DataHop: Spatial Data Exploration in Virtual Reality

Devamardeep Hayatpur
University of Toronto
Toronto, Canada
hayatpur@dgp.toronto.edu

Haijun Xia
University of California, San Diego
San Diego, California
haijunxia@ucsd.edu

Daniel Wigdor
University of Toronto
Toronto, Canada
daniel@dgp.toronto.edu

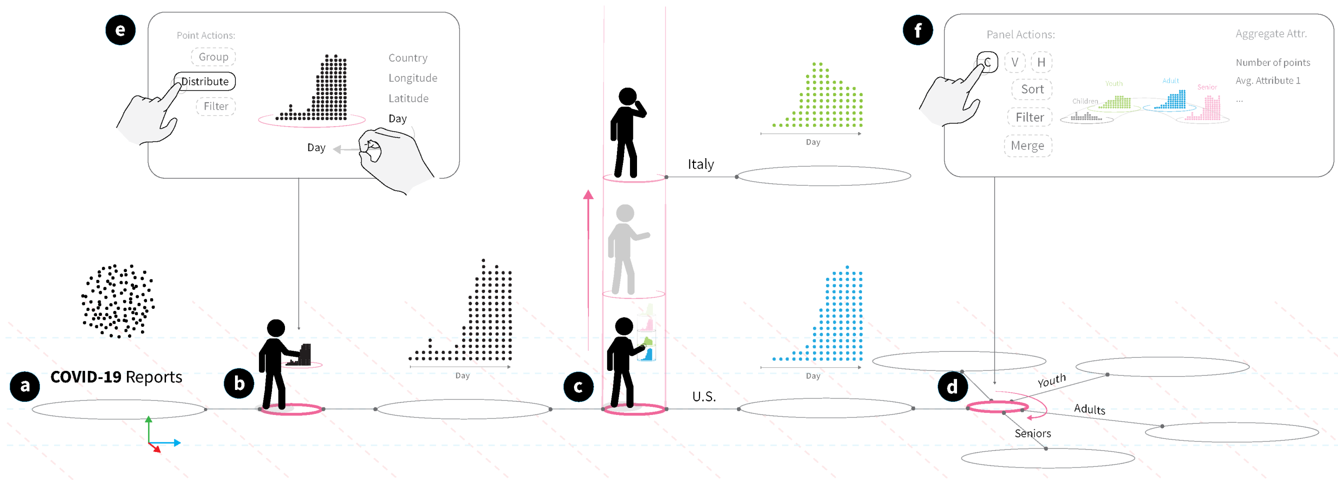

Figure 1. A user starts with the root *data panel* (a) and can modify it by passing its points through a *Filter* (b). *Filters* can be used to change the visual encoding of the input data panel (e) as well as control the placement of the output data panels (d, f). An elevator metaphor is used to navigate to different data panels (c).

ABSTRACT

Virtual reality has recently been adopted for use within the domain of visual analytics because it can provide users with an endless workspace within which they can be actively engaged and use their spatial reasoning skills for data analysis. However, virtual worlds need to utilize layouts and organizational schemes that are meaningful to the user and beneficial for data analysis. This paper presents *DataHop,* a novel visualization system that enables users to lay out their data analysis steps in a virtual environment. With a *Filter*, a user can specify the modification they wish to perform on one or more input *data panels* (i.e., containers of points), along with where output data panels should be placed in the virtual environment. Using this simple tool, highly intricate and useful visualizations may be generated and traversed by harnessing a user's spatial abilities. An exploratory study conducted with six virtual reality users evaluated the usability, affordances, and performance of DataHop for data analysis tasks, and found that spatially mapping one's workflow can be beneficial when exploring multidimensional datasets.

UIST 2020, October 20–23, 2020, Minneapolis, MN, USA.

ACM ISBN 978-1-4503-7514-6/20/10...$15.00.
DOI: https://doi.org/10.1145/3379337.3415878

Author Keywords

Immersive data visualization; spatial skills;

CSS Concepts

• **Human-centered computing~Visualization**; *Interaction Design*; User studies;

INTRODUCTION

Virtual reality (VR) enables users to be situated within vast virtual words that are often free of physical spatial constraints. The use of virtual environments for data visualization, i.e., *immersive analytics* (IA), has been an emerging data analysis technique because it provides opportunities for greater engagement, improved spatial reasoning, and an increased working area [25].

Current work in IA has harnessed users' spatial manipulation skills to edit visualizations [7], immerse one in their data [14], leverage 3D space to analyze high dimensional datasets, and create novel visualizations [6, 23]. Within these environments, however, the user is not encouraged to actively make use of the large amount of space offered by the virtual environment.

The spatial placement of data in virtual environments has previously been used to understand complex systems such as software source code, where each component of the system is mapped to a different spatial area in the virtual environment [16, 17, 24, 33, 36]. To encourage spatial recall, these layouts often use metaphors based on real world structures, such as buildings and roads [16, 17]. Users can thus leverage their spatial abilities to familiarize themselves with the underlying system. These structures, however, are often static and follow a pre-determined layout, which undermines a user's ability to explore areas that are most interesting to them, as well as removes any agency the user has over how the space is structured.

The use of virtual environments to dynamically create and navigate structures could aid users in understanding multidimensional datasets. This paper thus describes DataHop, a novel visualization system that spatially visualizes a user's steps during data analysis tasks. With DataHop, a user begins with a set of unit data points (Figure 1a) that they can transform by passing these points through a *Filter* (Figure 1b). The *Filter* provides a set of actions that can be used to edit the input data panel(s) (Figure 1e), and another to specify the placement of the output data panel(s) in the virtual world (Figure 1f). To assist a user in using their spatial skills as they navigate throughout the space, an elevator metaphor enables users to traverse from a *Filter* to any of its outputs (Figure 1c).

Output data panels can be placed in three different spatial organizations: (1) horizontally (i.e., left to right), (2) circularly (i.e., in a semi-circle around the user), or (3) vertically (i.e., on different floors). If one wants to analyze how various countries are containing the COVID-19 pandemic, for example, they could organize the countries vertically, such that each country would have its own dedicated floor within the environment. Then, each floor could be further populated by dividing the data by age group to, for instance, analyze the number of infections and recoveries across different age groups (Figure 1).

This paper describes the design and implementation of DataHop along with the findings from an exploratory user study. It thus contributes a novel visualization system for immersive data analysis and a qualitative exploratory study that demonstrates the usability and the utility of DataHop for understanding multidimensional datasets.

RELATED WORK

Of most relevance to the development of DataHop was research on immersive analytics, VR navigation techniques, and tools to visualize provenance.

