

FI3D: Direct-Touch Interaction for the Exploration of 3D Scientific Visualization Spaces

Lingyun Yu, Pjotr Svetachov, Petra Isenberg, Maarten H. Everts, and Tobias Isenberg, *Member, IEEE*

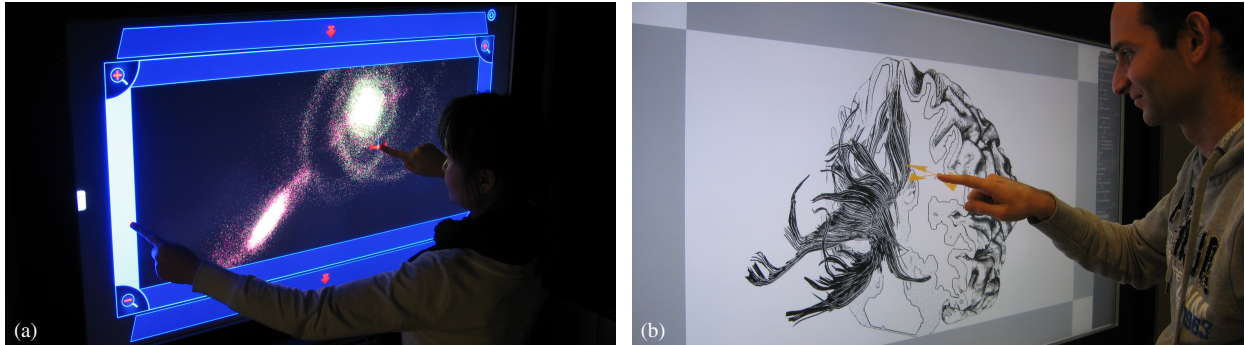


Fig. 1. Two case studies for a 3D visualization exploration widget that allows users to control the view in 7 degrees of freedom (DOF): (a) touch interaction with an astronomical simulation; (b) exploration of an illustrative 3D medical visualization.

Abstract—We present the design and evaluation of FI3D, a direct-touch data exploration technique for 3D visualization spaces. The exploration of three-dimensional data is core to many tasks and domains involving scientific visualizations. Thus, effective data navigation techniques are essential to enable comprehension, understanding, and analysis of the information space. While evidence exists that touch can provide higher-bandwidth input, somesthetic information that is valuable when interacting with virtual worlds, and awareness when working in collaboration, scientific data exploration in 3D poses unique challenges to the development of effective data manipulations. We present a technique that provides touch interaction with 3D scientific data spaces in 7 DOF. This interaction does not require the presence of dedicated objects to constrain the mapping, a design decision important for many scientific datasets such as particle simulations in astronomy or physics. We report on an evaluation that compares the technique to conventional mouse-based interaction. Our results show that touch interaction is competitive in interaction speed for translation and integrated interaction, is easy to learn and use, and is preferred for exploration and wayfinding tasks. To further explore the applicability of our basic technique for other types of scientific visualizations we present a second case study, adjusting the interaction to the illustrative visualization of fiber tracts of the brain and the manipulation of cutting planes in this context.

Index Terms—Direct-touch interaction, wall displays, 3D navigation and exploration, evaluation, illustrative visualization.

1 INTRODUCTION

Interactive 3D scientific visualizations have made a significant impact in many different disciplines. Yet, these systems are not typically regarded as being easy to learn or use [27]. Touch-based interfaces can potentially improve this situation as users of touch-based systems commonly associate them with being ‘intuitive’ and ‘natural.’ Part of the recent popularity of touch-based interaction is certainly due to the dedicated UI design and the novelty of touch as an interaction paradigm, but research has also shown that it can indeed be more effective than indirect forms of interaction. For example, touch interaction has been shown to outperform mouse input for the selection of targets on the screen [28], to facilitate awareness in collaborative settings [24], and to provide somesthetic information and feedback that is beneficial for effective interaction both in real and virtual environments [42].

As interactive displays become part of our everyday work environments, they provide ubiquitous new data analysis platforms that can encourage alternative forms of scientific data exploration and pro-

mote the use of scientific visualization techniques even by non-experts. Touch interaction has been the focus of previous research projects in the visualization context [14, 16, 25, 39], but much remains to be learned about the effects of providing touch input in scientific visualization. We need to learn more about how to re-design desktop and mouse-based systems for direct touch, for which scientific data analysis scenarios direct touch and traditional interfaces may be most suited as interaction alternatives, and, on a higher-level, how direct touch changes the ability of viewers to understand data and draw insights.

In an effort to explore this space we designed and studied FI3D (Frame Interaction with 3D spaces), a novel direct-touch technique that allows users to *explore three-dimensional data representations and visualization spaces*. This ability is essential to many tasks in scientific visualization, in particular for data such as medical volume scans, volumetric physical simulations, or models in astronomy and cosmology. Touch interaction for 3D data exploration is a challenging endeavor [49] because we have to deal with an under-constrained problem: mapping 2D input parameters to 3D transformations in space.

While much of the previous work on direct-touch data exploration has considered work with specific objects within a 3D space, our focus is on manipulating the space *as a unit* which is important for many scientific datasets, such as those found in particle simulations in astronomy. FI3D does not require separate menus or dedicated interaction widgets inside the space itself. Our goal is to ensure that the space itself is used solely for representing the data visualization itself and that the technique can be generically applied to different types of 3D scientific data. FI3D makes use of the visualization space’s borders

- Lingyun Yu, Pjotr Svetachov, Maarten H. Everts, and Tobias Isenberg are with the University of Groningen, the Netherlands; e-mail: {lingyun.yu@p.svetachov@student. | m.h.everts@isenberg@cs.}rug.nl.
- Petra Isenberg is with INRIA, France; e-mail: petra.isenberg@inria.fr.

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.

For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

and can be used on hardware that supports dual- or even just single-touch input. By focusing on a single- and dual-touch technique, we can take advantage of all modern types of touch surfaces and design our interactions to be fundamentally simple but easily extensible. Our interface allows for full 7 DOF manipulation using only single-touch. We support translation in x -, y -, and z -direction, orientation with respect to the 3D coordinate system, and uniform zoom. Furthermore, we present how an additional touch can be used to constrain certain interactions to allow for *precise* or *integrated* exploration of scientific data. We applied the interaction technique to two case studies and evaluated it in comparison to traditional mouse interaction. We show the utility of our technique for the exploration of particle simulation data from astronomy and for the investigation of fiber tract data in human brains. For the latter we also describe how to use an extension of the technique for the manipulation of cutting planes.

