DimpVis: Exploring Time-varying Information Visualizations by Direct Manipulation

Brittany Kondo and Christopher Collins

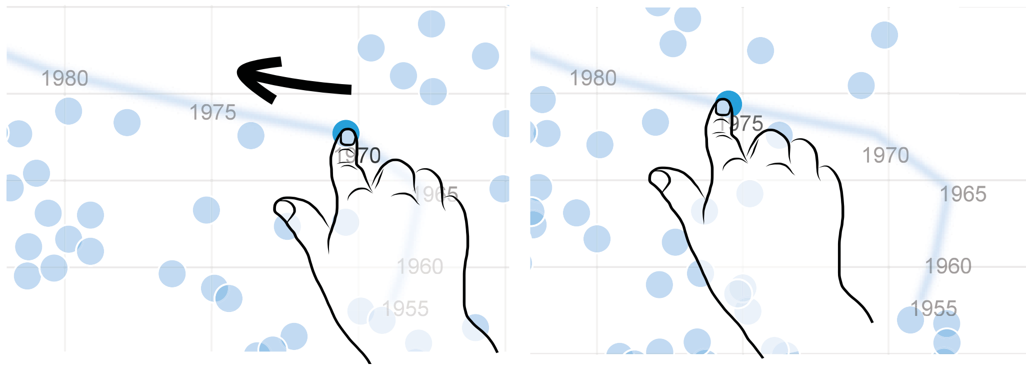

Fig. 1. DimpVis is an interaction technique for navigating time in information visualizations through direct manipulation of visualization objects. The hint path reveals the locations of a selected visual item over time. Navigation in time for scatter plots is achieved by dragging a selected point in 2D along its hint path.

Abstract—We introduce a new direct manipulation technique, DimpVis, for interacting with visual items in information visualizations to enable exploration of the time dimension. DimpVis is guided by visual hint paths which indicate how a selected data item changes through the time dimension in a visualization. Temporal navigation is controlled by manipulating any data item along its hint path. All other items are updated to reflect the new time. We demonstrate how the DimpVis technique can be designed to directly manipulate position, colour, and size in familiar visualizations such as bar charts and scatter plots, as a means for temporal navigation. We present results from a comparative evaluation, showing that the DimpVis technique was subjectively preferred and quantitatively competitive with the traditional time slider, and significantly faster than small multiples for a variety of tasks.

Index Terms—Time navigation, direct manipulation, information visualization

1 Introduction

Many types of data, such as census statistics, stock market prices, and twitter feeds change over time. Familiar chart types, such as bar charts and scatter plots can be used to represent this time-varying data. Changes in data values over time are most often shown through animation, usually paired with a separate time slider widget. Using this technique requires divided attention—manipulating the time slider while observing how items of interest change. Alternatively, images of the visualization at each moment in time can be presented side-by-side (known as small multiples [31]). However, images do not convey motion, which is important for investigating and understanding temporal trends. Dragicevic et al. created a non-linear video browsing technique, where any visual object can be dragged along its motion trajectory to navigate time [10]. Likewise, Wolter et al. designed a technique for dragging visualization objects along their 3D motion trajectories for navigating scientific visualizations [32]. While residing in different domains, both techniques were designed to solve a similar problem: the time slider is unsuitable for answering questions targeting the visual space, such as "Find the moment in the video when the car starts moving" [10], mainly because it is difficult to focus on the changes of individual visual objects.

- *Brittany Kondo is with the University of Ontario Institute of Technology. E-mail: brittany.kondo@uoit.ca.*
- *Christopher Collins is with the University of Ontario Institute of Technology. E-mail: christopher.collins@uoit.ca.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014. Date of publication 11 Aug. 2014; date of current version 9 Nov. 2014. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.
Digital Object Identifier 10.1109/TVCG.2014.2346250

We introduce *DimpVis* (*DimP* [10] for information visualizations), an object-centric technique for interacting with visual items in information visualizations to explore the time dimension (see Figure 1). DimpVis enables intuitive investigation of spatial queries. For example, to answer "Was this bar ever at 500?" in a time-varying bar chart, one simply has to drag the bar to that height. If a moment in time exists when the bar is at the height, the visualization is moved to that time. The interaction is guided by visual paths which reveal how a selected data item changes through the time dimension of a visualization. DimpVis is intended for local, temporal exploration of an individual data item's changing values.

We designed DimpVis to navigate time-varying bar charts, scatter plots, heat maps, and pie charts. Using the bar chart and scatter plot, we performed a quantitative evaluation comparing DimpVis to the traditional time slider and small multiples, when used to complete different types of tasks. Our results showed that DimpVis for the scatter plot was subjectively preferred overall, quantitatively competitive with the time slider, and significantly faster than small multiples. DimpVis for the bar chart, however, was less preferred, but still quantitatively competitive with the small mutliples.

Our main contributions are: (1) DimpVis, an object-centric temporal navigation technique for querying and exploring time-varying information visualizations, that reinforces focus on target visual objects, to intuitively answer spatial questions about the objects; (2) Four prototypes exemplifying how DimpVis can be designed to manipulate different temporally evolving visual variables (position, size, colour), in a variety of common chart types (scatter plot, bar chart, pie chart, heat map); and (3) A quantitative evaluation comparing DimpVis for scatter plots and bar charts to traditional temporal navigation techniques.

In the following sections, we will discuss the related research, present the DimpVis technique for scatter plots and its generalization to other visualization types, describe our comparative evaluation study, and offer some ideas for future work.

2 Background

We review related work in four areas: techniques for visualizing temporal trends of visual objects in 2D visualizations, techniques for temporal navigation in information visualizations, direct manipulation techniques, and a focus on object-centric interaction.

2.1 Trajectory Visualization

One common way to reveal temporal trends in varying data is to use a trace visualization, which explicitly overlays a data object's changing values at various time points, onto the same visualization. This technique has been used to illustrate trends of points in time-varying scatterplots. The popular Gapminder Trendalyzer system uses animation and traces which dynamically draws a point's position on the graph, showing its temporal evolution [15]. Trajectories of points in scatterplots have also been used in visual analytic domains, such as time-varying financial (e.g., [28]), or medical (patient) datasets (e.g., [23]). Using a stacked-bands approach atop a map, Tominski et al. used trajectory visualization to analyze spatio-temporal data, including attributes about data points in the trajectory [30].

In this work, we build on the idea of trajectories, by designing visual hint paths, or indicators of all visual states of an individual data object across time. However, instead of simply displaying the trajectory visualization, we enable temporal navigation along the trajectory, invoking a change in overall the visualization view.

2.2 Temporal Navigation in Information Visualizations

Existing techniques for exploring time-varying information visualizations include of variations of methods such as filters, animation, and series of static images.

Temporal filters, often employed as separate widgets (e.g., [8]), can isolate or aggregate views within ranges of time. However, since all views are not visible at the same time, it is difficult to observe how the visualization changes over time. The small multiples technique displays images of the visualization at each time slice, ordered by time in a matrix layout [31]. While this technique separates all time steps for easy viewing, reading values, and comparison, its effectiveness degrades as the time line and dataset sizes increase.

Animation techniques present each snapshot of the visualization one after the other. Smooth transitions can be used to ease or highlight the changes between time slices (Kriglstein et al. have contributed a survey [20]). However, tracking individual data objects can become cumbersome if too many objects are changing or when the visualization is highly cluttered, causing distraction (e.g., in animated scatterplots [24] and dynamic graphs [14]). Existing slider techniques facilitate global exploration, and the navigation control is decoupled from the changing visual elements, requiring shifting attention between the widget and the visualization [4].

