The Effects of Interactive Latency on Exploratory Visual Analysis

Zhicheng Liu and Jeffrey Heer

Abstract—To support effective exploration, it is often stated that interactive visualizations should provide rapid response times. However, the effects of interactive latency on the process and outcomes of exploratory visual analysis have not been systematically studied. We present an experiment measuring user behavior and knowledge discovery with interactive visualizations under varying latency conditions. We observe that an additional delay of 500ms incurs significant costs, decreasing user activity and data set coverage. Analyzing verbal data from think-aloud protocols, we find that increased latency reduces the rate at which users make observations, draw generalizations and generate hypotheses. Moreover, we note interaction effects in which initial exposure to higher latencies leads to subsequently reduced performance in a low-latency setting. Overall, increased latency causes users to shift exploration strategy, in turn affecting performance. We discuss how these results can inform the design of interactive analysis tools.

Index Terms—Interaction, latency, exploratory analysis, interactive visualization, scalability, user performance, verbal analysis

1 Introduction

One stated goal of interactive visualization is to enable data analysis at "rates resonant with the pace of human thought" [19, 20]. This goal entails two research directions: understanding the rate of cognitive activities in the context of visualization, and supporting these cognitive processes through appropriately designed and performant systems.

Latency is a central issue underlying these research problems. Due to the time required for query processing, data transfer, and rendering, data-intensive visualization systems incur delay. It is generally held that low latency leads to improved usability and better user experience. Unsurprisingly, multiple research efforts focus on reducing query and rendering latency for large datasets, which may include billions or more data points. Latencies in state-of-the-art systems can range from 20 milliseconds up to multiple seconds for a unit task [2, 28, 29].

Despite the shared goal of minimizing latency, the effects of interaction delays on user behavior and knowledge discovery with visualizations remain largely unevaluated. While previous research on the effects of interactive latency in puzzle solving [4, 17, 35, 36] and search [8] has shown that user behavior changes in response to millisecond-scale differences in latency, studies in other domains such as computer games report no significant effects [23, 39].

It is unclear to what degree these findings apply to exploratory visual analysis. Unlike problem-solving tasks or most computer games, exploratory visual analysis is open-ended and does not have a clear goal state. User interaction may be triggered by salient visual cues in the display, driven by *a priori* hypotheses, or carried out through exploratory browsing. The process is more spontaneous and is unconstrained by factors such as game rules.

How does latency affect user behavior and knowledge discovery in exploratory visual analysis? To answer this question, we conduct controlled experiments comparing two latency conditions, differing by 500ms per operation. We analyze data collected from both system logs and think-aloud protocols to test if (a) delay impacts interaction strategies and (b) lower latency leads to better analysis performance.

Our work makes the following contributions. First, we present the design and the results of a controlled study confirming that a 500ms difference can have significant impacts on visual analysis. Specifically, we find that (1) the additional delay results in reduced interaction and reduced dataset coverage during analysis; (2) the rate at which users make observations, draw generalizations and generate hypotheses (as determined using a think-aloud protocol) also declines due to the delay; and (3) initial exposure to delays can negatively impact overall performance even when the delay is removed in a later session. Second, we extend the insight-based evaluation methodology [37, 38] for comparative analysis of qualitative data regarding visualization use. We introduce a procedure for segmenting, coding and analyzing think-aloud protocols for visualization research. Our analysis contributes coding categories that are potentially applicable for future protocol analysis. Finally, our results show that the same delay has varying influences on different interactive operations. We discuss some implications of these findings for system design.

2 Related Work

Our research draws on related work in scalable visualization systems, cognitive science and domain-specific investigations on the effects of interactive latency. We review relevant literature below.

2.1 Scalable Data Analysis Systems

Building low latency analysis systems has been a focus for many research projects and commercial systems, spanning both back-end and front-end engineering efforts. Spark [44, 45] supports fast in-memory cluster computing through read-only distributed datasets for machine learning tasks and interactive ad-hoc queries. Nanocubes [28] contribute a method to store and query multi-dimensional aggregated data at multiple levels of resolution in memory for visualization. Profiler [26] builds in-memory data cubes for query processing. Tableau's data engine [1] optimizes both in-memory stores and live connections to databases on disk. imMens [29] decomposes multi-dimensional data cubes into binned data tiles of reduced dimensionality and performs accelerated query processing and rendering on the GPU.

In cases where long-running queries are unavoidable, sampling and online aggregation [22] are often used to improve user experience. BlinkDB [2] builds multi-dimensional, multi-resolution samples and dynamically estimates a query's response time and error. With online aggregation [22], visualizations of estimated results are incrementally updated as a query progresses. Studies suggest that data analysts can interpret approximate results visualized as bar charts with error bars to make confident decisions [16].

2.2 Time Scales of Human Cognition

Decades of psychology research have produced evidence that different thought processes operate at varying speeds [25]. Newell [33] provides a framework outlining proposed time scales of human cognition. Relevant to studies of human-computer interaction are the cognitive (100 milliseconds to 10 seconds) and rational (minutes to hours) time bands. Within the cognitive band, Newell identifies three types of time constants: deliberate act, operation, and unit task. Card et al. [11] make similar distinctions using a different terminology. Table 1 summarizes these scales, exemplary actions, and the time ranges during which these actions occur.

• *Zhicheng Liu is with Adobe Research. E-mail: leoli@adobe.com.*
• *Jeffrey Heer is with the University of Washington. E-mail: jheer@uw.edu.*

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

Newell's Term	Card et al.'s Term	Examples	Time Range
deliberate act	perceptual fusion	recognizing a pattern, tracking animation	~100 milliseconds
cognitive operation	unprepared response	clicking a link, selecting an object	~1 second
unit task	unit task	editing a line of text, making a chess move	~10 seconds

Table 1. Time scales at which human actions take place

Based on these frameworks, Card et al. [10] provide a high-level discussion of time constants in the context of information visualization. Fundamental task types in exploratory visual analysis include select, navigate, coordinate and organize [20]. Brushing and linking, for instance, coordinates multiple views such that selecting items in one view causes highlighting of corresponding items in the other views. Card et al. [11] place brushing and linking in the "perceptual fusion" time band, suggesting immediate visual feedback within 100ms. Little work elucidates the desired time scales for the other types of interactive techniques, however.

Ballard et al. [4] propose an adaptation of Newell's framework based on empirical observations. They argue that the ~100ms time scale for deliberate acts is an over-estimation, and adjust it to ~50ms. They strongly emphasize the time scale around ~300ms, termed the "embodiment level". Exemplary human actions at this time scale include eye saccades and body movements. By focusing on this level of time constant, they explain how a minor change in interface can lead to significant changes in cognitive behavior. For example, when the cost of acquiring a piece of information is increased from an eye movement to a head movement in a block-copying task, subjects change strategy and rely more on working memory [3].

These frameworks provide guidance for investigations of interactive dynamics in different domains, and anchor the interpretation of experimental results. In the next section, we review empirical findings on the effects of varying latencies in different domains.

2.3 Effects of Latency in Various Domains

Research into the effects of latency typically focus on the perception of delay, impacts on the outcome of an activity, and changes in user behavior. Michotte's [31] studies report that longer delay weakens the perception of causality. People will interpret the consecutive motions of two objects as causally related if the second object moves within 70ms after contact; such impressions are lost when the delay exceeds 160ms. For the perception of delay in task-oriented settings, studies find that users can detect latencies as low as 2.38 ms when dragging an object on direct touch interfaces [34].

