a query.

### 3 Hypothesis Formation and the cycle of visual analysis

#### 3.1 Visual Querying in the *Sensemaking Framework*

*zenvisage* is an example of a *precise visual querying system (PVQS)*, which accepts precise queries as an input, expressed through interactions or directly specified via the querying language. In developing *zenvisage*, we collaborated with scientists from astronomy, genetics, and material science in a year-long participatory design process [8]. In particular, we study how various features impacts analyst’s ability to rapidly generate new hypothesis and insights and perform visual querying and analysis. Our findings offered design guidelines for improving the usability and adoption of next-generation VQSs. More importantly, in this section, we focus on applying our findings on visual querying to the context of supporting the full cycle of visual data exploration.

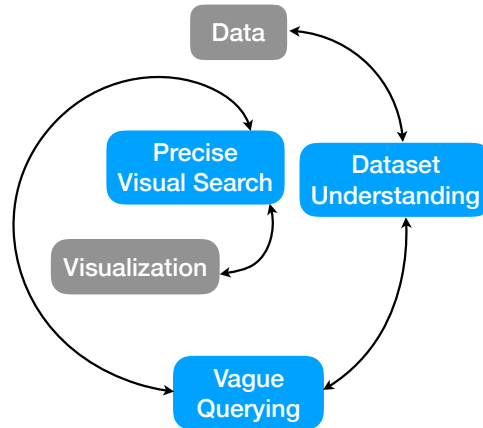


Figure 2: Cycle of visual data exploration.

#### The Need for Complex, Vague, and Expressive Querying

When users are querying with a PVQS, they often need to translate their ambiguous, high-level questions into an plan that consists of multiple interactions to incrementally address their desired query. Participants in our *zenvisage* design study often created unexpected workflows that chain together multiple analysis steps, including interactions, controls, and queries in order to address a higher-level research question. For example, the

geneticists in our study repeatedly explored representative trends to gain an overall sense of what typical profiles exist in the dataset and queried mainly through drag-and-drop of these representative trends. Variations to their main workflow also include changing cluster sizes and display settings to offer them different perspectives on the dataset and filtering on data attributes.

The expressiveness of PVQs comes from the multiplicative effect of stringing together combinations of interaction sequences into a customized workflow. Designing features that diversifies potential variations and helps the creation of multiple analysis workflows expands the space of actions that could be performed during the analysis. However, even with many supported interaction, there were still complex multi-step queries that could not be expressed through this framework. In Section 4, we highlight the challenges in PVQs in supporting these types of queries, highlight the need for tools that support complex and vague querying, and survey several promising directions of ongoing research in this area.

### **Top-down and Bottom-up Querying Modalities**

We employed Pirolli and Card’s [13] information foraging theory to contextualize our study results. Pirolli and Card’s notional model distinguishes between information processing tasks that are *top-down* (from theory to data) and *bottom-up* (from data to theory). In the context of visual querying, users employ top-down approaches by starting with a hypothesis on what patterns to look for and express it through sketching or inputting an equation (Figure 2.2b,d). On the other hand, bottom-up approaches originate from the data (or equivalently, the visualization). For example, the user may drag and drop a visualization of interest in the dataset as the input query or upload a visualization from an external dataset (Figure 2.2a,c). These two modalities of querying are reminiscent of the browsing and searching behaviors on the Web, which derives from the successful application of foraging theory to Web Search [11].

While the usage of each querying feature may vary from one participant to the next, our interactions with the scientists showed that *bottom-up querying via drag-and-drop was more intuitive and more commonly used than top-down querying methods when the users have no desired patterns in mind*, which is commonly the case for exploratory data analysis. One of the main reason why participants did not find sketching useful was that they often do not start their analysis with a pattern in mind. Later, their intuition about what to query is derived from other visualizations that they see in the VQS, in which case it made more sense to query using those visualizations as examples directly. Similarly, while functional fitting is a common operation in scientific data analysis, querying by equation is also unpopular, since it is challenging to formulate functional forms in an prescriptive, ad-hoc manner without seeing what the common patterns in the dataset are.

