Preserving Contextual Awareness during Selection of Moving Targets in Animated Stream Visualizations

Eric D. Ragan
University of Florida
eragan@ufl.edu

Andrew Pachuilo
Texas A&M University
apachuilo@tamu.edu

John R. Goodall
Oak Ridge National Laboratory
jgoodall@ornl.gov

Felipe Bacim
Virginia Tech
fbacim@gmail.com

ABSTRACT

In many types of dynamic interactive visualizations, it is often desired to interact with moving objects. Stopping moving objects can make selection easier, but pausing animated content can disrupt perception and understanding of the visualization. To address such problems, we explore selection techniques that only pause a subset of all moving targets in the visualization. We present various designs for controlling pause regions based on cursor trajectory or cursor position. We then report a dual-task experiment that evaluates how different techniques affect both target selection performance and contextual awareness of the visualization. Our findings indicate that all pause techniques significantly improved selection performance as compared to the baseline method without pause, but the results also show that pausing the entire visualization can interfere with contextual awareness. However, the problem with reduced contextual awareness was not observed with our new techniques that only pause a limited region of the visualization. Thus, our research provides evidence that region-limited pause techniques can retain the advantages of selection in dynamic visualizations without imposing a negative effect on contextual awareness.

CCS CONCEPTS

• **Human-centered computing → Empirical studies in visualization**.

KEYWORDS

Visualization, selection techniques, human-computer interaction, animation, streaming data, information visualization

ACM Reference Format:
Eric D. Ragan, Andrew Pachuilo, John R. Goodall, and Felipe Bacim. 2020. Preserving Contextual Awareness during Selection of Moving Targets in Animated Stream Visualizations. In *International Conference on Advanced Visual Interfaces (AVI '20), September 28-October 2, 2020, Salerno, Italy*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3399715.3399832

AVI '20, September 28-October 2, 2020, Salerno, Italy

ACM ISBN 978-1-4503-7535-1/20/09…$15.00
https://doi.org/10.1145/3399715.3399832

1 INTRODUCTION

In many types of interactive visualizations, it is often necessary to select dynamic or moving objects. For example, interactive maps often include animated elements representing moving people, vehicles, or weather systems. Other examples of dynamic visualizations include those focusing on streaming data—data that is continuously updated from sensors or systems as soon as they are available (e.g., [13, 16, 19, 32]). Different types of stream visualizations are commonly used for financial data (e.g., [8, 40]), social media updates (e.g., [11, 42]), or cyber security data (e.g., [32, 43]). In all of these areas, animation is a common method for portraying the passage of time [1, 27, 36], and selecting moving objects is often necessary in order to inspect additional properties of items.

While many researchers have designed interaction techniques to make selection easier in static scenarios (e.g., [4, 9, 18]), selection of moving objects involves different challenges. One straightforward approach for improving selection with moving targets is to pause all motion before selection [22, 24]. While effective for improving target selection, pausing animated content can disrupt the visualization and the ability of the human to continually monitor it. In the case of streaming data visualizations, for instance, one of the primary purposes is to support monitoring incoming data as it arrives and watching for patterns in real time [6, 20, 29]. If the goal is to monitor the data to interpret and respond as quickly as possible, pausing could cause users to miss or delay inspection of new data coming in, or it could cause users to have to fast-forward to "catch up" to real-time arriving data without missing anything.

Though researchers have designed selection techniques for moving targets, prior work has not tested how such techniques influence perception and awareness of dynamic visualizations during selection. Rather than focusing solely on selection performance, our study also considers awareness needs relevant to many visual analysis and monitoring scenarios. We studied a contextual awareness task with a simple animated visualization chosen for its generalizability to existing tools designed to support analysis of dynamic and streaming data (e.g., [3, 21, 36]).

To address possible disruption from pausing techniques in dynamic visualizations, we also investigated alternative methods that stop a subset of possible targets instead of stopping all targets. By limiting the effects of pausing to only the region the user is interested in, such techniques facilitate easier selection while reducing the disruption to the visualization. Based on this concept, we implemented several variations of local pause techniques, and we designed a study to assess how these techniques influence the ability to maintain awareness of visual state during selection.

Our experiment evaluates both selection performance and the ability to monitor the context of a visualization. We present a dual-task study that required participants to select targets and also remember objects on the screen. The study of different pause techniques also tests different visual cues that show the relationship between paused targets and their uninterrupted animated positions in the visualizations. With these factors, our study provides novel data about the trade-offs between selection performance and contextual awareness of dynamic visualizations.

2 RELATED WORK

Our work builds on previous work on selection techniques and visualizations for streaming and time-oriented data.

2.1 Stream Visualization and Analysis

Our study of the impact of selection techniques on contextual awareness of dynamic visualization was motivated by information visualization for streaming and time-series data. The visualization community has designed and evaluated a variety of visual representations for temporal data [1], but the preferred approach depends on the specifics of the data and analysis goals (e.g., [37]). Animation is a common method for visualizing changes over time [27], but it is not always the best approach for all forms of temporal analysis. For example, prior studies have found evidence that small-multiples representations are better than animation for comparing periods of time [35], while animation can offer advantages for detecting patterns and change [3, 17, 37, 39].

Our work is motivated by visualization scenarios involving streaming data where the task involves real-time monitoring data that is continually arriving and updating. When visualizing data in a dynamic, real-time fashion, it is important to simplify the presentation for the user. Fischer et al. [13] identified four criteria for the design and evaluation of real-time data visualizations: *interactive exploration*, *updatability in real-time*, *locality of changes*, and *preservation of temporal context*. *Interactive exploration* is required to explore data streams, which can be challenging when new data is added during exploration. The goal of *updatability* is to enable timely and accurate updates. By *locality of changes*, the arrival of new data or changes to existing data should not distract the user from exploration. Lastly, *preservation of temporal context* is important because the visualization should maintain its historical and current views. Based on these criteria, selection of moving targets should, ideally, avoid interfering with exploration and inspection of elements during real-time data analysis.