Lenses can be used to explore constrained regions of visual elements by showing visual representation alterations and exploring hypothetical visual states. Chronicle uses a temporal lens that records the creation of a graphical document and supports direct navigation through time[17]. Zhao et al. introduced Chronolenses, which facilitate exploratory and analytical tasks with time series data, visually filtering regions of interest with a lens and coupling analytical operations with direct manipulation techniques [33]. While powerful, lenses present subtle barriers to interaction by their constrained spatial extent, and they are generally used to isolate and temporally explore one area of a visualization while maintaining the global context. In contrast, in our work, direct temporal navigation is provided on all visual elements, and the changes are globally applied.

2.3 Direct Manipulation Interfaces

In direct manipulation interfaces, the visual objects of interest are represented consistently, physical actions are simple and support continuous flow of interaction, and immediate visual feedback is provided in response to physical actions [29]. For example, dragging a slider to navigate a timeline is a form of direct manipulation if the visualization updates in real time. Interaction techniques designed around these principles follow a user-centered model, providing capabilities to express intentions and manipulate objects to perform actions. The techniques can be analogous to real world manipulation, exhibiting implicit familiarity [29].

Beaudouin-Lafon defines interaction instruments as mediators between a user and an object of interest [4]. A good instrument is one with low indirectness, which integrates many actions with the input device, and provides highly similar mapping from physical actions to the object's response. For example, in interactive information visualizations, navigation and filtering instruments are often provided, such as zoomable viewports and sliders. To reduce the *temporal offset* between the instrument's action and the object's response, instant visual feedback is preferred.

Beyond the real-time feedback and feeling of engagement provided by direct manipulation interaction instruments (control widgets), direct interaction can be provided for the data items themselves. Existing interaction design approaches highlight the necessity for minimizing the distance between the interaction source and the target object (e.g., [4, 21]). Furthermore, several visualization-related interaction models have been derived from direct manipulation principles (e.g., [4, 11]). In the visual analytics domain, Endert et al. encourage direct manipulation of the visual representations of model outputs to adjust underlying model parameters, differing from the traditional method of using control panels [12].

2.4 Object-Centric Direct Manipulation

The principles of direct manipulation and recommendations for closer interaction between the visual object of interest and the user encourage object-centric interaction techniques. Our work is inspired by DimP, an interface for non-linear video browsing initiated by "relative flow dragging," an interaction technique for dragging objects in a video scene along their motion trajectories [10]. The DRAGON interface uses a similar approach for in-scene video scrubbing [18]. For temporal navigation of 3D scientific visualizations, Wolter et al. created a system where visualization objects can be dragged along their motion trajectories, invoking the corresponding movement in time [32].

The Design-by-Dragging interface is composed of "as-direct-as-possible" techniques (e.g., dragging along a visualized model) to explore effects of changing simulation input and outputs, and generate design alternatives [7]. DirectPaint merges the space and time controls for video animation authoring by using the visual element's motion trajectory as a basis for direct space-time manipulation [27].

Direct manipulation in the value domain for information visualizations was introduced by Perin et al. [22] to query time-varying data tables. The Drag-Cell technique is used to scan through the values of a data table cell across a time line. Upon releasing the manipulation, the entire table is updated. The Vis-Rank technique reveals a transient line chart which can be used to explore the time dimension. While related to our DimpVis technique, an important distinction is that several of our examples leverage "embodied interaction" [9] and a high degree of interaction compatibility [4]. Rather than adjusting abstract numbers, in DimpVis the finger or pointer remains connected to the data item as it is manipulated through the spatiotemporal value domain.

Direct manipulation can be used for data editing as well as exploration. Baudel presented a framework for directly manipulating data attributes, exemplified in the scatterplot view where point values are adjusted by dragging them [2]. Similarly, techniques in the form of dragging visual objects have been deployed in visual analytic systems as a means for altering the underlying model parameters (e.g., [6, 13]). This reduces the cognitive demand of learning complex models and their parameters, offering an intuitive method for model-steering [12].

Lastly, direct interaction with visualization objects for performing operations such as filtering and scaling has been employed using multi-touch gestures, combined with physics-based affordances [3, 26]. Related to the design of TouchWave, by eliciting *integrated interaction*, DimpVis provides a "hands on" data experience.

To our knowledge, direct object interaction has not been used as a temporal navigation technique in information visualizations.

3 Design of DimpVis

We extend previous work [10, 18, 32] by applying the direct temporal navigation technique to 2D time-varying information visualizations. We also demonstrate how DimpVis can be used to invoke temporal navigation by directly manipulating different visual variables beyond position, such as size and colour.

In creating DimpVis, we established the following design goals:

D1 Object-centric Navigation Temporal navigation occurs along the data trajectory.

D2 Navigation Flexibility Provide both controlled temporal navigation, as well as accelerated navigation shortcuts.

D3 Directness Direct connection to a data object of interest is maintained during navigation; the navigation control is embedded in the object.

D4 Interaction Consistency Navigation requires only a single finger or mouse pointer, without complex gestures or modes.

D5 Minimal Visual Change DimpVis can be added to existing visualizations without changing the underlying visual representation. Visual additions should be minimally distracting and removed when not in use.

DimpVis consists of two main components: *hint paths* [10], visualizing how an object changes over time, and *object-centric temporal navigation*, involving manipulating an object along its hint path. In the remainder of this section we discuss these components in detail. For clarity, the design components are first discussed in terms of designing DimpVis for scatter plots. Later, in Section 4, we describe how DimpVis was generalized and implemented for three other common visualization types.

3.1 Hint Path Design

A trajectory is an aggregated representation of all changes of a data object. In animated visualizations, trajectories are useful for trend analysis and pattern detection of time-varying data [20]. Visual feedback, or *hint paths* [10] can help guide interaction. Therefore we draw the hint path for the active data object (point in a scatter plot) to guide the interaction and provide contextual awareness during navigation.

To form a point's hint path, position is mapped to time. Following our design guidelines, the hint path should present the temporal evolution of a point in a clear, easily interpretable way, while also guiding fast and flexible temporal navigation (D1, D2). When not in use, hint paths are hidden (D5). We explored two design alternatives for the hint path: *time line* and *flashlight*.

Time line hint paths (Figure 2(l)): The positions of a point are linearly joined to form a path, ordered by time. Viewing this hint path reveals the patterns of change for a point over time. This design favours temporal trend legibility and navigation along the data trajectory (D1). It enables a user to trace the movement of a point through time, engendering a feeling for the data sequence through direct manipulation (D3). Navigation is linear in time, analogous to moving along a traditional timeline. Interaction flexibility (D2) is sacrificed in that navigation is constrained to the temporal order of the path, reducing the speed of long-distance temporal navigation.

Flashlight hint paths (Figure 2(r)): Similar to preview bubbles [7], the closest positions of the point are dynamically revealed as the navigation progresses. As the point is dragged, the positions nearest to the dragging direction, where the point exists at any time, are shown, regardless of temporal order. This design favours speed and flexibility of temporal interaction (D2). It enables fast navigation to moments in time where the point has a certain position (value) (D2, D3). However, since positions of the point are not connected in temporal order, the temporal trend is not apparent. While the flashlight supports exploration of data point positions in any temporal order, it violates D1.

Each hint path design offers advantages targeted at different analyst intentions. The time line is designed for understanding the temporal trend of a point, while the flashlight supports going directly to a time when a point exists at a certain position. We selected the time line design for further study, because it clearly illustrates a point's temporal trend, making its layout similar to a time slider. Navigation using the flashlight is more of a direct spatial query, as opposed to simulating a time slider embedded in the point. To overcome the main limitation of the time line design and support D2, a "fast-forwarding" feature was added to the path (Section 3.3), for quickly jumping to distant times.