2 RELATED WORK

Our work touches on existing approaches in several different domains. We first briefly talk about interactive visualization in general, including some techniques that employ direct touch. Then, we specifically discuss approaches for direct-touch interaction in 3D environments.

2.1 Interactive Visualization

In the field of information visualization, the challenges of interacting with data have been analyzed in several survey articles [29, 40, 54]. These surveys focus on specific data interaction techniques such as selection, exploration, change of representation, or filtering. Pike et al. [40] further distinguish between higher-level interactions (explore, analyze, etc.) and lower-level interactions (filter, sort, cluster, etc.), correlating those with representation intents (depict, identify, compare, etc.) and interaction techniques (dynamic query, zoom, brushing, etc.). While these overviews have targeted information visualization, interaction is no less important for scientific visualization [27]. We share the same fundamental notion that useful insight often only emerges out of the interactive data manipulation experience. Yet, in scientific visualization we focus on visualization spaces in which most of the data has preceived spatial meaning and, hence, many interactions for data exploration have to support explorations that match a user's existing (likely physically-based) mental model of a dataset [29].

In this paper we are particularly interested in the challenges of providing data exploration capabilities for 3D visualization spaces. In many scientific desktop-based systems widgets are used for navigating in and manipulating a 3D space. Typical techniques include the Arcball [45] or similar virtual trackballs [3], coupled with techniques for movements along all three spatial axes. Considerably more research has been conducted on 3D interaction in non-desktop based systems using dedicated input and output hardware [6] such as virtual reality environments like the Responsive Workbench [31] or the CAVE [7]. One important advantage of these virtual environments is that they afford direct manipulation with the 3D worlds [6, 9] because both stereoscopic projection of the virtual world and its manipulation (through tracking or by using haptic devices [48]) happen in-place.

For visualizations that are based on two-dimensional data, however, these disadvantages do not exist. Here, both projection and tracking can be realized on a 2D plane, using touch-sensitive surfaces. Scientific visualization systems that make use of this interaction metaphor include ones for interactive exploratory and illustrative visualization of 2D vector data through hand-postures with custom-drawn glyphs [25] or sketching [44]. Another example is Forlines et al.'s DTLens system [14]. Here, touch interaction with high-resolution 2D spatial data such as maps or large photographic astronomy sky data is made possible through mobile lenses that allow people to investigate detail in its context. Similar types of geospatial exploration using multi-touch interaction have also been explored in multi-display environments [12]. In all of these cases a direct manipulation of the presented data is key, mostly facilitated through direct touch. Similarly, we explore interaction with direct-touch technology but focus on interaction with 3D visualization spaces. We want to explore the advantages of data immersion through a direct-manipulation interface [42] and provide a

walk-up and use interface that does not require users to wear and be equipped with specific hardware to view and interact with the data. In the following section we discuss related work that directly relates to our goal of providing direct-touch 3D interaction capabilities.

2.2 Direct-Touch Interaction with 3D Environments

A relatively small area of previous work deals with touch-based interaction with scientific visualizations in 3D spaces. Direct-touch spatial interaction with 2D elements on a 2D surface are more common as the interaction is fairly straight-forward: x -/ y -motions of a single finger or pen can be directly mapped to matching translations of virtual 2D elements. Adding mode-less capabilities for object orientation to 2D spatial movement, however, then requires the mapping of 2 DOF input to 3 DOF of output (location in x and y and orientation) [23]. The rotation-and-translation (RNT) technique [30] solves this problem for touch-interaction by determining the orientation of an object based on the spatial movement of a touch-point on the object over time. TNT [32], a related technique, uses a pen's spatial orientation to set a virtual object's orientation. With the advance of *multi-touch* surfaces it also became possible to use more than one input point to control virtual objects; e. g., the popular two-touch interaction technique [23] (also named rotate-scale-translate, RST [41]) uses two fingers (4 DOF) to control location, 2D-orientation, and size of objects.

Designing interaction techniques for *three-dimensional* objects (as needed in many scientific visualization application scenarios) through direct-touch on a two-dimensional surface in a similar manner is not as straight-forward [49]. Here, the location and orientation of objects have 3 DOF each, thus, together with a (uniform) scaling we need to be able to control 7 DOF. Hancock et al. [21] presented an extension to the mentioned traditional 2D control techniques to support shallow-depth 3D interaction. In this setup, the 3D objects are restricted to locations on a 2D plane parallel to the touch surface. By using up to three fingers, people can have full control of 5 DOF of movement and rotation in a shallow 3D environment that does not allow for navigation in the z -direction. A related method by Martinet et al. [34, 35] supports two-finger 3 DOF control of the 3D position of objects. Recently, the shallow depth technique was extended to full 6 DOF control of both location and orientation in full 3D [22].

Hancock et al.'s [21, 22] extension of the concept of direct manipulation from 2D interaction to 3D provides users with precise control over the manipulated objects. However, sometimes it is desired to simulate physical interaction with 3D objects to make the experience more life-like. Inspired by this goal, Wilson et al. [53] use physics simulation to enable sophisticated casual interactions. This takes full advantage of multi-touch surfaces that not only provide single points of contact but can also capture the shape of a touch, such as the side of a hand. Wilson et al. thus model touch points as physical rays in the 3D scene which then interact with virtual 3D objects. This technique has recently also been extended to allow more control in the form of grasping [52]. In contrast to our own work, the goal for all these techniques is to interact with individual 3D objects within the 3D space. Our goal is to provide means for interacting with the 3D space itself as our target data does not necessarily contain individual objects that can be interacted with, such as in the example of particle simulation data.

Previously, this type of touch interaction with 3D spaces, in particular for navigation purposes, has been realized using gestural interaction on multi-touch surfaces. While Edelmann et al. [11] and Jung et al. [26] use dedicated definitions of gestures to navigate in 3D, Reisman et al. [41] use a constrained energy minimization approach to map the motions of contact points on the touch surface to transformations of objects in 3D space, essentially ensuring that touch points on the display surface as much as possible stay connected to the points on the objects that were initially touched ('sticky'). For moving the camera in a scene rather than the objects, Hachet et al. [19] use Navidget, a widget that can be accessed on demand to specify new views. However, all of these approaches also require that a dedicated navigation object in the 3D space is interacted with, as opposed to just the 3D space itself. We focus on interaction with the space itself in order not to occlude the main view of the data by navigation widgets. For scien-

tific visualization, two previous gesture-based interaction approaches permit interaction with aspects of space itself: Fu et al. [18] use gestural object-based interactions for most travel tasks and define a gesture to invoke powers-of-10 ladders to affect scale in an object-independent way; Forlines et al. [13] map hand postures and gestures to traditional 3D space interactions, such as rotation, translation, and scale. An off-screen touch-based technique of interacting with the space itself is discussed by de la Rivière et al. [10] who use a dedicated input device in addition to the wall that shows the 3D scene. The input device has the form of a cube with a touch surface and represents the space to be interacted with, and interactions on the cube are mapped to interactions with the 3D space shown on the separate wall display. In a way, our frame interaction could be thought of as a virtual variant of this cube.