Immersive Data Exploration and Analytics

The use of immersive technologies for data visualization has long been an area of study. A recent survey by Fonnet et al. provides an extensive overview of this research [10], however, the relevant visualization techniques that influenced the design of DataHop are discussed next.

Multidimensional Data

Analyzing multidimensional datasets to find insights and trends has been a popular topic within IA. The most common method to visualize multidimensional datasets has been to use a 3D scatterplot that maps one data attribute to each spatial axis [2, 28, 29, 7]. This approach can be particularly helpful for visualizing datasets with natural 3D embeddings such as flight paths [13, 23] or traffic forecasts [32]. One multidimensional data visualization technique that makes effective use of a large space is *small multiples*. Small multiples describe a set of visualizations that have the same visual encodings across different slices of a dataset. These visualizations are usually stacked alongside each other in a meaningful order so that comparisons can be made across the slices without overloading a user's working memory [31]. Small multiples have been explored in immersive spaces [13, 15, 23]. Liu et al., for example found that users were faster and more accurate when using a flat layout with a small number of multiples, but as the number of multiples increased, a half-circle layout provided the best performance [23].

While multidimensional visualization techniques enable users to see and interact with their data in new and efficient ways, they are mostly static. As the user is not encouraged to travel large distances within the environment, these techniques cannot capitalize on users' spatial capabilities like DataHop does. Indeed, in a recent field study evaluating the use of IA for economics analysis, Batch et al. found that participants did not use 3D space to its full extent, believing this *"points to the need for system support, such as constraints and organization frameworks, to help users organize their spaces"* [3].

Immersive Software Visualizations

Source code visualization systems have harnessed users' spatial recall capabilities to improve their understanding of complex systems [16, 17, 24, 33, 36]. Software systems are inherently multifaceted, with many elements such as programs, classes, methods, arguments, and so on, making it challenging for those unfamiliar with the source code to parse and understand it. *SoftwareWorld,* for example, sought to make comprehension easier by using different cartographic abstractions as metaphors for source code, i.e., buildings represented methods and districts represented a class [16, 17]. More recently, Souza et. al used the metaphor of a dynamically updating city to represent software source code commits over time [36].

By using urban metaphors, these visualizations leveraged users' spatial recall capabilities to relieve some of the cognitive workload associated with parsing and understanding software source code. However, with these approaches, even those that dynamically updated, a predetermined layout was used to populate the environment. While this can be useful within the domain of software system visualizations, it is not feasible to expect arbitrary multidimensional datasets to conform to such metaphors or spatial layouts. With DataHop, the user can create personalized, meaningful layouts to explore multidimensional datasets that leverage their spatial recall.

Unit Visualizations

DataHop visualizes multidimensional datasets by associating a unique mark to each data point in the dataset. This differs from other visualization techniques that aggregate multiple values together such as bar charts [10]. Drucker et al. refer to mark-based visualizations as *unit* *visualizations* [9]. These visualizations can enable more intuitive representations and interactions with an underlying dataset. Park et al. characterized the design space of unit visualizations and proposed a grammar, Atom, for creating such visualizations [30]. From their design space, they identified the four most common operations performed with unit visualizations, i.e., *Bin, Duplicate, Filter,* and *Flatten*. Recent work by Ivanov et al. [14] extended this concept into immersive spaces with *immersive unit visualizations.* These visualizations leverage the perceived presence and embodiment often found in VR to convey a sense of scale and magnitude that would not be achievable with aggregate visualizations or non-immersive spaces [35].

Navigation and Spatial Skills in Virtual Environments

Populating large environments with virtual content is a challenge, however, without effective navigation techniques, users will not be able to explore content quickly and easily in these environments. Many techniques have been proposed to improve navigation, such as jumping [4, 39], teleportation [5, 22], virtual body resizing [1, 18], and flying [26], among others.

Encouragingly, Krokos et. al. demonstrated that memory palaces (i.e. spatial mapping of information to features in the environment) performed better in VR than in desktop conditions, supporting that VR can offer superior memory recall [19]. Nevertheless, navigating large distances within the virtual environment can be disorienting and weaken spatial skills. Thus, many applications make use of a smaller scale model of the virtual environment, i.e., a World-In-Miniature (WIM) [37]. This handheld model allows users to manipulate objects within the model and have the manipulations reflected in the larger virtual environment or select locations or hotspots in the model to travel to. A WIM provides an at-a-glance overview of the surrounding area and prevents users from having to travel back and forth between different areas if they want to find patterns in content or make comparisons. This technique, along with an elevator metaphor, was used within DataHop to facilitate effective spatial navigation.

Provenance and Maintaining Workflows

Maintaining and referencing the steps in one's workflow is an important aspect of iterative problem solving [40]. Preserving and visualizing one's history has undergone extensive research in traditional visual analytics [12, 20, 34, 38] because it can provide an overview of the states that have been explored and reveal gaps in one's workflow. To maintain these steps, many systems model a workflow as a graph where nodes represent application *states* and edges represent *actions* used to transition between two states [12]. For instance, *Image Graphs* [20] used a graph structure to spatially visualize one's workflow during image creation, where a node was an image and an edge was an operation to transform one image into another. DataHop extends this graph concept into VR, where there is limitless space, thus enabling entire workflow graphs to be spatially represented for easy reference and understanding.

DATAHOP

Data analysis tasks require users to repeatedly triage, filter, search, combine, and synthesize large amounts of data throughout their workflow. DataHop harnesses the limitless space that virtual environments afford to enable users to visualize the mental steps they have taken throughout their workflows at distinct spatial positions in the virtual environment. Each step within one's workflow can be represented by a *Filter*, which takes in one or more *data panels* (i.e., containers of points) and outputs an incremental change to them in the virtual environment. This support for spatial remapping enables users to capitalize on their spatial recall capabilities by utilizing schemas and organizational strategies that are familiar to them.

As a motivating example exploring the use of DataHop, consider Jeffery, a data analyst and a movie enthusiast who would like to explore a multidimensional dataset of movies. Each movie in the dataset has various attributes including title, year released, genre, language, budget, revenue, rating, popularity, country of origin, and so on.

Basic Interactions and Design

The virtual environment is initially composed of a single root data panel placed in the center of an infinitely large floor. Data panels and *Filters* are indicated with circular boundaries on the floor (Figure 2a). Through empirical observations, a data panel was chosen to be 1.5 x 1.5 meters.