Further, visualizations designed for analysis of streaming data are often designed to support two modes of analysis: real-time monitoring of incoming data, and analysis of existing data for context (e.g., [16, 20, 29, 32, 34]). To take advantage of real-time updates, it is necessary to maintain situational awareness of the trends and the current state of the system in order to respond quickly and make appropriate real-world reactions. Continuous monitoring and maintaining a sense of the environment is crucial to such situational awareness [12] to be able to make fast real-time decisions. When interacting with dynamic visualizations in such stream analysis scenarios, it is often important for a user to continue to monitor the state of the data stream [34]. Woods [41] discusses the cognitive challenges of monitoring alarm systems and how it is necessary to maintain an understanding of the state and the interplay among multiple types of signals evolving over time. At times, a specific alarm or event may require more attention for a specific signal, but interpreting that specific event requires an understanding of the context in which the signal was found. For example, in network monitoring for cyber security, when an analyst moves from continuous monitoring to a deep analysis of an event, she needs to continue to keep abreast of new updates that may be even more important than the current focus of her analysis [15].

Such issues are prevalent in various task domains that employ animated visualizations to help monitor and analyze dynamic data sets in real time (e.g., [21, 28, 36]). For example, Sigovan et al. [36] used animated scatterplots for live performance monitoring of high-performance computing systems. In other work, Harrison et al. [21] presented a cyber security tool that used multiple animated scatterplots to support situation awareness and monitoring of network traffic and cyber alerts. Air traffic control is yet another example, as controllers use 2D radar visualizations when making decisions about flight recommendations [5].

For our research of dynamic visualizations, we chose a basic visualization similar to animated scatterplots for two primary reasons: (1) for the simplicity and generalizability to various visual analysis scenarios requiring contextual awareness (e.g., [21, 36]), and (2) to maintain similarity and comparability to existing studies of selection techniques (e.g., [2, 22]).

2.2 Modified Selection Techniques

Many researchers have designed and studied modified techniques to make it easier to select targets. While selection by direct pointing is perhaps the most commonly used approach for selection of static targets, the selection of small targets or selection in dense environments can be challenging because users must slow down to avoid errors. This trade-off between speed and accuracy is described by Fitts' law, which is commonly used to model and compare selection techniques [14, 30].

Numerous selection techniques aim to reduce this effect by increasing the target size or cursor size, or by decreasing movement distance (e.g, [4, 26, 31]). One well-known example is *bubble cursor* [18], which dynamically re-sizes the cursor's activation area based on its nearest target. *Bubble cursor* targets one object at a time based on the cursor's position, and it resizes the cursor based on its distance to the nearest target. A related technique is *DynaSpot* [9], a technique with a dynamic activation area based on the cursor's movement speed. The area can increase up to a given maximum size at high speeds, and the size shrinks when stopped. *Implicit fan cursor* [38] is another cursor technique that takes the cursor's movement direction into account. With this technique, a fan-like selection region grows in the direction of movement with a size based on cursor speed.

Rather than changing the cursor's activation area based on cursor movement, another approach is to dynamically expand the size of targets as the cursor moves toward them, as done by McGuffin and

Balakrishnan [31]. Alternatively, *Drag-and-Pop* [4] moves potential targets closer to the cursor when it is dragging an object, thus allowing faster drag-and-drop functionality by reducing distance the between the target and cursor.

While empirical studies have shown evidence that many variations of specialized selection techniques can work well for static targets, moving targets introduce additional challenges.

2.3 Selection of Dynamic Targets

Selection becomes more difficult when the targets are moving, and such scenarios are not captured well by Fitts' law. As an alternative model, Jagacinski et al. [25] proposed a formulation based on a target's velocity, width, and distance from the cursor. Other models include that of Port and Lee [33], which addresses target prediction and interception strategies.

Researchers have begun exploring various ways to make the selection of moving targets easier. For example, Haan et al. [10] used a volumetric cursor and a method that first calculates a ranking of objects inside the volume based on time and proximity to its center. Then, the cursor snaps to the highest ranking object. Perhaps a more common approach for selection of dynamic targets is the use of pause. For instance, *Click-to-Pause* [2, 24] aids in the selection of moving targets by pausing an entire visualization, thereby reducing the task to that of static target selection. In this technique, depressing the mouse button causes the screen to pause, then selection is triggered when the mouse button is released over a target. While effective and straightforward, stopping the entire visualization can disrupt the experience or result in a loss of temporal information.

Hasan etl al. [22] presented two techniques to partially address this issue: *Target Ghost* and *Comet. Target Ghost* follows the same principle of pausing all moving objects to facilitate easier selection, but it also shows how movement would continue if the object had not stopped. When targets are stopped with *Target Ghost*, static proxy targets are left in place for selection while dimmed versions ("ghosts") of the original objects continue to move along their trajectories. The other technique, *Comet* [22], renders a selectable trail behind each target. The length of the trail is proportional to the object's speed, so target size is increased for faster-moving targets. While the *Comet* technique makes target acquisition easier, the trails of nearby objects can overlap, making them difficult to select. In addition, the added visuals can clutter the visualization.

Such previous work has demonstrated useful ways of selecting moving data, but further investigation is required to understand their implications for disruptions and the ability to maintain contextual awareness of dynamic content.

3 TECHNIQUE DETERMINATION

To enable the evaluation of the effects of pausing selection techniques on contextual awareness in dynamic visualizations, we implemented several techniques.

3.1 Pausing Selection Techniques

Here, we describe multiple variations of techniques used to pause subsets of targets. Figure 1 shows visual representations of how different designs work, demonstrating how pause regions limit the amount of objects that are paused. For all of the pause region techniques, a gray background is drawn to explicitly show the region where the data is paused.