3 FI3D: A NEW TECHNIQUE FOR FRAME INTERACTION WITH 3D VISUALIZATION SPACES USING TOUCH

In the design of our technique we were guided by several complementary goals. We designed our technique to:

- G1: encapsulate all seven common degrees of freedom for 3D interaction in one joint interaction paradigm,
- G2: support the manipulation of the space itself rather than specific objects within it,
- G3: not require intermediate in-space interaction proxies,
- G4: require only spring-loaded modes,
- G5: be generic to be applicable to many different 3D scientific data representations,
- G6: allow for both large-scale and precise interactions,
- G7: in its base form require just one touch-point,
- G8: be easily extensible,
- G9: be intuitive and require little learning time, and
- G10: be ‘competitive’ with current techniques for 3D scientific data exploration.

These goals set our technique apart from previous approaches. We specifically do not require dedicated objects to be present inside the visualized data space (G3). This is a crucial design decision since many data sources in scientific visualization inherently do not have (large) dedicated objects (e. g., particle clouds, 3D flow fields, or voxel data) which could be used as interaction proxies. Therefore, we enable interaction with and exploration of the space itself (G2), as opposed to objects within the space as done previously. This goal implies that, for example, we cannot rely on a surface being present to extract the constraints for the 3D interactions (e. g., [11, 26, 41]). We also strive for our 3D interactions to be intuitive extensions (G9) of 2D RST direct touch manipulations [22, 41]. Both constraints result in that we have to find meaningful heuristics, for instance, to determine the axes for rotations or virtual planes where touch interactions are ‘sticky’ [22, 41]. At the same time, we do not want to rely on a large number of gestures (G1, G9) that users have to learn and remember or system-controlled modes (G4) that have negative usability implications. Another design criterion is that we want to be able to perform both broad interactions to explore large-scale structures and precise and constrained interaction (G6) to examine fine details. The separation of spatial manipulations required for this goal is often challenging in multi-touch environments [37]. Finally, we want to enable users to control all degrees of freedom on single-touch surfaces (G7) so that our technique can take advantage of all types of currently available touch-surfaces. At the same time, we designed the technique to be easily extensible (G8) so that it can take advantage of multi-touch interaction where available.

Next, we discuss how we realized these goals in two stages: (1) transitioning from 2D direct-touch interaction to 3D space manipulation and (2) mapping specific direct-touch interactions to specific 3D manipulations using frame interaction with FI3D.

3.1 From 2D Planes to 3D Space Manipulation

To inform the design of FI3D, we first looked at related traditional mouse-based interfaces such as 3D viewers. Here, different interaction modalities—often combined with key combinations—are used to map the 2 DOF mouse input to 7 DOF navigation in 3D: motion along the

x -, y -, and z -axes, rotation around the x -, y -, and z -axes, and (uniform) scale or zoom. We thus chose to incorporate these techniques in our direct-touch technique. Before describing the final interaction design, however, we examine the different necessary interactions individually:

Translation of the 3D space parallel to the view plane: Mouse input is directly mapped to virtual motion along the x - and y -axes. One problem that arises in 3D but not 2D spaces is that, for perspective projection, the interaction point typically does not stay directly connected to locations on 3D objects which may have been initially touched. The interaction is only ‘sticky’ for a single plane parallel to the viewplane located at a certain distance from the viewer, depending on the control/display ratio [5] or control gain [33]. A suitable distance is typically chosen by the interaction designer or given by a center point of the dataset. Realizing x -/ y -translation with touch input is possible in the same way: the motion of the touching point is mapped to a x -/ y -translation in 3D space in a shallow depth fashion [21].

Rotation around x - and y -axes: In traditional mouse-based interfaces, this type of rotation is often achieved with a trackball/arcball metaphor [3, 45]. In a touch-interface, this type of rotation can be easily achieved by treating a touch input like a mouse input. Similar to the translation case, it is necessary to pre-select a specific location for the center of rotation. If dedicated objects exist in space, typically their center of mass or a specified pivot point is used. In our case—without such dedicated objects—we have to rely on a heuristic. We use half of the depth range covered by the *visible* part of our dataset. This means that those parts of the dataset which are in front of the near clipping plane are not considered in this case. We also use the same heuristic for determining the ‘sticky’ plane for x -/ y -translation.

Translation in z and rotation around z : The transformation along or around the z -axis can be controlled individually (i. e., by involving a mode switch) by mapping one dimension of the 2 DOF of touch input to the specific transformation: for example, the 2D y -motion of touch can be mapped to z -translation, while the angle of a circular motion around the screen center can be mapped to a rotation around the z -axis. The latter can be mapped directly, while the former needs to employ a certain control/display ratio.

Scaling and Zooming: In perspective projection, z -translation results in objects increasing or decreasing in their visible size on the screen. Other possibilities to achieve a similar effect but without moving the camera are to enlarge the dataset itself (scaling) or to change the *field of view* angle of the virtual camera (dolly-zoom). In the context of 3D exploration, both have advantages and disadvantages. Scaling, for instance, requires a center of transformation that is located exactly at the point of interest; otherwise focus objects may move further away or closer to the viewer. The dolly-zoom, in contrast, has upper limits (an angle of 180°) but is independent of the depth of the focus objects. This last consideration lead us to using dolly-zoom in our interaction in addition to z -translation.

RST Interaction: The two-touch pinching gesture [23, 41] has been popularized in the media and is perceived by the general public to be an integral part of multi-touch interaction. Therefore, we also make use of this 4 DOF technique for the interaction with 3D spaces, comparable to the 2D RST interaction [23]. We realize this RST mapping by combining the individual mappings for x -/ y -translation, rotation around the z -axis, and zoom the same way as the 2D mapping, simply taking the 3D control/display ratio considerations into account. The center of transformations is always the middle between the two touching points, and the translation is ‘sticky’ for the plane located at half the space interval that is taken up by the visible part of the dataset.

