FiberClay: Sculpting Three Dimensional Trajectories to Reveal Structural Insights

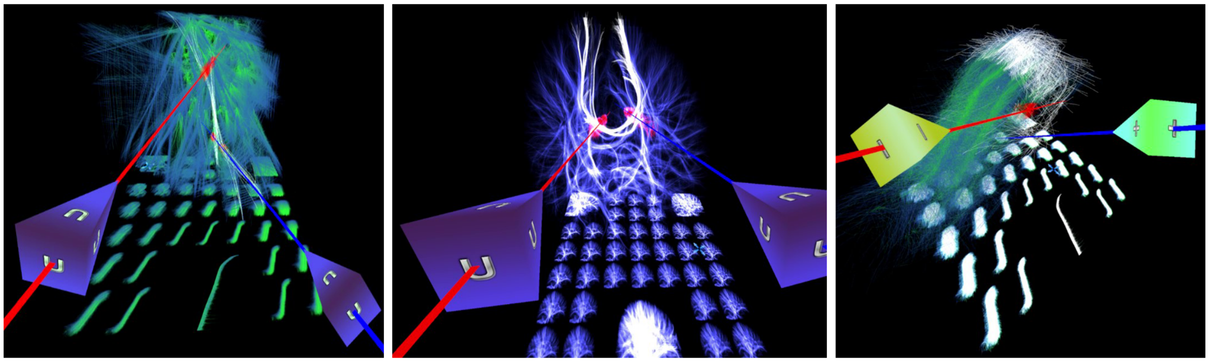

Fig. 1. FiberClayis an immersive multidimensional visualization system. The user can navigate into trail sets to gain a better understanding of dense and complex datasets. Left, the user activated the union brushing interaction to select trails intersected by the two beams in a recorded aircraft trajectory dataset. Middle, the user investigates DTI fiber tracks connecting the two hemisphere of the brain. Right, the user refine a data selection by visually sculpting the query with the two brush interactions while navigating into the different visual mappings (e.g. presets) of the investigated dataset.

Abstract— Visualizing 3D trajectories to extract insights about their similarities and spatial configuration is a critical task in several domains. Air traffic controllers for example deal with large quantities of aircrafts routes to optimize safety in airspace and neuroscientists attempt to understand neuronal pathways in the human brain by visualizing bundles of fibers from DTI images. Extracting insights from masses of 3D trajectories is challenging as the multiple three dimensional lines have complex geometries, may overlap, cross or even merge with each other, making it impossible to follow individual ones in dense areas. As trajectories are inherently spatial and three dimensional, we propose FiberClay: a system to display and interact with 3D trajectories in immersive environments. FiberClay renders a large quantity of trajectories in real time using GP-GPU techniques. FiberClay also introduces a new set of interactive techniques for composing complex queries in 3D space leveraging immersive environment controllers and user position. These techniques enable an analyst to select and compare sets of trajectories with specific geometries and data properties. We conclude by discussing insights found using FiberClay with domain experts in air traffic control and neurology.

Index Terms—Immersive Analytics, 3D Visualization, Dynamic Queries, Bimanual Interaction, Multidimensional Data

1 Introduction

One hundred and eighty years have passed since Charles Weatstone invented the stereoscope [52]: a device leveraging binocular depth perception and enabling humans to perceive 3D objects by looking at two 2D images. The technology has evolved tremendously since then and we now have immersive headsets capable of displaying digital 3D objects in a virtual or augmented environment while providing controllers to interact with the objects in physical space. As these devices become more popular and the effectiveness of input devices and the quality of their output has increased, the visualization community has started paying more attention to the opportunities that these environments can provide for exploring and analyzing data [11].

While the benefits of visualizing inherently spatial data as 3D digital objects has been demonstrated in numerous scientific visualization software (e.g. brain models [17]), there is still a debate on the pros and cons of 3D representations of abstract data [9, 35, 43]. In this paper, we tackle a hybrid type of data, composed of trajectories (inherently spatial in three dimensions) and associated to non-spatial data.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

Representing trajectories in a 3D space-time cube is natural [5], nevertheless this representation raises a number of issues. In particular, given the often dense and overlapping nature of many trajectories, effectively analyzing large numbers of trajectories can require novel techniques for interacting with them. For example, the user must compose complex queries in order to retrieve specific geometries or to identify clusters with similar properties. Performing such interactions is particularly challenging in immersive 3D environments as users have to simultaneously manipulate 6 degrees of freedom input devices, potentially to point to a specific pixel in a tangle of curves tightly interleaved in space.

To solve this problem, we present FiberClay, an immersive visualization system that renders masses of 3D trajectories in interactive time and enables bi-manual interaction to compose complex queries in 3D space. Its novelty lies in the simultaneous combination of both 3D brushing, for selection, and spatial navigation. Using this interaction, users can progressively filter data by quickly adding or subtracting trajectories to a selection, while easily navigating to different views and between different combinations of data attributes. Our case studies implementing this technique show that users can quickly grasp structural overviews of datasets; gain insights on the data distribution in both space and time; and investigate global and local trajectory patterns and anomalies.

We also describe technical considerations for addressing scalability as 3D computations in immersive environment can require considerably more computational power. We conclude by showcasing insights gained by 20 domain experts (air traffic controllers, medical professionals and data scientists) using FiberClay to explore their data.

2 Related Work

There has been a surge of interest in the past couple of years from the visualization community towards leveraging immersive environments for visualizing data [4, 13]. We first revisit inherent issues of these environments due to representing data in three dimensions and report on empirical evidence gathered through perception studies. We then summarize prior techniques for enabling users to interact and perform complex, multidimensional queries and highlight unique challenges and opportunities for composing queries in immersive environments.

2.1 The Case for Immersive Analytics

Immersive environments attempt to make use of the physical space a user is in, namely interacting within a three dimensional physical room, rather than limiting interaction to working on a flat 2D screen. Visual representations of data that have three dimensions, such as the positions and orientations of physical objects, are particularly appropriate to work with in these environments since they parallel the real experiences humans have in the natural environment. While the benefits of representing inherently 3D data as 3D representations are plain, there has been a lively and ongoing debate [9, 35, 43] as to whether using a third dimension can be advantageous for analyzing non-spatial (abstract), multidimensional data as it might potentially reveal more complex correlations and patterns.

Prior work has shown that 3D visualizations generally suffer from multiple issues: perspective distortion and occlusion on monoscopic 2D flat screens [37], and difficult navigation in 3D space using standard input modalities (mouse and keyboard) [36]. Findings from prior studies indicate that 3D representations are less effective than 2D representations of abstract data [40, 51]. In some cases, 3D visualizations were even found to be less effective than 2D ones for representing inherently spatial data (such as human organs) depending on the task [8].

A number of research projects have reported on the potential benefit of 3D representations of paths, graphs and networks. For example, dating back to 1993, Sollenberg and Milgram [44] and Ware et al. [49] collected evidence that the number of errors for following paths could be reduced by using stereoscopic displays. Ware and Mitchell [50] further observed this benefit for path tracing by combining stereo and motion. More recent work by Kwon et al. [33] introduced a head-centered spherical projection layout and a set of interaction techniques for graph visualization in VR. Their study showed evidence that the immersive 3D spherical layout led to faster answers than traditional 2D graph layout for typical graph visualization tasks and caused fewer errors in the case of large graphs. Another recent study by Usher et al. [46] also found that neuron path tracing using direct manipulation with hand controllers in VR was faster and less fatiguing than using commercial state-of-the-art dedicated software on 2D devices.

This evidence suggests that immersive environments may have significant advantages over existing 2D visualizations for representing and tracing trajectories. However, analyzing masses of trajectories requires interacting with these three dimensional objects to select subsets and compose complex queries, potentially involving non-spatial attributes. We tackle this challenge in this paper and propose FiberClay, a scalable system for representing and interacting with masses of 3D trajectory data also associated with multidimensional abstract data.

2.2 Queries in Multidimensional Data Visualization

Visual exploration of data relies on visualizing the data from multiple perspectives and interacting with it to perform different selections, filtering and queries, thus revealing patterns and outliers in the data. Seminal techniques to achieve these interactions include brushing and linking [6, 10] and dynamic queries [2].

2D multidimensional data queries. For multidimensional data, generalized selections [20] and iterative query building in scatterplot matrices with scatterdice [14] provide powerful interaction methods to craft complex queries involving multiple attributes. Several research projects also focused specifically on composing spatio-temporal queries such as TimeSearcher [21], query-by-trace [16] or model based queries [18], but none of them investigated their usage in an immersive environment.