3.1.1 Whole Screen Pausing. Perhaps the most straightforward method for pausing dynamic visualizations is simply to pause the entire view, stopping all items on the screen on key press. After pausing, users can use the cursor normally to select the desired target. This method was previously demonstrated in *Target Ghost* [22] but was not evaluated for potential effects on contextual awareness for monitoring dynamic information.

3.1.2 Cursor Proximity Pausing. Rather than pausing all moving targets, we implemented a *cursor proximity* technique to stop only the targets in a circular area surrounding the cursor. Once targets in the proximity radius are paused, they can then be selected normally with the cursor. The biggest advantage of the cursor proximity method is it is easy to understand. Users can move the cursor towards the intended target per usual, and once the cursor is close, items can be stopped to make selection easier.

The size of the proximity radius can depend on the size and density of available targets. Radius size could also be dynamic, as with modified cursors such as *Bubble Cursor*, *Dynaspot*, or *Implicit Fan Cursor*. However, from pilot testing, we found it important that users understand when targets are pausing; otherwise, dynamic pausing can be distracting and confusing. In addition to using a static radius, we also explored a nearest-neighbor region similar to the *Bubble Cursor* [18], where the pause region was centered around the target closest to the cursor rather than around the cursor itself. The technique takes advantage of nearby targets to predict an area to pause before the cursor moves to that location, but our pilot testing, found participants grew frustrated with the pause region "snapping" to different targets.

To avoid such frustration or confusion with the more advanced variations of proximity techniques, our experiment only tested a simple implementation *cursor proximity* with a constant radius.

3.1.3 Trajectory Pausing. This technique is similar to the *implicit fan cursor* [38] but for creating a pause region instead of a selection region. *Trajectory* pausing uses an angular pause region that reaches in the direction of the cursor's movement. To facilitate selection of far-away targets, the pause region extends to the edge of the window along the cursor trajectory. Though the angle of the pause region could be dynamic, our experiences suggested that technique consistency was important, so we opted for a fixed angle in our final design. The angular region always starts behind the cursor in order to maintain a persistent pause region around the cursor. This design mitigates target overshooting and adds base functionality provided by *cursor proximity* pausing.

While *trajectory* pausing provides the benefit of more-easily pausing distant targets, the technique is more complex than *cursor proximity* due to the dynamically changing trajectory regions. Trajectory pausing can also stop more targets at a time, which could cause further disruption to awareness of the visualization space.

3.2 Context Trails

Simply pausing a target makes it difficult to keep track of where the target would have been if it maintained its original movement. The *Target Ghost* technique [22] is one approach to address this issue

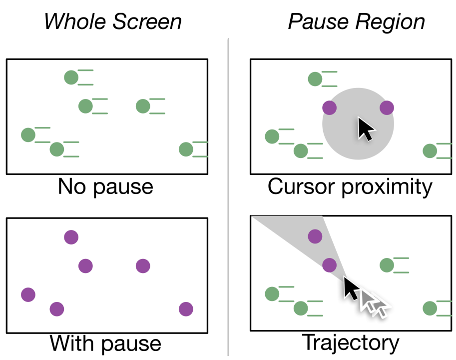

Figure 1: Representation of pause techniques that stop target movement within regions. Moving objects are shown as green circles, and paused objects are shown in purple. On the right, the gray background areas denote the pause region.

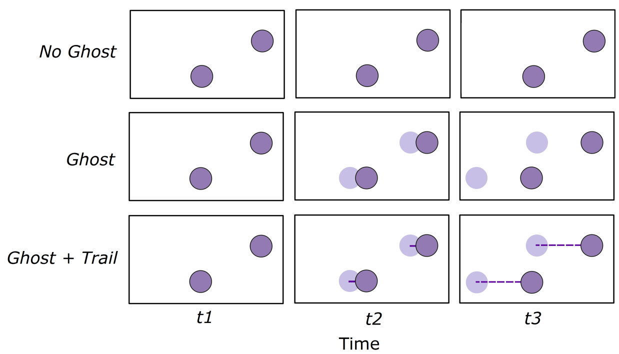

Figure 2: Visual representation of contextual cues that indicate movement of targets when they are paused for selection. Progression of time is shown in the sequence of images from the left to the right. The darker circles are paused, so their positions do not move in each frame, but ghosts and trails show continued movement over time.

by pausing moving targets as a selectable static proxy target while dimmed versions of the original items continue to move normally. However, in dense data sets, it can be difficult to match a paused target proxy to its corresponding moving ghost. To address this concern, we tested techniques with and without ghosting. We also implemented *context trails* as an additional visual cue to augment target ghosting. *Context trails* were dashed lines that connected the dynamic ghosted object to the paused target. When the ghosted object moved beyond the screen, its trail was removed, which helped to represent stale data in the visualization. This approach can help to match the static proxy and the moving object.

4 EXPERIMENT

We ran a controlled experiment with variations of selection techniques to test for trade-offs between selection performance and the ability to maintain contextual awareness of the visualization.

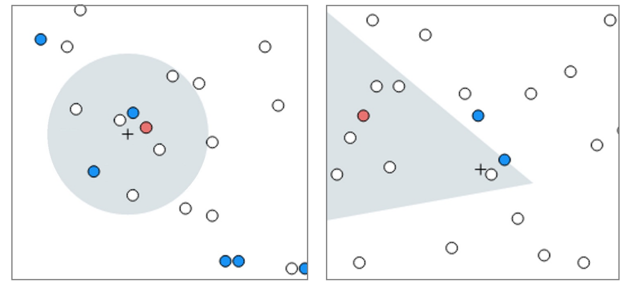

Figure 3: Appearance of study application with *no ghost*. Cursor Proximity Pausing is shown on the left, and Trajectory Pausing is shown on the right.