Technique Integration: The 4DOF dual-touch technique for x -/ y -translation, rotation around the z -axis, and zooming can easily be combined with the 2DOF single-touch control of trackball x -/ y -rotation. This results in a combined single- and dual-touch interaction for controlling a single large 3D space. This 6 DOF technique, however, has a number of problems in our context of scientific visualization. While the single-touch rotation can control the orientation without affecting the zoom parameter, this is not easily possible with the two-touch technique. As both fingers are controlled by the user to some degree independently, it is difficult to achieve translation or rotation while leaving

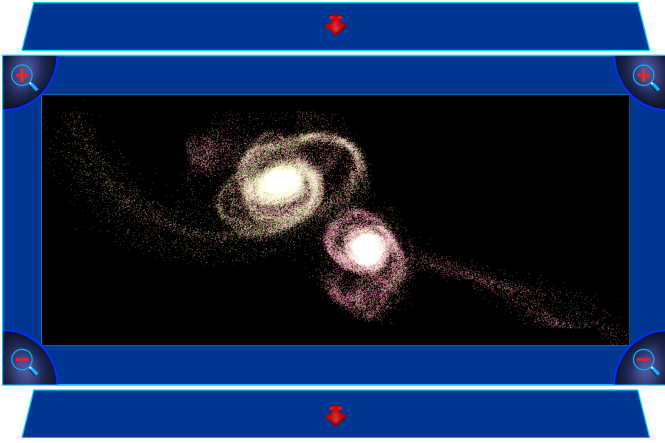


Fig. 2. Screenshot of frame-based 3D visualization exploration widget.

the scale of the space unaffected. Similarly, translation in x and y cannot be performed independently from rotation around z and zooming. Finally, we are only able to control 6DOF and it is not possible to manipulate the space’s location along the z -axis. Therefore, we need to involve other elements to integrate this last DOF (G1) while also allowing single-touch-only interaction (G7). This is described next.

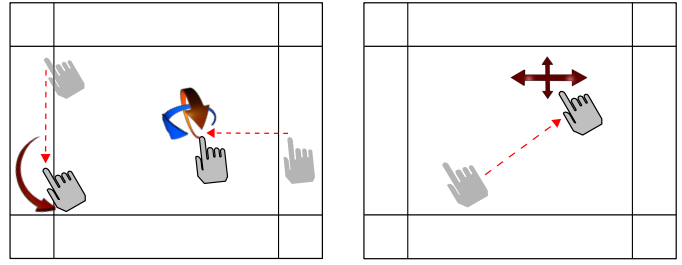
3.2 Frame Interaction for 3D Visualization

The techniques described in the previous section allow us to control up to 4DOF simultaneously by directly interacting on the visualization display space. To support full 7DOF interaction (G1) and address our remaining goals (G2–G10), we introduce a *frame-based interaction* technique. This method is inspired by previous interfaces [38, 55] that used widget borders to control aspects of the interaction. For instance, Zeleznik and Forsberg’s [55] UniCam permits the use of a number of gestures with a single touch point to control several parameters of a virtual camera, a goal related to our own (G7). Most interestingly, however, they use the border region to start free rotations around the viewer’s center which they determine as the depth of the closest object located along the view ray. For a 2D sketching interface, Nijboer et al. [38] use the frame of their interface (and that of custom selections) to control both rotation and translation in 2D. If one starts an interaction along the main axis of the frame side, the content of the frame is rotated, if one starts perpendicular then the content is translated.

We combine both approaches to allow interaction with 3D visualization spaces (Fig. 2). Similar to Nijboer et al.’s [38] interface, we differentiate interaction by movement direction with respect to the frame. We map motion initially parallel to the frame to rotations around the z -axis (Fig. 3(a)). The reason for this choice is that it is reminiscent of what one would expect in 2D (G9), similar to RNT touch interaction with 2D [30] or 3D [21] shapes. Also, for 3D exploration of visualizations this has the advantage that it allows constrained z -rotation, an interaction typically not required for regular camera control (G6).

For frame interactions that are started perpendicular to the main axis of a frame side we have two options: mapping them to x -/ y -rotation like Zeleznik and Forsberg [55] or mapping them to x -/ y -translation like Nijboer et al. [38]. We experimented with both and chose to map to x -/ y -rotation (Fig. 3(a)) because this lets users associate interactions starting on the frame to rotations, leaving single-touch interaction in the center of the widget to be mapped to x -/ y -translation (Fig. 3(b)).

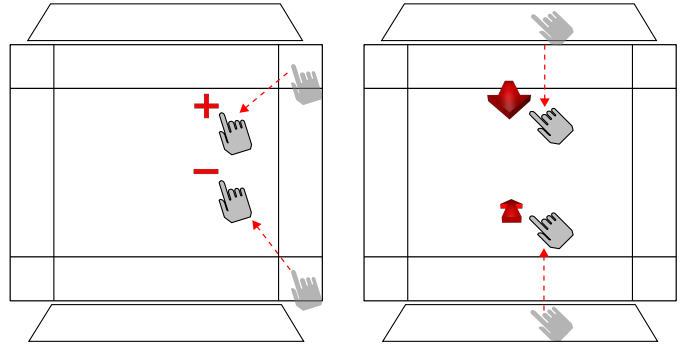
To be able to accommodate controls for the remaining two parameters—zoom and z -translation—we add additional regions to the frame. For zooming we use the four corners of the regular frame (Fig. 4(a)). Downward motions started from the top corners zoom in, subsequent upward motions zoom out. The bottom corners have the opposite behavior, initial upward motions zoom in and subsequent downward motions zoom out. Translation along the z -axis is made possible using two extra frame elements at the top and the bottom of the widget (Fig. 4(b)). Here, the perspective shape of the extra frame elements suggests the direction of movement. Starting motions from the



(a) z -rotation and trackball rotation.

(b) x -/ y -translation.

Fig. 3. General frame interaction technique.



(a) Zooming.

(b) z -translation.

Fig. 4. Zooming and z -translation with extra frame regions.

top downward moves the camera away from the dataset, while motions from the bottom initially upward move the camera closer to dataset.

All these specific regions act as spring-loaded modes for adding these additional degrees of freedom (G4). To further support usability, we always display icons associated to the chosen actions (as shown in Fig. 3 and 4) to make it easier for users to remember the mappings.