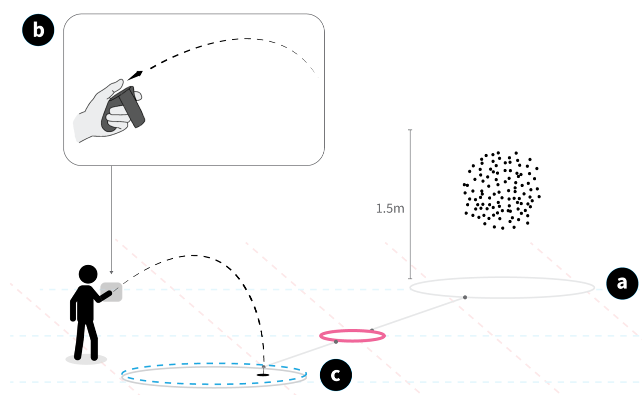

Figure 2. Data panels and *Filters* are shown with circular boundaries on the floor (a). A user can point at the floor by pushing their thumb stick forward (b) to select a data panel or a *Filter* (c).

To interact with the environment, a parabolic ray can be invoked by pushing the thumb stick forward (Figure 2b). This will cause a cursor to become visible at the end of the ray. To interact with an object, i.e., a data panel or a *Filter*, the user needs to place the cursor near or inside the object boundary (Figure 2b). Similarly, a user can navigate along the floor by pointing the parabolic ray at an empty location and clicking the index trigger to teleport to that location.

Filters and Branching

Filters are composed of a set of actions and a list of parameters that are used as inputs to the actions. Multiple actions can be selected at a time, but only one set of parameters can be used as input to these actions, which inherently limits each *Filter* to one iterative step. An action can be selected by tapping on it and data attributes can be dragged in as input to the action. Similar to the edges in Image Graphs [20], *Filters* are non-destructive and can be modified at any time such that a change to a *Filter* will propagate through the rest of its outputs.

When beginning to explore his dataset, Jeffery *branches out* from the root data panel. To do so, he points at the circular boundary of the root data panel (Figure 3a), squeezes the index trigger, and drags his cursor to specify the placement of the new *Filter* (Figure 3b). When satisfied with the placement, he releases the index trigger to confirm the creation of the *Filter*. Multiple data panels can also be assigned as input into a single *Filter* by first branching out from multiple data panels and then dragging one's cursor into the same *Filter*.

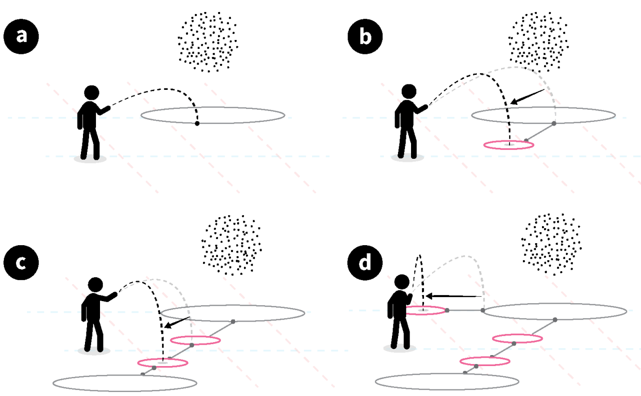

Figure 3. A user can branch out from a data panel by (a) pointing at its circular boundary and then (b) dragging to specify the position of the *Filter*. Multiple *Filters* can be chained together (c) and a user can also branch out in multiple directions from the same data panel (d).

Because a *Filter* represents a single step in the workflow, to perform multiple steps, Jeffery can *branch out* from a *Filter*. This creates a new *Filter* which uses the outputs of the old *Filter* as inputs. The output data panel(s) will now be the composition of these two *Filters* (Figure 3c).

It is important note that users can branch out in multiple directions from a single data panel (Figure 3d). In this way, users can carry out alternative explorations in different regions of space while preserving their existing workflow.

Actions Within Data Panels

If Jeffery wishes to transform the visual encoding of points in a data panel, DataHop provides three basic actions: group, distribute, and filter. He can pass parameters of each data point as input to these actions (e.g., *group* by *title*, *distribute* by *release year*, *filter* by *genre*, etc.).

While using these actions, a WIM visualization appears and displays the single output being acted on (Figure 4). This visualization ensures that Jeffery does not need to frequently change his focus by looking at the output in the world when experimenting with the *Filter*.

Grouping

The group action assigns a common color to data points that are in the same category or bin of values. If no previous distribution is specified, i.e., through the distribute action, then the group action also clusters the points together (Figure 4).

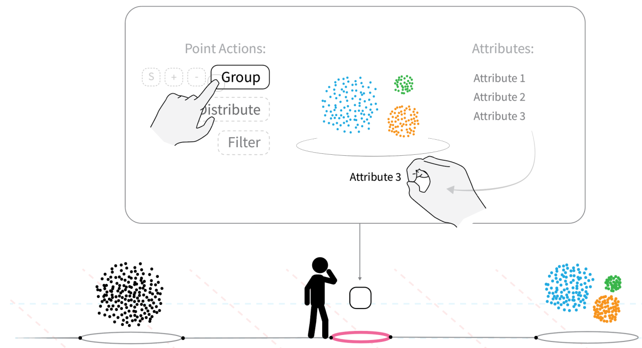

Figure 4. The group action assigns the same color to common points and clusters them together.

The group action also allows users to divide groups such that there is a unique output data panel for each group of points (Figure 5), i.e., a user can divide a single input into multiple data panels so each contains a group of data points.

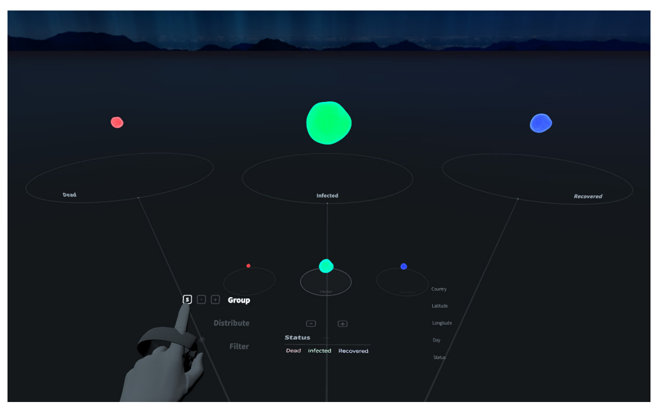

Figure 5. An input data panel of reports of COVID-19 being divided by the status of the report, creating a separate output data panel for infections, recoveries, and deaths.

While Jeffery is using the movies dataset, he thinks that it would be interesting to see the state of cinema at different periods in time so he groups the data by decade by selecting the group action and dragging in the year property. After adjusting the bin size to 10 years, he has a group of points for every decade, as indicated by them having a common color and being located near each other. Jeffery now divides the groups into distinct data panels so he can take a closer look at each decade.

Distributing

To distribute the points along one or more axes, the user can use the distribute action (Figure 6). Distributing data using a single property creates a histogram, whereas distributing data using two properties creates a 2D scatter plot with the added option to view a 3D histogram. Data that is distributed along three properties creates a 3D scatter plot.