4.1 Goals and Hypotheses

Pausing techniques can support faster and easier selection of moving targets, but they can also interfere with perception of animated content. With new pausing techniques designed to limit the amount of targets that are stopped, we hypothesized we could retain the benefits of pausing while reducing visual disruption or distraction. Thus, hypothesized techniques with limited pause regions would make it easier to monitor the context of a dynamic visualization but at the cost of reduced selection performance.

We also hypothesized the addition of contextual cues with pause techniques (such as "ghosts" or "trails") can make the techniques easier to understand and enable continued awareness of the context of the visualization during pausing. However, because such methods add clutter that could exacerbate distraction, we expected them to negatively influence the ability to maintain contextual awareness during a selection task.

4.2 Task

We designed the task for to study the effects of pause techniques in scenarios requiring contextual awareness while both monitoring and interacting with an interactive visualization. The visual basis for the task was an animated field of circles, which was chosen to retain similarity to other researchers' prior work on selection techniques (e.g., [2, 22]). The design is also similar to an animated scatterplot, a common and fundamental representation that others have used for visualizations of streaming data (e.g., [21, 23, 36]).

We used a dual task setup to simulate a monitoring scenario that required users to maintain awareness of the overall visual state of the visualization while selecting specific targets. To this end, the primary task was presented as a selection task: participants were asked to continually select red targets from a field of white and blue circles during a brief timed period. At the end of the period, the visualization was hidden, and participants were immediately asked to estimate the number of blue circles that were on the screen when the period ended. With this design, participants were encouraged to pay attention to or recall the visual state of the application so they could complete the secondary counting task.

In the application, white and blue circles continually moved from the right side of the screen to the left. The number of white circles on the screen was dependent on the density level of the experimental condition (independent variables are explained in the following section). The number of blue circles on screen at any given time ranged from 4 to 16 blue dots, with the count being independent of experimental condition.

Users selected circles by using a mouse to move a crosshair (point cursor) over a circle and clicking. All circles were outlined with a black stroke that thickened when the user hovered over the circle to indicate target acquisition for selection.

Figure 3 shows the application with examples using the *ghost* contextual cue for paused objects. As part of the experimental task, the field of white and blue circles contained one red circle at any given time. Participants were instructed to select as many red circles as possible before they ran out of time. When clicked, the red circle turned white, and a different white circle on the screen would turn red and become the new target. The new target was randomly selected from the white circles on the screen.

If the red target moved beyond the edge of the screen before it was selected, then a new target was created from an existing white circle. However, in trials with target ghosts or contextual trails, it was possible for a selectable proxy to remain paused on the screen after a ghosted red dot exited the screen. In such cases, the application provided a two second window to select the proxy target before a new target was chosen.

Each trial lasted 5–10 seconds. The exact time varied per trial to prevent participants from getting used to counting the blue circles right before time expired. At the end of the time period, the screen immediately changed from the field of moving circles to a new screen asking how many blue circles were on the screen at the end of the trial. Participants answered by clicking on a number from a list of numerical buttons ranging from 0 to 20.

Participants were instructed to prioritize correct target selection were reminded to do so throughout the experiment.

4.3 Experimental Design

The experiment controlled four independent variables: pause technique, contextual cue, target speed, and target density. The variations of pause techniques provided different mechanisms for determining the pause region. The four *pause techniques* tested in the final experiment were:

- Whole Screen Pausing: All items on the screen could be paused.
- Cursor Proximity Pausing: The pause region was a 2.2 cm (70 px) radius area around the cursor. The pause region was visibly shown to users as a transparent gray overlay.
- Trajectory Pausing: The pause region was a 50 degree angular area that extended from behind the position of the cursor towards the direction of the cursor's movement trajectory. The pause region was shown with a transparent gray overlay. In the implementation for the experiment, the trajectory region started 1.5 cm (48 px) behind the cursor. To smooth the movement of the trajectory, we used a Catmull-Rom spline and defined the trajectory based on the four most recent positions of the cursor from the mouse polling.
- Baseline (No Pause): No pausing technique was available to aid selection.

For the experiment, all techniques were provided with manual control of pausing. Participants could click the *shift* key of the keyboard to active pausing at the time of key press. Pressing *shift* again cancelled the previous pause effect and update a new pause. Participants could also clear all pausing by using the *C* key. With this design, participants could use the left hand to control the pause functionality and the right hand to control the mouse (all participants were right handed).

In addition to the pause techniques, three types of *contextual cues* were tested with the pause techniques:

- Ghost: When targets were paused, a dimmed "ghost" version of the object continued moving along its path, as done in the *Target Ghost* technique [22].
- Ghost+Trail: With this cue enabled, paused targets had the ghost cue, and an additional dashed line was rendered to connect the moving ghost to the paused target.
- No Ghost: When targets were paused, no ghosted version was shown.

As another independent variable, the *target density* of circles in the application was varied with two levels: high and low. The distribution of circles was generated with 0.36 cm (11.7 px) between circles in the low density variation and with 0.15 cm (4.7 px) between circles for high density. Low density conditions had approximately 77 circles on screen at once, and high density conditions had approximately 194 circles at once.

Two *target speeds* were also tested: slow and fast. Circles moved at a speed of 1.41 cm/second (45 px/second) with the slow setting, and they moved at 3.7 cm/second (119 px/second) on the fast setting.

The experiment followed a within-subjects design so all participants used all combinations of the variations. If all combinations of all treatment levels were possible, the experiment would have a 4x3x2x2 design. However, the *baseline* technique did not allow pausing and therefore did not have *ghost* or *ghost+trail* variations.

For condition ordering, all variations of each pause technique were completed together. We used a Latin square design to balance the orderings of the four pause techniques for each participant. For each pause technique, the ordering of the three contextual cues was randomly determined at run time. The exception was the *baseline* technique with no pause functionality and only the *no ghost* version. Thus, the combinations of pause technique and contextual cue yielded a total of 10 technique configurations (3x3+1). Within each configuration, a random ordering was determined for the four combinations (2x2) of speed and density. For each condition, participants completed the task five times, yielding a total of 200 trials per participant (not including practice trials).