A key design principle that came from this study was the need for visual query systems to provide visualization recommendations that can help analysts jumpstart their exploration. We found that many users made use of the representative trends and outliers visualizations provided by *zenvisage* as contextual information to better understand their data (e.g. after a filter is applied) or to query based on these recommended visualizations (e.g. find visualizations that are similar to the one in the largest representative clusters).

As evident from the representative and outliers in *zenvisage*, recommendations facilitate smoother flow of analysis by closing the loop between the two modalities of querying and exploration, thus ensuring that user is never stuck or out of ideas at any point during the analysis. Typically, visualization recommendation systems seeks to accelerate the process of discovering interesting aspects of the data by broadening exploration. In Section 5, we advocate recommendation systems should not only focus on data discoverability aspect of exploration, but also contribute towards helping users gain better awareness and understanding of the scope and context of their analysis and data.

## 4 Vague and Complex Querying

### 4.1 Challenges

The need for vague and complex querying stems from the inevitable design trade-off between the query expressivity and interface usability [9, 7]. For example, the *zenvisage* interface was unable to support high-level queries that involved the use of vague descriptors for matching to specific data characteristics, such as finding light curves that are flat and ‘without noise’, or patterns that ‘exhibits irregularities’. While this could be expressed through user-defined functions in underlying querying language ZQL, the learning curve and engineering cost is high.

This tradeoff is observed not only in visual data exploration systems, but also true for general ad-hoc data querying. For example, while querying language such as SQL are highly expressive, formulating SQL queries that maps user’s high-level intentions to specific query statements is challenging. Therefore, query construction interfaces have been developed to address this issue to enable direct manipulation of queries through graphical representations [1], gestural interaction [10], and tabular inputs [24]. Form-based query builders often consist of highly-usable interfaces that ask users for a specific set of information mapped onto a pre-defined query. However, form-based query builders are often based on templated queries with limited expressiveness in their semantic and conceptual coverage, which makes it difficult for expert users to express complex queries. The extensibility of these systems or querying language also comes with the high engineering cost, as well as potentially overloading the users with too many potential options to choose from. There is a need for tools that enables users to formulate rich and complex queries, yet highly usable even for novices.

### 4.2 Ongoing research and opportunities

Given the tradeoff between expressivity and usability, we can not assume a one-size-fit-all PVQS that could fit the need for users of different expertise levels and workload. In this section, we discuss a growing class of *intelligent visual querying system* (IVQS) that works around this problem by taking into account a wide spectrum of queries of different with varying degrees of specificity. We describe three categories of challenges in vague querying. While our analysis is not limited to natural language interfaces, we use this classification scheme to provide an analogy for types of vague queries that an analyst might specify.

**Lexical Ambiguity:** Lexical ambiguity involves the use of vague descriptors in the input queries. Our previous example of queries that could not be expressed in the *zenvisage* interface showcases one example of lexical ambiguity, where the PVQS can not map the vague term ‘irregular’ or ‘noisy’ into the appropriate series of analysis steps required to find these patterns. [Doris: Not sure if its worthwhile to include a couple sentences about Tarique’s regex/NLP work here. e.g. the ‘up and then down’ example.](#) Resolving these lexical ambiguities has been a subject of research in natural language interfaces (NLIs) for visualization specification, such as DataTone [4] and Eviza [17]. These NLIs detects ambiguous quantifiers in the input query (e.g. “Find large earthquakes near California”), and then displays ambiguity widgets in the form of a widget to allow the user to specify what is the definition of ‘large’ in terms of magnitude and how many miles radius do we consider ‘near’. These ambiguity widgets not only serve as a way to provide feedback to the system for lexically vague queries, but also is a way for displaying interpretable explanations of how the system is interpreting the input queries. An IVQSs can be thought of as a layer on top of PVQS (for functionalities such as shape-matching, filtering), which chooses appropriate *parameters* to the PVQS to achieve the user’s desired querying effects.