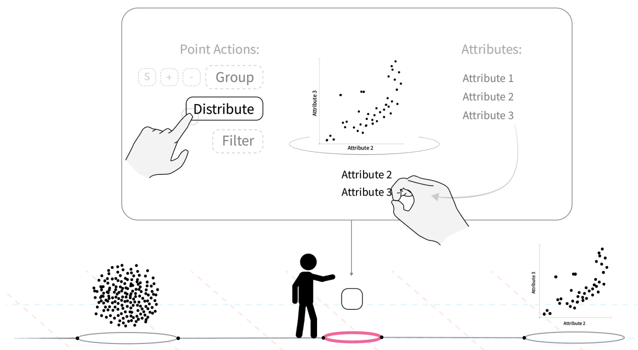

Figure 6. Distributing data along two attributes displays a 2D scatterplot of those attributes.

Jeffery can use this action to explore the distribution of different properties, for example, by year. The resulting histogram shows Jeffery that this dataset has a rich archive of movies, dating all the way back to late 19th century. Jeffery is now curious if this catalogue of movies also contains non-Western cinema. So, he distributes the data by the longitude and latitude of where the movie was produced, creating a 2D map of movies from around the world. Because the points overlap each other, Jeffery cannot see the number of points at different locations, so he looks at the 3D histogram to see the density of points at different latitude and longitude positions. While this dataset is dominated by Western cinema, there seems to be quite a few movies from South Asia as well.

Property-Based Filtering

The filter action can be used to mask data points that have certain properties. Categorical parameters can be filtered by category, whereas a range of values can be specified for continuous parameters. For instance, Jeffery wonders if there was a boom in history films following the second World War. To answer his question, he first filters out all other genres except *historical*, and then chains together another *Filter* which distributes them by year.

Actions Across Data Panels

Filters can produce multiple output data panels so sometimes it may be useful to edit the placement and arrangement of these panels. DataHop provides another set of actions for this, which can be activated from a *Filter*.

Layout Schemes

DataHop supports three different layout schemes, i.e., horizontal, circular, and vertical, however all layouts constrain the resulting output to be in front of the user, in the direction of the branch. As such, the user will always face their current path of exploration and the additional structures will not be generated outside their field of view.

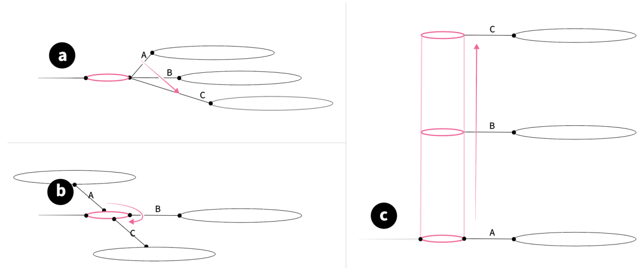

Figure 7. DataHop supports three layout schemes, i.e., (a) Horizontal, (b) Circular, and (c) Vertical.

The ***Horizontal*** layout places output data panels from left to right in the virtual environment (Figure 7a). It is an easy to recognize configuration that works well for side-by-side comparisons of a small number of outputs. For example, when visualizing a dataset of federal election results between three political parties, if a *Filter* created one output for each party, it would be trivial to layout and make quick comparisons across data panels. However, when there are more than a few outputs, outputs at the extremities can be too far away for the user to meaningfully parse.

The ***Circular*** layout places output data panels in a semi-circle a uniform distance from the user (Figure 7b). This removes any bias of certain outputs being spatially further away than others and allows for the easy comparison of trends. For example, with a movie dataset where one output represents a decade of movies, if outputs were sorted chronologically using the circular layout, a user could easily compare adjacent outputs and discern overall trends.

The ***Vertical*** layout stacks output data panels on top of each other, on separate floors (Figure 7c). It is a powerful layout because each output has its own dedicated floor, thus acting like a new workspace. With the movies dataset, for example, if we wanted to observe language-based patterns, then creating an output for each language and arranging it vertically would provide ample room to independently analyze each language. This layout does, however, makes it unfeasible to perform comparisons across outputs because they are located on different vertical levels.

Output Modifications

In addition to specifying the placement of data panels in the environment, users can also make use of four available actions to further modify a data panel's output, i.e., sorting, filtering, merging, and reusing.

The sort action arranges data panels with respect to the number of points in a panel or by averaging a data attribute. (Figure 8). Optionally, the user can choose to reverse the sorting order. If Jeffery had an output for each decade of cinema, for example, he could sort them by the average run time to determine when lengthier films were more common.

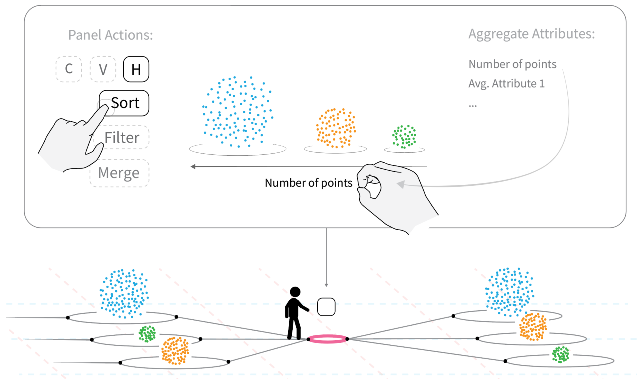

Figure 8. Users can sort their outputs according to the number of points in an output, or by the average attribute of points in an output.

Like property-based filtering, output-based filtering can be used to mask entire outputs. For example, if Jeffery had an output for each language, he could filter out those outputs with fewer than 100 movies to simplify his analysis.

The merge action combines all inputs into a single output. For instance, Jeffery could merge educational genres (e.g., documentaries, historical, etc.) and compare these movies against specific entertainment genres (e.g., action, adventure, etc.).

Reusing Filters

Filters can be reused by copying and pasting them. This can be beneficial if, for instance, a user wishes to transfer the analysis they performed from one data panel to another without having to redo their steps. A user can copy a *Filter* by pointing in the center of its circular boundary on the floor and holding the grip trigger, they can then paste this *Filter* near an output by clicking the index trigger. To store *Filters* for later use on different floors or after Jeffery has travelled a long distance, he can copy a *Filter* and then pull back his thumb stick. This, and other stored filters, will then become visible near his virtual hand.

Navigation

To navigate throughout DataHop while preserving spatial memory, two techniques are available, the Elevator and a Mini Map.