Assessment of selection performance used the rate of successful selections from the number of targets selected per second. We similarly recorded the rate of selection errors; an error was any selection attempt that did not select the red target. An error was also counted if the target moved off the screen and could not be selected (that is, the target was not paused at the time). To assess contextual awareness, we measured the error of the estimated number of blue circles as compared to the actual count.

4.4 Apparatus and Implementation

We ran our tests using a mouse and keyboard setup with a 27-inch monitor with 1920x1080 screen resolution. Participants used a Dell MS111-P corded mouse with 1000 DPI (dot per inch) and mouse acceleration set to zero. The experiment was created using the D3 JavaScript library [7], which ran in Chrome Stable. The experiment application ran inside a 28x11 cm (900x360 px) SVG window. Target graphics were presented as circles with a radius of 0.15 cm (5 px).

4.5 Procedure

The study was approved by the Institutional Review Board (IRB) at our organization. At the beginning of the study, participants were asked to complete a short questionnaire about their age, gender, education, occupation, computer usage, and video game experience. We then provided instructions and explained the goal was to prioritize selecting red targets, but they would also be asked to estimate how many blue objects were on the screen when time expired.

Before each new combination of pause technique and contextual cue, participants were required to practice with the technique to get used to performing the tasks. The first practice trial was one minute long with randomly selected levels of speed and density. Then, four more mandatory practice trials followed; each was 5–10 seconds long with a random ordering of the four possible combinations of speeds and target densities. Following these five mandatory practice trials, participants were permitted to complete additional practice trials, but they were not required to do so.

Participants then completed 20 trials for the combination of technique and contextual cue (five trials for the each of the 2x2 speed and density combinations). Participants were also required to take at least two breaks during the study. Break periods were scheduled for before the beginning of each new pause technique (excluding the block of baseline trials). After finishing all trials for each combination of technique and contextual cue, participants were asked to complete a brief questionnaire to provide feedback about each technique. The average amount of time for an entire participant session was approximately 80 minutes.

4.6 Participants

The experiment had 20 participants (16 males, 4 females). Ages ranged from 20 to 30 with mean age and median age both being 24 years. Participants were students in varying undergraduate and graduate degree programs. Academic disciplines included a wide range of fields, but the most common area (50%) involved graphic and animation. Eleven participants were gamers, and six reported regularly playing video games using a mouse each week.

5 RESULTS

We used statistical testing and graphical analysis to interpret the results of the experiment. In the following sections, we prioritize reporting statistically significant effects and differences. We also note that in all charts, error bars denote standard error.

5.1 Statistical Testing Approach

We used repeated measures ANOVAs for the statistical analysis. Full factorial testing was not appropriate due to the unbalanced design because it was not possible to have the ghost and ghost+trail cues in the baseline technique with no pausing functionality. Therefore, we decided to use two ANOVAs: (1) a test including the baseline technique but ignoring differences in contextual cues, which allowed us to compare the different techniques, and (2) a full factorial test that excluded the baseline technique but tested interactions between contextual cue and pause technique. We also note that not all metrics met the assumption of sphericity for parametric testing. In such cases, we present corrected degrees of freedom and adjusted p values using Greenhouse-Geisser estimates (noting $\epsilon < 0.75$ for all outcomes).

5.2 Selection Performance Results

Average performance outcomes per technique and contextual cue are summarized in Figures 4 and 5, which shows mean rates of successful selection and mean rates of selection errors, respectively. As can be seen in Figure 4, rate of successful selections was fairly consistent, averaging near one successful selection every two seconds. Among the pause techniques, Figure 4 does show a slower selection rate with the trajectory technique. It is surprising to note that the average selection rate for the baseline technique without the pause functionality was similar to that of the other pause techniques. However, the selection error results provide a more complete picture (Figure 5), showing that the baseline technique had over five times more errors than the pausing techniques.

The ANOVA testing with all techniques found a significant main effect of technique for both selection rate, with $F_{(3,57)} = 32.00$ and $p < 0.001$, and error rate, with $F_{(1.04,19.80)} = 28.36$ and $p < 0.001$. Bonferroni-corrected pairwise comparisons found that the baseline technique had a significantly higher error rate than all other techniques ($p < 0.001$), and the trajectory technique had a slower successful selection rate than the baseline as well as both the proximity cursor and whole-screen pause techniques ($p < 0.001$).

These test results agree with the graphical analysis interpretation that the baseline technique had a similar selection rate but more errors than the other techniques. Additionally, the results indicate that the trajectory technique was slower than the other pause techniques, which we expect is related to the extra split second needed for the pause region to update based on the cursor trajectory.

The ANOVA that accounted for contextual cues found a significant effect of cues on selection rate with $F_{(2,38)} = 7.19$ and $p < 0.005$, and pairwise comparisons (with Bonferroni correction) found that selection rate was significantly slower with *ghost+trail* than either *ghost* cues or *no ghost* cues ($p < 0.05$). Graphical analysis shows the differences due to contextual cues to be small (see Figure 4), and the effect is most noticeable for the *whole screen pausing* technique. This suggests that although the addition of contextual cues can make it easier to understand what is happening with pause techniques, the additional visual information may invoke additional perceptual distraction that can slow selection performance—at least with the addition of too much supplemental contextual markup.