The combination of these frame interactions with single-touch x -/ y -translation started from the center of the widget allows us to provide control for all main 7DOF necessary for the exploration of 3D visualization spaces (G2) in one integrated paradigm (G1), without requiring dedicated objects to be present (G3). Compared to the integrated 6DOF single-/dual-touch technique discussed in Section 3.1, however, the frame-based interactions allow us to separate out the interactions and, consequently, permit users to control the exploration more precisely (G6). For instance, users can affect the rotation around z independent from x -/ y -translation and both without affecting the zooming, and vice versa. Furthermore, we are able to control all 7DOF with only a single touch (G7) and, therefore, our technique can be used on touch-displays that only provide a single simultaneous touch location. Nevertheless, many people nowadays expect two-point RST interaction in touch-enabled interfaces. Our technique was designed to be easily extensible (G8) and we provide RST interaction (Fig. 5) when people start their interaction with two fingers inside the frame and if the used touch technology has this capability. This way we give the user the choice of either fast and integrated interaction with two fingers or precise and separate interaction with the frame.

Because precise control is of high importance in scientific visualization, we explored the possibilities of frame interaction for constraining selected transformations further. While so far it is possible to single out rotation around the z -axis, rotation around the x -/ y -axes has been integrated: the motion during rotation determines an axis parallel to the x -/ y -plane around which the rotation occurs. To instead permit constrained rotation around either x - or y -axis we propose to employ dual-touch and the frame sides. Here we make use of the fact that the four frame sides are perfectly vertical or horizontal. We let users specify an axis around which one aims to rotate by touching one side of the frame with their non-dominant hand while the dominant hand then can be used to perform the rotation (Fig. 6). For example, for rotating

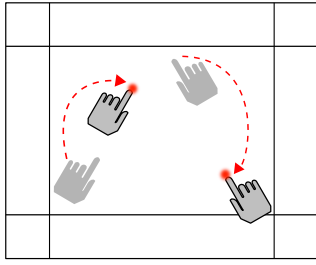
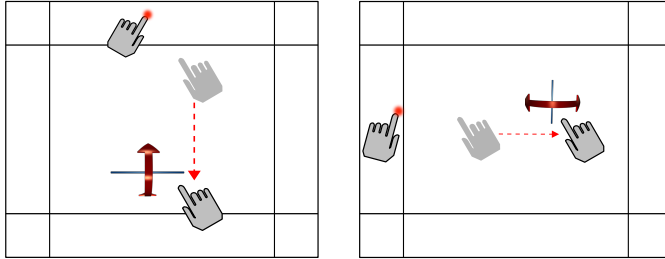


Fig. 5. Two-touch RST interaction: translation, z -rotation, and zoom.



(a) Constrained x -rotation.

(b) Constrained y -rotation.

Fig. 6. Constrained rotation techniques for the x - and y -axes.

only around the y -axis one would place one finger on either vertical frame side and then use the other finger (inside the frame or when also starting from one of the vertical frames) to rotate a constrained trackball (i. e., a virtual track-cylinder). Similarly, when one has already started trackball rotation through perpendicular motion originating in any frame side, one can constrain this interaction at any time by placing another finger in one of the four frame sides, the horizontal ones for rotation around the x -axis or the vertical ones for y -rotation.

4 USER STUDY

To understand how people would perform with and rate our frame technique, in particular related to our goals G9 and G10, we conducted a comparative study. Since the mouse is currently the standard interface for 3D interactive desktop applications in scientific visualization it was chosen as the baseline and compared to the frame technique based on speed, accuracy, and qualitative feedback for eight travel tasks and one wayfinding task. Since these two technique are considerably different in interaction style, a comparison can show where tradeoffs exist and how our technique can be further improved. Based on previous work on the comparison of touch and mouse interaction [15, 24], we hypothesized that touch interaction would not outperform the mouse for tasks that required 3D interaction according to one interaction type (translation, zoom, rotation only) but that the touch technique would outperform the mouse for tasks in which multiple integrated types of interactions were required. We further hypothesized that the touch-technique would score higher on questions related to how immersed the participants felt in the data and would be generally preferred.

Participants. Twelve members (6 male, 6 female) from our local university participated in the study. Seven participants reported prior experience with 3D computer games. The experience varied from none (5 participants) to once a day with five participants reporting at least weekly experience. Ages ranged from 19 to 39 ($M = 27.25$, $SD = 5.29$). All participants were right-handed. Eight participants were students from different disciplines and four non-students.

Apparatus. The experiment was performed on a 52" LCD screen with full HD resolution (1920 \times 1080 pixels, 115.4 cm \times 64.5 cm). The display was equipped with a DViT overlay [46] from Smart Technologies, capable of recognizing two independent inputs. The display was positioned so that the center of the display was at a height of 1.47 m above the ground. Participants interacted in a 3D view covering 575 \times 575 pixels (34.5 cm \times 34.5 cm) throughout the study so that we could show information relevant to the task side-by-side. The setup



Fig. 7. Example setup for travel task, (a) touch and (b) mouse condition.

Table 1. 3D interaction mappings for the mouse condition.

translation x/y	left-click
translation z	middle- or 2-button click
rotation trackball	right-click
rotation constrained in $x/y/z$	right-click+{X Y Z} key down
zoom	mouse wheel

Table 2. Sequence of 3D interaction tasks per condition.

1	translation x/y	center object in target area
2	translation $x/y/z$	center, fill object in target area
3	rotation z	rotate object to face screen
4	rotation trackball	rotate object to face screen
5	rotation x	rotate object to face screen
6	rotation y	rotate object to face screen
7	zoom	zoom object to fill target area
8	rotate-scale-translate (RST)	center, fill object in target area

for the frame interaction was described in Section 3 (Fig. 7(a)). We removed the interaction frame for the mouse condition (Fig. 7(b)). In contrast to typical 3D scientific visualization applications, we decided not to use menus or buttons to switch between translation, rotation, or zoom modes when using the mouse to avoid measuring additional travel distance and time necessary to reach these buttons. Instead we chose mouse+keyboard combinations as shown in Table 1. The system ran on Windows 7, and the mouse pointer speed was set to the average speed (half-way between slow and fast). In both conditions participants stood at approximately arm-length away from the display. The 800 dpi optical mouse was placed on a side-table of 1.03 m height.