Interactive trajectories selection and visualization. GeoTime [31] is an early desktop based system to visualize 3D space-time trajectories. Trajectory selection is performed using a mouse to create a rectangular selection region and time navigation is performed with a range slider. FromDaDy [30] is a 2D multidimensional data visualization tool that uses direct manipulation to select aircraft trajectories with a free-hand mouse-based brushing selection. This system allows successive selections of trails sets across several view configurations to achieve detailed selections.

Selection techniques and immersive visualization systems. While these interaction techniques work well in 2D, making them effective in immersive 3D environments is challenging. Selection of graphical objects in virtual environments is well explored [34], but there is a lack of immersive visualization-specific research in this domain. Previous research has focused on 3D point-based selection with a 2D touch surface [53], CT scan volume visualization selection in VR [45] and immersive network visualizations [23] in a HMDs. Our design of an abstract multidimensional data visualization tool in virtual reality is, itself, somewhat novel. Filonik et al. [19] developed Glance, a technical framework to produce visualizations of multidimensional, multivariate data in AR and VR environments. However, Glance does not include selection interactions. ImAxes [12] is a virtual reality based multidimensional data visualization system that leverages the user's space to build 2D and 3D visualizations with embodied interaction. In ImAxes a user can make multidimensional queries by connecting and linking visualizations. However in ImAxes, there is no direct manipulation to select data.

FiberClay is, to our knowledge, the first multidimensional data visualization tool that provides selections of 3D trails with a bi-manual, iterative, brushing technique and continuous navigation between visual configurations.

3 Design Challenges

We set out to address several problems inherent in working with large sets of 3D trajectories. Specifically, we wished to address the following design challenges:

DC1. Trajectory datasets are inherently multidimensional. The projection of these datasets into 2D for viewing on a conventional screen can hide interesting and important features within the data. While some previous work has focused on finding effective projections [48], this work focuses instead on enhancing the interaction techniques for immersive 3D environments operating on trajectory data.

DC2. Dense overlapping trajectories can visually occlude other features. Given the crowded and tangled nature of many trajectory datasets, there need to be effective mechanisms for isolating interesting features for further exploration.

DC3. Accurate selections can be difficult to perform in 3D. It may be hard to see what gets selected from a single point of view. Combinations of views, or views that are based on different data-mappings may be required to create the precise, desired selection.

DC4. Selections need to be progressively combined. Usually no single selection will isolate the desired selection so a combination of selections both through addition and subtraction need to be supported.

DC5. The system must be responsive even with large datasets. Interaction, as a key to data exploration, must be fast enough to be perceived as instantaneous to every user request.

4 FiberClay

In order to address the design challenges from the previous section, we propose FiberClay, a system that uses a set of interaction and visualization techniques to explore datasets in 3D immersive environments. FiberClay enables users to quickly reveal structural insights on datasets by allowing query construction using multiple input modalities (e.g. hands, head) and visualization facets (e.g. maps, small multiples, space-time cube). The name *FiberClay* suggests that the user starts with a whole dataset as *raw sculpting material* that is progressively refined until producing the desired goal. The clay metaphor affords important interaction opportunities: users can make mistakes that can be easily fixed; users can either remove material or add material much as sculpting with real clay; they can walk around the sculpture to obtain the best angle with which to work; or position the sculpture in a convenient position; and furthermore, the fibrous nature connotes that the entire material is made of tangled weaves of paths in a 3D space. In a similar fashion, users can progressively interact with a dataset, sculpting selections from appropriate views which can help users gain valuable insights into the data.

While our specific implementation of FiberClay uses a Samsung Odyssey virtual reality headset and a pair of hand-held controllers, the overall techniques it uses are applicable in a variety of contexts. The headset, composed of two screens, each facing one eye, enables binocular depth perception (*e.g.* enables the viewers to perceive digital objects in 3D). It is also equipped with motion sensors to detect changes in position of its wearer and thus enable a user to navigate in the virtual environment by physically moving in space. The pair of controllers, one in each hand, enables bi-manual interaction with 3D objects displayed. Note that the shape of controllers allows the viewer to comfortably hold them in each hand while moving in space all the while accessing joysticks and buttons using minimal finger movements.

4.1 Multidimensional Trajectory Visualization

FiberClay supports the rendering of element-based plots. As such, a trajectory set is a dataset $D = (N, E)$ with nodes $n \in N$ and edges $e \in E$. Both nodes and edges can have data attributes, thereby making D a multidimensional dataset. A 3D trajectory drawing is a visualization $V(D) = (V(N), V(E))$, where $V(D)$ is typically a 3D lineplot of line $V(n)|n \in N$ and $V(E)$ is a set of straight line segments or points $V(e)|e \in E$. To address **DC1**, FiberClay visualizes multidimensional data composed of 3D trajectories associated with abstract data attributes in an immersive environment. Given the inherent match in the number of dimensions being explored, a 3D immersive environment is particularly appropriate: FiberClay represents triplets of data values as series of x, y, z positions in digital space, thus displaying sets of curves (polylines) in three dimensions. For instance, a dataset which contains recorded aircraft trajectories has many dimensions that include the location of the moving aircraft (latitude, longitude and altitude), the recorded time stamp, the aircraft identifier (i.e. unique value to discriminate a trajectory), the airline, and possibly many others data characteristics such as speed and acceleration. The standard visual mapping for aircraft trajectory is to assign the x,y and z respectively to the latitude, longitude and altitude data dimensions. Other mappings (e.g. assigning different z coordinates to the type of carrier instead of the altitude which would lead to different layers of 2D trajectories per carrier) are possible as explained in section 4.4.

Trajectory representation: FiberClay supports multiple styles of representation for a trajectory with lines, points or animated particles (figure 2). While the line representation makes sense to display trajectories, the point representation is preferable to display non continuous data like the airline carrier. In the animated particle mode, the user can investigate flows and their directions [22, 39]. Each particle corresponds to an instance of a moving object, but many instances of the same moving object are visible at the same time (typically 10 instances per trajectory).

Trajectory color blending: In FiberClay the user can also map a data dimension to color. For instance, aircraft trajectories can change color with respect to the altitude dimension (e.g. a color gradient from green to blue to correspond with low to high altitudes, or based on the time along a trajectory to create a gradient from the start to the end of a trajectory). Since our system displays many sets of trajectories, they can intermix and occlude each other. To help address this problem (**DC2**), FiberClay supports three modes of color blending modes, with the ability to switch between them at interaction time: transparent, additive, solid (figure 2). The transparent mode displays trajectories as semi-transparent curves, thus enabling viewers to identify dense areas (i.e. where many trajectories overlap) as they become visually

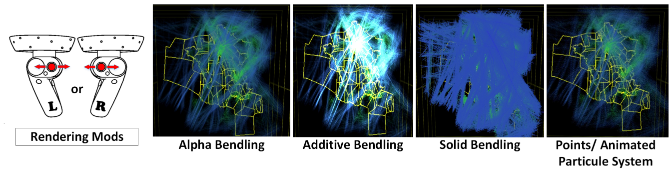

Fig. 2. 3D rendering. The left and right trigger on the controller joystick switches between the following mods: transparent, additive, solid and particle system rendering.

more salient as many trajectories overlap. The additive mode also displays transparent trajectories but dense areas show as a white solid color resulting from the additive blending. Finally, the solid rendering mode displays opaque trajectories which may make it easier to follow a specific trajectory, although occlusion can still make it difficult to display a large number of trajectories at once.

4.2 Navigation

Efficient navigation within these dense trajectory sets is particularly important since different point of views are necessary both to better understand the data (**DC2**) and to select items to filter them out (**DC3**). One distinct advantage of head-tracking in immersive environments is that movements of the head naturally change the displayed point of view within the virtual environment. While the user can also physically move in space, there are many times where the user may wish to stay in a fixed position while manipulating the view itself. It may be more comfortable to remain seated or in a fixed position to avoid fatigue by constantly moving back and forth. Furthermore, the position of interesting features within the environment may necessitate movement to awkward positions or obstacles within the physical environment may prevent users from moving to the appropriate location. To address all these issues, in addition to head movements, the user can use the hand controllers to change the view, translating, rotating, or scaling it to convenient sizes and locations.

Translation : The user can translate the entire dataset using a *grab* metaphor. The user reaches into the visualization and presses the right controller grip button. While keeping the grip button pressed, the user can move the controller in any direction and the data moves accordingly. The translation interaction stops when the user releases the grip button.

Rotation : The user can rotate the data set along a vertical axis by pressing the left and right grip buttons, and then pulling or pushing on the left or right controllers. The amount of rotation is dependent on whether the interactions take place inside or outside the bounding box of the data. If the user is far away from the data (outside of the bounding box), the rotation axis is placed at the center of the data, making it easy to spin the data to alternate viewpoints. Once within the bounding set, the rotation axis is located 50cm in front of the user. The helps in smaller refinements of the viewpoint as items closer to the axis will have reduced movements.