Overall, selection performance was worse with the faster speed (Figure 6). Rates of successful selections were significantly worse for the faster speed with the ANOVA with all techniques yielding $F_{(1,19)} = 4.94$ and $p < 0.05$. Faster speeds also had significantly more error, with $F_{(1,19)} = 28.42$ and $p < 0.001$. However, differences were small except for the baseline case without pausing, and the interaction between technique and speed was significant for both selection rate, with $F_{(3,57)} = 3.64$ and $p < 0.05$, and error rate, with $F_{(1.18,\ 22.48)} = 27.71$ and $p < 0.001$. The posthoc test confirmed that the only significant pairwise differences due to speed were for the

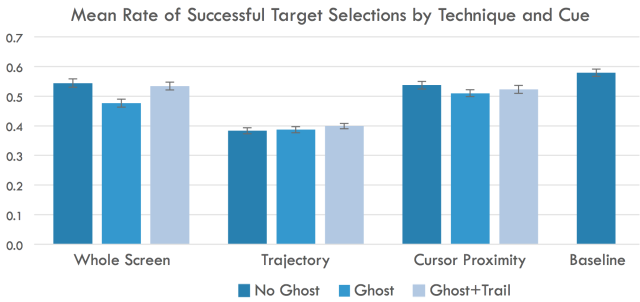

Figure 4: Rate of successful (correct) target selections per second. Higher bars indicate better performance.

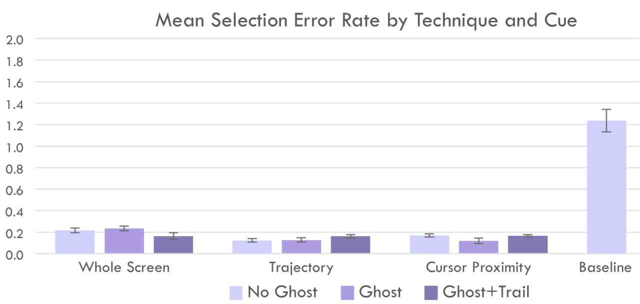

Figure 5: Selection error rate (errors per second). Higher bars indicate worse performance.

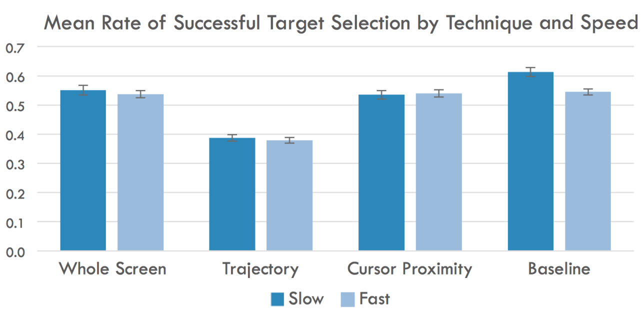

Figure 6: Successful (correct) selections per second by speed and technique. Higher bars indicate better performance.

baseline. Thus, this results demonstrates that all pause techniques were fairly robust for target selection with different speeds, but selection performance is more difficult without support for pausing.

Not surprisingly, increased object density resulted in significantly worse selection performance. Test results yielded $F_{(1,19)} = 12.23$ and $p < 0.005$ for selection rate, and the results were $F_{(1,19)} = 7.74$ and $p < 0.05$ for error rates. No interaction effects were detected involving density.

5.3 Contextual Awareness Results

Average counting error was used as the dependent outcome for contextual awareness. For reference, Figure 8 shows average counting errors as a percentage of the error in the baseline condition. When testing for effects with the test including cues, a significant

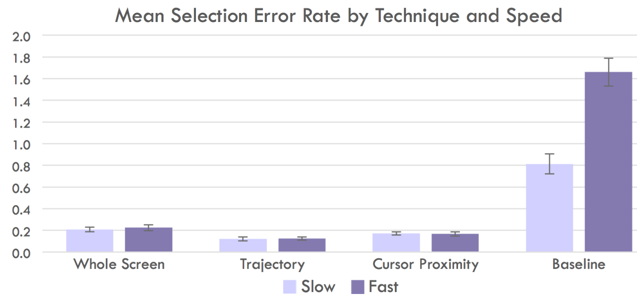

Figure 7: Selection error rate (errors per second) by speed and technique. Higher bars indicate worse performance.

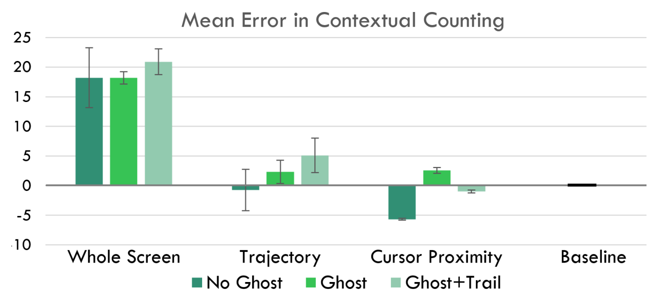

Figure 8: Counting error as a percentage of the baseline error. Bars indicates distance from the correct number. Higher bars indicate worse counting than the baseline; bars below the 0 line indicate lower error than the baseline.

effect of technique was found with $F_{(2,38)} = 5.67$ and $p < 0.01$. Graphical inspection shows that the baseline, trajectory, and cursor proximity techniques had similar counting error, and the whole screen pause technique had higher (near 20%) error (see Figure 8). Bonferroni-corrected posthoc testing only detected significant differences between the whole screen and cursor proximity techniques ($p < 0.01$). These results suggest that contextual awareness was more difficult with the pausing techniques that interfered with the entire screen, but the localized pause techniques were beneficial for overall awareness.

There was a significant effect on counting due to speed, with $F_{(1,19)} = 12.67$ and $p < 0.005$. As would be expected, there were more counting errors with faster speed, and this effect was consistent across all techniques.

Contrary to our expectations, there was no significant effect of contextual cue on counting errors. That is, the addition of visual ghosts and trails did not cause significant penalties for contextual awareness. Also, no significant effects on counting errors were detected due to density, and no interaction effects were detected regarding the counting metric.

6 DISCUSSION

Our work provides new knowledge about the use of pause techniques to aid selection of moving targets.