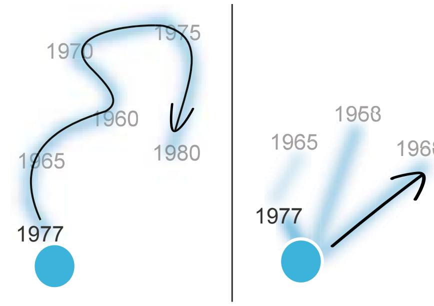

Fig. 2. The *time line* hint path follows temporal sequences (left), and the *flashlight* hint path connects to spatially adjacent time points (right).

Labels are added along the path to mark the point's position at each time interval [18], to show temporal location. Following D5, the hint paths should be subtle, therefore we blur them and use faint colours. Hint paths are only revealed for selected objects, and removed when interaction ends.

3.2 Object-Centric Temporal Navigation with Dragging

Object-centric navigation allows the user to explore how a point changes over time, while remaining focused on it. To navigate time, a point is dragged along its hint path, and the rate of dragging controls the speed of temporal navigation. Dragging has a high degree of compatibility, since the target object closely follows the action, and it lowers separation from the object of interest [4]. The position of the finger is projected onto the path according to the minimum-distance point. The temporal direction is indicated by the time labels along the hint path. While a point is dragged, the global time of the visualization is updated accordingly, and all other points are updated to their new positions.

3.2.1 Touch Input

In time-varying visualizations, depending on the rate and types of changes occurring in a data item's visual attributes, complex, curved trajectories may form. The mouse is a precise input device for target selection (pointing) tasks. However, it is unsuitable for precisely controlling and following a path of movement, such as in drawing tasks [16]. Additionally, a mouse presents a secondary barrier which separates the user from the data, decreasing the level of directness and transparency of the interface [21]. On the other hand, gestural interfaces reduce this separation, resulting in easier and more natural manipulation of data objects. Through directly touching and moving a data point along a trajectory, somatic feedback about the data values is received. Interaction techniques which engage people in a physical experience of connection with and direct manipulation of data facilitate the creation and communication of meaning through doing [9]. Therefore, the need for precise navigation and directness motivates us to prefer touch input for selecting and dragging a point along its hint path. However, mouse input is also supported.

3.2.2 Temporal Ambiguity and Interaction Detours

Temporal ambiguity has been recognized as a challenge for direct manipulation video browsing techniques [19]. Object-centric navigation along a hint path becomes ambiguous when the point's position does not change across two or more consecutive time points. A user may still want to navigate through the ambiguous time points, rather than skip them, in order to see how other non-stationary points in the visualization change during that time period. To resolve this issue, we use *interaction detours* integrated into the hint path, at areas of temporal

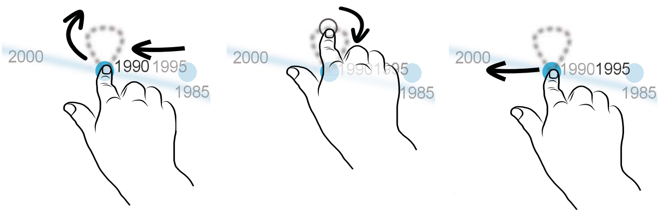

Fig. 3. Temporal ambiguity occurs in a scatter plot when a point does not move between time steps. Temporal navigation capabilities are provided in ambiguous using loops in which one transit around the loop corresponds to one time step.

ambiguity. The detours are designed to maintain the flow of dragging (D1) without a loss of directness (D3) or a need for disruptive gestures or multi-touch interaction (D4).

When a point does not change position, we insert loops into the hint path [19]. When dragging around a loop, the point does not move, but an outline of it is dragged around the loop, to maintain connection between the finger and its temporal position on the hint path. To enter a loop, the user continues dragging in the current 2D direction of motion, moving in a continuous temporal direction (Figure 3). The dragging direction can be reversed inside the loop, to reverse the temporal direction. One full rotation around the loop moves forward or backward, by one time point. Labels are placed near the stationary point to show the time points covered by the loop. The current time point's label is highlighted.

In areas of the hint path where the differences in point positions are very small, temporal navigation is challenging. In these regions, we also insert interaction detours. Therefore, in the scatter plot, loops are inserted where sequential points are too close together.

Alternative designs include a "sticky motion" effect [19], which could be used to skip through ambiguous regions. However this does not support detailed temporal navigation. A second finger could also be used to activate and drag along a smaller, separate time slider. However this disconnects the finger from the point, and potentially requires more mental effort, violating D3 and D4.

3.2.3 Interaction Ambiguity

We define interaction ambiguity as points when the interaction (dragging) cannot be resolved to a unique temporal direction. Interaction ambiguity mainly occurs when cusps are formed along a hint path [10], where dragging in a certain direction can indicate navigate in both directions in time. For example, if a scatter plot point's hint path doubles back on itself, at the point of reversal dragging along the hint path is temporally ambiguous.

Interaction ambiguity is resolved by maintaining *temporal continuity* at cusps. That is, we continue the navigation in the same *temporal direction* as it was moving prior to reaching the cusp. Consequently, temporal direction cannot be reversed at a cusp. When interaction starts at a cusp, there is no information for temporal continuity. In this case, forward time navigation is assumed. The direction can be reversed by changing dragging direction in a non-ambiguous area.

At some cusps, such as a sharp peak in the hint path of a point, our early testing showed it was difficult to bring the dragged object exactly to the cusp before reversing dragging direction to transit the point around the cusp. Frequently, users reversed dragging direction slightly before the point reached the cusp, leading to an unwanted reversal in the time direction. To ameliorate this problem, *tolerance regions* are applied to the cusp in which temporal continuity is enforced when the dragging direction changes near the peak. In this way, the cusp is 'rounded off' in interaction space and the user need not actually reach the peak in order to transit across it. The size of this region depends on sharpness of the peak; the tolerance region increases as the angle decreases.

3.3 Additional Features

In order to support the design goals, several additional design elements are included in all implementations of DimpVis:

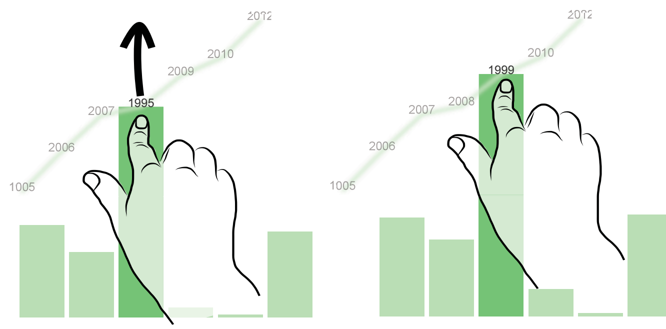

Fig. 4. Navigation in time for bar charts is achieved by dragging a selected bar vertically along its hint path. The hint path slides horizontally to stay connected with the bar and finger.

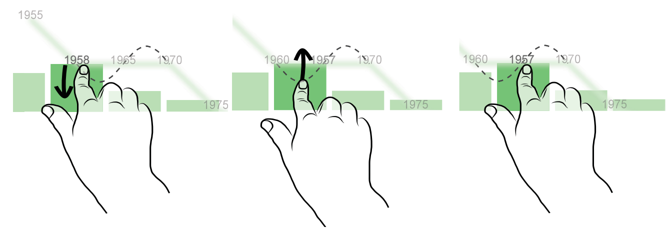

Fig. 5. Sine waves are introduced to provide an interaction technique to navigate through time periods where the bar height is not changing. The period of the wave is set so that the finger returns to the bar at each time step.

Time Slider: For high-level temporal navigation, DimpVis is paired with a traditional time slider widget. To move forward or backward in time, a small triangle tick is dragged horizontally.

Flexible Dragging: The finger can deviate away from the path during dragging, as though an elastic were connected to the nearest point. This is beneficial when using touch screens, where the hands may occlude the visualization.

Snapping to Time Points: As data values for a point exist only at labelled positions along the hint path, after a point is released from dragging, it is automatically re-positioned to the closest time position on the path.