Investigating the effect of varying latency between 1 and 50 ms on touch interfaces, Jota et al. [24] find that user performance decreases as latency increases. In particular, input latency significantly affects the last stage of pointing tasks. Beigbeder et al. [7] also find that latency as low as 100ms can significantly degrade shooting precision in the first-person shooting game "Unreal Tournament 2003".

Not all studies find a negative correlation between latency and user performance. Sheldon et al. [39] report that latency as high as several seconds does not significantly affect user performance in real time strategy games such as Warcraft III. While perhaps surprising, this result can be explained by the observation that tasks such as understanding game situation and conceiving strategy play a more important role in affecting the outcome of a game. These tasks take place at a larger time scale, and so are less sensitive to typical network latencies.

Research in cognitive science finds that changes in interaction cost can change user behaviors during problem-solving. This change in behavior in turn impacts the final outcome. O'Hara and Payne [35] show that by changing the cost of moving a tile in an 8-puzzle game (shifting from pressing a key to typing a command), participants resort to more mental planning and solve the puzzle in fewer moves. In a related study, they impose a 4-second additional lockout delay in each tile move. This change results in less manipulation of the tiles and more search of an internally-represented problem space [36]. O'Hara and Payne's experiments arguably involve conditions that span multiple time scales in Table 1.

Evidence from longitudinal studies of user search behavior reveals that seemingly insignificant time differences for a unit task can impact user behavior, too. Brutlag [8] observes that a small difference of 300ms in web search delay reduces the number of searches users perform. Even after the latency is restored to previous levels, this reduction in user activity persists for days. In contrast, players of Half-Life, a first person shooting game, do not appear to be affected when more than 40% of them experience delays of over 225ms. Players tolerate the delays and keep returning to the games [23].

It is not clear if these findings can be simply extrapolated to a new domain. Unlike problem solving or computer games, exploratory visual analysis is open-ended and has no clearly formulated goal state. Conducting a targeted study for visual analysis is thus necessary and presents new challenges in experiment design and data analysis.

3 Experimental Method

The goal of this research is to investigate whether differences in interactive latency influence behaviors and outcomes during exploratory visual analysis. To this end, we use a 2 (datasets) x 2 (latency conditions) experiment design. We induce two latency conditions in the imMens [29] system, differing by 500 milliseconds per operation. The subjects analyze two datasets and report their findings through a think-aloud protocol. We log user interaction and record verbal data for analysis. In this section, we describe the system used in the study, datasets and visualizations, participants, tasks, and study procedures.

3.1 Experimental System

We adapt the open-source imMens system [29] for our study. imMens can interactively query and visualize more than millions of data points at higher frame rates than other existing systems. The low interactive latency in imMens allows the creation of different latency conditions by injecting precise delays into the interactive operations.

imMens aggregates data at multiple scales of binning resolution and visualizes the aggregates as histograms, bar charts, line graphs, binned scatterplots and geographic heatmaps. The system supports four main interaction techniques for the binned plots: select, pan, zoom, and brush & link. To ensure real-time interaction with the visualizations, imMens pre-computes data tiles, which are 3- or 4-dimensional projections decomposed from the full data cube. When visualizations are created or viewports are modified via panning or zooming at the client, imMens loads relevant data tiles from the server and updates the visualizations. To reduce network latency, imMens pre-fetches data tiles for application states reachable from the current state and caches the data tiles at the client side. The browser-based client uses WebGL to perform parallel query processing and rendering on the GPU. Benchmarking tests show that imMens is able to sustain a performance of 50 frames per second for interactive brushing & linking of 1 billion data points across 25 visualizations. Prior work by Liu et al. [29] describes the design and implementation of imMens in greater detail.

We added two user interface controls to the imMens interface to support log scale transforms and color scale adjustment. Given a large number of data elements and a limited number of pixels for each visualization, a brushing selection of a small number of data points can be difficult to see, as the projected sums in linked views may be quite small. In response, we provided a checkbox for users to toggle between linear and log scale for histograms and bar charts. In addition, an outlying density value in a two-dimensional plot can skew the color mapping. Users can use a range slider control to define the lower and upper bounds of color clamping within binned scatterplots and geographic heatmaps.

3.2 Latency Conditions

We considered multiple choices when designing our latency conditions. One approach is to include multiple latencies in small increments, which is useful for identifying time scale thresholds for each interactive operation. Assessing thresholds, however, is not the focus of our study, and often requires conducting studies with highly-controlled, low-level tasks. We are more interested in understanding the effects of latency on various dimensions of exploratory visual analysis. Thus a more ecologically valid setting, in which users perform open-ended exploratory analysis, is appropriate. However, studying ecologically valid behavior imposes practical constraints. Exploratory visual analysis is a complex process, requiring careful analysis of both quantitative interactive event log data and qualitative data concerning insight discovery. We also anticipate that datasets with different semantics can lead to different user behaviors, so it is necessary to include dataset and visualization configuration as a factor and repeat the latency conditions in more than one analysis scenario. As a result, we decided to use a 2 (datasets) x 2 (latency conditions) mixed design.

Table 2 summarizes the latency for the primary interactive operations supported in imMens (brushing and linking, selecting, panning and zooming) in the two latency conditions. In the control condition, the latency is simply the time taken by imMens to fetch data tiles, perform aggregation (roll-up) queries and re-render the display. In the delay condition, we injected an additional 500 milliseconds for each of these operations. We experimented with different delays in pilot studies. Initially we chose to inject an additional delay of 1 second, based on the representative latencies of related data-processing systems. Our pilot subjects found the system unusable, especially for operations like brushing and linking. We thus reduced the additional delay to 500ms. Since there is little prior work on the time scales of different interactive operations in visual analysis, we applied the same amount of delay for all four operations to see if the operations have varying sensitivity to the same delay.

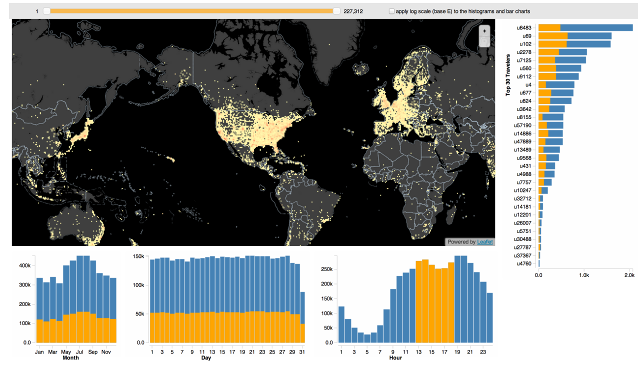
To ensure the usability of the system in the delay condition, we implemented throttling and debouncing in imMens. Throttling prevents repeated firings of the same event. For example, mouse movements within the same bar only trigger a single brushing event. Debouncing maintains a queue of events being fired, delays processing by 500ms, and drops unprocessed events when a new event of the same kind arrives. The injected delay per operation thus does not result in a growing accumulation of unprocessed events, preventing cascading delays and thus substantial usability problems.

Both log transform and color scale adjustment are client-side rendering operations that do not incur data processing latency. We chose not to inject delays into these two operations to maintain ecological validity. It is also beneficial to include both low- and high-latency operations so that we can examine if subjects preferentially use low-latency operations in favor of higher-latency ones.