Scaling : The user can scale the entire dataset larger or smaller by pressing the left grip controller button and moving the right controller relative to it. The scale varies according to the relative distance between the left and the right controller positions. The system maintains a congruency between the controller movement and the axis orientation.

4.3 Brushing of 3D displayed trajectories

A fundamental component of FiberClay is the ability to create complex visual queries by progressively refining a selection of trajectories (**DC3**, **DC4**). To create a selection, a ray is activated by pressing the trigger button (figure 3) which results in a selection ray being attached to the hand-held controllers. When the ray intersects a segment of a trajectory, the entire trajectory is selected and turn to solid white color. We empirically defined a radius tolerance parameter defined to be 2 centimeters. We turn sections of trajectories red when they fall into the beam tube to help users better understand how their interactions

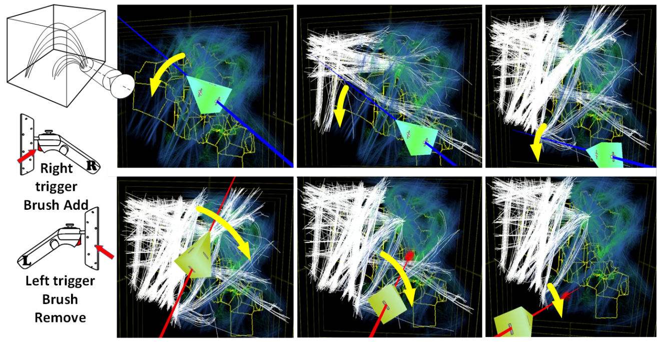

Fig. 3. In the Add/Remove Brush, the user can select trajectories with the right hand-held controller. Each trajectory is brushed when it crosses the ray. Similarly, the user can remove a selection by using a unselect ray from the left controller.

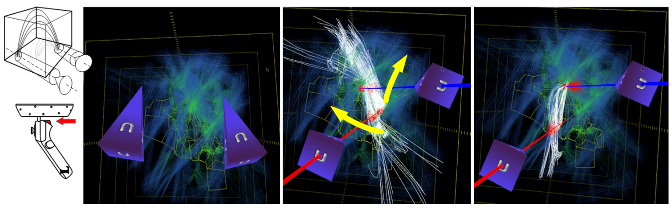

Fig. 4. In the intersection mode, the user can use this bi-manual technique to select trajectories which intersect both the left and the right controller rays. This technique can be used to refine trajectories when switching from the Add/remove brush mode and removing unwanted trajectories (DC4).

generated selections. Our system provides two brushing modes which can be combined to interactively sculpt user queries.

Add and remove brush: This most fundamental brushing paradigm is to add or remove trajectories to the current selection. Pressing the trigger button on the right controller and moving the beam adds trajectories to the selection, while pressing the same button on the left controller and moving its beam subtracts items from the selection (figure 3).

Intersection brush: A more advanced selection paradigm allows the user to isolate only those trajectories that are touched simultaneously by both the right and left beams. This makes it easy to select trajectories that start in a certain specified location and end in another specified location (figure 4). This trajectory selection technique requires bi-manual interaction.

4.4 Changing Visual Mappings

As noted in design challenge **DC3** we need to have convenient ways of changing both the view and visual data mappings in order to facilitate selection. We do so by pre-generating a set of view positions and data-mappings and can navigate between them naturally within the virtual environment without any discontinuities.

Lets consider a trajectory dataset $D = \{d_i\}$, $1 \le i \le N$, containing n-dimensional elements $d_i \subset \mathbb{R}^n$. We note $P = \{p_i\} \in \mathscr{P}$ the set of parameters that controls a visualization and $V(P,D) \in \mathscr{V}$ the resulting view for the given dataset D. The exploration of such a dataset requires finding a relevant V defined by a preset P (DC1).

Many techniques have been developed to navigate between presets on a 2D screen. Scatterdice [14] and graphdice [7] represent presets in a matrix scatterplot and help user to navigate between them. FromDady [30] uses the same principle with large trajectory set explorations. Color tunneling [29] addresses scalability issue and provides a general framework for dual preset navigation. More recently, Kruiger et al. [32] provide a grid with user defined presets where the user can navigate and transition between them. All these preset navigation techniques have shown their efficiency, but none have been investigated in the context of an immersive environment where depth perception can play a significant role in the data exploration process.

4.4.1 Small multiple grid computation

In FiberClay , we developed an extended 3D version of the Inverse Distance Weighting (IDW) interpolation using Shepard's method [41] to provide a continuous preset navigation process.

This computation is given using equation (1). Considering $x \in \mathbb{R}^2$ a point in the 2D preset grid space, we compute the elements $V(x,d_j)$ of the interpolated view at position x as:

$$V(x,d_j) = \begin{cases} \dfrac{\sum_{i=1}^{N} w_i(\mathbf{x}) V_i(d_j)}{\sum_{i=1}^{N} w_i(\mathbf{x})}, & \text{if } \|\mathbf{x}-\mathbf{x}_i\| \neq 0 \text{ for all } i \\ V_i(d_j), & \text{if } \|\mathbf{x}-\mathbf{x}_i\| = 0 \text{ for some } i \end{cases} \tag{1}$$

x_i are the centers of the preset V_i in the grid. $\|\cdot\|$ is the 2D Euclidean distance, and $w_i(\mathbf{x}) = \frac{1}{\|\mathbf{x}-\mathbf{x}_i\|^p}$ the interpolating basis function controlled by the power parameter $p > 1$ (we use $p = 2$ leading to a classical inverse quadratic Shepard interpolation).

This preset exploration can produce an infinite number of view spaces $\mathscr{V}$ between provided presets following the original preset controller principle [47]. In FiberClay , up to five predefined presets are available and are placed on a grid where the system will display thumbnails of the intermediate steps (Figure 5) to give a better idea of the possible transitions. These presets are customized for each dataset.

4.4.2 Small multiples navigation

In order to choose the displayed visual configuration, the user presses the left touch-pad joystick (figure 5). The small multiple grid is subsequently displayed where a rotating *cross* shows the current visual interpolated configuration. The movement of the left controller is mapped to the *cross* movements which enables navigation through the interpolation space. Releasing the touch-pad button stops the transition and maintains the currently displayed visual configuration. While this small multiple navigation provides a smooth transition between an infinite space of visual configurations, the user has to look at the small multiple grid during this transition to accurately position the *cross*. The *cross* movements are recorded and can be replayed when pressing the right touch-pad joystick and moving the controllers closer or further apart. This replay interaction helps the user investigate details of the transitions while focusing on the 3D display rather than on the visual configuration grid.

4.5 Technical considerations

We used the Samsung Odyssey headmounted display as our immersive display. The software was built in C# using the DirectX graphical API. We tested our system with various datasets of up to 10 million records. Our system maintains a frame rate of 90 Hz with a screen resolution of 1440x1600 pixels running on a laptop equipped of a Nvidia 1080 GTX graphic card with 8 Go graphical memory on a Intel Core i7-7820HK 2.9 Ghz.

To implement our system we had to address data scalability issue, where large datasets must be displayed while allowing the user to interactively explore and select parts of it (**DC5**). We made sure that interaction (e.g. brushing, navigation) will not affect the frame rate.

Rendering response time: Using VR technology can necessitate a great deal of computational power to display data since we need high resolution, high frame rates to avoid simulator sickness (when the display lags relative to user movement), and two views (one view rendered for each eye to achieve binocular disparity). Our device can render images at a maximum frame rate of 90 frames per second. In order to maintain this high frame rate, we had to develop our own rendering engine. Other solutions like Unity, or the Unreal engine are not optimized for data visualization and cannot ensure such a high frame rate with large datasets of potentially changing data. We implemented

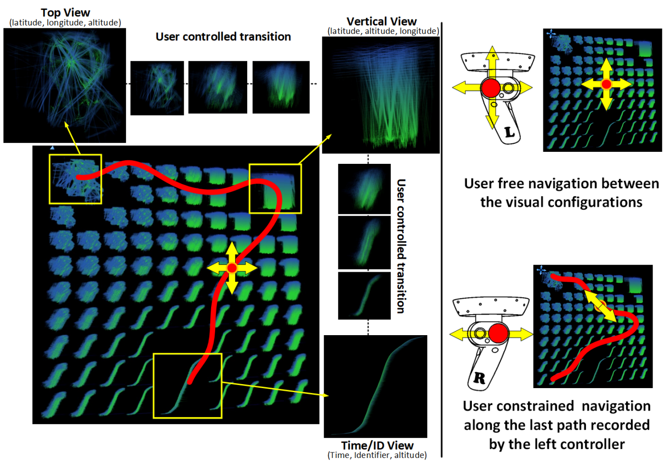

Fig. 5. This image shows our preset selection navigation principle [47]. The user can navigate between the preset by pressing the left touch pad button and then moving the controller in any direction to navigate through the presets. The yellow cross show the current display interpolated preset thanks to our 3D implementation of the Inverse Distance Weighting (IDW) interpolation [41]. The user can replay the last transition by pressing the right touch pad button and moving the two controller apart.

a standard visualization pipeline using VBO (Vertex Buffer Objects) to store the data, and vertex, geometry and pixel shaders to render them onto the device.