Fast-forwarding: The hint path can facilitate both ordered navigation (dragging) and jumping across time. Fast-forwarding through time is invoked by tapping any time label on the path.

4 Generalizing DimpVis

In addition to the scatter plot, which applies DimpVis to navigate 2D position, we have generalized DimpVis to three other existing data visualizations, exemplifying direct manipulation of two additional visual variables: size (bar chart, pie chart) and colour (heat map). Each web-based prototype was implemented using the D3 toolkit [5]. For each prototype, we discuss the hint path design, dragging method used for object-centric navigation, interaction detour design (for temporal ambiguities) and techniques to resolve interaction ambiguities.

4.1 Bar Chart

Bar charts encode scalar data values in the height of bars, one for each data item or category.

Hint Path: All heights of a bar over time are connected by linearly interpolated lines to show variations in height, forming a line chart of heights over time from left to right (Figure 4). As a bar is dragged vertically towards an adjacent height in time, the hint path translates horizontally. When the next time point is reached, its label is centered on the dragged bar.

Dragging Bars: To navigate time, a bar is dragged vertically, according to the hint path. Horizontal dragging could be used, since the hint path is presented as a horizontal time line. However, this would violate our directness guideline (D3) as the finger would leave the bar.

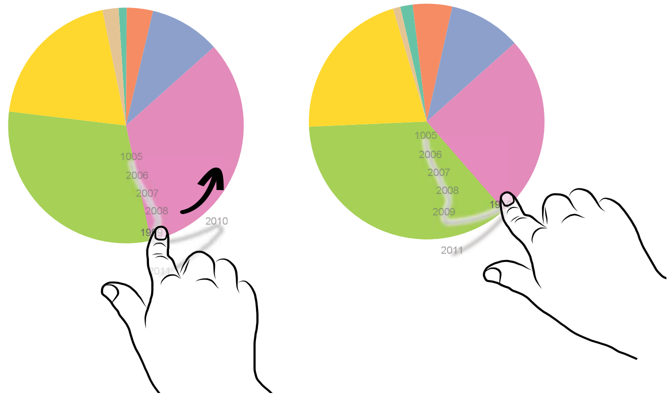

Fig. 6. Pie charts are adjusted in time by dragging an edge of a segment in angular directions along its hint path. Here, the purple segment is updated. The hint path slides in and out along the radius to remain connected with the finger position and edge of the segment.

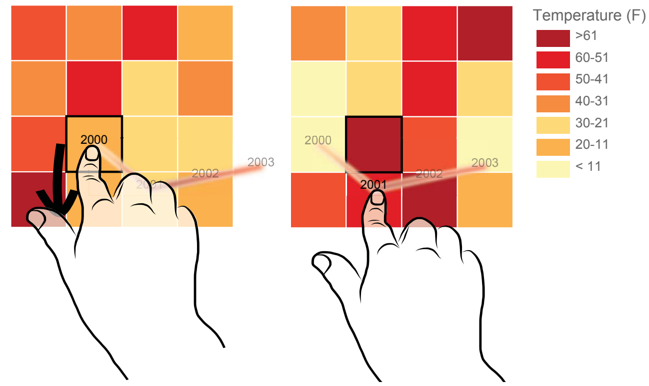

Fig. 7. Heat maps are adjusted in time by dragging vertically in the space of the colour scale. The hint path slides horizontally to stay connected with the cell and finger.

Vertical dragging corresponds directly to the changing value. So, the horizontal hint path translation is synchronized to the vertical dragging motion so that the finger and hint path always intersect at the top of the bar and the current time point.

Temporal Ambiguity: To navigate time spans when the bar stays at the same height, sine waves are drawn on top of the hint path as dotted lines. The period of the wave is set so that the finger returns to the bar at each time step, maintaining directness (D3). For example, when encountering an upwards peak, the user must first drag up, then down, to move to the next time (Figure 5). When the apex of the peak is reached, the user is halfway between time points. Due to directional ambiguity at peaks, temporal continuity is enforced. Therefore, navigating along a sine wave maintains interaction consistency (D4), by using only vertical dragging motion.

Interaction Ambiguity: Temporal continuity and tolerance levels are used when cusps are formed on the hint path. Sine waves are inserted to ease navigation across very close heights along the hint path. Additionally, when the bar has a zero value, a short, faded, grey bar is used as a placeholder to initialize interaction.

4.2 Pie Chart

Pie charts are used to display parts of a whole, such as percentage information. Angluar sizes of segments encode scalar values.

Hint Path: All angles of a pie chart segment over time are drawn and connected by angular paths (Figure 6). The angles of the hint path are placed outward on different radii: radius encodes time. When the segment is dragged, the path is animated in the radial direction. Using the chart's center as a reference, the path shrinks inwards when moving forward in time and expands outwards when moving backward.

Dragging Segments: One side of the segment remains stationary while the other side can be dragged to resize the segment's angle. While a segment is dragged, all other segments are resized according to their values at the updated time point. The radial hint path translation is synchronized to the angular dragging motion so that the finger and hint path always intersect at the edge of the segment and the current time point.

Temporal Ambiguity: Sine waves in the hint path are used as detours, using angular motion to navigate through them. One time step corresponds to half a period of the wave, so that the finger always returns to the dragged segment at each time point.

Interaction Ambiguity: Ambiguous interaction occurs whenever a cusp is formed on the hint path, indicating a change in angular dragging direction. Therefore, temporal continuity is enforced at cusps.

4.3 DimpVis for Non-Spatial Visual Variables

An important distinction between DimpVis and the DimP video browsing technique is that DimpVis provides a temporally ordered visual scan of all values of a data item. When the changing visual variable has spatial motion, dragging is conveniently mapped to spatial location, creating a direct correspondence between the visual feedback and user interaction. However when there is no motion, such as changing colour over time, the correspondence must be reinforced by the hint path, to maintain connection between the finger and the data item. This interaction design is similar to the DRAG-CELL technique, where values over time are browsed by dragging in the value domain [22]. The design challenge here is finding an appropriate mapping between the interaction (dragging) and the changing variable.

4.4 Heat Map

In general, a heat map encodes information using colour. In this section, we refer to a correlation matrix plot, where information corresponding to the strength of connections corresponds to row/column intersections is encoded using colour. A time-varying correlation matrix could be used to study data such as the changing strength of connections between friends in a social network.

Hint Path: Using the same design as the bar chart's hint path, the data values are plotted along a horizontal time line, where the y-position is relative to the corresponding vertical position on the colour scale (Figure 7). The hint path is also coloured to show the colour at each time point, and a gradient is used along the interpolated segments to show the transition.

Dragging Coloured Cells: Dragging is initiated by touching a cell. As colour doesn't have an inherent spatialization, we use the arrangement on the colour scale to provide one, thus dragging occurs vertically for a vertical colour scale. As dragging occurs, the colour of all cells is linearly interpolated to represent the current point in time. The hint path translates horizontally to maintain the direct connection of the finger to the hint path while dragging in the data-space direction. Note that direct connection to the cell may be lost, violating D3, as the cell itself does not move with the finger.

Temporal and Interaction Ambiguity: Ambiguities in the heat map are handled in the same way as for the bar chart.

5 Evaluation

We performed a comparative evaluation between three interaction techniques: DimpVis, traditional time slider, and small multiples, measuring their performance (time and error rate) when used to complete tasks involving reading values and observing trends. Additionally, we created a smaller set of extra tasks (not included in our analysis of performance measures) using the interaction detours (loops and waves). To keep experimental sessions a reasonable length of time, we decided to evaluate two representative visualization types (bar chart and scatter plot) from our four prototypes.