Tasks. We tested eight travel and one longer wayfinding tasks. Travel is characterized by low-level actions that control position and orientation of the 3D space, while wayfinding requires a cognitive component and may include planning, higher-level thinking, or decision-making [6]. Wayfinding is important to scientific analysis but we also wanted to test travel tasks as a fundamental component of wayfinding to see more specifically how our technique supported these individual lower-level aspects. Table 2 summarizes the eight travel tasks. We tested the four main interaction techniques translation, rotation, zoom, and the integrated RST technique. As constrained rotation requires

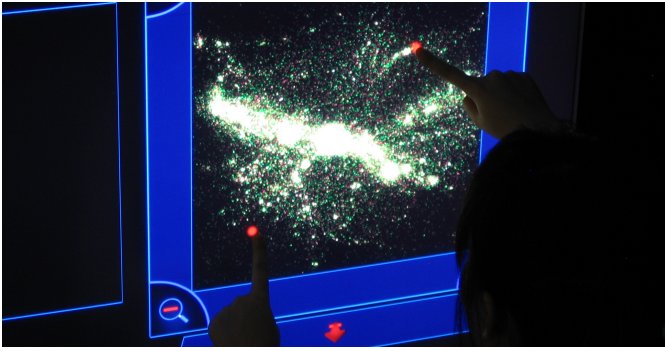


Fig. 8. A person performing the wayfinding task.

participants to perform a bi-manual task by touching the frame on different sides of the 3D display, we tested all three individually to understand if difficulties would arise due to the spatiality of this interaction.

At the start of each travel task, the data space was repositioned and a target image was shown (see Fig. 7(a) and 7(b)). Participants were asked to reposition the space as quickly and accurately as possible to visually match the target image. As participants would always reposition the whole space itself to achieve a matching we did not have to separately measure target selection and travel times for these tasks. To aid the matching process, a transparent red target area was displayed that served as a reference frame into which the target had to be moved. When participants let go of the display or mouse button, we calculated whether the target image had been matched and in this case stopped the trial automatically. If participants felt that they could not match the target image they could abort the trial, but this never occurred.

The wayfinding task required participants to explore a 3D astronomical dataset consisting of a particle simulation representing different masses (Fig. 8). We asked them to examine the data for five minutes, exploring different parts and different scales, and to report any interesting or strange aspects they noticed. In particular, we asked them to explore and describe the 3D shape of the clusters in the core region.

Design. We used a repeated-measures design with the within-subject independent variable *input device* (frame, mouse). Each participant performed 4 runs of 4 trials for each input device and task. For each run we chose 4 unique starting positions of the space per task and varied their order between runs using a Latin square. For rotation tasks we additionally chose two rotation directions per run which was also varied using a second Latin square. Tasks were always performed in the same order (Table 2) and the order of presentation of the two input devices was counterbalanced among participants. The first two runs were discarded as practice runs for the final analysis of the data.

In summary, the design was 12 participants \times 2 input devices \times 8 tasks \times 4 runs of trials \times 4 trials = 3072 interactions in total. The dependent variable measured was the time to reposition the space. Before each task for each input device participants were introduced to the technique through a set of practice trials. They moved on to the experiment after they reported to feel that they had understood how to perform the tasks. After completing all tasks with one input device, participants were asked to complete a questionnaire to rate the usability of the technique in terms of ease of use, ease of remembering, precision, efficiency, and difficulty on a seven step Likert scale. After the second condition, participants were also asked to compare both techniques, voice their preference, and give qualitative feedback. During the wayfinding task, we collected only qualitative feedback and took notes on participants' interactions. Finally, after finishing the wayfinding task, participants were asked to fill in the final part of the questionnaire to comment on which technique they preferred and why, and whether this technique allowed them to explore the data as they desired. Also, they filled in their demographic background information and were asked to provide additional verbal feedback on their experience of which we took notes.

Table 3. Significance scores for the four rotation tasks.

Task:	trackball	x-rotation	y-rotation	z-rotation
$F(1, 11) =$	8.040	18.967	41.837	7.439
$p <$.016	.001	.001	.020

5 RESULTS

After the task completion times collected during the study were log-transformed to comply with the normality assumption of the data analysis, time was analyzed using a repeated-measures ANOVA (frame, mouse). The results are broken into the four main types of travel tasks.

Translation. Tests showed no significant effect of input device on the translation time for x - y -translation ($F(1, 11) = .075; p = .79$) with mean completion times increasing from 5.49 s for the touch condition to 5.86 s for the mouse. Similarly, the test for Task 2 (z -translation) showed no significant effect between the two input devices ($F(1, 11) = 3.57; p = .08$) with mean time of 9.34 s for mouse and 10.63 s for the touch condition. A practical equivalence test with a threshold of 0.12 (3% of the mean) was significant ($p = .019$) for x - y -translation. Therefore, the two techniques can be considered equivalent for this task. In the post-session questionnaire participants were asked to rate the two input techniques on a 7-point Likert scale according to whether they thought translation in/out and left/right was easy to perform. Both techniques scored highly with a median of 6 (agree) for the mouse and 6.5 (agree-strongly agree) for touch.

Rotation. The analysis of task completion time showed a significant effect for all four rotation tasks with the mouse condition being significantly faster than touch in all cases. Table 3 gives an overview of the significance scores for the rotation tasks. Hence, for rotation we did not achieve our goals of providing a competitive alternative to (good) mouse-based interaction in terms of speed. We discuss our hypotheses of these results in the discussion section. The post-study questionnaire asked participants to rank the two techniques on a 7-point Likert scale according to whether rotation was easy to perform. Both techniques scored a median if 6 (agree) on this question.

Zoom. The analysis of task completion time for the zooming task showed a significant difference between both input types ($F(1, 11) = 64.70; p < .001$) with mouse being significantly faster. During this task we asked participants in the touch condition to use one of the four corners of the frame to perform a zoom operation. This proved to be significantly slower than simply turning the scroll wheel on a mouse to perform the same operations. Yet, the touch condition has the advantage that it provides smooth zooming steps compared to the mouse wheel. In addition, our technique allows for alternative means to zoom into the data by using the two-touch RST technique. In the post-session questionnaire participants rated the zoom technique on ease of use on a 7-point Likert scale. Both scored a median of 7 (strongly agree). While the specific zoom functionality was not faster on touch, we believe that our interaction design still offers competitive alternatives in *functionality and ease of use*.

Rotate-Scale-Translate. We did not observe a significant effect of input device on task completion time in the final task ($F(1, 11) = .982; p = .343$) with mean task times increasing from 11.86 s (touch) to 14.62 s (mouse). This surprised us since we had hypothesized that the integrated gesture would show significant performance benefits.