Interactive response time: The system must be as responsive as possible to ensure usability. Significant latencies can drastically impede human's ability to interact with the scene. This is an additional challenge since the computation time for user interactions needs to be combined with the rendering time and thus potentially create a degraded user experience. In our system, the frame rate is maintained at 90 Hz even while the user performs brushing interactions. We tested our system with large datasets of up to 20 millions records. Our only limitation is the available memory in the graphic card since the entire dataset needs to fit into it. We use advanced GPGPU techniques to render and to brush displayed items. To achieved a complexity of O(n) where n is the number of records, we used the instantiation technique where data is displayed while any computations are performed (e.g. the brushing or visual configuration interpolation). The rendering pipeline computes the display of every item location at every frame. Since we also use this location to detect if an item is selected, we can use a vertex shader to compute if a selection beam is close to the item to be rendered. With a multiple render target technique (2 textures, one for each eye and two brushing texture, one for each wand), the system simultaneiously computes the image to be displayed and the brushed items. This computation and rendering technique is inspired by the image based techniques [29] and extended to multiple render targets.

5 Use Cases

To observe how FiberClay can be used in practice, we collaborated with different domain experts (air traffic controllers, engineers and medical doctors) conducting collaborative sessions where we demonstrated the tool and they used it to explore their data. Over 20 experts tried the system (only two with previous experience in interacting in a virtual environment), successfully composing queries after 15 minutes of guidance and experimentation. While we did not conduct a formal user study, we did observe experts extract insights during these explorations. We describe the most interesting patterns and interactions they used to find them in this section. Note that it is particularly challenging to illustrate 3D patterns as 2D images in this paper, as well as interactions and their impact in three dimensions. Thus we strongly encourage reviewers to watch our accompanying video [1].

5.1 Unique Patterns Revealed in Air Traffic Data

The first use case we present is our own exploration of an existing dataset. It is interesting to note that we had extensively explored this dataset in the past using the FromDady system. Yet, using FiberClay we found unique three-dimensional patterns that we had not identified previously (figure 6).

Dataset: This dataset contains 22,720 trajectories recorded over France during one day (Friday, the 22nd of February, 2008) consisting of 3,914,905 records (after re-sampling). Each aircraft has a recorded position (latitude, longitude, altitude) every 3-4 minutes and an associated unique numerical identifier.

Findings: By visualizing this dataset in FiberClay and interacting with it to reproduce prior insights discovered with FromDady, we also were surprised to discover three novel patterns:

1. A *crenelated trajectory* alternating between two locations (figure 6), indicating that the aircraft suddenly jumped and dropped multiple times during flight. Upon closer examination of the data, we realized this pattern is caused by an error in the data collection process, in which points from two different trajectories were merged. Dissociating these trajectories led to two aircrafts following the same route at different altitudes. This pattern is particularly difficult to identify in sets of 2D projections as (longitude by altitude) show a single route, and (time by altitude) shows multiple crossings between two altitudes, which could, in effect, be two different trajectories.

2. A *looping trajectory*, resembling a vertical square indicating that the aircraft would have performed a series of loops in the sky. As we could not explain this rather uncommon behavior for an aircraft, we consulted with three air traffic controllers experts. Upon investigation they concluded that this specific aircraft was taking photographs of a specific area of land at different altitudes (to better assess the differences in size of objects in the ground), thus flying back and forth over the region at different altitudes during each segment. Similar to the first pattern, this problem was difficult to discover from multiple 2D views where lot of occlusion occurred.

3. A *star-shaped trajectory*, which upon investigation is caused by an artifact of the data sampling. Indeed, aircraft location is provided every three to four minutes. If an aircraft is executing holding patterns, waiting for a window to land for example, this low sampling rate leads to this particular star shape. In figure 6), caption "star Shape", we compare the extracting trajectory in our system and with FromDaDy [30]. This shape is small and difficult to detect on a 2D screen while it easier to find in an immersive environment. The combination of a large field of view and the flexibility to quickly change the point of view helps to detect it.

5.2 Traffic analytics

The second use case we present also deals with air traffic data (see Figure 8). We present insights gathered from four experienced air traffic controllers (3 Male, 1 Female, Age from 31 to 53), with ten to twenty years of experience in the profession and no previous experience with immersive devices.

DataSets: We used the air traffic dataset described above as well as a more recent dataset of flight over the world, containing one week of recorded aircraft trajectories between the 1st and the 7th of January 2018. This data is extracted from the on-board ADSB system emitted by every aircraft, collected by the OpenSky Network [38]. It is composed of 55,557 flights with 1,787,000 records. Each record contains the aircraft unique identifier, its location (latitude, longitude, altitude), its speed and its direction.

Findings: In this section, we report insights gained by our collaborators using FiberClayand describe interactions they performed to discover or further investigate them.

1. *A vase-shaped traffic pattern around Paris.* As noted by our collaborators, one of the most challenging area to analyze is the dense air traffic around Paris and its three different airports, including Charles De Gaulle, a key hub in Europe. Since it is forbidden to fly over Paris,

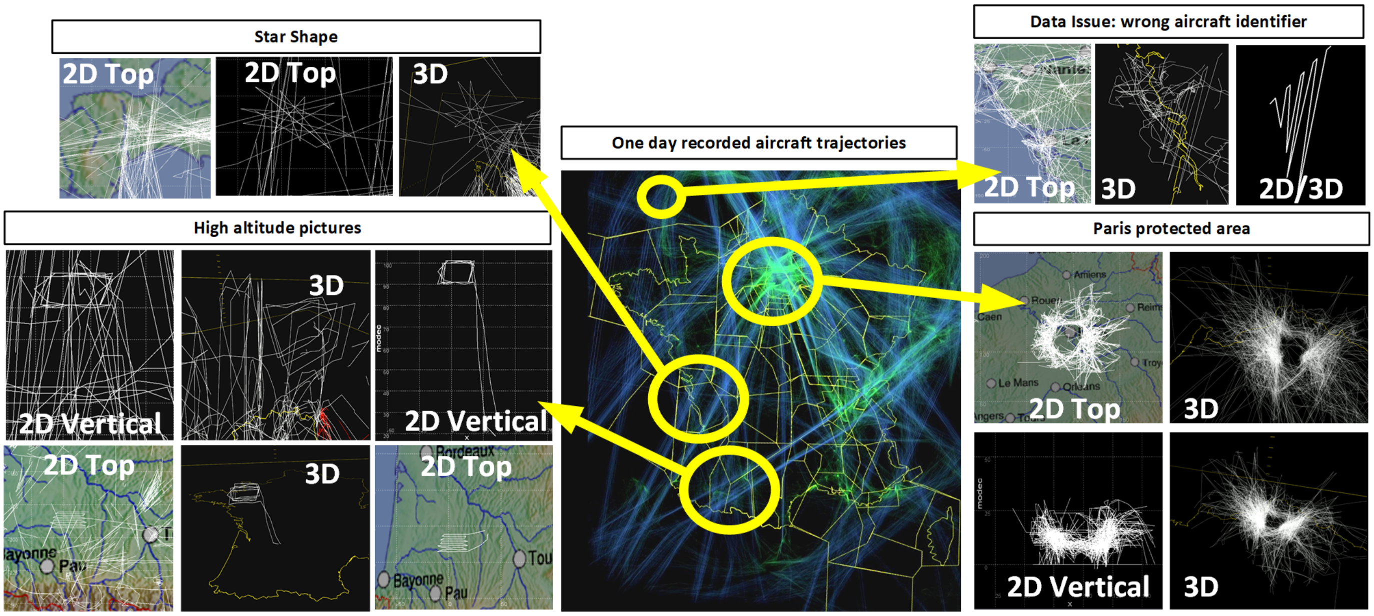

Fig. 6. This set of screen captures shows how our system helped the user to visually explore our trajectory dataset. 3D images have been produced by our immersive system, while 2D images have been produce by an 2D trajectory visualization tool [30].The center image gives an overview of one day of recorded aircraft trajectories over France. The user can explore this data set using the grasp and rotation interaction. Furthermore, user can use the two brushing modes (additive/subtractive and intersection) to select and highlight specific trails. As such the user detected wait patterns in aircraft which create star shape trails. Many aircraft went around Paris to bypass the forbidden flyover city area and created a 3D vase shape. Data fusion errors creates jumping trajectories. One aircraft did vertical square shapes during a high altitude picture shooting.