Elevator

An elevator metaphor is used to support vertical navigation. It helps decrease nausea and should better match the natural pacing of data analysis tasks than other techniques such as jumping or teleporting. A *Filter* displays a World-In-Miniature view of its output data panels, resembling an elevator interface. Tapping on one of the miniature outputs renders an elevator that moves the user to the respective output (Figure 9).

The elevator moves gradually through space, allowing the user to take in the structure as they are navigating it. The time taken in the Elevator is proportional to how far away the target destination is and the speed of the elevator eases in and out. The Elevator may also be used for navigation horizontally along the floor to data panels that are organized horizontally or circularly. In this case, the metaphor transitions to a moving walkway.

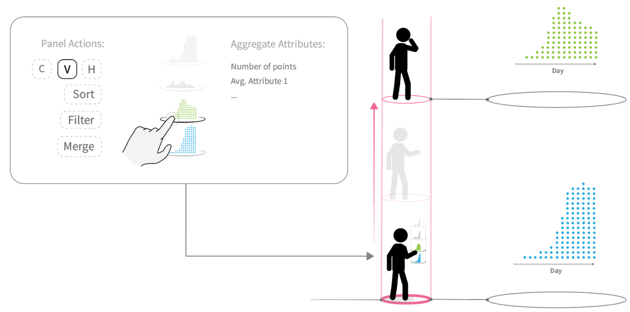

Figure 9. Tapping on one of the outputs in the Filter's WIM navigates the user to that output using an Elevator.

Mini Map

A World-In-Miniature view of the surrounding environment can be rendered whenever the user tilts their left hand towards themselves, similar to the motion one makes to look at their watch. This mini map acts like a compass in that it always faces the global forward direction. A green arrow indicates the user's position and direction in the miniature view (Figure 10).

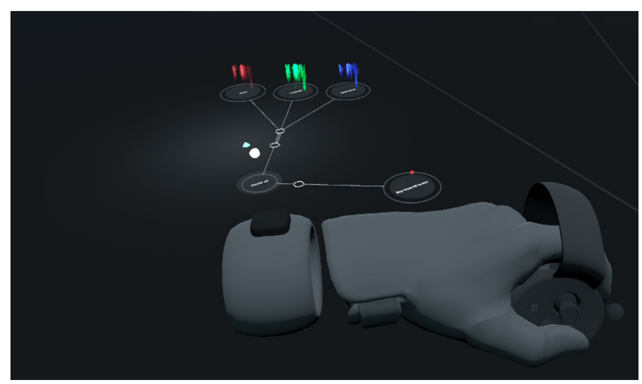

Figure 10. A mini map is shown whenever the user tilts their left hand and displays a WIM view of the surrounding area.

With the Mini Map, not only do users obtain a spatial understanding of where they are in the structure they have built, but they can also use it to invoke the elevator and traverse to a data panel by tapping on the data panel's miniature version.

USER STUDY

A remote exploratory user study was conducted with VR users to gather feedback about the utility of DataHop to explore and understand multidimensional datasets. We wished to investigate the new workflows that DataHop enables and how users utilize layout schemes to facilitate their data explorations. During the study, participants investigated two multidimensional datasets in virtual reality using DataHop's techniques. The study took approximately one hour, and participants were compensated $30 CAD.

Participants

Six users who had an Oculus headset were recruited (2 females; μ = 30 years, range = 22 to 39 years). Participants took part in the study over a remote video call and they were required to use an Oculus headset to ensure that the overall characteristics of the controllers were consistent across participants. On average, participants had 1.5 years of experience using VR devices. Participants self-evaluated their data visualization expertise on a 5-point Likert Scale ranging from *Novice* (None)*, Beginner* (P5)*, Competent* (P1, P3, P6)*, Proficient* (P2)*, and Expert* (P4).

Experimental Environment and Hardware

To evaluate the design of DataHop, a prototype application was developed using the Oculus Rift. It was implemented using the Unity Game Engine and Steam VR plugin.

Study Setup

For the duration of the study, the experimenter was on remote video call with the participant. The participant also shared their screen so that the experimenter could guide them through the prototype. The call was recorded by the experimenter for transcription purposes.

Participants were asked to stand up when using the prototype. Each participant had varying amounts of physical space to walk around in, but because the system supports short distance navigation by grabbing the air, participants did not need to move significant distances in their physical space.

Procedure

The study consisted of the following phases.

Pre-study

Before the study, the participant was asked to download the prototype and digitally sign a consent form.

Introduction (5 minutes)

The participant joined a remote video call and filled out a demographic questionnaire. The goal and purpose of the study was then introduced to them.

Video Tutorial (5 minutes)

Before putting on their headset, participants were shown a brief video tutorial about what to expect in the virtual environment and were introduced to the controls. This video tutorial covered navigating by grabbing the air, pointing at a location to teleport, as well as explaining how to branch out from a data panel.

Guided Scenario #1 (20 minutes)

During the first scenario, participants were guided through the key features of DataHop using a dataset of the 2016 US Election [21], where each unit represented an electoral precinct (N=140,000).

Guided Scenario #2 (20 minutes)

The second scenario used a COVID-19 dataset [8], where each point represented 10 reports of an infection, death, or recovery from COVID-19 (N=430,000). Each unit had the following dimensions: country, longitude, latitude, day, status. The status property encoded whether the report was an infection, death, or recovery. With this scenario, participants were provided with more freedom to explore the data at their own pace. If they became stuck at any point, they were tasked with investigating weekly trends in the worst hit countries.

Interview and Questionnaire (15 minutes)

Following the guided scenarios, an interview was conducted where participants were asked high level questions regarding the underlying features and metaphors within the system. After the interview, the participant completed a set of 7-point Likert Scale questions about the usability and usefulness of DataHop.

Results

Overall, participants found DataHop to be useful to maintain and understand their spatial workflow when exploring data. The exploratory interview data, along with the questionnaire results, highlighted the important role of vastness during exploration, how filters encourage users to transfer their existing knowledge into new immersive analytic environments, and the importance of using navigational techniques that match task expectations and pacing, among others.

Vastness Encourages Exploration

Participants were able to maintain their spatial understanding throughout the whole session (i.e., 1 - Neutral, 1 - Slightly Agree, 2 - Agree, 2 - Strongly Agree). The third dimension afforded by VR was found to be helpful because *"if you have an infinite 2D canvas, it's difficult to navigate, because in 3D I can look around in all directions, but 2D is limited to the canvas size. Seems like you have more ability to navigate or like find out history, than if this were in 2D"* (P3). Other participants enjoyed being able to spatially see where they had already navigated to, e.g., *"being able to see this [mini map] view and see oh I already tried that in this direction, and that got me here. Being able to go back and say what if I changed this back here, I think that is super important for exploring*" (P1). For these participants, the use of spatial mapping enabled them to quickly scan throughout the entire environment to see all their data and workflow history, something that was important to them, and in the case of P3, was perceived as impossible if an endless 2D environment was used.