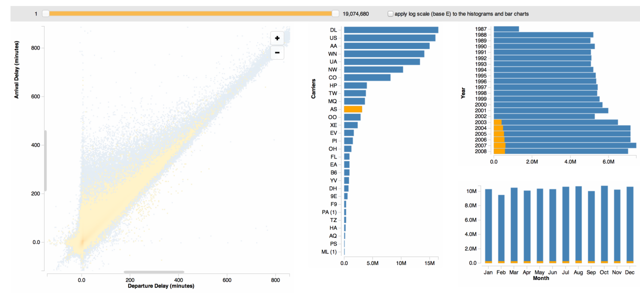
3.3 Datasets and Visualizations

We use two publicly available datasets from different domains. One contains 4.5 million user check-ins on Brightkite [13], a location-based check-in service similar to Foursquare, over a period of two years. We visualize this dataset using five linked components (Figure 1(a)): a multi-scale geographic heatmap showing the locations of the checkins, three histograms showing the number of check-ins aggregated by month, day and hour, and a bar chart showing the number of check-ins by the top 30 travelers whose check-ins span the greatest geographic bounding box. The geographic heatmap has 8 zoom levels.

The other dataset consists of 140 million records about the on-time performance of domestic flights in the US from 1987 to 2008 [9]. Subjects explore this dataset using four linked visualizations (Figure 1(b)): a binned scatterplot showing departure delay against arrival delay, two bar charts showing the number of flights by carrier and year, and a histogram showing the distribution of flights across months. The binned scatterplot has 5 zoom levels.



(a) Five coordinated visualizations showing geographical and temporal distribution of user checkins and top users.



(b) Four linked visualizations showing departure and arrival delays, carriers, yearly and monthly distribution of flights.

Fig. 1. Visualizations for the datasets used in the study.

Operation	Control Condition	Delay Condition
brush & link	20 ms	520 ms
select	20 ms	520 ms
pan	100 ms	600 ms
zoom	1000 ms	1500 ms

Table 2. Average latencies for interactive operations, across conditions.

3.4 Study Procedure

We recruited 16 subjects from the San Francisco Bay Area. All participants had experience analyzing data using systems such as Excel, R and Tableau. We instructed the participants to perform two analysis sessions, one dataset each. Every participant experienced both latency conditions, but not all combinations of latency and dataset; the same dataset cannot be reused for different latency conditions due to learning effects. For each subject, one dataset had the default latency and the other dataset had the injected 500 millisecond delay. To control for order and learning effects, half of the subjects experienced delay in the first session and the other half experienced delay in the second session. The order of the dataset analyzed was also counterbalanced.

We first gave each subject a 15-minute tutorial on imMens for each of the two analysis scenarios, teaching them how to interact with the visualizations under the respective latency condition. Subjects then spent approximately one hour exploring both datasets. They could spend a maximum of 30 minutes on a single dataset, but could stop their analysis at any time if they felt nothing more could be found. At the end of each study, we conducted an exit interview. We did not inform the subjects about the injected delay in one of the two sessions.

We considered carefully the challenge of evaluating subjects' performance when designing the study procedure. Compared with solving a tightly-specified problem, visual analysis is open-ended and lacks clear-cut performance metrics. To this end, we were inspired by the insight-based evaluation methodology proposed by Saraiya et al. [37, 38]. A fundamental premise of visualization research is that "the purpose of visualization is insight, not pictures" [10]. Insight-based evaluations collect qualitative data about the knowledge discov-

Event	Explanation	Parameters	Querying Event
brush	mouse over a single bin in a one-dimensional plot	bin index, data dimension	yes
select	click on a single bin in a one-dimensional plot	bin index, data dimension	yes
range select	click and drag over multiple bins in one or more dimensions	bin indices, data dimensions	yes
zoom	navigate to higher or lower levels of data resolution	zoom level, data dimensions	yes
pan	navigate to areas of a visualization outside the current viewport	number of bins panned for each data dimension	yes
unique data tile	fetch a data tile from the server to update visualizations, this event can be indirectly triggered by any of the above events	data tile ID	no
clear	remove selection	none	no
log transform	toggle the scales of visualizations between linear and log-scale	none	no
color slider	change the upper and lower bounds of color mapping	upper and lower bounds	no

Table 3. Application events. Only querying events are subject to additional latency. We logged both triggered and processed querying events.

ery process, either through a think-aloud protocol [15] or by asking participants to keep a diary in a longitudinal study.

As asking subjects to take notes could disrupt the flow of analysis, we decided to use a think-aloud protocol and record the subjects' verbal utterances. During each session, we observed the analysis process and took notes to triangulate with audio-recorded data.

We asked subjects to report anything they found interesting, including salient patterns in the visualizations, their interpretations, and any hypotheses based on those patterns. Findings might include surprising events, data abnormality, or confirmations of common knowledge and intuition. The subjects did not have to verify the accuracy of these findings. To help orient subjects, we provided three hypothetical examples of findings for each dataset. The subjects were told that these findings were not necessarily true and should be taken only as examples of potential observations and insights. We encouraged the subjects not to be constrained by these examples, and the subjects reported diverse findings beyond the scope demonstrated by the examples. For the Brightkite dataset of mobile check-ins, the examples included:

- Brightkite has a significant presence in Japan despite Japan's relatively small population.
- A user checked in along the African coast over a period of three months.
- User A makes most of his check-ins in Alaska, while user B checks in all over the world, mostly in the summer.

Examples for the dataset of airline flight delays included:

- Airline X has more flights than any other airline, except in 2008.
- Specific flights in the 1990s depart late but arrive early.
- Airline Y stopped operating in 2001, and re-emerged in 2006.

All sessions were run in Google Chrome v.29.0.1547.65 on a quad-core 2.3GHz MacBook Pro (OS X 10.8.2) with per-core 256K L2 caches, shared 6MB L3 cache and 8GB RAM. The test machine had an NVIDIA GeForce GT 650M graphics card with 1024MB video RAM.

3.5 Interaction Logs

To analyze user behavior, we recorded all the *mouse events* along with timestamps. In addition, we logged nine higher-level, more interpretable *application events* together with associated parameters. These application events are: brush, select, range select, pan, zoom, clear, unique data tile, log transform, and color slider. Table 3 explains the meaning of these application events and their associated parameters. Five of these application events (brush, select, range select, zoom, and pan) are *querying events*, which involve both data query processing and rendering. Three other events (clear, color slider, and log transform) only require repainting the visualizations. Due to the use of debouncing (see Section 3.2), a querying event may not actually be processed by imMens and result in visual feedback. For example, when users brush over a histogram at a rate much faster than the delay condition frame rate, debouncing will take effect and drop some of these brushing events to keep the visual interface usable and responsive. We ensured that non-querying events were always processed.

We kept two separate logs for the application events in each session: a log of all the application events *triggered* by a user, and a log of events *processed* by imMens. Triggered events are indicators of user behavior, while processed events reflect the visual feedback that was actually seen by the subjects.

4 Analysis of Interaction Event Logs

In this section we present the methods and results of analysis on user behavior based on logged mouse events and application events. We use linear mixed effects models to statistically analyze the effects of latency on event rates. We report the coefficients of the latency factor and p-values for significance tests. In addition, we provide an assessment of user behavior by analyzing event transitions across different experimental treatments.

4.1 Statistical Analysis: Mixed Effects Models

It is common for psychology and HCI studies to apply analysis of variance (ANOVA) for significance testing. Two main factors in our study are latency and analysis scenario (a combination of the dataset and the configuration of interactive visualizations, sometimes shorthanded as "dataset"). As no single subject experienced all combinations of factors, repeated measures ANOVA is subject to sampling bias and higher chances of type I error [32]. An alternative is to treat the analysis scenario as a fixed effect and perform two-way repeated measures ANOVA to model the subject variance. However, treating the scenario as a fixed effect limits our ability to generalize the results to analysis scenarios not included in the study [14].