the controllers were not surprised by the empty inner space and trajectories appearing to circle over the region to avoid it. However, as one controller physically moved in space to look at the 3D trajectories from multiple angles, he reported an interesting volumetric pattern roughly in the shape of a vase. He explained that this pattern was composed of two types of trajectories: low-altitude aircrafts aiming at landing in one of the airports close to the border of the forbidden area, and high-altitude aircrafts bypassing Paris on their way to different regions, and thus further away from the center. The controller was able to isolate the different sets of trajectories he was interested in by using FiberClay brushes to compose complex queries. For example, by placing the first beam on the left side of Paris and the second one on its right side, he isolated trajectories where aircrafts were bypassing the area. Changing his point of view, he then removed low-altitude flights from his selection, effectively including only aircrafts that were bypassing the region without intending to land. figure 6 shows the extracted trails and its comparison with FomDaDy visualization system. Only stereoscopic system like FiberClaycan make this vase shape emerge.

2. *Flight route complexity:* In figure 8, an air traffic controller used our system to identify airspace where air traffic could be optimized to be more fuel-efficient. She completed a similar query several times in different locations in order to find the patterns she wanted to showcase. Each time she proceeded by placing beams in two different geographic locations to isolate a specific air traffic flow. She then changed the point of view by moving using the small multiples enabling her refine her selection by altitude levels. She then used the removal brush to deselect very low altitude aircrafts, revealing small gradual changes of high-altitude flights. She moved back to her previous position and scaled her selection to see with greater precision how aircrafts changed altitude. After a couple of queries in different locations, she pointed out a traffic flow where aircrafts had inconsistent staircase-shaped climbing patterns (figure 8). She explained that for safety reasons, aircrafts must remain at a given distance from each other, thus, as one aircraft changes altitude, it may slightly impact other aircrafts routes or their altitudes which can rapidly cause a chain reaction. Thus, optimizing when each aircraft should adjust its altitude is important as it could lead to less fuel consumption. However, this task is extremely hard to perform without the 3D view of the data as the task involves to study a set of 3D trajectories (latitude, longitude and altitude), and understanding the distance between each of their points. We observed that it required the controller to almost constantly move her head in space to understand this complex geometries.

5.3 Wind extraction

This section provides another type of usage of our system. While we previously demonstrated how FiberClay facilitates in the exploration of geo-spatial temporal data, in this section we look and exploring additional multidimensional attributes of a geo-spatial dataset.

DataSets: The dataset for this scenarios is based on one week of world wide aircraft trajectories. As shown in figure 9, many areas are not covered by aircraft trajectories such as over oceans and seas. This dataset only shows trajectories where ADSB ground stations captured aircraft positions in their vicinity (additionally, there are no ADS-B receptors installed in the desert, mountains or unpopulated areas). Each record contains the 3D location of the recorded aircraft, its time and its direction and speed. The coupling of aircraft direction and speed is of interest since it has been previously shown that it codes for the actual wind faced by an aircraft [24]. Aircrafts flying directly into the wind have a lower speed that aircrafts flying with the wind. The wind triangle principle shows that a sinusoidal relationship exists between the aircraft direction and its speed.

Findings: Thanks to the ability of FiberClay to smoothly transition between mappings of different visual attributes, the user can go back and forth between the world view (i.e. latitude, longitude) to a *wind view* (aircraft direction and speed). The wind view shows the specific sinusoidal wave. When transitioning between the two views (world and wind view), the user can follow the curve transformation and figure out that this shape mainly corresponds with aircraft in Europe and the US. European and US flights have the majority of ADS-B sensors which partially explain the data density. The second reason is due to the jet stream, a very strong high altitude wind between US and Europe. This shape results from the impact of this wind on aircraft speed. Aircraft flying east are flying faster than aircraft flying west. While this is

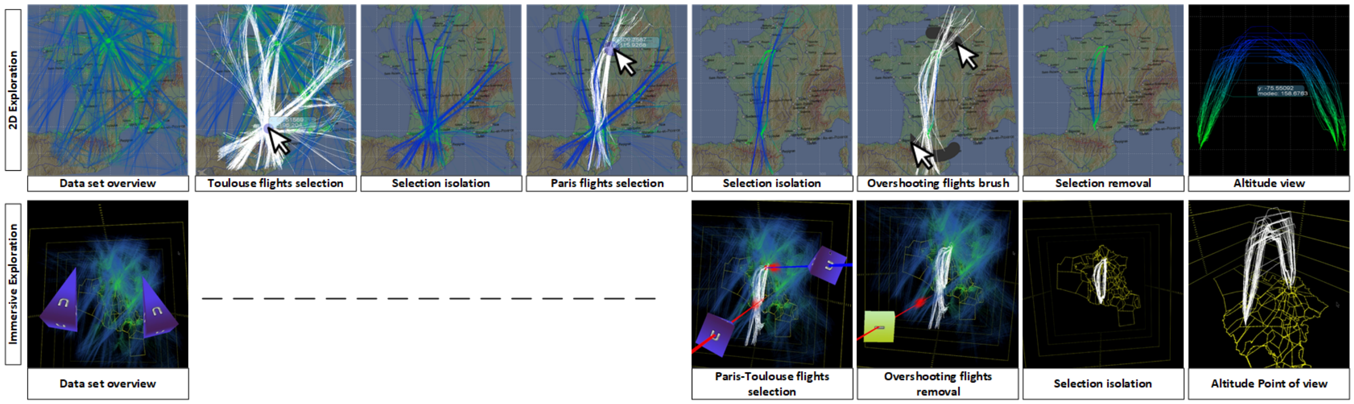

Fig. 7. These images shows the comparison between standard trail exploration system and our tool. In this example, the user wanted to explore the fights between Toulouse and Paris, two major cities in France. While standard tools use brushing and linking technique, our system uses a bi-manual interaction plus the aspects of the immersive environment such as depth perception and head movements (i.e. intrinsic user view modification due to head movements). As a result, fewer user actions were required to fulfill the same task task in the immersive environment as compared to standard interactive technique [30].

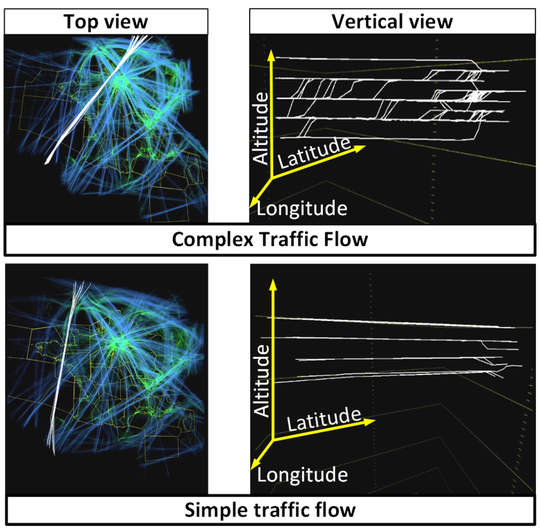

Fig. 8. These screen shots shows two different types of traffic complexity. Simple traffic shows aircraft which did not change their altitude and followed their route. Complex traffic patterns show many aircraft which changed their altitude and thus created many crossing lines.

the case over the US and Europe, wind speeds and directions are not uniform around the globe. When transitioning between the world and wind views, the user can notice many secondary wave shapes which are occluded by the main jet stream wave. These intermediate animated steps help locate their origin. With interactive brushing, the user can select aircraft around the equator and highlight them. This selection helps the secondary wave shape emerge more clearly which has an opposite phase shift in comparison with the jet stream wave. This shows that the wind around the equator has an opposite direction than the jet stream and it also three times weaker. As a final step the user can display the evolution over time of wind impact by using the z axis to map the time. Depth perception (via both binocular disparity and motion parallax) helps to see that the equatorial wind was lower at the beginning of the week, then increased and finally decreased.

This scenario does not necessarily provide new insights into the data, since the jet stream and other wind parameters are well-known, but the scenario does illustrate some of the novel potentials of our data immersive system. The continuous view transformations and the ability to make and refine selections at any point during that transition can help the user detect emerging patterns that would be difficult to see in more standard 2D systems.