Wayfinding Task. In the wayfinding task we let participants freely explore the dataset with an interaction method of their choice. Participants had the freedom to use either touch or mouse or a combination of both. 75% of participants chose to only use touch, while 25% of them chose to use a combination of both. Participants typically began exploring the dataset by zooming in and looking at the center in more detail. To explore the general shape of the datasets they then used trackball rotation and constrained rotation to get a more precise understanding. To perform bigger space transformations, participants tended to use the two-point RST interaction, moving interesting regions of a cluster to the center and simultaneously re-orienting the 3D space. For smaller changes, in contrast, participants used translation and constrained rotation around one of the axes separately. The two-

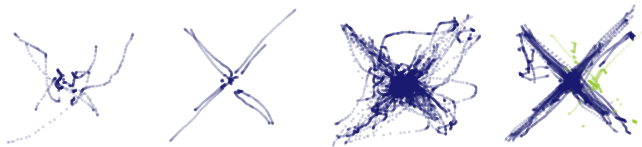


Fig. 9. Translation for mouse and touch for a single participant (left two images, resp.) and for all participants overlaid (right two images, resp.).

point RST interaction was the most frequently used tool. When asked about the reason for choosing a particular interaction technique, participants who solely worked with touch generally felt that the experience was more interesting using this input (5×) or that touch was easy and intuitive (4×). People who chose to use both wanted to take advantage both of the easier zooming with the mouse and the easier rotation with the touch (1×) or were interested in exploring the differences further (1×). All participants reported that the techniques they chose for the wayfinding task were useful for exploring the data and that they allowed them to see those parts of the space they were interested in.

Overall, participants were able to explore the space effectively and could name several interesting aspects of the data based on overviews as well as detailed views. We observed some people trying unsupported gestures such as twisting their fingers to make small rotations. Our technique could be extended to include new types of input gestures on multi-touch screens but further work needs to determine the most useful multi-finger gestures for these types of space transformation.

Overall Preferences. After all travel tasks had been completed for each interaction technique, participants were asked to compare both techniques. At this time, 66% of participants chose the mouse as their preferred input type. They reported that the main reason for this choice was that under time pressure it was easier due to familiarity (2×), speed (2×), and less physical involvement (4×). Out of the four participants who preferred touch, three named intuitiveness, and the natural and closer feel of the touch condition as the main reason.

After the more exploratory wayfinding task, we asked participants again about their overall preference for one technique. At this time, their preference changed. Now, 75% of the participants chose touch over mouse. The main reasons for choosing touch at this stage were perceived immersion (2×), intuitiveness and ease of learning (2×), and a general feeling of having “things under control” (1×). The three people who preferred the mouse did so because they felt to be more precise and because it felt easier due to familiarity of mouse interaction.

Ease of Use. One of our goals was to design an intuitive, easy to use and learn touch-interaction technique for 3D data. On a 7-point Likert scale participants agreed (median 6) that touch was easy to use, and disagreed (median 2) that it was difficult to remember how to interact with the frame and to use it in general. Given these positive responses we feel that our G9 was achieved.

6 DISCUSSION AND LESSONS LEARNED

Our experiment aimed to evaluate how our touch interaction compared to a traditional mouse interface in terms of both quantitative and qualitative measures. The touch technique was rated highly for its overall ease of use and for each individual travel task. Participants found that it is easy to remember how to perform interactions, that they were able to interact precisely, and efficiently. Touch seemed to invoke a sense of more direct connection to the data and increased immersion with the displayed information. Touch interaction also provided additional integrated interaction functionality compared to the mouse: the RST technique was frequently used and highly valued by participants.

We found that participants’s acquaintance and years of practice with the mouse influenced their performance. Participants preferred the mouse for time-pressured tasks due to its familiarity but preferred touch for open-ended exploration. In terms of speed, the touch technique did not show improvement over the mouse but generally also did not incur large time penalties for the travel tasks. Yet, we found that both touch and mouse could be considered practically equivalent in terms of speed only for the 2D translation task. The interaction logs revealed that participants made more precise and straight movements

in the touch condition (Fig. 9) while mouse movement was considerably more noisy. The proprioception afforded by each technique likely played a role here. It would be interesting to further test the value of touch interaction for tasks in which precise movements along specific paths are required. A further investigation in this direction could also shed more light on why some participants preferred touch due to an increased sense of “having things under control.”

The mouse was significantly faster than touch for the rotation tasks which we explain largely with hardware issues. The DVIT technology we used to capture inputs [46] relies on triangulating shadows of touching fingers in front of a strip of IR-LEDs. This results in less responsiveness in certain touch configurations which was apparent, in particular, in the rotation tasks in the study where participants tended to touch the screen often inadvertently with parts of the whole hand instead of one or two fingers only. During these rotation tasks we thus noticed interference between the two simultaneous touches, with one input having an occasional response delay of up to 0.5 s. This delay prohibited participants from finishing the tasks quickly. We hypothesize that with more reliable sensing the timing results will differ. For the zooming task we did not expect touch to outperform the mouse as turning a scroll wheel is faster than a touch+drag action. Yet, our zoom technique provided additional functionality. Two participants commented that they preferred touch due to the continuous nature of the zoom which the mouse interaction was not able to provide. While participants were ca. 3 s faster in the RST task using touch, we did not find a significant difference. We have no direct explanation but suspect that fatigue played a role. Forcing participants to take longer breaks between tasks may have helped to get a clearer answer for this task.

Overall, the role of touch for scientific data exploration will have to be explored further. The tradeoffs of speed over precision, ease of use, and perceived immersion require further attention for scientific data exploration. Participants named several preferences for touch which warrant further analysis, such as: embodiment or a sense of feeling connected to the data, walk-up and use scenarios for different types of user groups as touch was rated “intuitive” and “easy to use,” long-term vs. short term usage scenarios in terms of “fatigue,” and new audiences for scientific data exploration as some participants described touch to be more “fun,” “refreshing,” and “innovative.”

Our study also showed areas for improvement of our techniques. First, we plan to experiment with heuristics to detect erroneous input. We also found ways to improve our chosen heuristic for determining the depth of the rotation axis for x - y -rotation which also serves as the depth of the ‘sticky’ plane for x - y -translation. It became apparent that this is an issue, in particular, in the free exploration task when people tried to rotate around the most prominent cluster of the particle cloud which was not in the center of the dataset. This means that—even if this cluster of the dataset was in the middle of the widget—a rotation is almost always performed around an axis behind or in front of the cluster, resulting in the cluster moving away from the middle of the widget. This issue can be solved for particle-based datasets by determining the rendering depth of all of the particles within the view volume before a rotation or translation interaction is initiated. Then, the average depth of these, potentially weighted by their 2D screen position, is used to determine the rotation and translation depth. For particle datasets with dense clusters this results in a stable rotation behavior according to what is shown on the screen. We implemented this technique and in our experiments it solves the mentioned issue.