Linear mixed effects models are a robust and interpretable alternative to ANOVA for many cases where the study design does not meet the assumptions of ANOVA [5, 18, 27]. While mixed effects models are not (yet) as widely used in HCI and InfoVis research, they are preferable due to their ability to handle missing values. In our case, we treat the latency condition and the order of analysis (whether a dataset is the first or the second seen by the subject) as two fixed effects, and data and subject as random effects modeled using random intercept terms. We are thus able to better generalize the results across subjects and analysis scenarios.

The common practice for assessing significance within mixed effects models is to use likelihood-ratio tests [43]: we build a full model (with the fixed effect in question) and a reduced model (without the fixed effect in question), and compare these models to obtain p-values. We use the lme4 R package [6] for our analysis.

4.2 Results

Each analysis session is identified by a subject ID, the analysis scenario and the latency condition. For each session, we logged three types of event sequences ordered by timestamps: mouse events (consisting of mouse down, mouse up, mouse click), triggered application events, and processed application events. Table 3 gives detailed descriptions of the application events. For each type of event, we derived the event rate (per minute) as dependent variables by normaliz-

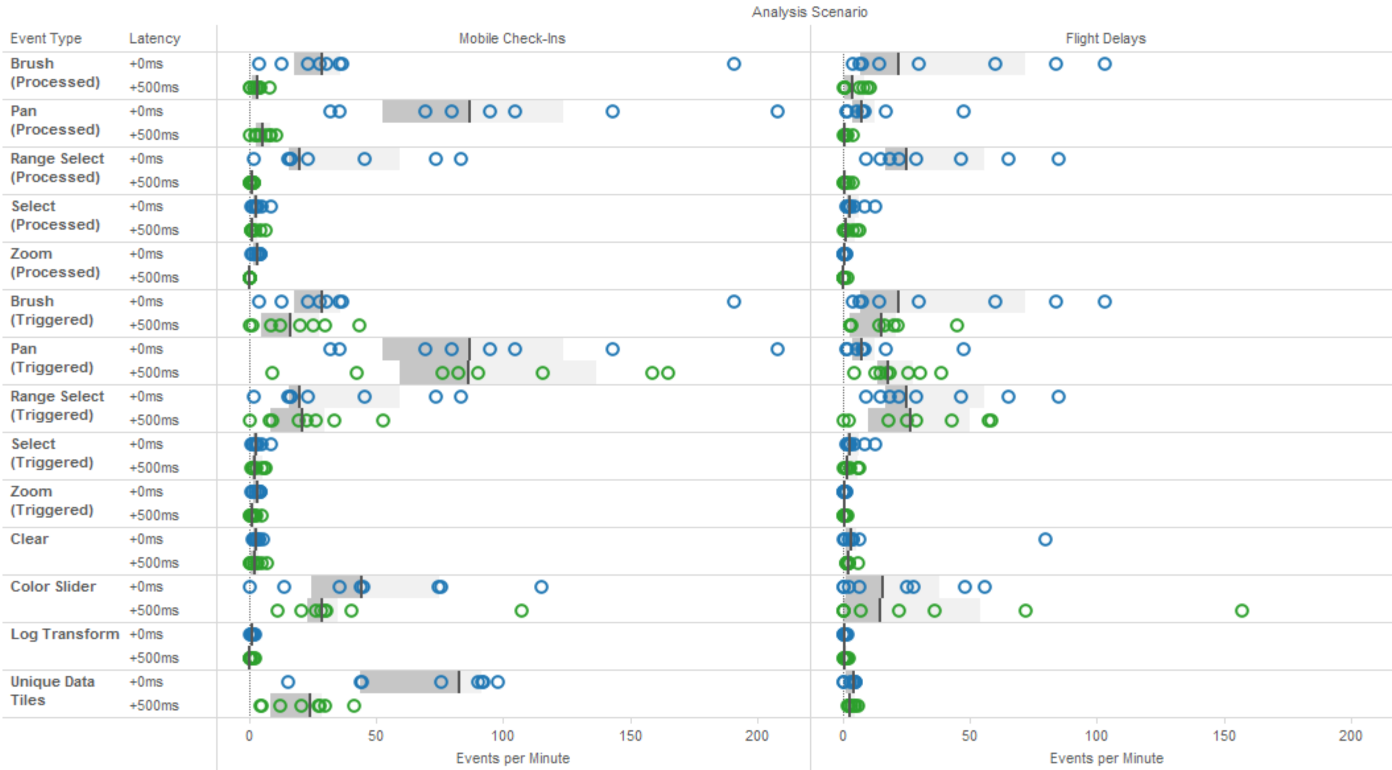


Fig. 2. Event rate data for triggered and processed application events by latency and analysis scenario, annotated with medians and interquartile ranges. Log transform, clear, color slider and unique data tile cache events are always processed.

ing event counts for each session against the session length. Figure 2 shows plots of these metrics across 32 sessions.

We first built mixed effects models to assess the effect of latency on the time spent in an analysis session. The motivating question is: does an additional 500ms delay affect the duration of user engagement? Using likelihood-ratio tests we found that latency had no significant effect on session length. Examining the raw data, most of the subjects adhered to the 30-minute-per-session guideline and tried to spend as much time as allowed to analyze the datasets.

We then assessed the latency effect for each of the event types. In total, we analyzed seventeen metrics as dependent variables: fourteen application events as shown in Figure 2 and three mouse events. For each dependent variable, we built the reduced and full models, noted the coefficients of delay in the full model (Figure 3), and performed likelihood-ratio tests to assess significance. To sanity check, we also performed a two-way repeated measures ANOVA with dataset as a fixed effect. The significance results are identical with the mixed effects model analyses.

4.2.1 Mouse Movements Increase with Lower Latency

We found a strong effect for latency on the rate of mouse movement: $\chi^2(1, N=32) = 7.861$, $p < 0.01$. With increased latency, users decrease their activity, moving the mouse less. In this data, mouse movements are most often low-level indicators of brushing and linking. We did not find significant effects on mouse down and mouse up events.

4.2.2 Latency Affects Triggered Brushes, Shifts User Strategy

We found a significant main effect for latency on triggered brush rate: $\chi^2(1, N=32) = 5.2932$, $p < 0.05$. We did not find significant main effects on the other triggered event rates, although the effect on zooming rate was marginally significant: $\chi^2(1, N=32) = 3.0228$, $p < 0.1$. Figure 3 shows the latency coefficients for each event type in the mixed effects models. The coefficients represent the changes in dependent variables as latency decreases from the delay condition to the control condition. Overall, users shift strategies in response to varying latency. The rates for brushing and range selection increase as latency decreases. This pattern is less obvious in navigational events such as zooming and panning. While the frequencies of panning and color slider events are lower (negative coefficient) under the low latency condition, the effects are not significant.

4.2.3 Processed Querying Events Increase with Low Latency

For processed events, we found significant main effects of latency for all event types except clear, log transform and color slider. In addition, latency had a significant main effect on the number of unique data tiles cached, implying reduced dataset coverage in the delay condition. We did not find any significant effects for order, or for interaction effects between latency and order. Figure 3 reports the detailed test scores.[1] Due to the added latency and event debouncing, we expect decreased rates for processed events in the delay condition and so less responsive visual feedback to the subjects. The positive delay coefficients for processed events in Figure 3 corroborate these findings.

4.3 Event Sequences

We also analyzed the ordered event sequences for behavioral patterns. Plotting event timelines resulted in excessive visual information and no salient patterns emerged. Instead, we construct transition graphs where the nodes are the event types and the links represent aggregated first-order transitions between events across all sessions. We visualize the resulting directed graphs using a matrix diagram (Figure 4).