5.4 DTI extractions of neuronal fibers

The last use case scenario shows that FiberClay can be applied to completely different types of trajectories than objects moving through a volume over time. In this case, we explore neuron fiber tracks from different regions of a brain visualized as streamlines of the major eigenvector of a diffusion tensor imaging (DTI) field. Such datasets show a spatially complex structure which makes them hard to explore [3]. We validated this scenario with a professor at the hospital who specializes in brain diseases.

Dataset: This dataset is extracted from a $128 \times 128 \times 51$ DTI volume (dataset from [17]). We extracted 150352 fibers from high fractional anisotropy areas and filtered out fibers shorter than 2mm which results in 120,593 displayed fibers (1,998,652 sample points). Each record contains the 3D location of the sample and the numerical identifier of the fiber.

Findings: Compared to previous work dealing with fiber selection [42], our system offers real-time rendering where the user can interactively sculpt queries to isolate specific fiber subsets and better investigate them. In figure 10, we explore the original dataset and the result of two different bundling algorithms (KDEEB [27] and FDBEB [28]). The user can navigate between the original 3D fibers and their two different simplifications (see companion video). Furthermore, the user can select a fiber subset and explore how these simplification algorithms operated. The gray fibers in figure 10 shows one original data subset and two visual simplifications. The KDEEB simplification shows many distortions while the FDBEB one shows smooth aggregated lines. This example shows how the user can choose appropriate edge bundling techniques and associated parameters. KDEEB shows more details while FDBEB shows greater simplification. Our system can interpolate between both simplifications so that the appropriate view for a particular selection can be used.

To achieve smooth animation betweens different edge bundling technique, we augment the original dataset with a 3D position for every fiber point in each simplification technique, thus we have the original position of the point, its position based on KEDDB, and one based on FDBEB algorithm. The user can navigate between the different data dimensions using the standard, small multiple interaction previously discussed. Compared to previous work, we are not aware of any previous techniques dealing with animated transitions to visualize multidimensional dataset. This example shows the potential of this style of interaction.

6 Discussion

In this section we give additional details and research perspectives of our work compared to existing techniques and we discuss feedback we collected from the users who tested our system.

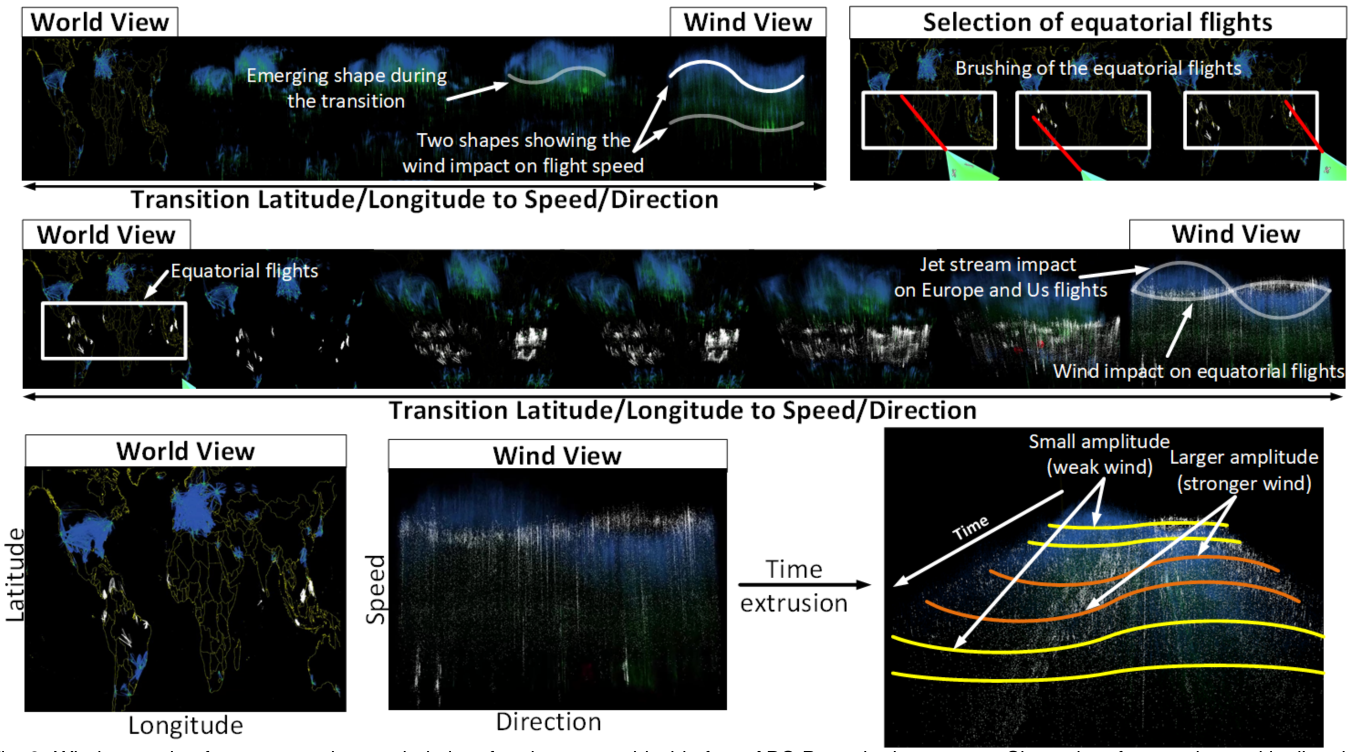

Fig. 9. Wind extraction from one week recorded aircraft trajectory world wide from ADS-B monitoring system. Since aircraft ground speed is directly affected by the wind, aircraft flying with the wind fly faster than aircraft facing the wind. Using FiberClay, the user can select specific aircraft and show different wind parameters around the world. Equatorial wind is opposite in direction than the jet stream affecting aircraft in the US and Europe. Finally the time visualization shows how this wind evolved other time. The equatorial wind was weaker at the beginning and end of our recording.

While four experienced air traffic controllers helped us to validate our data exploration scenario, we tested this system with more than 20 additional users from various domains: air traffic controllers, medical experts, computer scientists, and managers.

6.1 Comparison with existing techniques

Many multidimensional data exploration and visualization techniques already exist. Most of those that are dedicated to exploring trajectories are limited to interaction on 2D display devices. To the best of our knowledge, no previous work has investigated immersive devices to display and explore trajectory datasets. This work tries to fill that gap by providing examples of efficient interaction techniques on immersive displays.

Figure 7 compares the extraction of specific flights between airports using the FromDaDy system [30] (a trajectory and multidimensional data exploration tool using 2D displays) and our system. This comparison shows how our system reduces the number of interactions the user has to perform to fulfill fundamental data selection tasks. The immersive, bi-manual interaction techniques used in FiberClay compare favorably to using a single mouse pointer on a 2D screen. The ability in the immersive environment to directly change the camera position with head movements, and smoothly use two controllers to refine selections keeps mode switching necessitated by a single mouse pointer to a minimum. This interaction is enhanced by current improvements in resolution and field of view of devices making and binocular disparity can help improve depth perception as well.

The combination of the immersive display characteristics, bi-manual interaction for selection refinement, and natural changes of view afforded by head movements (and our enhancements to enable natural changes data attribute mapping as well) are formidable assets for 3D dataset exploration. For instance, in this paper, we have investigated the "France trajectory" dataset several times previously explore in many previous works [15, 24–30, 39]. Even though this dataset has already been intensively explored, our system helped us to discover many additional features especially with respect to insights about data errors, emerging trajectory shapes and dimension correlation. These all help with a more thorough understanding of the underlying data. While proving that this system provides quantitative performance improvements beyond 2D systems is beyond the scope of this paper, the results of the case studies above were encouraging.