Others found the WIM views to be useful when exploring the vastness of the environment. For example, seeing the *"smaller map on Filter is very useful in conjunction with the larger one I see in the space*" (P5) and when *"actually interacting with the Filters, I am more looking at the tiny view. But I guess when the Filter is completed and I want to look at it in more depth, then I like concentrating on the large-scale visualization"* (P4). For these participants, the *Filter's* WIM view was a helpful aid for accessing and reinforcing what was in their working memory.

There were, however, concerns about the environment being too vast and users becoming disoriented. e.g., *"I think spatially it gets too vast for me to navigate properly after a while, even in the mini map you only see surrounding data"* (P5). All participants except P2 mentioned that the Mini Map view helped them better understand the distribution and the layout of the environment. As noted by P3 however, *"when there are too many groups, it seems like mini world [map] is still not adequate"* P6 felt disoriented using the system, noting that since the *"scene was blank, it was hard to remember spatial position. If you have something with more variation, you can maybe color the ground inside the game scene, just something to remember that by*". The inclusion of markers or waypoints within the environment were proposed as ways to alleviate these concerns, in addition to an indicator specifying which dataset was the root dataset and the ability to have variable zoom levels to pick from when using the WIM widgets.

Filters Support the Transfer of Existing Workflows

Participants liked the *Filter* as an iterative step for data analysis and found its use helped them transfer the steps they normally take within traditional data analysis workflows, i.e., *"if you do stats in R[1], it is how people do it. You don't call it a Filter, it is typically called a pipe, and you perform functions and steps to transform the data … [DataHop] can borrow a lot from existing data analysis tools, like not just group data but create data, like a classic operation is to mutate data"* (P3). For this user, the *Filters* within DataHop enabled them to harness the knowledge and common workflow activities that they perform in non-immersive analytics tasks. This is a welcome finding, as it suggests the potential usefulness of creating other visual representations of common analytic task tools within DataHop in the future.

Although *Filters* were well liked and participants appreciated their iterative nature, participants did not like the overhead necessary to create them, e.g., *"the overhead of creating them and moving to them is what would change my preference"* (P6). As suggested by P5, it may be useful to "*think that Filter is one horizontal line across the sand, so that I don't have to keep moving forward, and moving back, I would like to have it displayed horizontally".*

Different Layouts for Different Tasks

The different layouts used by DataHop to visualize branches were generally found to be sufficient (i.e., 1 - neutral, 1 - slightly agree, 1 - agree, 2 - strongly agree), but *"they all had advantages and disadvantages"* (P6) for different tasks and goals.

The horizontal layout was perceived as useful for comparisons among smaller number of outputs as *"if there are a smaller number of data groups, I want to compare them side by side which is a natural behavior of people"* (P3). However, all participants stated that it would not be effective for viewing more than a few outputs, even if they *"really like [horizontal layout] when everything can stay within [their] field of view, once it goes beyond it, it is kind of hard to compare spatially"* (P4), so the circle layout would be better, i.e., "*the circle uses your full field of view rather than just having them laid out side by side*" (P1).

The vertical layout was noted as the most useful when looking at output separately. It seemed *"to have a unique function, vertical layouts create these different planes, almost like a Photoshop layer if I had to describe it, because that's how it felt to me when I looked up and saw the China layout [one of the explorations on another floor] that I made five minutes before"* (P2). Participants also mentioned that while *"you can leverage the height to expand your space by a huge factor"* (P3), it nullifies the possibility for comparisons with *"anything beyond three levels up [because it] is impossible to see"* (P3). P1 noted that the vertical layout gave *"the clearest visualization of it being the same data set just split up. Having them stacked like that makes a lot of sense spatially".*

One interesting finding that emerged was how spatially mapping one's workflow gave rise to the desire to utilize more intricate and complex structures, like coming up "*with elaborate cool looking data-vis forests, that you could walk around in. Because I think that it is a really amazing thing for yourself and to see them, but if you apply and arrange them in a creative way, it can become this VR poster, that you can walk around in and share with other people."* (P2). Hybrid layouts were also suggested to facilitate rapid side-by-side comparisons, e.g., *"a horizontal layout within the circular layout if I want to say, compare two elements within the circular layout"*. Others suggested to distributing the outputs in a matrix or cube (P1) or on polar coordinates about the center of a *Filter* (P5).

These results suggest that there is need to include multiple layout options within immersive analytic environments so that users can choose the one that best suits the task at hand, or a system could make layout recommendations based on underlying dataset properties. It is also encouraging that participants ideated on other creative layouts that could help them understand and explore their own personal workflows, suggesting the further possibly for tools that enable a user to create ad-hoc hybrid layouts if needed.

Data Splitting Does Not Lead to Clutter and Confusion

If a user divides groups into outputs, they subsequently populate the environment with multiple new data panels, which could eventually lead to clutter and confusion. Participants, however, reported that they easily maintained a sense of spatial understanding after dividing data. P4 and P5 were concerned that in the case of many outputs being created it could feel *"like when it's off to the distance on the sides, it's hard to parse"* (P4). These comments suggest that it may be prudent to limit the number of groups that are

[1] R: A language and environment for statistical computing. http://www.R-project.org/

created within a visualization or develop algorithms that re-flow generated data panels into hybrid horizontal-vertical layouts when the number of visualizations falls out of one's field of view or distance threshold they wish to travel.

The Elevator as a Transitional Navigation Metaphor

All participants found the elevator to be a helpful metaphor for navigation (i.e., 3 - Slightly Agree, 1 - Agree, 2 - Strongly Agree) and none of the participants reported that the elevator caused them motion sickness. We acknowledge that these participants had previous experience using VR devices (on average, 1.5 years of experience), thus, are less susceptible to motion sickness than novice VR users.

The elevator was perceived as helpful because it allowed participants to "*see how you are transitioning"* (P4) and did so at a speed that was appropriate for their tasks, i.e., *"experience the in-between … the experience of going up the elevator is slow enough that spatially in my mind, I am maintaining all the connections."* (P4). P5 found that the *"elevator is actually the most useful function for me creating this spatial relationship. I think organizationally, it's very useful"*. P2 also mentioned that invoking the elevator from the *Filter* instead of arbitrarily from any position was *"good because it support[ed] the idea that it is an elevator which has a fixed thing [location on the horizontal plane]"* (P2). By returning to the elevator, participants could maintain their bearings when navigating between different floors.

DISCUSSION AND FUTURE WORK

Results of the user study demonstrated that new visualization and navigation paradigms such as DataHop can enable users to understand and explore their datasets in novel, more meaningful ways. The results also revealed the importance of larger themes such as the need for spatial cues and the importance of flexible layouts.