We find that subjects in the delay condition are less likely to trigger consecutive brush actions. A mixed effects model for the count of per-user consecutive brushes confirms that this effect is significant:

[1]The interaction logs are available at `http://goo.gl/fGh1NM`. The aggregated data file and R scripts to reproduce the results are available at `http://goo.gl/aj0myF`

Event Type	likelihood-ratio test: Chisq(1, N=32)	p value	significance	Latency Coefficient
Brush (Processed)	9.8585	0.00169	**	37.82
Pan (Processed)	14.935	0.0001113	***	50.50
Range Select (Processed)	19.451	0.0000103	***	34.04
Select (Processed)	7.5031	0.006159	**	1.55
Zoom (Processed)	21.881	0.0000029	***	1.49
Brush (Triggered)	5.2932	0.02141	*	25.09
Pan (Triggered)	0.0414	0.8387		-2.30
Range Select (Triggered)	2.3448	0.1257		10.01
Select (Triggered)	2.3645	0.1241		0.79
Zoom (Triggered)	3.0228	0.0821	.	0.54
Clear	1.4405	0.2301		5.28
Color Slider	0.0291	0.8644		-1.22
Log Transform	1.6139	0.2039		0.14
Unique Data Tiles	13.25	0.0002726	***	23.77

Fig. 3. Significance test results and latency coefficients for each event type. Significant effects of latency are observed for triggered brush, processed brush, processed select, processed range select, processed zoom and processed pan. A positive latency coefficient indicates increased event frequency as latency decreases. Significance: $***\ p < 0.001$; $**\ p < 0.01$; $*\ p < 0.05$; $.\ p < 0.1$;

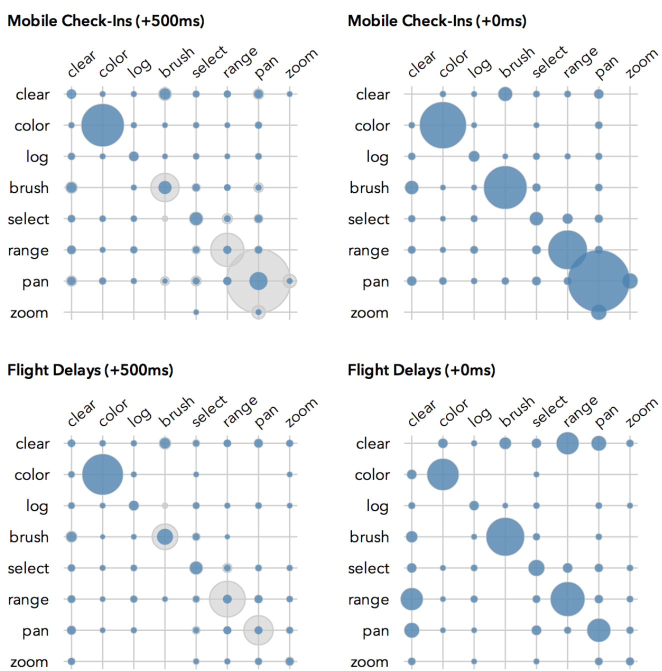


Fig. 4. Transitions between application events by analysis scenario and latency condition. Circular area represents the number of transitions between pairs of event types. Gray circles represent transitions between triggered events; blue circles between processed events. Rows represent source nodes and columns represent target nodes.

$\chi^2(1, N = 32) = 5.4258,\ p < 0.05$. Under the delay condition, user strategies tend to shift towards continuous adjustment of color sliders, which is not affected by the additional delay because it only involves repainting the visualizations without any data processing.

It is also interesting to observe the role played by the dataset in shaping user behavior. For the mobile check-ins dataset, zooming and panning interleave significantly when there is no delay. This pattern is not observed in the flight delays dataset.

5 Analysis of Verbal Data

Mouse and application event rates provide insight into users' behavior patterns. To evaluate the outcome of visual analysis, however, we need to go beyond event logs and analyze qualitative data. Based on audio recordings of subjects' think-aloud protocol and our notes taken during the study sessions, we conduct verbal analysis to assess the effects of latency on knowledge discovery. We segment and code the protocols to produce quantified measures of cognitive events. We then build mixed effects models to assess significance and visualize transitions between verbal categories to understand exploration strategy.

5.1 Segmentation

We manually transcribed the audio recordings to text scripts. The first step to analyze these transcripts is to determine the appropriate level of granularity by segmenting the protocol [12]. For example, consider the following utterance:

"The day histogram doesn't seem to be too useful except for letting me know that there are normally 30 days in a month."

We can treat this sentence as a unit of analysis, or break it down into two propositions (a proposition is usually in the form of a subject-verb-object structure):

"The day histogram is not too useful"

"There are normally 30 days in a month"

The choice of a grain size can be a subtle and complex decision [12]. We chose a proposition as a unit of analysis because coarser grains such as sentences contain varying amounts of information, making calibration across subjects difficult. We made the decision that the number of entities in a predicate should not affect the segmentation process. For example, consider the following sentence:

"the most traveled traveler actually traveled to China, Alaska, Japan, US on the west coast like San Francisco maybe, and then South American like Brazil maybe, I'm not too sure... oh Mexico. And then went to Australia, well not really in that particular order."

We treated the entire sentence as one proposition because the subject was really focusing on a single composite: the places the traveler had been to. Furthermore, the number of places enumerated here could vary depending on the underlying data and how many the participant felt like reporting. On the other hand, in the case of

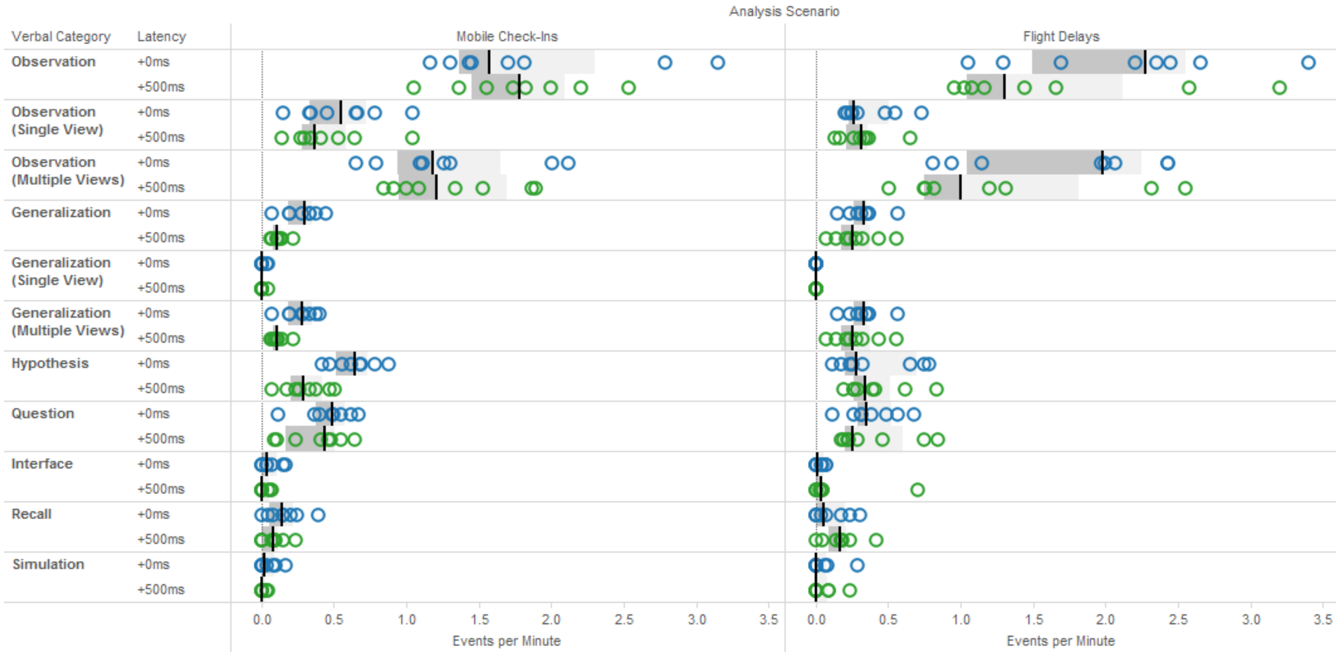


Fig. 5. Verbalization rate for each category by latency and analysis scenario, annotated with medians and interquartile ranges.