5.1 Task and Dataset Design

To evaluate the performance of each interaction technique, we created a set of short analytical tasks. Our tasks targeted the visual space, as opposed to the temporal space, coinciding with related experiments [10, 32]. We characterize visual space tasks as finding when a certain data object has a specified visualized property, concentrating on observing the data object's value (e.g., "When is bar A at height 3?"). Conversely, temporal space tasks involve finding a specified data object's value given a moment in time, concentrating on navigating the time line (e.g., "At 1995, what is the height of bar A?").

We designed tasks which required a participant to observe and quantify a data object's individual value or its trend of values, changing over time. Additionally, we included tasks for comparing changing values of multiple data objects. While DimpVis is an object-centric technique intended to focus on changes of single data objects, comparison tasks were added to test the effect of divided attention between navigation and other changing objects, on technique performance. Our tasks were mainly derived from a taxonomy of low level analytical tasks [1]. We generated four types of tasks:

Retrieve Value (RV): Read the changing value of a data object. *Example: "When is point A at age=50 and height=5?"*

Comparison (CO): Compare the changing values of two data objects. *Example: "When is point A's age and height greater than point B's?"*

Characterize Distribution (CD): Identify characteristics of the overall trend of a data object's changing value. *Example: "After the age and height of point A have been increasing, find the first year when they are both decreasing."*

Outlier Detection (OD): Find when a data object's changing value deviates from the overall trend of all other data objects. *Example: "Find the first year when point A is moving in the opposite direction of the other points."*

RV and CO are value-reading task types, while CD and OD are trend analysis task types. We generated three objective and three practice versions of each task type, per visualization and interaction technique. Each task version's difficulty was assessed during pilot testing to ensure participants were able to comprehend and complete it. In pilot testing the OD task was frustratingly difficult with the bar chart for all conditions and was removed for that visualization type.

To ensure the data exhibited realistic behaviour, we started with real datasets and made adjustments to ensure each task had a unique correct answer. The specific data items and question details were varied across all interaction techniques, and the practice and objective tasks. In each task version, the target data object was always different. Therefore, a participant never encountered a task targeting the same data object (with the same values). The time pointer was set to the starting year at the beginning of each task. For all tasks, the correct solution (year) was placed somewhere in between the middle to the last year. This ensured temporal navigation was required to complete the task. The hint paths of data objects involved in tasks were kept simple, avoiding loops and zigzags, such as a point whose path crosses over on itself (except for CD and OD tasks, where the data object changes trends).

For the scatter plot, datasets always contained 20 points (axes artificially labelled as age and height), over 10 years, and datasets for the bar chart always contained 13 bars, over 10 years. Data labels were also artificially created as 2000–2009 (for years) and randomly assigned letters of the alphabet (for the data objects).

In summary, we used a 3 *technique* (DimpVis, slider and small multiples) x 2 *visualization* (scatter plot, bar chart) within subjects design. The order of *technique* and *visualization* were counterbalanced with two participants for each ordering, resulting in a total of 12 participants. For bar charts there were 3 *task types* (RV, CO, CD), and for scatter plots there were 4 *task types* (RV, CO, CD and OD). *Task type* ordering was randomized across participants. In total, there were *technique* (3) x *task type* (4) x *task versions* (3) x 12 participants= 432 trials for the scatter plot and *technique* (3) x *task type* (3) x *task versions* (3) x 12 participants= 324 trials for the bar chart. In addition, for the DimpVis technique for each visualization type, participants completed three task versions for the RV and CD tasks (total 6 trials per visualization) on datasets with temporal ambiguities using interaction detours. These tasks trials were only used to inspire subjective feedback and not included in quanitative analysis.

5.2 Procedure

The following procedure was carried out twice for each participant, once for each visualization type. Half the participants started with scatter plot, and half with bar chart.

Prior to using each interaction technique, participants were given an explanation of how to use it and a demonstration of how it works. The participant was instructed to complete each task as quickly and as accurately as possible. Participants were able to skip tasks, but they could not re-do them. With each interaction technique, the participant first engaged in a set of practice tasks, followed by the objective tasks (used in our analysis). We ensured that participants never encountered an objective task type they have not previously practiced by using the same amount and type of tasks in both the practice and objective task sets. The participant was not informed which tasks were for practicing.

At the start of each task, the participant was given as much time as needed to read the task description, which remained visible during task completion. When the participant pressed a "ready" button, the visualization was displayed. Data objects involved in tasks were highlighted in orange during the initial time step, allowing participants to pre-attentively locate them [25]. After passing the second time step, all data objects faded to the same colour, permanently. Solutions to the tasks were submitted as views of the visualization at a certain year, using the assigned interaction technique to navigate time.

We measured task completion time and error rate. Time was recorded as the difference between displaying the visualization and submitting the task solution. Error rate was measured by the amount of incorrect trials out of the total number of task trials.

For example, if using DimpVis for the scatter plot, a participant would drag the point to a position which they thought answered the task. Task completion time was measured from when the "ready" button was pressed to when the answer was submitted (pressing a "submit" button). All completion times, submitted answers and user interactions were logged by the system. Participants were also video recorded, from over-the-shoulder. On screen feedback about correct and incorrect answers was provided after task submission.

After completing all objective tasks, participants were invited to rate each technique subjectively, using a 5-point Likert scale (1-Strongly Disagree to 5-Strongly Agree). Although the interaction detours are an extension of DimpVis, during pilot testing we found that participants had specific comments about them. Therefore, the interaction detours were rated separately.

Participants were then invited to use a full-featured version of the DimpVis technique in the bar chart and scatter plot prototypes, to explore real datasets. We provided some open-ended questions focusing on temporal trends of the data to inspire exploration (e.g., "Is there any common trend across all programs of how enrollment varies over time?"). Participants were instructed to freely explore the data by dragging the points or bars while speaking aloud.

Following the exploration, participants completed a questionnaire about the hint path and the study concluded with an interview. The entire procedure was repeated for the second visualization type.

5.3 Interface Designs

We created three technique interfaces for each visualization type:

DimpVis: DimpVis with a restricted version of the time line hint path, revealing only the immediately adjacent time steps during dragging. A non-interactive time slider is included to show temporal location.

Time Slider: An interactive horizontal time slider is placed underneath the visualization, where years are shown at each tick mark.

Small Multiples: Equal-sized static images of the visualization at each year are displayed on the screen. The images are ordered by time, spanning from left to right, then top to bottom. Year labels are placed on top of the images. We ensured that data objects necessary to complete a task were clearly visible.

A constrained version of the DimpVis hint path was used for the experiment in order to focus on the feature of most interest — object-

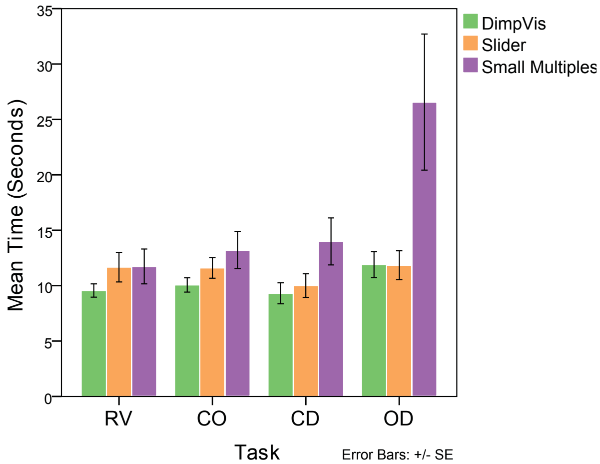

Fig. 8. Scatter plot task completion times.