Spatial Cues

Although vast environments encourage exploration, a few participants expressed concerns about losing their bearings or feeling lost, particularly when they had travelled to a higher floor or far away from the root dataset. Disorientation may be due to the lack of variety in the virtual space. When different regions look or feel similar, it can become easy to confuse them up and get lost. Spatial cues could be used to stimulate and maintain a person's spatial memory to overcome these challenges. For example, visual markers could be placed by the system (e.g., at the root dataset) or manually by the user at places of interest. Waypoint markers could then be displayed on WIMs to further reinforce their spatial locations or markers could also use spatial audio cues to passively communicate their position. Environmental spatial cues could also be employed, such as landmarks in the distance, elevation-based visual effects, or varied textures on floors. It may also be helpful to display a user's movement history by preserving and visualizing their footsteps as they explore the environment and use these footsteps as a reference to the areas of space they have explored and navigated to.

Flexible Layouts

While participants generally found the three provided layouts to be sufficient for their visualization tasks, there is an opportunity to provide the user with more agency over the layouts that are available. In doing so, users can create expressive visualizations that better reflect their intent such as creating hybrid layouts or adjusting continuous parameters such as the curvature [23]. Creating more intricate 3D layouts, where outputs are spatially distributed in a matrix or a cube could also be interesting as these layouts would enable additional properties such as inter-panel spacing or cellular proximity to be harnessed. They do, of course, introduce additional challenges such as how to overcome multiple outputs being located in the same spatial location [27].

Future Work

This research has opened up several interesting directions for future work. For one, it has only focused on unit data representations. Data that is structured in other forms, such as node-link diagrams, may also be effectively mapped to immersive spaces using provenance. It may also be interesting to visualize timeseries datasets temporally as well as spatially.

While the current implementation of DataHop enforces a consistent user scale, future work should explore the use of dynamic user scales to enable for direct interaction with data at various scales, from individual datapoints, to groups of datapoints and even whole datasets. This could open new forms of immersive data exploration, sensemaking, analysis, and storytelling.

VR is a useful tool for collaboration as users have a sense of each other's presence in virtual space. DataHop allows users to analyze data by incrementally creating structures from it, which can naturally lend itself to multi-user collaboration. Since these structures are a representation of one's thought process, being able to save and share them could be very powerful. Others not only see the resulting intricate visualizations but can also observe each step spatially to see how it was derived and constructed.

CONCLUSION

This paper presented a novel method to view and understand multidimensional datasets that leverages users' spatial skills. DataHop enables users to view, manipulate, and traverse each step of their data analysis workflow in virtual space. From a qualitative user study, we found that being able to spatially layout one's workflow can be beneficial for data exploration and data understanding. This work provides a glimpse into future virtual reality interfaces that may utilize our spatial cognition skills for effective information visualization.

ACKNOWLEDGEMENTS

We would like to thank Dr. Michelle Annett for editing, the anonymous reviewers for their valuable feedback in helping refine this work, and members of the DGP lab for their help and input.

REFERENCES

[1] Parastoo Abtahi, Mar Gonzalez-Franco, Eyal Ofek, and Anthony Steed. 2019. I'm a Giant: Walking in Large Virtual Environments at High Speed Gains. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19), 1–13. https://doi.org/10.1145/3290605.3300752

[2] Ayman Ammoura, Osmar R. Zaíane, and Yuan Ji. 2001. Immersed Visual Data Mining: Walking the Walk. In *Advances in Databases* (Lecture Notes in Computer Science), 202–218. https://doi.org/10.1007/3-540-45754-2_13

[3] Andrea Batch, Andrew Cunningham, Maxime Cordeil, Niklas Elmqvist, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2020. There Is No Spoon: Evaluating Performance, Space Use, and Presence with Expert Domain Users in Immersive Analytics. *IEEE Transactions on Visualization and Computer Graphics* 26, 1: 536–546. https://doi.org/10.1109/TVCG.2019.2934803

[4] Benjamin Bolte, Frank Steinicke, and Gerd Bruder. 2011. The Jumper Metaphor: An Effective Navigation Technique for Immersive Display Setups. *Proceedings of Virtual Reality International Conference*.

[5] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. 2016. Point & Teleport Locomotion Technique for Virtual Reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY '16), 205–216. https://doi.org/10.1145/2967934.2968105

[6] Simon Butscher, Sebastian Hubenschmid, Jens Müller, Johannes Fuchs, and Harald Reiterer. 2018. Clusters, Trends, and Outliers: How Immersive Technologies Can Facilitate the Collaborative Analysis of Multidimensional Data. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (CHI '18), 1–12. https://doi.org/10.1145/3173574.3173664

[7] Maxime Cordeil, Andrew Cunningham, Tim Dwyer, Bruce H. Thomas, and Kim Marriott. 2017. ImAxes: Immersive Axes as Embodied Affordances for Interactive Multivariate Data Visualisation. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17), 71–83. https://doi.org/10.1145/3126594.3126613

[8] Ensheng Dong, Hongru Du, and Lauren Gardner. An interactive web-based dashboard to track COVID-19 in real time. The Lancet infectious diseases 20.5 (2020): 533-534. https://doi.org/10.1016/S1473-3099(20)30120-1

[9] Steven M Drucker and Roland Fernandez. A Unifying Framework for Animated and Interactive Unit Visualizations. *Microsoft Res.* 2015

[10] M.C. Ferreira de Oliveira and H. Levkowitz. 2003. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics* 9, 3: 378–394. https://doi.org/10.1109/TVCG.2003.1207445

[11] Adrien Fonnet and Yannick Prié. 2019. Survey of Immersive Analytics. *IEEE Transactions on Visualization and Computer Graphics*: 1–1. https://doi.org/10.1109/TVCG.2019.2929033

[12] Jeffrey Heer, Jock D. Mackinlay, Chris Stolte, and Maneesh Agrawala. 2008. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. In *Proceedings of IEEE Information Visualization 2008, IEEE.*

[13] Christophe Hurter, Nathalie Henry Riche, Steven M. Drucker, Maxime Cordeil, Richard Alligier, and Romain Vuillemot. 2018. FiberClay: Sculpting Three Dimensional Trajectories to Reveal Structural Insights. *IEEE transactions on visualization and computer graphics*. https://doi.org/10.1109/TVCG.2018.2865191

[14] Alexander Ivanov, Kurtis Danyluk, Christian Jacob, and Wesley Willett. 2019. A Walk Among the Data. *IEEE Computer Graphics and Applications* 39, 3: 19–28. https://doi.org/10.1109/MCG.2019.2898941