6.1 Overall Feasibility of Pause Techniques

The study evaluated pause techniques and focused on variations that only pausing limited regions of the dynamic visualization to facilitate selection. The results provide evidence that although whole-screen pausing significantly improves selection performance over the baseline, pausing the entire visualization can significantly interfere with the ability to maintain awareness of the state of the visualization. We demonstrated that it is possible to pause subsets of moving targets with varying methods to control the techniques' pause activation areas, and we investigated how additional visual cues added by pause techniques may cause problems for contextual awareness. The tested limited-area pause techniques (*cursor proximity* and *trajectory* pausing) support the selection benefits of whole-screen pausing while making it significantly easier to maintain awareness (see Figure 8)

Our experiment demonstrates that all types of pause techniques made selection easier than with no pause functionality. In terms of selection performance with the different techniques, the study showed that only the trajectory pause technique was significantly different from the other techniques (including the baseline), as shown in Figure 4. Despite being designed to define the pause region based on the direction of cursor movement, the selection-rate results indicate that the trajectory technique was somewhat confusing due to the angular selection region adjusting during the first several mouse polls in a new movement direction.

The negative perception of the trajectory technique could have been due to confusion while the trajectory region aligned with movement direction. Taking this into consideration, we designed a potential design improvement that combines the benefits of the trajectory technique's predictive pausing and the proximity cursor's simpler control. The alternative design would have a cursor proximity region as the default state when the cursor was static or moving slowly, but the pause region could branch out along the movement direction for bigger or faster mouse movements. This design would improve the brief early stage of the original trajectory design, during which the pause region changes rapidly to adjust to the new movement direction. We are interested in testing such a design with different target speeds and densities, as well as for more complex target movement patterns.

6.2 Selection and Contextual Awareness

More broadly, the evaluation provides findings about the general use of pause techniques for interaction with dynamic data visualization. Overall, selection error rate was significantly higher with the baseline selection without pausing, with error rate being approximately five times worse without any pausing (see Figure 5). Thus, the results clearly show that pause techniques make selection of moving targets easier. Similar results were found by other tests with pause techniques [22]; however, by adding the contextual awareness task, our study was able to assess other tradeoffs of pause techniques related to perception of the visual state of an animated visualization. Contextual counting errors were noticeably worse when pausing the entire visualization as compared to the other techniques (see Figure 8).

It would make sense that the additional visual representation of the paused visualization could have caused some interference with visual memory of the visualization, and this would be worse with the addition of the contextual cues (ghost and trails). While these added contextual cues do make it possible to track movement of paused targets, such contextual tracking was not necessary or beneficial for either selection or the counting task in this study. So, while it seems logical that many practical applications might want to provide contextual cues for pause techniques, designers should also be aware of the potential negative effects. Additional research could investigate whether different types of contextual cues impact tasks where it is helpful to maintain awareness of object trajectory and continued movement.

Visualization of temporal or streaming data is one application area that could benefit from continual situational awareness during data monitoring or analysis [6, 20, 32]. Numerous previous visualizations require maintaining awareness while objects continually move across a visual field, with examples including radar monitoring (e.g., [5, 28]), cyber security (e.g., [15, 21]), and computational performance monitoring (e.g., [36]). Considering general similarities with the multi-task scenario of our study, we predict that the results could be similar to such real-world scenarios that involve inspecting individual items (e.g., planes, cyber alerts, computational processes) while also monitoring the overall state of the system. In future studies, we plan to study how selection techniques and visual cues might influence performance on data analysis tasks and more complex situational awareness tasks. It will also be important to consider similar contextual awareness with dynamic visualizations in more domain-specific applications, but the more general approach taken through our research provides findings that are more accessible to the broader visualization community.

7 CONCLUSION

We studied how motion-pausing selection techniques influence both selection performance and contextual awareness of a dynamic visualization. All tested variations of pause techniques improved selection performance, and we learned about the advantages and limitations of our *trajectory* and *cursor proximity* designs through empirical evaluation. While pausing the entire visualization can negatively influence contextual awareness, pausing partial regions of the visualization can retain benefits of pausing while allowing users to better maintain awareness of the overall visual state.

ACKNOWLEDGMENTS

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan http://energy.gov/downloads/doe-public-access-plan. This research is supported in part by the DARPA XAI program under Grant N66001-17-2-4032.

REFERENCES

[1] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. 2007. Visualizing time-oriented data—A systematic view. *Computers & Graphics* 31, 3 (2007), 401–409.

[2] Abir Al Hajri, Sidney Fels, Gregor Miller, and Michael Ilich. 2011. Moving target selection in 2D graphical user interfaces. In *Human-Computer Interaction–INTERACT 2011*. Springer, 141–161.

[3] Daniel Archambault, Helen Purchase, and Bruno Pinaud. 2011. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics* 17, 4 (2011), 539–552.

[4] Patrick Baudisch, Edward Cutrell, Dan Robbins, Mary Czerwinski, Peter Tandler, Benjamin Bederson, Alex Zierlinger, et al. 2003. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch-and pen-operated systems. In *Proceedings of INTERACT*, Vol. 3. 57–64.

[5] Richard Bentley, John A Hughes, David Randall, Tom Rodden, Peter Sawyer, Dan Shapiro, and Ian Sommerville. 1992. Ethnographically-informed systems design for air traffic control. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. ACM, 123–129.

[6] Daniel M Best, Shawn Bohn, Douglas Love, Adam Wynne, and William A Pike. 2010. Real-time visualization of network behaviors for situational awareness. In *Proceedings of the seventh international symposium on visualization for cyber security*. ACM, 79–90.

[7] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on* 17, 12 (2011), 2301–2309.

[8] Badrish Chandramouli, Mohamed Ali, Jonathan Goldstein, Beysim Sezgin, and Balan Sethu Raman. 2010. Data stream management systems for computational finance. *Computer* 12 (2010), 45–52.

[9] Olivier Chapuis, Jean-Baptiste Labrune, and Emmanuel Pietriga. 2009. DynaSpot: speed-dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1391–1400.