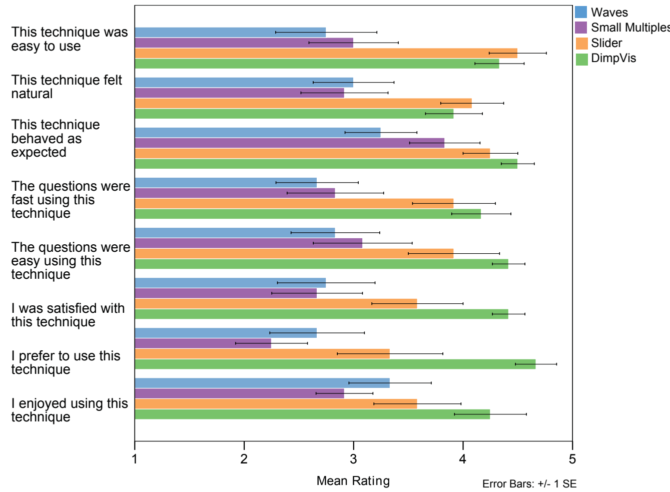

Fig. 9. Scatter plot subjective ratings.

centered dragging. This hint path indicated available dragging directions in the immediate area of interaction, but the whole hint path was not shown. This prevented simple reading of the complete hint path to answer task questions. No additional interaction capabilities (pan, zoom, filter, fast-forwarding) were provided. All colours were colour-blind friendly. The task question was displayed in a left-hand sidebar, and main part of the screen was used to show the visualization. Axes lines and data object labels were added to the visualizations.

5.4 Experimental Setup

All experiment sessions were conducted in the same, private laboratory, with lower lighting conditions. A standard workstation computer and a wall-mounted 46-inch Phillips TV with a PQ Labs multi-touch overlay were used to run and display the visualization prototypes in a Google Chrome browser window. Participants used the touch screen to complete the tasks while standing. On average, the study lasted two hours. Participants were allowed to take breaks between tasks as needed, and received a gift card as compensation.

5.5 Participants

We recruited 13 participants (11 male and 2 female), aged between 19 and 30 years, from our university and surrounding area. One female participant's data was excluded from analysis due to an observed lack of effort to correctly complete the tasks, resulting in frequent incorrect answers and many skipped tasks. All remaining twelve participants self-declared as at least beginners in reading both bar charts and scatter plots, reading them for visual analysis at least a few times a year. All participants used touch screens daily (mainly phones or tablets), and the DimpVis technique was new to them.

5.6 Hypotheses

The tasks fall into two main categories: reading values of data objects (RV, CO) and observing the trend of a data object's changing values (CD, OD - scatter plot only). We argue that while DimpVis may not present a faster or more accurate method for reading values from visualizations, it may be more efficient for characterizing the trends of data objects. We suspected that the small multiples technique would perform better for reading values from a visualization, as opposed to observing trends, because motion is not apparent. Conversely, the slider and DimpVis might perform better for characterizing trends. Although sliders are effective for understanding global changes, tracking changes of individual data objects can be difficult due to the simultaneous motion of other data objects. DimpVis, however, reinforces focus on target data objects. Therefore, we hypothesize that, for both the bar chart and the scatter plot:

H1: Overall, DimpVis will be the fastest and most accurate for completing tasks, followed by the slider and then small multiples.

H2: Overall, tasks involving reading values (RV, then CO) will be faster and more accurate than trend-based tasks (CD, then OD).

H3: For reading values tasks (RV, CO), small multiples will be the fastest and most accurate, followed by the slider and then DimpVis.

H4: For trend-based tasks (CD, OD - scatter plot only), DimpVis will be the fastest and most accurate, followed by the slider and then small multiples.

6 Results

Below, we present our quantitative (time and error rate) and qualitative (interview, subjective ratings and observations) results, as well as observations gathered during exploratory periods.

6.1 Scatter Plot: Quantitative Results

6.1.1 Task Completion Time

A two-way repeated measures ANOVA, with factors *technique* (3 levels) and *task type* (4 levels) was performed. The dependent variable, time (measured in seconds), was log-transformed to reduce skewing in the data caused by outliers. We adjusted significance values for all post-hoc pairwise comparisons using the Bonferroni correction.

The results are summarized in Figure 8. The main effect of *technique* was significant ($F_{2,22} = 11.942, p < 0.05$), with post hoc tests showing that DimpVis ($M = 10.2s$) and slider ($M = 11.3s$), were significantly faster than small multiples ($M = 16.4s$). However, the difference between DimpVis and slider was not significant. Therefore, H1 is only partially supported. There was also a significant main effect of *task type* ($F_{3,33} = 13.104, p < 0.05$), with post hoc tests showing that RV ($M = 11s$), CD ($M = 11.1s$), and CO ($M = 11.6s$) were significantly faster than the OD task ($M = 16.7s$). However, the differences between RV, CO and CD were not significant, only partially supporting H2. Lastly, the interaction effect between *technique* and *task type* was significant ($F_{6,66} = 3.844, p < 0.05$), due to the differences between DimpVis ($M = 11.9s$) and small multiples ($M = 26.6s$) in the OD task, as well as, slider ($M = 11.8s$) and small multiples in the OD task, as determined by post hoc tests. This result only partially supports H4, and H3 is rejected.

Small multiples required participants to scan through each image to locate the target point, resulting in slower completion times. Unsurprisingly, animation seemed to accelerate trend-based tasks, since times were faster for the CD and OD tasks using DimpVis and slider. The OD task was significantly slower using small multiples, and was tedious to answer, according to participants. This suggests that the small multiples technique becomes distinctly slower as the amount of moving points to observe increases.

6.1.2 Error Rate

Overall, error rates were low for each technique (DimpVis$= 3/144$, small multiples$= 4/144$, and slider$9/144$) and nearly uniformly dis-

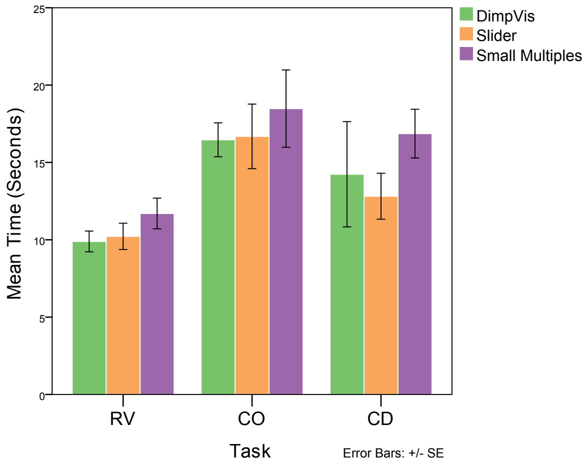

Fig. 10. Bar chart task completion times.

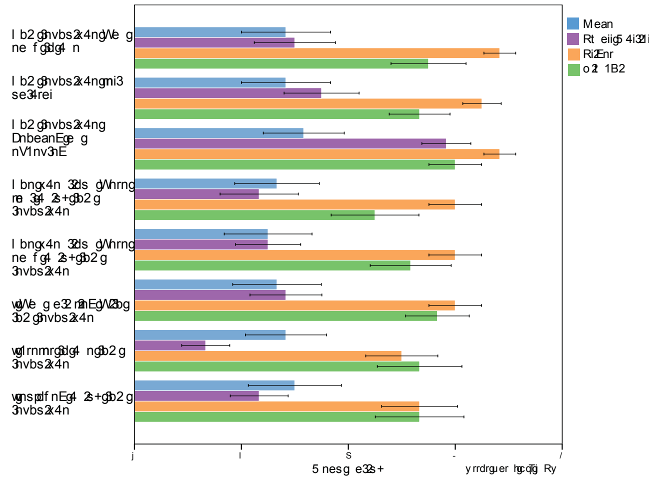

Fig. 11. Bar chart subjective ratings.

tributed, therefore, no significant differences were found. Very few errors were produced by each technique in the RV and CO tasks (2/108), however for trend-based tasks (CD and OD), error rates were higher (7/108). Trend tasks were intended to be more difficult, as they require locating a specific time while interpreting the motion of a point, whereas RV tasks involve moving points directly to a known position.