**Syntactic Ambiguity:** Syntactic ambiguity is related to the vagueness in the specification of how the query should be structured or ordered. For example, DataPlay introduced the idea of syntax non-locality in SQL, in which switching from an existential (at least one) to a universal (for all) quantifier requires major structural changes to the underlying SQL query. They have built a visual interface that allowed users to directly manipulate the structure of the query tree in tweaking the query to its desired specification. IVQSs that resolve syntactic

ambiguities either map the vague queries into to *a series of multi-step workflows* to be executed in the PVQS or allow users to tweak the query representation directly. The query specification and tweaking is done in a declarative manner in that the mechanism in which the visualized workflow gets translated to the underlying language is hidden from the end-user.

**Semantic Ambiguity:** Semantic ambiguity arises when the user does not specify their intent completely or explicitly, which is often the case in the earlier stages of the visual data exploration. Natural language interfaces for visual data exploration such as Evizeon [6] makes use of anaphoric references to fill in incomplete follow-on queries. For example, when a user says ‘Show me average price by neighborhood’, then ‘by home type’, the system interprets the latter partial specification as continuing the context of the original utterance related to average price on the y-axis. Semantic ambiguity can also be thought of as being composed of one or more lexical and syntactical ambiguity. For example, a user can specify a vague, high-level query such as ‘Create a classifier’, then the conversational agent makes use of nested conversations to inquire about what type of classifier to chose and what features to use in the model to fill in the details of the structure and parameters required, thus resolving the syntactic and semantic ambiguity. A semantically vague query may or may not be expressible through a single PVQS, since the operations involved in the query may not be covered by the limited workflow combinations in the PVQS. [Doris: Might be good to include three screenshot side-by-side of each of these types of vagueness in querying, situation in which people do vague querying, and systems that tries to address it, are we allowed to take screenshots from other people’s papers or do we need to ask them for permission.](#)

## 5 Towards Dataset Understanding

One of the key goals of visual data exploration is to promote a better understanding of the dataset to enable users to make actionable decisions. While our focus in the previous sections have focussed on intention-driven queries, where users have some knowleldge of what types of questions he may be interested in. This section discusses systems that helps users become more aware of their dataset and where they are in their analysis workflow.

Situations where there is an absence of explicit signals from the user can happen in two scenarios: 1) user is at the beginning of their analysis (commonly known as the ‘cold-start’ problem) and 2) user doesn’t know what to query for, which is the situation derived from our *zenvisage* finding in Section 3. In this section, we will describe STORYBOARD, a system that provides data summaries and guides users through informative subsets of data, as an example of a system that promotes distribution awareness in a query-free scenario. Then, we will discuss two other types of data understanding during dynamic visual data exploration to highlight the challenges and opportunities ahead in this space.

### 5.1 STORYBOARD: Promoting Distribution Awareness of Data Subsets with Summary of Visualizations

Common analytics tasks, such as causal inference, feature selection, and outlier detection requires studying the distributions or patterns at different levels of data granularity [2, 23, 5]. However, it is often hard to know *what* subset of data contains an insightful distribution to examine. In order to explore different data subsets, a user would first have to construct a large number of visualizations corresponding to all possible data subsets, and then navigate through this large space of visualizations to draw meaningful insights. The lack of a systematic way to perform these excercises makes the process of manually exploring distributions from all possible data subsets tedious and inefficient [14, 15].

To this end, we present STORYBOARD, an interactive visualization summarization system that automatically selects a set of visualizations to summarize the distributions within a dataset in an informative manner. Figure ??

illustrates an example dashboard generated by STORYBOARD from the Police Stop Dataset [12], which contains records of police stops that resulted in a warning, ticket, or an arrest. The attributes in the dataset include driver gender, age, race, and the stop time of day, whether a search was conducted, and whether contraband was found. We requested STORYBOARD to generate a dashboard of 9 bar chart visualizations with x-axis as the stop outcome (whether the police stop resulted in a ticket, warning, or arrest/summons) and y-axis as the percentage of police stops that led to this outcome. First, at the top of our dashboard, STORYBOARD highlights three key data subsets that results in a high arrest rate, which looks very different trend than the overall (where the majority of stops results in tickets). Following along the leftmost branch, we learn that even though in general when a search is conducted, the arrest rate is almost as high as ticketing rate, when we look at the asian population, whether a search is conducted had less influence on the arrest rate and the trend resembles more like the overall distribution.