[10] Gerwin de Haan, Michal Koutek, and Frits H. Post. 2005. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Proceedings of the 11th Eurographics Conference on Virtual Environments* (Aalborg, Denmark) *(EGVE'05)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 201–209.

[11] Nicholas Diakopoulos, Mor Naaman, and Funda Kivran-Swaine. 2010. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*. IEEE, 115–122.

[12] Mica R Endsley. 1995. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 37, 1 (1995), 32–64.

[13] Fabian Fischer, Florian Mansmann, and Daniel A Keim. 2012. Real-time visual analytics for event data streams. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, 801–806.

[14] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.

[15] John R. Goodall, Wayne G. Lutters, and Anita Komlodi. 2009. Developing expertise for network intrusion detection. *Information Technology & People* 22, 2 (2009), 92–108.

[16] John R Goodall, Eric D Ragan, Chad A Steed, Joel W Reed, G David Richardson, Kelly MT Huffer, Robert A Bridges, and Jason A Laska. 2018. Situ: Identifying and explaining suspicious behavior in networks. *IEEE transactions on visualization and computer graphics* 25, 1 (2018), 204–214.

[17] Amy L Griffin, Alan M MacEachren, Frank Hardisty, Erik Steiner, and Bonan Li. 2006. A comparison of animated maps with static small-multiple maps for visually identifying space-time clusters. *Annals of the Association of American Geographers* 96, 4 (2006), 740–753.

[18] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 281–290.

[19] Chetan Gupta, Song Wang, İsmail Arı, Ming Hao, Umeshwar Dayal, Abhay Mehta, Manish Marwah, and Ratnesh Sharma. 2011. Chaos: A data stream analysis architecture for enterprise applications. (2011).

[20] Ming Hao, Daniel A Keim, Umeshwar Dayal, Daniela Oelke, and Chantal Tremblay. 2008. Density displays for data stream monitoring. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 895–902.

[21] Lane Harrison, Jason Laska, Riley Spahn, Mike Iannacone, Evan Downing, Erik M Ferragut, and John R Goodall. 2012. situ: Situational understanding and discovery for cyber attacks. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*. IEEE, 307–308.

[22] Khalad Hasan, Tovi Grossman, and Pourang Irani. 2011. Comet and target ghost: techniques for selecting moving targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 839–848.

[23] Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. 2013. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2446–2455.

[24] Michael Victor Ilich. 2009. *Moving target selection in interactive video*. Ph.D. Dissertation. Simon Fraser University.

[25] Richard J Jagacinski, Daniel W Repperger, Sharon L Ward, and Martin S Moran. 1980. A test of Fitts' law with moving targets. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 22, 2 (1980), 225–233.

[26] Paul Kabbash and William AS Buxton. 1995. The "prince" technique: Fitts' law and selection using area cursors. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 273–279.

[27] Simone Kriglstein, Margit Pohl, and Claus Stachl. 2012. Animation for time-oriented data: An overview of empirical research. In *Information Visualisation (IV), 2012 16th International Conference on*. IEEE, 30–35.

[28] Thomas J Laffey, Preston A Cox, James L Schmidt, Simon M Kao, and Jackson Y Readk. 1988. Real-time knowledge-based systems. *AI magazine* 9, 1 (1988), 27.

[29] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey P Lankford, and Donna M Nystrom. 2004. Visually mining and monitoring massive time series. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 460–469.

[30] I Scott MacKenzie. 1992. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction* 7, 1 (1992), 91–139.

[31] Michael McGuffin and Ravin Balakrishnan. 2002. Acquisition of expanding targets. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 57–64.

[32] Peter McLachlan, Tamara Munzner, Eleftherios Koutsofios, and Stephen North. 2008. LiveRAC: interactive visual exploration of system management time-series data. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 1483–1492.

[33] Nicholas Lindman Port, Daeyeol Lee, Paul Dassonville, and Apostolos P Georgopoulos. 1997. Manual interception of moving targets I. Performance and movement initiation. *Experimental Brain Research* 116, 3 (1997), 406–420.

[34] Eric D Ragan, Andrew S. Stamps, and John Goodall. 2020. Empirical Study of Focus-Plus-Context and Aggregation Techniques for the Visualization of Streaming Data. In *Proceedings of the 2020 International Conference on Advanced Visual Interfaces (AVI)*. 1–5.

[35] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. 2008. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics* 14, 6 (2008), 1325–1332.

[36] Carmen Sigovan, Chris W Muelder, and Kwan-Liu Ma. 2013. Visualizing Large-scale Parallel Communication Traces Using a Particle Animation Technique. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 141–150.

[37] Terry Slocum, Robert Sluter, Fritz Kessler, and Stephen Yoder. 2004. A qualitative evaluation of MapTime, a program for exploring spatiotemporal point data. *Cartographica: The International Journal for Geographic Information and Geovisualization* 39, 3 (2004), 43–68.

[38] Xiaojun Su, Oscar Kin-Chung Au, and Rynson WH Lau. 2014. The implicit fan cursor: a velocity dependent area cursor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 753–762.

[39] Tatiana Tekušová and Jörn Kohlhammer. 2007. Applying animation to the visual analysis of financial time-dependent data. In *Information Visualization, 2007. IV'07. 11th International Conference*. IEEE, 101–108.

[40] Martin Wattenberg. 1999. Visualizing the stock market. In *CHI'99 extended abstracts on Human factors in computing systems*. ACM, 188–189.

[41] David D Woods. 1995. The alarm problem and directed attention in dynamic fault management. *Ergonomics* 38, 11 (1995), 2371–2393.

[42] Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems* 6 (2012), 52–59.

[43] Álvaro Herrero, Emilio Corchado, and José Manuel Sáiz. 2006. MOVICAB-IDS: Visual Analysis of Network Traffic Data Streams for Intrusion Detection. In *IDEAL 2006. LNCS*. Springer, 1424–1433.