"in the hour histogram, 5am is the lowest, 7-8pm is the highest",

we treated the sentence as two propositions because it contains two pieces of information: the hours at which the minimum and maximum number of check-ins occurred.

5.2 Coding

After segmenting the protocols, we focused on coding the propositions into categories. Initially, we tried to extract pieces of knowledge ("insight"), following the insight-based methodology [37]. Two potential problems surfaced. First, it did not seem that all insights should be considered equal. For example, three different subjects made each of the following reports about airlines (identified by two-letter codes):

- *Many new airlines emerged around the year 2003.*
- *HP started in 2001, AS in 2003, PI in 2004, OH in 2003.*
- *OH started in 2003, and they're doing pretty well in terms of delays.*

The three sentences are similar yet contain important differences. The first sentence is a basic generalization; the second is a chain of related specific propositions; and the third contains two propositions, one about the new airline, another about the airline's delay performance. We thought it was especially crucial to distinguish the first two cases because counting the propositions alone would only capture one aspect of the insights uncovered.

Secondly, since we are interested in understanding the potential impacts of latency on analytical behavior and our goal is not to evaluate a visualization design or system, focusing exclusively on "insights" can be limiting. We thus need to consider coding categories that account for interesting information about subjects' cognitive behavior.

Based on these considerations and on our collected transcripts, we iteratively developed a coding scheme with seven categories: observation, generalization, hypothesis, question, recall, interface, and simulation. These categories align well to codes used in prior work for content analysis of comments posted in collaborative visual analysis systems [21, 41]. We elaborate on these categories below.

- *Observation*: an observation is a piece of information about the data that can be obtained from a single state of the visualization system. An observation can be made at the visual level or the data level. For example, *"I see a bump here around 1-2 pm, people are checking in a bit after lunch"*. Here there are two propositions, the first one is an observation directly read off of the visualization, the second is an observation at the data level. We made the decision to code such propositions as a single observation for two reasons. First, the propositions are really describing the same thing; second, some subjects may not literally verbalize what they perceive and only report at the data-level.

- *Generalization*: a generalization is a piece of information acquired from multiple visualization states, often via interaction. Examples of generalizations include *"Most of these airlines have pretty solid flight numbers month over month"*, or *"I don't see any difference between the people who are more frequent of the top 30 and the less frequent of the top 30"*.

- *Hypothesis*: a hypothesis is an inference or a conjecture about the data. A hypothesis may be made before a unit task to steer exploration or as a conjecture to explain an observation or generation. Examples include *"more like I think [the number of check-ins] is more correlated with Internet use or some component of economic development"*.

- *Question*: a question is an indication of desire to examine certain aspects of the data. A question does not have to end with a question mark. For example, in the following sentence, *"I wonder if also people just stopped delaying flights but just start canceling them, so then we might want to look at data that shows the number of cancellations, so that in combination with this."*. The participant first proposes a hypothesis, then poses a question.

- *Recall*: a recall is prior knowledge or personal experience brought into working memory, for example, *"I just happen to know it's [Thanksgiving] the most traveled day of the year"*.

- *Interface*: subjects sometimes make comments about how to improve or revise the visualization interface, for example, *"just from a UI standpoint, I don't need this precision here, if it's just 1k, I think it's OK for it to say 1 [instead of 1.0]"*.

- *Simulation*: subjects perform mental simulations [30, 40] of alternative visual representations not shown in the study to aid the

Verbal Category	likelihood-ratio test: Chisq(1, N=32)	p value	significance
Observation	5.4812	0.01922	*
Observation (Single View)	1.5706	0.2101	
Observation (Multiple Views)	3.3119	0.06878	.
Generalization	8.9763	0.002735	**
Generalization (Single View)	0.2641	0.6073	
Generalization (Multiple Views)	8.5054	0.003541	**
Hypothesis	8.3999	0.003752	**
Question	0.7416	0.3891	
Interface	0.4651	0.4953	
Recall	0.0202	0.8869	
Simulation	0.6983	0.4033	

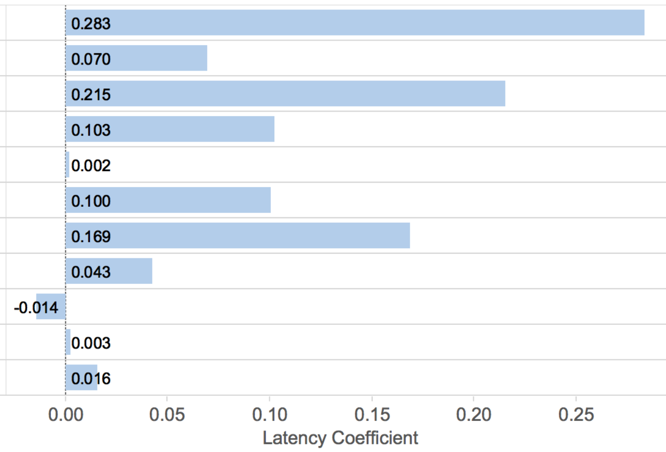


Fig. 6. Significance test results and latency coefficients for each verbal category. Significant effects of latency are found for observation rate, generalization rate, hypothesis rate, generalization rate (multiple visualizations), and observation rate (multiple visualizations). A positive latency coefficient indicates increased event frequency as latency decreases. Significance: $*** \ p < 0.001$; $** \ p < 0.01$; $* \ p < 0.05$; $. \ p < 0.1$;

thinking process, for example, *"so you know if I imagine a 3D histo... a frequency distribution diagram, like a normal distribution, it's like the normal distribution is really wide, like the variance is really really wide here"*.

The first author performed the bulk of the coding. To reduce bias, the coder consulted with the second author and iteratively revised finished codes to ensure consistency across coding sessions. We counted the number of propositions per session for each category, and normalized by session duration to compute event rates. We additionally subdivided the rates of observation and generalization by the number of visualizations involved. When an observation or a generalization is made based on a single view, no interaction is required. Insights drawn from multiple views require brushing and linking across the visualizations. Unlike Saraiya et al. [37], we did not assign quality scores to the coded protocols, as the goal of visual analysis is open-ended.[2]

5.3 Results

Our analysis focused on eleven metrics: observation rate, observation rate (single view), observation rate (multiple views), generalization rate, generalization rate (single view), generalization rate (multiple views), hypothesis rate, question rate, recall rate, interface rate, and simulation rate. Figure 5 plots these metrics across 32 sessions. Not surprisingly, observations and generalizations accounted for the majority of subjects' verbalizations.