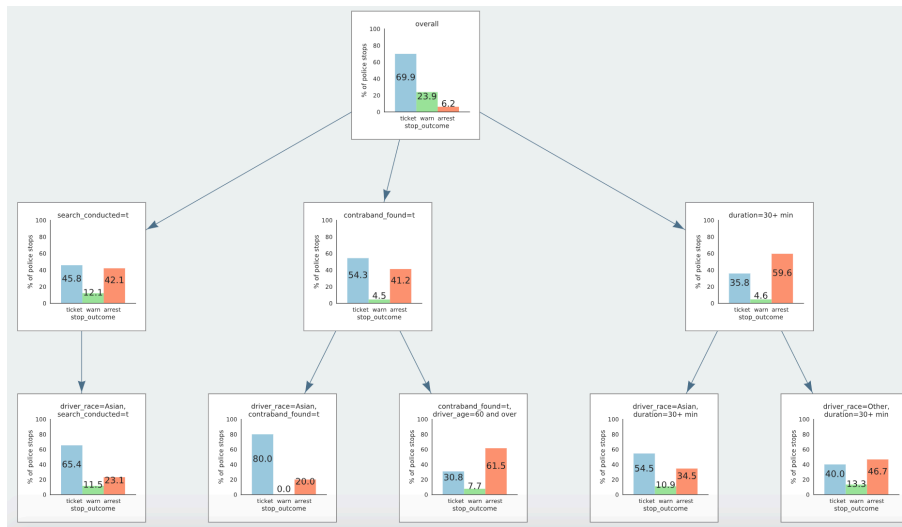


Figure 3: Example dashboard generated by STORYBOARD summarizing the key insights in the Police dataset.

While such summary dashboards are useful for making sense of relationships between data subsets, finding effective visualizations to summarize a dataset is not as trivial as picking individual visualizations that maximizes some statistical measure, such as deviation [20], coverage [16], or significance testing [2], which can often result in misleading summarizations. The key insights of our work is realizing how users formulate their expectations regarding an unseen visualization in a *data subset lattice*. We make use of the idea of a data subset lattice from data cube literature to organize the relationships between different visualization. A visualization is a *parent* of another visualization if the latter visualization can be derived from the first visualization by adding one additional filter constraint. Our formative user study showed that people naturally form their expectations based on one or more observed parents and that seeing a parent that well describes the unseen visualization leads participants to better estimate the unseen visualization. More importantly, in the absence of an informative parent or in the presence of multiple parents, participants can be misled to form an inaccurate expectation that exhibit higher variance. Given these insights, the goal of our system is to select *interestingness* and *informativeness* visualizations that can help them make more accurate predictions regarding the unseen visualizations. To model the informativeness of an observed parent in the context of an unseen visualization, we characterize the capability of the parent in predicting the unseen visualization. Our study shows that a visualization is *informative* if its data distribution closely follows the data distribution of the unseen child visualization, since the visualization helps the analyst form an accurate mental picture of what to expect from the unseen visualization. While informative parents contribute to the prediction of an unseen visualization, the most interesting visualizations to recommend are those for which *even the informative parents fail to accurately predict or explain the visualization*. To model



the interestingness of an unseen visualization in the context of an observed parent, we characterize the deviation between their data distributions using a distance function. The unseen visualizations whose data distributions deviate from the observed informative parents are *interesting*. Detailed treatments of our metrics and algorithms can be found in our technical report. [Doris: \(CITE PLACEHOLDER\) Can we put a version of the STORYBOARD paper on arxiv so that we can cite it?](#)

The effectiveness of STORYBOARD largely comes from how it helps analysts become more *distributionally aware* of the dataset. We define *distribution awareness* as the aspect of data understanding in which analysts make sense of the key distributions across different data subsets and their relationship in the context of the dataset. So that even though it may be infeasible to examine all possible data subsets, with distribution awareness, the analyst will still be able to draw meaningful insights and establish correlations about related visualizations by generalizing their understanding to make predictions regarding the unseen visualizations. Our user study evaluations show that facilitating distribution awareness through STORYBOARD guides users to make better predictions regarding unseen visualizations, ranking attribute importance, and retrieval of interesting visualizations compared to the baselines.