[15] Seth Johnson, Daniel Orban, Hakizumwami Birali Runesha, Lingyu Meng, Bethany Juhnke, Arthur Erdman, Francesca Samsel, and Daniel F. Keefe. 2019. Bento Box: An Interactive and Zoomable Small Multiples Technique for Visualizing 4D Simulation Ensembles in Virtual Reality. *Frontiers in Robotics and AI* 6. https://doi.org/10.3389/frobt.2019.00061

[16] C. Knight and M. Munro. 1999. Comprehension with[in] virtual environment visualisations. In *Proceedings Seventh International Workshop on Program Comprehension*, 4–11. https://doi.org/10.1109/WPC.1999.777733

[17] C. Knight and M. Munro. 2000. Virtual but visible software. In *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*, 198–205. https://doi.org/10.1109/IV.2000.859756

[18] Andrey Krekhov, Sebastian Cmentowski, Katharina Emmerich, Maic Masuch, and Jens Krüger. 2018. GulliVR: A Walking-Oriented Technique for Navigation in Virtual Reality Games Based on Virtual Body Resizing. In *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play* (CHI PLAY '18), 243–256. https://doi.org/10.1145/3242671.3242704

[19] Eric Krokos, Catherine Plaisant, and Amitabh Varshney. 2019. Virtual memory palaces: immersion aids recall. *Virtual Reality* 23, 1: 1–15. https://doi.org/10.1007/s10055-018-0346-3

[20] Kwan-Liu Ma. 1999. Image graphs-a novel approach to visual data exploration. In *Proceedings*

Visualization '99 (Cat. No.99CB37067), 81–88. https://doi.org/10.1109/VISUAL.1999.809871
[21] MIT Election Data and Science Lab. 2019. U.S. President Precinct-Level Returns 2016. https://doi.org/10.7910/DVN/LYWX3D
[22] Eike Langbehn, Paul Lubos, and Frank Steinicke. 2018. Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking. In *Proceedings of the Virtual Reality International Conference - Laval Virtual* (VRIC '18), 1–9. https://doi.org/10.1145/3234253.3234291
[23] Jiazhou Liu, Arnaud Prouzeau, Barrett Ens, and Tim Dwyer. Design and Evaluation of Interactive Small Multiples Data Visualisation in Immersive Spaces. 10.
[24] J.I. Maletic, J. Leigh, A. Marcus, and G. Dunlap. 2001. Visualizing object-oriented software in virtual reality. In *Proceedings 9th International Workshop on Program Comprehension. IWPC 2001*, 26–35. https://doi.org/10.1109/WPC.2001.921711
[25] Kim Marriott, Falk Schreiber, Tim Dwyer, Karsten Klein, Nathalie Henry Riche, Takayuki Itoh, Wolfgang Stuerzlinger, and Bruce H. Thomas (eds.). 2018. *Immersive Analytics*. Springer International Publishing. https://doi.org/10.1007/978-3-030-01388-2
[26] Daniel Medeiros, Antönio Sousa, Alberto Raposo, and Joaquim Jorge. 2019. Magic Carpet: Interaction Fidelity for Flying in VR. *IEEE Transactions on Visualization and Computer Graphics*: 1–1. https://doi.org/10.1109/TVCG.2019.2905200
[27] Wouter Meulemans, Jason Dykes, Aidan Slingsby, Cagatay Turkay, and Jo Wood. 2017. Small Multiples with Gaps. *IEEE Transactions on Visualization and Computer Graphics* 23, 1: 381–390. https://doi.org/10.1109/TVCG.2016.2598542
[28] Henrik R. Nagel, Erik Granum, and Peter Musaeus. 2001. Methods for visual mining of data in virtual reality. In *In Proceedings of the International Workshop on Visual Data Mining, in conjunction with ECML/PKDD2001, 2nd European Conference on Machine Learning and 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 9–12.
[29] Henrik R Nagel, Michael Vittrup, Erik Granum, and Søren Bovbjerg. Exploring Non-Linear Data Relationships in VR using the 3D Visual Data Mining System. 17.
[30] Deokgun Park, Steven M. Drucker, Roland Fernandez, and Niklas Elmqvist. 2018. ATOM: A Grammar for Unit Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 24, 12: 3032–3043. https://doi.org/10.1109/TVCG.2017.2785807
[31] Matthew D. Plumlee and Colin Ware. 2006. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *ACM Transactions on Computer-Human Interaction* 13, 2: 179–209. https://doi.org/10.1145/1165734.1165736
[32] Arnaud Prouzeau, Anastasia Bezerianos, and Olivier Chapuis. 2016. Towards Road Traffic Management with Forecasting on Wall Displays. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (ISS '16), 119–128. https://doi.org/10.1145/2992154.2992158
[33] C. Russo Dos Santos, P. Gros, P. Abel, D. Loisel, N. Trichaud, and J.P. Paris. 2000. Mapping information onto 3D virtual worlds. In *2000 IEEE Conference on Information Visualization. An International Conference on Computer Visualization and Graphics*, 379–386. https://doi.org/10.1109/IV.2000.859785
[34] Claudio T. Silva, Juliana Freire, and Steven P. Callahan. 2007. Provenance for Visualizations: Reproducibility and Beyond. *Computing in Science Engineering* 9, 5: 82–89. https://doi.org/10.1109/MCSE.2007.106
[35] M. Slater and M. Usoh. 1993. Presence in immersive virtual environments. In *Proceedings of IEEE Virtual Reality Annual International Symposium*, 90–96. https://doi.org/10.1109/VRAIS.1993.380793
[36] Rodrigo Souza, Bruno Silva, Thiago Mendes, and Manoel Mendonça. 2012. SkyscrapAR: An Augmented Reality Visualization for Software Evolution. In *Proceedings of 2nd Brazilian Workshop on Software Visualization (WBVS 2012)*.
[37] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '95), 265–272. https://doi.org/10.1145/223904.223938
[38] Jurgen Waser, Raphael Fuchs, Hrvoje Ribičič, Benjamin Schindler, Gunther Blöschl, and Eduard Gröller. 2010. World Lines. *IEEE Transactions on Visualization and Computer Graphics* 16, 6: 1458–1467. https://doi.org/10.1109/TVCG.2010.223
[39] Dennis Wolf, Katja Rogers, Christoph Kunder, and Enrico Rukzio. 2020. JumpVR: Jump-Based Locomotion Augmentation for Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (CHI '20), 1–12. https://doi.org/10.1145/3313831.3376243
[40] H. Wright and J. P. R. B. Walton. 1996. HyperScribe: A Data Management Facility for the Dataflow Visualization Pipeline. In *IRIS Explorer Technical Report IETR/4, NAG Ltd*.