6.2 User Feedback

During our investigation and software development, we regularly tested our system with end-users. Computer scientists, engineers, medical doctors, and air traffic controllers all tried using the system. In total, more than 20 users participated in this process where only 2 had previous experience in virtual environments or immersive technology. A systematic investigation of user performance is not the primary purpose of this paper and will be done in a future work, but we can here summarize some of our findings. Of the 20 users, two users had to stop because they did not feel comfortable after 5 minutes of manipulation (a common problem in many immersive environments). Every other user managed to interact and accomplish tasks within our system. They quickly learned the full capabilities of the system and accomplished tasks of varying degrees of difficulty. We first instructed the users by demonstrating each feature, starting with data grasping (i.e. panning), followed by rotation, scaling, intersection selection, additive and subtractive brushing, intersection brushing, and finally changing the visual configuration. Every user managed to learn the full set of buttons and their combinations. Many users followed a quick learning phase, followed by a drop in manipulation capability when trying more complex interactions). All users managed to adapt to more complex bi-manual interactions after using the system for 15 minutes. They were able to navigate the system and select a few targeted items. Further investigation is necessary to validate these findings, and quantify the performance characteristics, but every single user was able to effectively manipulate and interact with the displayed datasets. Several users commented that the grasping metaphor for panning the view of the dataset was quite natural. Rotation was a bit more difficult to master as it necessitated using the tool controllers as if a wheel was connected between the two controllers. Still, users were able to master this relatively quickly as well. Additive and subtractive selection was immediately

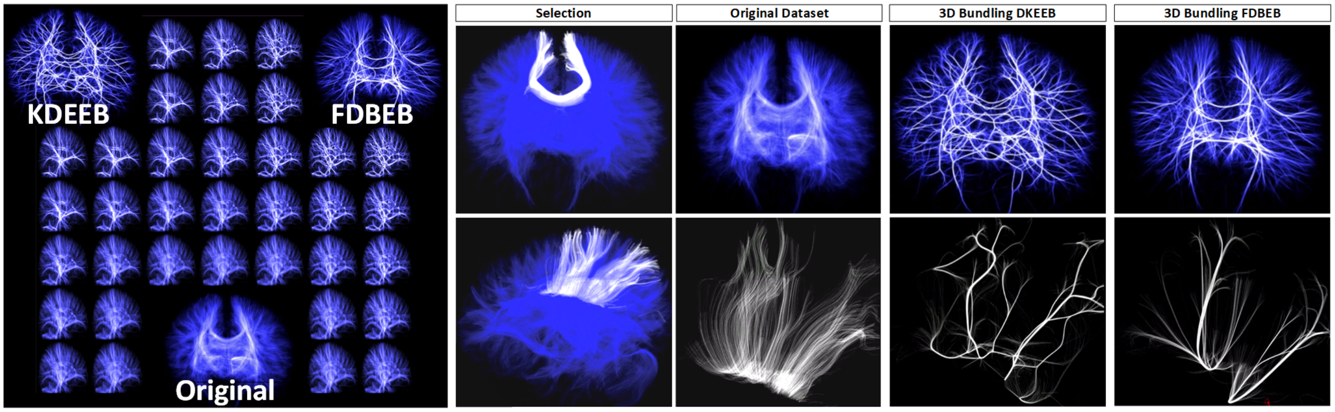

Fig. 10. Immersive visualization of DTI fiber extraction from a brain Scan. The small multiple visualization shows the original dataset and two different 3D edge bundling algorithm (an extended 3D version of KDEEB [27] and Functional Decomposition Edge Bundling [28]). Our system helps to select a subset of the 3D fibers with our 3D brushing techniques (KDEEB and FDBEB). Then the user can investigate how this subset is bundled through the two different 3D edge bundling methods. As a result, FDBEB proposed less distortion compared to KDEEB but with less visual simplification. Our system help to define the correct balance between edge bundling simplification and distortion in a 3D space.

grasped, and this in combination with view manipulation allowed users to quickly accomplish the given tasks. Users were impressed with how rapid the system responded to selections and were excited about the new insights they were able to gain in looking over the data. Finally, navigation between the visual configurations was the most complicated and was not readily explained by any natural metaphor in the real world (grasping or wheel turning). However the visual feedback from the cursor on the small multiples helped users to understand the process. Using the left controller to navigate freely between available configurations and the right controller to replay or control the distance along the transition was again mastered within the 15 minute instructional period.

6.3 Guiding principles

Our previous experiences in working in virtual environments and iterative design techniques helped inform our design of the system. While thorough validation of those principles are beyond the scope of this paper, it's appropriate to reflect on some of those ideas here.

No standards for immersive User Interfaces: Immersive environments have been studied for some time, yet no standard has emerged for interaction. While conventional WIMP techniques (windows, icon, mouse, pointers) such as buttons, menus and list can be displayed, it is not clear how efficient they are when used within an immersive environment in comparison to 2D interfaces. In general, where possible, we eliminated standard 2D interface techniques and relied on mapping actions to controls on the controller devices or natural movement within the space. While this can add some learning and memory burden, we found that the actions were surprisingly easy to learn and remember by our users.

Eliminate modes when possible: Since one of the benefits of the system was seen as the interleaving of view manipulation with selection modification, creating a modal barrier between these interactions would significantly slow user performance. This went hand in hand (so to speak) with mapping actions directly to the buttons for the controllers (4 buttons and 2 joysticks per hand). The resultant efficiency in interaction was one of the primary strengths of the system.

Make view navigation easy: Since the main asset of immersive device is the user's ability to change the point of view, special care has to be taken with respond to data navigation. In our system, the user can freely move around and into the data, but many users desired to remain seated when exploring the data. In this case, additional navigation techniques were required to move the data and rotate it. While the pan (in our case the data grasping) and its metaphor were easy to master by the user, our initial implementation of data rotation needed additional refinements. We first constrained rotation to only be around the vertical axis, and we subsequently modified the magnitude of rotation depending on user's position with respect to the data. We found that this allowed for both rapid, overall changes of the view as well as smaller refinements for exploring specific features.

Judicious use and display of presets: By pre-calculating different combinations of view data mappings, and embedding 2D representations of those mappings on the ground, we found that we could enable convenient re-mapping of data-attributes while exploring the data. We used the principle of small multiples with smooth animated transitions between them to help visually link items between different visual mappings. While our IDW (Inverse Distance Transform) interpolation may not be the provably optimal transition, it helps to navigate between visual configurations and link visual items across views without any discontinuities.

One primary view: Our current system only offers one primary visualization of the data. While technically, we can easily extend it to multiple views, the added complexity of trying to portray multiple linked views, make for natural movements around the views, and convey linking between the views convinced us to focus first on creating an effective system with one primary view and leave exploration of multiple, linked views for future work.

7 Conclusion

In this paper, we report our investigations to support the exploration of multidimensional trajectory datasets in an immersive environment. This work shows how we designed interaction techniques to effectively select and to explore data subsets. Thanks to a bi-manual intersection brushing technique, the user can efficiently select trails, refine that selection by adding or subtracting trails from the selection set. This selection can be done while navigating in the 3D space to find a suitable view point for both selection and trail comprehension. We also provide a smooth transition technique to navigate between different visual configurations (i.e. data attribute mapping). The user can define the navigation path through a predefined set of visual configurations and control the transition to help better understand the relationships between visual items. Our system is scalable and is capable of managing a large dataset limited only by what fits into the GPU memory. To demonstrate the utility and efficiency of our system, we show four concrete examples of usage though different application domain scenarios: anomaly detection, trajectory analysis, wind extraction, and investigation of fiber track simplification algorithms. Finally we discuss our findings through qualitative user feedback and research perspectives.

As future work, we plan to assess and extend our work in many directions. Various application domains where trajectory analysis play an important role can take advantage of our system: stream lines for flow visualization, network analysis, tensor flow investigations. Finally, additional UI refinements and directions have been suggested by users including convenient undo/redo to help users return to previous selection states, and multi-user collaboration techniques where different users might simultaneously interact with the same dataset.

REFERENCES

[1] Companion video. https://www.youtube.com/watch?v=8j9r9JKIqww&feature=youtu.be. Accessed: 2018-03-31.

[2] C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *The Craft of Information Visualization*, pp. 7–13. Elsevier, 2003.

[3] Y. Assaf and O. Pasternak. Diffusion tensor imaging (dti)-based white matter mapping in brain research: A review. *Journal of Molecular Neuroscience*, 34(1):51–61, 2008. doi: 10.1007/s12031-007-0029-0

[4] B. Bach, M. Cordeil, T. Dwyer, B. Lee, B. Saket, A. Endert, C. Collins, and S. Carpendale. Immersive analytics: Exploring future visualization and interaction technologies for data analytics.

[5] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A descriptive framework for temporal data visualizations based on generalized spacetime cubes. *Computer Graphics Forum*, 36(6):36–61. doi: 10.1111/cgf.12804

[6] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[7] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. D. Fekete. Graphdice: A system for exploring multivariate social networks. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'10, pp. 863–872. The Eurographs Association & John Wiley & Sons, Ltd., Chichester, UK, 2010. doi: 10.1111/j.1467-8659.2009.01687.x

[8] M. Borkin, K. Gajos, A. Peters, D. Mitsouras, S. Melchionna, F. Rybicki, C. Feldman, and H. Pfister. Evaluation of artery visualizations for heart disease diagnosis. *IEEE transactions on visualization and computer graphics*, 17(12):2479–2488, 2011.

[9] R. Brath. 3d infovis is here to stay: Deal with it. In *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, pp. 25–31, Nov 2014. doi: 10.1109/3DVis.2014.7160096

[10] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on*, pp. 156–163. IEEE, 1991.