## 5.2 From Distributional to Contextual Awareness: Challenges and Opportunities

The notion of distribution awareness is useful when we are looking at user understanding at one static point in time of the analysis (e.g. during cold start). In this section, we introduce a complementary notion of data understanding called *contextual awareness*, which is essential when considering the dynamic setting of visual data exploration in the context of an analytic workflow.

Contextual awareness is the aspect of data understanding related to understanding the *situation* (what is the information that I’m currently looking and how did it come about?) and *provenance* (what have I explored in the past and where should I look next?) of the data. Situational understanding involves recognizing what data is in the current scope of analysis, including making sense of the data attributes and schema and keeping track of what filter or transformations have been applied to the displayed data. Provenance understanding is related to the user’s past analysis actions performed to the data. As an example, an analyst may be interested in how sales price of a product changes as a function of other dimensions variables, such as geographic location, year sold, and product type. Situation information informs him that he is looking at a bar chart with  $x = \text{product type}, y = \text{AVG}(\text{PRICE})$ , whereas provenance information points to the fact that he should explore the geographic dimension, since he has already explored the temporal attribute YEAR.

Mechanisms that facilitate distribution awareness for users can effectively couple with contextual awareness in dynamic exploration situations to help update the user’s mental model on the current data context. For example, the representative and outlier patterns in *zenvisage* provides summaries of data in context. When a dataset is filtered, the representative trends are updated accordingly. By being aware of both the context and the distributions, the users becomes distributionally aware of how the typical patterns and trends of the distributions changes in a particular context.

Within a dataset, provenance is essential to help users navigate and provide users with sense of coverage and completion. The notion of adding navigational cues to guide exploration in visual information spaces was first proposed by Willet et al.[22] in work on *scented widgets*. Scented widgets are inspired by Pirolli and Card’s theory of ‘scent’ in foraging. The widgets adds to existing interfaces by embedding visualizations that provide informational ‘scents’, such as histogram distributions of how popular a particular value is among users or using color to encode the size of a dataset in a dropdown menu. Recently, Sarvghad et al. [16] have extended the idea of scented widgets to incorporate dimension coverage information during data exploration, including which dimensions have been explored so far, in what frequency, and in which combinations. Their study show that visualizing dimension coverage leads to increased number of questions formulated, more top-level finding, and broader exploration than participants who had no access to coverage information. Interpretable and non-disruptive cues that enables users to visualization provenance history helps sustain contextual awareness and



guides users towards more informative next steps in their analysis.

Note that while the discussion above have been focussed on how to design systems that can help facilitate these aspects of user's awareness in dataset understanding, these ideas can be generalized to principles in designing the types of intelligent querying systems discussed in Section 4. An intelligent visual exploration system needs to be distributionally, contextually and situationally aware, by make use of information about the data (distribution awareness), the analytic context, and situation jointly in making timely recommendations. For example, contextual awareness can inform the system that the user's current situation (x,y, encoding, etc.), while a distributionally aware system may recommend a highly-skewed data subset as interesting, a situational aware system may realize a variable have been explored extensively in the past and recommends it accordingly. In other words, these intelligent visual query system not only needs to facilitate these aspects of data understanding, but also need to make use of this information to make inference and recommendations in an interpretable manner that can guide analysts towards meaningful stories and insights for further investigation.