6.2 Scatter Plot: Subjective Feedback

The majority of participants agreed that DimpVis was easy to use (see Figure 9). However, opinions on the loops were divided: half the participants reported them as generally easy to use (some even found them fun), whereas the other half felt frustrated. Once participant suggested that it would be nice to be able to fast-forward through ambiguous regions, because they slow down navigation. Some participants clarified that different techniques were more useful for different types of tasks. Specifically, two participants found that DimpVis was easier for completing the CD and OD tasks, compared to the slider. Lastly, participants expressed their excitement for DimpVis, that they "would use it to look at data,", it "increased engagment with the data", and that it could be "useful for teaching [charts to] students."

6.3 Bar Chart: Quantitative Results

6.3.1 Task Completion Time

A two-way repeated measures ANOVA with factors *technique* (3 levels) and *task type* (3 levels) was performed. The dependent variable, time (measured in seconds), was log-transformed to reduce skewing in the data caused by outliers. We adjusted significance values for all post-hoc pairwise comparisons using the Bonferroni correction.

The results are summarized in Figure 10. On average, slider was the fastest ($M = 13.2s$), closely followed by DimpVis ($M = 13.5s$) and then small multiples ($M = 15.7s$). However no signficant differences were found for *technique*. There was a significant main effect of *task type* ($F_{2,22} = 62.313, p < 0.05$), with post hoc tests showing that the differences between the CO ($M = 17.2s$), CD ($M = 14.6s$) and RV ($M = 10.6s$), tasks were all significant. No significant interaction effect between *technique* and *task type* was found. Therefore, H4 is only partially supported and all other hypotheses are rejected.

The CO tasks were consistently the slowest across all interaction techniques. During these tasks, we noticed that some participants stepped away from the display to compare the bars (only when using slider and DimpVis), which may have lead to slower times.

6.3.2 Error Rate

Overall, error rates were low for each technique (Dimpvis$= 0$, slider$=$ 4/108 and small multiples$= 7/108$), and nearly uniformly distributed. Therefore, no significant differences were found. Error rates varied between each type of task (RV$= 1/108$, CO$= 4/108$ and CD$= 6/108$) and DimpVis produced no errors for all tasks.

6.4 Bar Chart: Subjective Feedback

The subjective feedback indicates participants generally preferred DimpVis over the small multiples (see Figure 11). Three participants mentioned that dragging the bars was confusing. One participant stated that it was "difficult to understand the simultaneous height and time change." The majority of the feedback pertained to the wave interaction detour. Only three participants commented that the waves were easy to use. The remaining seven participants expressed various concerns regarding the usability of the waves, such as feeling "lost in the wave," or how the wave "refused to respond during dragging." Overall, majority of participants stated that they preferred the slider, mainly because it was easier to use and learn quickly (e.g., horizontal dragging motion).

Despite being informed that dragging must follow the path, we observed at least four participants attempting to drag away from it, multiple times during the tasks. For instance, in the RV task, they would try to drag directly to the height, even though the path indicated a different dragging direction. Five participants also used a second finger to either drag other bars not involved in the task or mark a significant spot on the chart, such as another bar in CO task, or the target height of an RV task. Additionally, some interesting attempts to accelerate navigation were observed, such as: swiping up to reach far heights, horizontal dragging along waves or shorter bars, and even tracing the wave. These may suggest ways to refine the design.

6.5 Exploratory Period

For the scatterplot, a dataset representing total internet users for some of the world's major economies from `http://gapminder.org` was used. For the barchart, we used a dataset showing total enrollment in different programs at our university from `http://cudo.cou.on.ca`. These datasets were chosen as they may be interesting to our participant population.

Common strategies were observed for exploring the data. Some participants (3-bar chart and 4-scatter plot) would focus mostly on reading the trends of individual points or bars using the hint paths, performing less temporal navigation. However, other participants (3-bar chart and 5-scatter plot) preferred to drag only a few points/bars, while examining the motion of other data objects in the visualization. Lastly, the remaining participants almost evenly divided their attention between the hint path and dragging; by first examining a hint path and then dragging along it to explore time (6-bar chart and 3-scatter plot). Notably, one participant used a storytelling approach, where they narrated the trends of the points and bars, while dragging them. The hint paths and interaction technique seemed to supplement the story.

Subjective feedback on the hint path was uniformally positive across both visualization types. The hint path was rated as beneficial ($M_{scatterplot} = 4.8$, $M_{barchart} = 4.7$), helpful ($M_{scatterplot} = 4.6$, $M_{barchart} = 4.6$) and useful ($M_{scatterplot} = 4.8$, $M_{barchart} = 4.2$) during exploration. Generally, participants did not find the hint paths distracting ($M_{scatterplot} = 1.3$, $M_{barchart} = 1.4$) or confusing ($M_{scatterplot} = 1.4$, $M_{barchart} = 1.3$) when exploring the visualizations.

All participants agreed that DimpVis was a suitable interaction technique for touchscreens, mainly because it seemed to "enhance engagement with the data." Three participants suggested that a mouse may be more suitable for navigating the loops and waves, because they felt more precision was needed for navigating along them. The visual complexity of the detours may give the impression that more precision is required (e.g., following the sine waves closely), when in fact they require a similar dragging motion as the regular hint paths.

7 Discussion

The main goal of our evaluation was to determine the benefits of using DimpVis for answering questions targeting the visual space, and compare its performance to the time slider and small multiples technique.

DimpVis for the scatter plot was significantly faster than the small multiples and subjectively preferred by participants. DimpVis did not significantly out-perform the time slider, however there was no loss in time or accuracy. This null result suggests that DimpVis may be useful for supporting some object-centric tasks that are cumbersome using the slider or small multiples, without any significant loss in performance.

We noted several implications for the design of future DimpVis interfaces:

Hint Paths Aligned with Dragging Direction: We expected that DimpVis for the bar chart would to be easier to learn and use, because it requires only vertical dragging and a single changing visual variable over time, whereas the scatter plot has two changing data dimensions over time. All participants appeared to use DimpVis for the scatter plot consistently: dragging along the path using one finger. Whereas, for the bar chart, a diverse set of unsuccessful interaction actions were observed, including using a second finger and dragging in a direction opposite to the path. This indicates that participants may have expected DimpVis to have different capabilities, suggesting that dragging the bars was a less intuitive interaction than dragging the points. The mix of vertical dragging and horizontal hint path translation seems to cause some confusion (as opposed to the scatter plot's stationary hint path).

This may indicate that restricting dragging to the direction of the hint path is preferrable. This is a difficult design challenge, as it competes with our design goal of keeping the finger on the bar at all times (meaning only vertical motion is possible for bar charts). Given these constraints, for bar charts this would mean hint paths should clearly indicate vertical motion. One approach is to provide vertical arrows, scaled to the amount of change, which indicate at any instant in which temporal direction dragging will move time (however, this would not support fast-forwarding). Another idea is to provide a vertical hint overlay using the flashlight hint path design (Figure 2(r)), with year labels showing bar heights at each year, allowing for direct dragging to any year of interest. However, this does not restrict navigation to following the temporal sequence.

Multi-touch Ambiguity Resolution: The usability of the wave interaction detours was a concern raised by some participants. This may have been due to the lack of an anchor to explicitly indicate the position along a wave, which was provided for the loop. Also, better tuning of the tolerance region around the peak of the wave may help prevent unwanted direction reversals. However, a more complete solution may be to take a different approach to temporal ambiguities, such as relaxing our goal of interaction consistency (D4) in favour of introducing a second finger for scrolling through time steps where the value does not change.