5.3.1 Lower Latency Encourages Insight Generation

We again used mixed effects models to test the effects of latency. The models found significant main effects of interactive latency on four metrics: observation rate, generalization rate, hypothesis rate and generalization rate (multiple views). The effect on observation rate (multiple views) was marginally significant. Figure 6 presents the test results.[3] Two-way repeated measures ANOVA produces identical significance results.

Figure 6 shows the latency coefficients for each coded category in the mixed effects models. The coefficients represent the changes in dependent variables as latency decreases from the delay condition to the control condition (+0ms). In general, the subjects perform more verbal reports across all categories except one. The subjects tend to comment more on the visualization interface under the delay condition.

[2] The transcripts are available at `http://goo.gl/mdyQ1T`, and the coding results are available at `http://goo.gl/EOBFPM`

[3] The aggregated data file and R scripts to reproduce the results are available at `http://goo.gl/aj0myF`

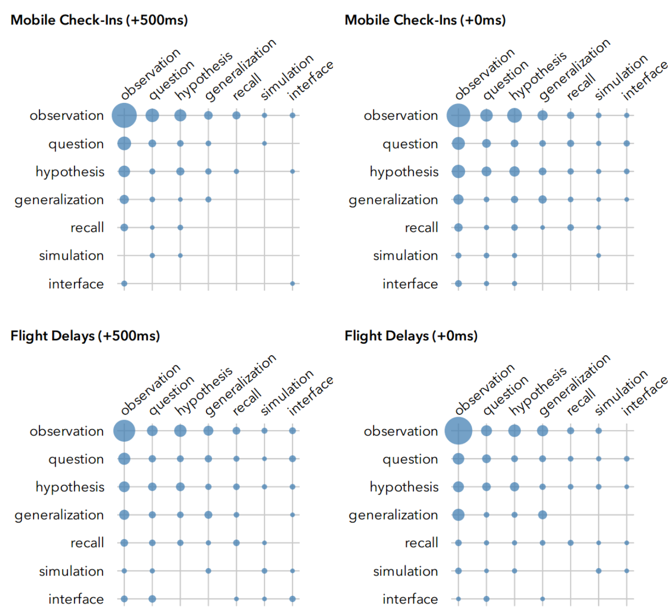


Fig. 7. Transitions between verbal categories by analysis scenario and by latency condition. Circular area represents the number of transitions between from category to another. Rows represent source nodes and columns represent target nodes.

5.3.2 Initial Exposure to Delay Dampens Later Performance

We found significant interaction effects between latency and order for observation rate ($\chi^2(1, N = 32) = 4.803$, $p < 0.05$, interaction coefficient $= -1.2944$) and generalization rate ($\chi^2(1, N = 32) = 4.7204$, $p < 0.05$, interaction coefficient $= -0.20485$). Delays experienced in the first session affect subjects' subsequent performance; even when the delay is removed in the second session, a negative impact on user performance persists.

5.4 Transitions between Verbal Categories

To investigate potential changes in users' exploration strategies, we calculated aggregate transitions between verbal categories, shown as a matrix diagram in Figure 7. Observation is the most common category and users tend to report multiple observations in sequence. Interestingly, the rate of transitions between observation and other data-related categories ("generalization", "hypothesis" and "question") in both directions are significantly reduced in the delay condition ($\chi^2(1, N = 32) = 12.309$, $p < 0.001$). This result indicates that not only do subjects perform knowledge discovery at a higher rate under low latency conditions, their explorations are arguably more dynamic, engaging in sensemaking loops of observing, generalizing and hypothesizing.

6 Discussion

Our experimental results for both event logs and verbalizations provide new insights into the effects of latency on exploratory visual analysis.

6.1 Exploration Strategy

Our analysis of interaction event logs provides evidence of users shifting strategies when the latency changes. We report significant effects in which increased latency reduces mouse movement and triggered brush rates. On the other hand, the effects on zooming and panning rate are not significant.

We hypothesize that panning rate is not affected because the delay only applies to the plotting of data, not the background map images. User thus could reposition the map interactively in a relatively unencumbered fasion. Zooming, on the other hand, may fall into a different cognitive time scale, as discussed in Section 2.2. Users thus may have higher tolerances for the delay. According to Card et al.'s definition, brushing is a "perceptual fusion" act: when brushing results in visual feedback within 100ms or less, consecutive visual updates fuse into one percept. Subjects' verbal reports corroborate this finding: "this is like a nice movie here" (P6), "I think it's obvious that once I have an animated gif I can stare at it" (P4). The added delay likely impedes this fusion effect. The injected delay also slows the system response time by a factor of 25 over the ~20 ms baseline. As a result, when it becomes costly to perform brushing and range selection, users change their behavior to reduce short-term effort. In contrast, zooming may take place at the "unprepared response" time scale; the relative increase in latency accounts for only half of the baseline.

A potential reason for decreased performance in the delay condition is that the less responsive interface dampens user motivation. However, latency had no significant effect on session length or question rate, an indication that users had similar levels of interest in the data under both latency conditions. We hypothesize that the shift in exploration strategy together with the effect of debouncing impacts user performance. Since we found no significant effects on observation rate and generalization rate based on a single visualization view, the differences in performance can be attributed to observation and generalization rates involving multiple views. Given that insights drawn from multiple visualizations depend heavily on brushing and selecting, user performance is adversely affected with the additional delay.

6.2 User Perception of Delay

At the end of each study, we asked subjects if they noticed anything different about the system between the two sessions. While many participants clearly observed the difference in latency, 6 out of 16 subjects did not report a noticeable difference in terms of system responsiveness. Out of these six subjects, half experienced delay in the first session and half analyzed the mobile check-in dataset first. One subject (P9) even remarked:

"One thing that jumps out at me then is the fact that both of these things (datasets) were equivalently responsive, despite the scale of the second [dataset] is like .. you said, one versus a hundred? but yeah in terms of responsiveness the two kind of feel the same."

We informed subjects at the end of the study that we were investigating potential impacts of latency on user behavior. 15 out of 16 subjects did not think the delay, whether perceptible or not, had any influence on their interactions. Our interpretation of such feedback is that the subjects did not deliberately choose different strategies. The affordances and constraints of interactive interfaces, however, often influence cognitive behavior without the need for mindful planning and deliberation [42].

6.3 Implications for System Design

Our study confirms that an injected delay of half a second per operation adversely affects user performance in exploratory data analysis. To conclude that high latency is bad, however, would be an over simplification. As the experiment results demonstrate, some operations, such as zooming, are less sensitive to delay than others. In optimizing system performance, we can take such observations into consideration. Traditionally, visualization system design often takes a modular approach: the visualization pipeline is divided into stages with dedicated components for data management, scene graph construction and rendering. Optimization efforts have largely centered around each of these pieces separately. For example, a number of efforts are concerned with speeding up data processing with little consideration of the corresponding user interface design. Our study suggests the value of taking a user-centric approach to system optimization. Instead of uniformly focusing on reducing latency for each of the processing stages in the visualization pipeline, a potential optimization strategy is to analyze the interaction space supported by the visual interface and balance computational resources between interactive operators. For example, more aggressive caching or prefetching methods may be employed for operations sensitive to small variations in latency, such as brushing and linking.

7 Future Work and Conclusion

In this research, we have found that interactive latency can play an important role in shaping user behavior and impacts the outcomes of exploratory visual analysis. Delays of 500ms incurred significant costs, decreasing user activity and data set coverage while reducing rates of observation, generalization and hypothesis. Moreover, initial exposure to higher latency interactions resulted in reduced rates of observation and generalization during subsequent analysis sessions in which full system performance was restored.