## 6 Concluding Remarks

### References

- [1] Azza Abouzied, J Hellerstein, and A Silberschatz. Dataplay: interactive tweaking and example-driven correction of graphical database queries. *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pages 207–217, 2012.
- [2] Anushka Anand and Justin Talbot. Automatic Selection of Partitioning Variables for Small Multiple Displays. 2626(c), 2015.
- [3] Drlica Wagner et al. Dark Energy Survey Year 1 Results: Photometric Data Set for Cosmology. 2017.
- [4] T.a Gao, M.b Dontcheva, E.a Adar, Z.b Liu, and K.c Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, pages 489–500, 2015.
- [5] Jeffrey Heer and Ben Shneiderman. Interactive Dynamics for Visual Analysis. *ACM Queue*, 10(2):30, 2012.
- [6] Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman. Applying Pragmatics Principles for Interaction with Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics*, (c), 2017.
- [7] H. V. Jagadish, Adriane Chapman, Aaron Elkiss, Magesh Jayapandian, Yunyao Li, Arnab Nandi, and Cong Yu. Making database systems usable. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, SIGMOD '07*, pages 13–24, New York, NY, USA, 2007. ACM.
- [8] Doris Jung-Lin Lee, John Lee, Tarique Siddiqui, Jaewoo Kim, Karrie Karahalios, and Aditya G. Parameswaran. Accelerating scientific data exploration via visual query systems. *CoRR*, abs/1710.00763, 2017.
- [9] Kristi Morton, Magdalena Balazinska, Dan Grossman, and Jock Mackinlay. Support the Data Enthusiast: Challenges for Next-Generation Data-Analysis Systems. *Proceedings of the VLDB Endowment, Volume 7, pp. 453–456, 2014*, 7:453–456, 2014.
- [10] Arnab Nandi, Lilong Jiang, and Michael Mandel. Gestural query specification. *Proceedings of the VLDB Endowment*, 7(4):289–300, 2013.

- [11] Christopher Olston and E D H Chi. ScentTrails: Integrating browsing and searching on the Web. *ACM Transactions on ComputerHuman Interaction TOCHI*, 10(3):177–197, 2003.
- [12] E. Pierson, C. Simoiu, J. Overgoor, S. Corbett-Davies, V. Ramachandran, C. Phillips, and S. Goel. A large-scale analysis of racial disparities in police stops across the united states, 2017.
- [13] Peter Pirolli and Stuart Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. A Notional Model of Analyst Sensemaking.
- [14] S Sarawagi, R Agrawal, N Megiddo, Generalitat Valenciana Ajuntament Valencia Univ Politecn Valencia, and E T H Zurich Oracle Sybase Softlab Iberia Edbt Fdn. Discovery-driven exploration of OLAP data cubes. *6th International Conference on Extending Database Technology (EDBT 98)*, pages 168–182, 1998.
- [15] S. Sarawagi. User-adaptive exploration of multidimensional data. *Proc of the 26th Intl Conference on Very Large*, pages 307–316, 2000.
- [16] Ali Sarvghad, Melanie Tory, and Narges Mahyar. Visualizing Dimension Coverage to Support Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):21–30, 2017.
- [17] Vidya Setlur, Sarah E Battersby, Melanie Tory, Rich Gossweiler, and Angel X Chang. Eviza: A Natural Language Interface for Visual Analysis. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, pages 365–377, 2016.
- [18] Tarique Siddiqui, Albert Kim, John Lee, Karrie Karahalios, and Aditya Parameswaran. Effortless data exploration with zenvisage: An expressive and interactive visual analytics system. *Proc. VLDB Endow.*, 10(4):457–468, November 2016.
- [19] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multi-dimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–14, 2002.
- [20] Manasi Vartak, Samuel Madden, and Aditya N Parmeswaran. SEEDB : Supporting Visual Analytics with Data-Driven Recommendations. 2015.
- [21] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Berlin, Heidelberg, 2005.
- [22] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.
- [23] Eugene Wu and Samuel Madden. Scorpion: Explaining Away Outliers in Aggregate Queries. *Proceedings of the VLDB Endowment*, 6(8):553–564, 2013.
- [24] Moshe M Zloof. Query by Example. *National Computer Conference*, pages 431–438, 1975.