[11] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive analytics. In *2015 Big Data Visual Analytics (BDVA)*, pp. 1–8, Sept 2015. doi: 10.1109/BDVA.2015.7314296

[12] M. Cordeil, A. Cunningham, T. Dwyer, B. H. Thomas, and K. Marriott. 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, pp. 71–83. ACM, New York, NY, USA, 2017. doi: 10.1145/3126594.3126613

[13] T. Dwyer, N. H. Riche, K. Klein, W. Stuerzlinger, and B. Thomas. Immersive Analytics (Dagstuhl Seminar 16231). *Dagstuhl Reports*, 6(6):1–9, 2016. doi: 10.4230/DagRep.6.6.1

[14] N. Elmqvist, P. Dragicevic, and J. D. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, Nov 2008. doi: 10.1109/TVCG.2008.153

[15] O. Ersoy, C. Hurter, F. Paulovich, G. Cantareiro, and A. Telea. Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2364–2373, Dec 2011. doi: 10.1109/TVCG.2011.233

[16] M. Erwig and M. Schneider. *Query-by-Trace: Visual Predicate Specification in Spatio-Temporal Databases*, pp. 199–218. Springer US, Boston, MA, 2000. doi: 10.1007/978-0-387-35504-7_14

[17] M. H. Everts, E. Begue, H. Bekker, J. B. T. M. Roerdink, and T. Isenberg. Exploration of the brain's white matter structure through visual abstraction and multi-scale local fiber tract contraction. *IEEE Transactions on Visualization and Computer Graphics*, 21(7):808–821, 2015. doi: 10.1109/TVCG.2015.2403323

[18] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, Dec 2013. doi: 10.1109/TVCG.2013.226

[19] D. Filonik, T. Bednarz, M. Rittenbruch, and M. Foth. Glance–generalized geometric primitives and transformations for information visualization in ar/vr environments. In *Proceedings-VRCAI 2016: 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry*, vol. 1, pp. 461–468, 2016.

[20] J. Heer, M. Agrawala, and W. Willett. Generalized selection via interactive query relaxation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 959–968. ACM, 2008.

[21] H. Hochheiser and B. Shneiderman. Dynamic query tools for time series data sets: Timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004. doi: 10.1057/palgrave.ivs.9500061

[22] D. Holten, P. Isenberg, J. J. van Wijk, and J. D. Fekete. An extended evaluation of the readability of tapered, animated, and textured directed-edge representations in node-link graphs. In *2011 IEEE Pacific Visualization Symposium*, pp. 195–202, March 2011. doi: 10.1109/PACIFICVIS.2011.5742390

[23] Y.-J. Huang, T. Fujiwara, Y.-X. Lin, W.-C. Lin, and K.-L. Ma. A gesture system for graph visualization in virtual reality environments. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 41–45, April 2017. doi: 10.1109/PACIFICVIS.2017.8031577

[24] C. Hurter, R. Alligier, D. Gianazza, S. Puechmorel, G. Andrienko, and N. Andrienko. Wind parameters extraction from aircraft trajectories. *Computers, Environment and Urban Systems*, 47:28 – 43, 2014. Progress in Movement Analysis Experiences with Real Data. doi: 10.1016/j.compenvurbsys.2014.01.005

[25] C. Hurter, G. Andrienko, N. Andrienko, R. Gting, and M. Sakr. *Air Traffic Analysis*, p. 240258. Cambridge University Press, 2013. doi: 10.1017/CBO9781139128926.013

[26] C. Hurter, S. Conversy, D. Gianazza, and A. Telea. Interactive image-based information visualization for aircraft trajectory analysis. *Transportation Research Part C: Emerging Technologies*, 47:207 – 227, 2014. doi: 10.1016/j.trc.2014.03.005

[27] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation. *Comput. Graph. Forum*, 31(3pt1):865–874, June 2012. doi: 10.1111/j.1467-8659.2012.03079.x

[28] C. Hurter, S. Puechmorel, F. Nicol, and A. Telea. Functional decomposition for bundled simplification of trail sets. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):500–510, Jan 2018. doi: 10.1109/TVCG.2017.2744338

[29] C. Hurter, R. Taylor, S. Carpendale, and A. Telea. Color tunneling: Interactive exploration and selection in volumetric datasets. In *2014 IEEE Pacific Visualization Symposium*, pp. 225–232, March 2014. doi: 10.1109/PacificVis.2014.61

[30] C. Hurter, B. Tissoires, and S. Conversy. Fromdady: Spreading aircraft trajectories across views to support iterative queries. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1017–1024, Nov. 2009. doi: 10.1109/TVCG.2009.145

[31] T. Kapler and W. Wright. Geotime information visualization. *Information visualization*, 4(2):136–146, 2005.

[32] J. F. Kruiger, A. Hassoumi, H.-J. Schulz, A. Telea, and C. Hurter. Multidimensional data exploration by explicitly controlled animation. *Informatics*, 4:26, 2017.

[33] O. H. Kwon, C. Muelder, K. Lee, and K. L. Ma. A study of layout, rendering, and interaction methods for immersive graph visualization. *IEEE Transactions on Visualization and Computer Graphics*, 22(7):1802–1815, July 2016. doi: 10.1109/TVCG.2016.2520921

[34] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[35] J. P. McIntire and K. K. Liggett. The (possible) utility of stereoscopic 3d displays for information visualization: The good, the bad, and the ugly. In *2014 IEEE VIS International Workshop on 3DVis (3DVis)*, pp. 1–9, Nov 2014. doi: 10.1109/3DVis.2014.7160093

[36] T. Munzner. *Visualization analysis and design*. CRC Press, 2014.

[37] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 189–194. ACM, 1991.

[38] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm. Bringing up opensky: A large-scale ads-b sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pp. 83–94, April 2014. doi: 10.1109/IPSN.2014.6846743

[39] R. Scheepens, C. Hurter, H. V. D. Wetering, and J. J. V. Wijk. Visualization, selection, and analysis of traffic flows. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):379–388, Jan 2016. doi: 10.1109/TVCG.2015.2467112

[40] M. Sedlmair, T. Munzner, and M. Tory. Empirical guidance on scatter-

plot and dimension reduction technique choices. *IEEE transactions on visualization and computer graphics*, 19(12):2634–2643, 2013.

[41] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings* 23*rd* *ACM National Conference*, pp. 517–524, 1968.

[42] A. Sherbondy, D. Akers, R. Mackenzie, R. Dougherty, and B. Wandell. Exploring connectivity of the brain's white matter with dynamic queries. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):419–430, July 2005. doi: 10.1109/TVCG.2005.59

[43] H. S. Smallman, M. S. John, H. M. Oonk, and M. B. Cowen. Information availability in 2d and 3d displays. *IEEE Computer Graphics and Applications*, 21(5):51–57, Sep 2001. doi: 10.1109/38.946631

[44] R. L. Sollenberger and P. Milgram. Effects of stereoscopic and rotational displays in a three-dimensional path-tracing task. *Human Factors*, 35(3):483–499, 1993.

[45] M. Sousa, D. Mendes, S. Paulo, N. Matela, J. Jorge, and D. S. Lopes. Vrrroom: Virtual reality for radiologists in the reading room. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4057–4062. ACM, 2017.

[46] W. Usher, P. Klacansky, F. Federer, P. T. Bremer, A. Knoll, J. Yarch, A. Angelucci, and V. Pascucci. A virtual reality visualization tool for neuron tracing. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):994–1003, Jan 2018. doi: 10.1109/TVCG.2017.2744079

[47] J. J. van Wijk and C. W. A. M. van Overveld. *Preset Based Interaction with High Dimensional Parameter Spaces*, pp. 391–406. Springer US, Boston, MA, 2003. doi: 10.1007/978-1-4615-1177-9_27

[48] J. Vesanto. Som-based data visualization methods. *Intell. Data Anal.*, 3(2):111–126, Mar. 1999. doi: 10.1016/S1088-467X(99)00013-X

[49] C. Ware and G. Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics (TOG)*, 15(2):121–140, 1996.

[50] C. Ware and P. Mitchell. Visualizing graphs in three dimensions. *ACM Trans. Appl. Percept.*, 5(1):2:1–2:15, Jan. 2008. doi: 10.1145/1279640.1279642

[51] S. J. Westerman, J. Collins, and T. Cribbin. Browsing a document collection represented in two-and three-dimensional virtual information space. *International Journal of Human-Computer Studies*, 62(6):713–736, 2005.

[52] C. Wheatstone. *Contributions to the Physiology of Vision: Part the First: On Some Remarkable and Hitherto Unobserved Phenomena of Binocular Vision*. 1838.

[53] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. Efficient structure-aware selection techniques for 3d point cloud visualizations with 2dof input. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2245–2254, Dec 2012. doi: 10.1109/TVCG.2012.217