In addition, our work demonstrates an operationalized procedure for the analysis of verbal data collected through think-aloud protocols. We contribute coding categories that capture various types of verbal utterances. These categories can potentially be useful in future studies of qualitative experiment data from exploratory data analysis.

One limitation of our study is that we only collected event logs and verbal data. Gaze data from an eye-tracker could be a valuable adjunct. Analyzing where the subjects were looking might provide additional insights into their exploration strategy. For example, under higher latency, subjects might plan more by scrutinizing visualizations carefully and relying more on working memory. We leave such eye-tracking studies to future work.

Our study also presents evidence that the effects of interactive latency on exploratory visual analysis go beyond the simplistic assertion that low latency uniformly improves usability and performance. The same amount of delay can exert varying degrees of influence on different interactive operators, in turn affecting higher-level cognitive strategy. An important task for future work is to investigate more precise time scale thresholds for each of the major interaction techniques in visualization. Another interesting study is to analyze user behavior with the injected additional delays as a fixed percentage of the time scale thresholds identified for each operation. When designing interactive analysis systems, researchers should not only try to reduce latency, but also be aware of differential effects for different interactive operations. When computational resources are constrained, careful decisions are needed to determine an optimized allocation of resources.

Finally, the exploratory analysis tasks are completely open-ended in our study. Users thus may abandon an exploration path if the interaction cost appears too high. It would be interesting to investigate user behavior with more constrained tasks in follow-up experiments.

8 Acknowledgments

We thank Stu Card for the insightful discussion on time scales in human cognition and Andreas Paepcke for his help in participant recruitment. This research was supported by the Intel Science & Technology Center (ISTC) for Big Data and DARPA XDATA.

REFERENCES

[1] Tableau software. http://www.tableausoftware.com.

[2] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. BlinkDB: Queries with bounded errors and bounded response times on very large data. In *Proc. ACM European Conference on Computer Systems (EuroSys)*, pages 29–42, 2013.

[3] D. H. Ballard, M. M. Hayhoe, and J. B. Pelz. Memory representations in natural tasks. *J. Cognitive Neuroscience*, 7(1):66–80, Jan. 1995.

[4] D. H. Ballard, M. M. Hayhoe, P. K. Pook, and R. P. Rao. Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, 20(04):723–742, 1997.

[5] D. J. Barr, R. Levy, C. Scheepers, and H. J. Tily. Random effects structure for confirmatory hypothesis testing: Keep it maximal. *Journal of Memory and Language*, 68(3):255–278, Apr. 2013.

[6] D. Bates, M. Maechler, and B. Bolker. lme4: Linear mixed-effects models using S4 classes. 2012.

[7] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in Unreal tournament 2003. In *Proc. ACM SIGCOMM Workshop on Network and system support for games*, pages 144–151, 2004.

[8] J. Brutlag. Speed matters for google web search. *Google. June*, 2009.

[9] Bureau of Transportation Statistics. Airline on-time performance data, 2013. http://www.transtats.bts.gov /DL_SelectFields.asp?Table_ID=236.

[10] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[11] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction*. Routledge, 1983.

[12] M. T. Chi. Quantifying qualitative analyses of verbal data: A practical guide. *The Journal of the Learning Sciences*, 6(3):271–315, 1997.

[13] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1082–1090, 2011.

[14] H. H. Clark. The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior*, 12(4):335–359, 1973.

[15] K. A. Ericsson and H. A. Simon. *Protocol Analysis*. MIT press, 1985.

[16] D. Fisher, I. Popov, and S. Drucker. Trust me, I'm partially right: Incremental visualization lets analysts explore large datasets faster. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 2012.

[17] W. D. Gray and D. A. Boehm-Davis. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied*, 6(4):322, 2000.

[18] S. Green, J. Heer, and C. D. Manning. The efficacy of human post-editing for language translation. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, CHI '13, pages 439–448, New York, NY, USA, 2013.

[19] P. Hanrahan. Analytic database technologies for a new kind of user: the data enthusiast. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 577–578, 2012.

[20] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *ACM Queue*, 10(2):30, 2012.

[21] J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1029–1038, 2007.

[22] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. *ACM SIGMOD Record*, 26(2):171–182, 1997.

[23] T. Henderson. Latency and user behaviour on a multiplayer game server. In *Proc. Third International COST264 Workshop on Networked Group Communication*, NGC '01, page 113, London, UK, UK, 2001. Springer-Verlag.

[24] R. Jota, A. Ng, P. Dietz, and D. Wigdor. How fast is fast enough?: a study of the effects of latency in direct-touch pointing tasks. In *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2291–2300, 2013.

[25] D. Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[26] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proc. Advanced Visual Interfaces (AVI)*, pages 547–554, 2012.

[27] C. Krueger and L. Tian. A comparison of the general linear mixed model and repeated measures ANOVA using a dataset with multiple missing data points. *Biological Research for Nursing*, 6(2):151–157, Oct. 2004. PMID: 15388912.

[28] L. Lins, J. Klosowski, and C. Scheidegger. Nanocubes: Fast visualization of large spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 2013.

[29] Z. Liu, B. Jiang, and J. Heer. imMens: real-time visual querying of big data. *Computer Graphics Forum*, 32(3pt4):421–430, 2013.

[30] Z. Liu and J. T. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008, 2010.

[31] A. Michotte. *The Perception of Causality*. Basic Books, Oxford, England, 1963.

[32] K. E. Muller and C. N. Barton. Approximate power for repeated-measures anova lacking sphericity. *Journal of the American Statistical Association*, 84(406):549–555, 1989.

[33] A. Newell. *Unified Theories of Cognition*, volume 187. Harvard University Press, 1994.

[34] A. Ng, J. Lepinski, D. Wigdor, S. Sanders, and P. Dietz. Designing for low-latency direct-touch input. In *Proc. ACM Symposium on User Interface Software and Technology (UIST)*, UIST '12, pages 453–464, New York, NY, USA, 2012.

[35] K. P. O'Hara and S. J. Payne. The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology*, 35(1):34–70, 1998.

[36] K. P. O'Hara and S. J. Payne. Planning and the user interface: the effects of lockout time and error recovery cost. *International Journal of Human-Computer Studies*, 50(1):41–59, Jan. 1999.

[37] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456, Aug. 2005.

[38] P. Saraiya, C. North, V. Lam, and K. A. Duca. An insight-based longitudinal study of visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1511–1522, 2006.

[39] N. Sheldon, E. Girard, S. Borg, M. Claypool, and E. Agu. The effect of latency on user performance in warcraft III. In *Proc. 2nd Workshop on Network and system support for games*, page 314, 2003.

[40] J. G. Trafton, S. B. Trickett, and F. E. Mintz. Connecting internal and external representations: Spatial transformations of scientific visualizations. *Foundations of Science*, 10(1):89–106, 2005.

[41] F. B. Viegas, M. Wattenberg, M. McKeon, F. Van Ham, and J. Kriss. Harry potter and the meat-filled freezer: A case study of spontaneous usage of visualization tools. In *Proc. Hawaii International Conference on System Sciences (HICSS)*, pages 159–159. IEEE, 2008.

[42] T. D. Wilson. *Strangers to Ourselves: Discovering the Adaptive Unconscious*. Harvard University Press, June 2009.

[43] B. Winter. Linear models and linear mixed effects models in R with linguistic applications. *arXiv:1308.5499 [cs]*, Aug. 2013.

[44] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proce. 9th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, page 22, 2012.

[45] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *Proc. 2nd USENIX Conference on Hot Topics in Cloud Computing*, page 1010, 2010.