Provide Time Line and Flashlight Interaction: Initially, we selected the time line hint path design because clearly illustrating temporal trends was considered an important requirement for guiding temporal navigation (D1). However, during our evaluation, some participants attempted to drag bars directly to a desired height, an action that may be better supported by our flashlight hint path design (Section 3.1). Different types of tasks may be better supported by different hint path designs. Additionally, both detailed (dragging) and accelerated (fast-forwarding) temporal navigation are important (D2), since one participant found the interaction detours slowed down navigation, and wanted to quickly skip through them. In the study we did not provide the fast-forwarding feature in the evaluation as we wanted to focus on the dragging interaction.

Scalability Issues: We have not investigated the scalability of DimpVis in detail, but have some thoughts about it given our experience with the prototypes. Larger datasets generally make trend detection and tracking items of interest more difficult in animated plots, and this may be true for the scalability of DimpVis too. Techniques for filtering or focusing on subsets of data items of interest could be used, such as lenses or highlights. DimpVis may encounter interaction usability challenges in dense regions of a visualization. For instance, if a scatter plot is too dense to perceive individual points, then directly interacting with them may not be feasible without filtering the dataset. Also, if the time scale of a dataset is large, the hint path will become long and potentially cluttered. A cluttered hint path may require aesthetic (e.g., thinner lines, aggregating time points) or functional (e.g., a scrolling hint path) enhancements. Additionally, DimpVis does not support comparison of multiple data items, and it may be difficult or distracting to focus on the rest of the visualization, while dragging the target item. In both cases, the time slider may be more suitable.

8 Conclusion and Future Work

Temporal navigation controls for exploring dynamic information visualizations are typically disjoint from the changing visualization objects. Therefore, the user must shift their focus between observing an object of interest (to see visualized temporal changes) and using a separate control, such as a time slider (to see temporal location). We introduced DimpVis, an object-centric temporal navigation technique that narrows the gap between the user and visual objects of interest, by directly manipulating objects along their hint paths. We implemented DimpVis for touch interaction with bars (bar chart), points (scatter plot), coloured cells (heat map) and angular segments (pie chart).

In our comparative evaluation, while DimpVis did not significantly out-perform the traditional time slider, DimpVis for the scatter plot was subjectively preferred by participants overall and was significantly faster than the small multiples technique. Participant feedback and results from the evaluation motivate us to further explore and evaluate different design components, such as alternative techniques for handling temporal ambiguity and different hint path designs. Finally, our study indicated the need for both time line and flashlight style hint paths. We would like to investigate new hint path designs which may support both styles of interaction, as well as how to transition between them smoothly.

In future work, we plan to apply DimpVis to other types of dynamic information visualizations, with different types of changing visual variables. For instance, items in a time-varying bubble chart (e.g., [15]) often have two changing visual variables: size and position. Using DimpVis, a user could drag a bubble's position and/or its radius to explore temporal change. In addition, we would like to study the usefulness of DimpVis for storytelling or presentations of dynamic data visualizations, as hinted at by our subjective feedback. For instance, when a presenter wishes to draw emphasis to a data item's trend they can show the hint path and then navigate to time points when the item has an interesting value, while maintaining focus on the item. Multi-touch gestures may also expand interaction capabilities, for example enabling temporal queries by moving multiple data objects.

Acknowledgments

We are grateful to the reviewers, study participants, the other members of the Visualization for Information Analysis lab, and to Fanny Chevalier, Anastasia Bezerianos, Petra Isenberg, and Pierre Dragicevic for their helpful insights and comments. This work is supported by NSERC SurfNet and The Italian Cultural Centre of Durham.

References

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 111–117. IEEE, 2005.

[2] T. Baudel. From information visualization to direct manipulation: Extending a generic visualization framework for the interactive editing of large datasets. In *Proc. of the ACM Symposium on User Interface Software and Technology*, pages 67–76. ACM, 2006.

[3] D. Baur, B. Lee, and S. Carpendale. Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the International Conference on Interactive Tabletops and Surfaces*, pages 255–264. ACM, 2012.

[4] M. Beaudouin-Lafon. Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 446–453. ACM, 2000.

[5] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[6] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proc. of IEEE Visual Analytics Science and Technology (VAST)*, pages 83–92. IEEE, 2012.

[7] D. Coffey, C.-L. Lin, A. G. Erdman, and D. F. Keefe. Design by dragging: An interface for creative forward and inverse design with simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2783–2791, 2013.

[8] M. Dork, S. Carpendale, C. Collins, and C. Williamson. VisGets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1205–1212, 2008.

[9] P. Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. MIT Press, 2001.

[10] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. Video browsing by direct manipulation. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, page 237, New York, New York, USA, Apr. 2008. ACM Press.

[11] N. Elmqvist, A. V. Moere, H.-C. Jetter, D. Cernea, H. Reiterer, and T. Jankun-Kelly. Fluid interaction for information visualization. *Information Visualization*, 10(4):327–340, 2011.

[12] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, 2013.

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

[14] M. Farrugia and A. Quigley. Effective temporal graph layout: A comparative study of animation versus static display methods. *Information Visualization*, 10(1):47–64, 2011.

[15] G. Foundation. Gapminder trendalyzer. `http://www.gapminder.org`.

[16] G. Goth. Brave NUI world. *Communications of the ACM*, 54(12):117–119, 2011.

[17] T. Grossman, J. Matejka, and G. Fitzmaurice. Chronicle: Capture, exploration, and playback of document workflow histories. In *Proc. of the ACM Symposium on User Interface Software and Technology*, pages 143–152. ACM, 2010.

[18] T. Karrer, M. Weiss, E. Lee, and J. Borchers. DRAGON: A direct manipulation interface for frame-accurate in-scene video navigation. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 247–250. ACM, 2008.

[19] T. Karrer, M. Wittenhagen, and J. Borchers. Draglocks: Handling temporal ambiguities in direct manipulation video navigation. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 623–626. ACM, 2012.

[20] S. Kriglstein, M. Pohl, and C. Stachl. Animation for time-oriented data: An overview of empirical research. In *Proc. of the 16th International Conference on Information Visualisation (IV)*, pages 30–35. IEEE, July 2012.

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

[22] C. Perin, R. Vuillemot, and J.-D. Fekete. À table!: Improving temporal navigation in soccer ranking tables. In *Proc. of SIGCHI Conference on Human Factors in Computing Systems*, Apr. 2014.

[23] A. Rind, W. Aigner, S. Miksch, S. Wiltner, M. Pohl, F. Drexler, B. Neubauer, and N. Suchy. Visually exploring multivariate trends in patient cohorts using animated scatter plots. In *Ergonomics and Health Aspects of Work with Computers*, pages 139–148. Springer, 2011.

[24] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[25] S. Rufiange and M. J. McGuffin. Diffani: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2556–2565, 2013.

[26] J. M. Rzeszotarski and A. Kittur. Kinetica: naturalistic multi-touch data visualization. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 897–906. ACM, 2014.

[27] S. Santosa, F. Chevalier, R. Balakrishnan, and K. Singh. Direct space-time trajectory control for visual media editing. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1149–1158. ACM, 2013.

[28] T. Schreck, T. Tekušová, J. Kohlhammer, and D. Fellner. Trajectory-based visual analysis of large financial time series data. *ACM SIGKDD Explorations Newsletter*, 9(2):30–37, 2007.

[29] B. Shneiderman. Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proc. of the 2nd International Conference on Intelligent User Interfaces*, pages 33–39. ACM, 1997.

[30] C. Tominski, H. Schumann, G. Adrienko, and N. Adrienko. Stacking-based visualization of trajectory attribute data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012.

[31] E. R. Tufte. Envisioning information. *Optometry & Vision Science*, 68(4), 1991.

[32] M. Wolter, B. Hentschel, I. Tedjo-Palczynski, and T. Kuhlen. A direct manipulation interface for time navigation in scientific visualizations. *2009 IEEE Symposium on 3D User Interfaces*, pages 11–18, 2009.

[33] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan. Exploratory analysis of time-series with ChronoLenses. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2422–2431, 2011.