Another extension we added post-study allows users to specify a center of rotation: one static touch determines the center, the other touch starts on the frame to initiate trackball or z -rotation. This technique integrates nicely with RST and frame-initiated rotations and also uses the aforementioned heuristic to determine the rotation’s z -depth.

7 CASE STUDY: BRAIN ANATOMY EXPLORATION

In this section we present a second case study to demonstrate that the concept of frame interaction with 3D spaces can also be employed in visualization domains other than particle simulation. Specifically, we show how we applied the concept to an illustrative medical visualization tool [50] and discuss some adaptations specific to this domain.

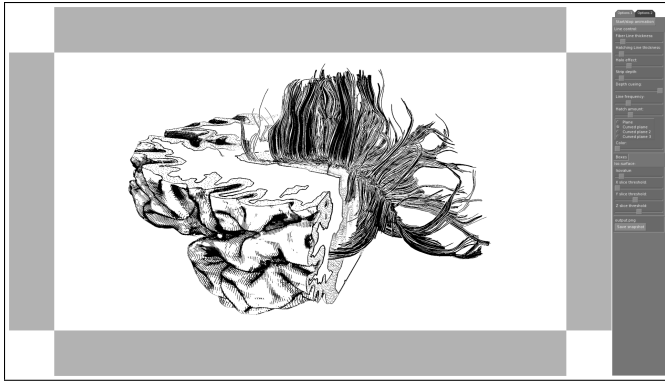


Fig. 10. Screenshot of the medical illustrative visualization application.

7.1 Adaptation of Interaction Mappings

For our medical case study we adjust the frame interaction to the exploration of an illustrative visualization of DTI fiber tracts and the surface of the brain [50] (see Fig. 10). The requirements for the interactive exploration of such medical data typically differ from those for the exploration of particle-based datasets we initially discussed in several ways. Important differences include the following:

- instead of interacting with a large space without any dedicated objects, we now have a space with a major central object (the brain) while there still exist many small sub-objects (brain fibers),
- for such medical datasets it is often easily possible to define a center point around which manipulations such as rotations are being carried out, making it unnecessary to define a heuristics for determining the center of rotation,
- navigation in terms of moving a camera through the dataset are typically not required, and manipulations can be reduced to rotations, some translations in the x - y -plane, and zooming, and
- the exploration of datasets often requires the interaction with additional elements such as the manipulation of cutting planes or the selection of sub-parts of the dataset to display or highlight.

Based on these requirements we can adjust the mappings of our interaction widget. In particular, due to rotation being more important as a transformation than translation, we switch the mappings for trackball rotation and x - y -translation. We now initiate trackball rotation for single-finger interactions started in the center of the widget, while interactions that start on the frame and move perpendicular to it, are mapped to x - y -translation. The effect of this changed mapping is that that it rather resembles Wilson et al.'s [53] physics-based interaction with a cube, as opposed to what would happen using 3D shallow-depth RNT [21]. Translation along the z -axis is no longer needed, so we removed the two dedicated regions previously used for this interaction. Zooming is still necessary so we continue to use the corners for this purpose. Two-point RST interaction, however, seem to be less important because we always rotate around the dataset's center and because most datasets have a clear up-direction, so we can map two simultaneous touches to other forms of data exploration.

In addition to these basic interactions we also need to provide mappings for two more exploration tasks: the manipulation of cutting planes and the selection of a sub-section of fiber tracts.

7.2 Manipulation of Cutting Planes

Interactions that allow people to place and manipulate cutting planes have been employed for the exploration of scientific datasets for a long time (e. g., [8, 17, 36]). This interaction is also important to understand the extracted DTI fiber tracts in relation to the brain's structure in our examples. Therefore, we explored how to allow users to manipulate axis-aligned cutting planes using direct-touch interaction on a 2D surface. This means we need to provide a means of creating cutting planes on all six sides of the dataset and to drag them along the three axes.

For this purpose we employ a technique similar to the previously used constrained rotation interaction. Touching the frame with one

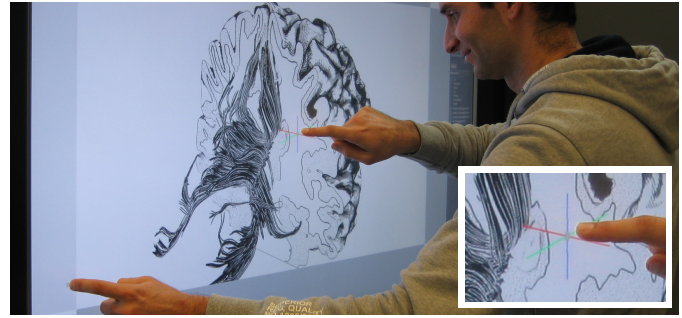


Fig. 11. Coordinate axis selection (detail) for cutting plane interaction.

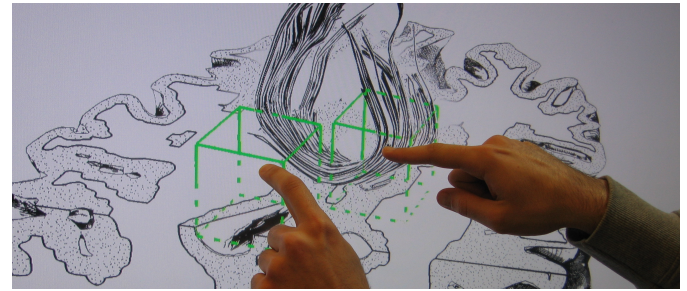


Fig. 12. Selecting fibers that pass through two regions in 3D space.

finger allows users to lock the dataset itself in place while the second finger is placed in the center of the widget to initiate the cutting plane interaction. As soon as this finger touches the center part, three colored crossing lines are shown (Fig. 11), each one in the direction of one coordinate axis, projected to screen-space. The next motion of the touching finger selects the coordinate axis along which the cutting plane will be moved depending on which of the colored lines is closest to the finger's 2D motion. This is similar to methods employed by 3D modeling tools with 2D input such as Blender to select translation directions. Afterward, motions of the second finger on the 2D surface can easily be mapped to translations of the cutting plane in 3D space.

This technique allows users to interact with three axis-aligned cutting planes, one for each of the coordinate axes. However, we also want to be able to distinguish between cutting away from the front and cutting away from the back of a dataset. To enable users to make this distinction we allow them to start cutting away from both sides and to move the cutting planes through the entire dataset. The side from which a cutting plane is started is the side that is initially cut away, but this interaction can be started from both sides. Also, if the cutting plane leaves the dataset on one side, while a user continues the interaction, a new plane is started at the opposite side.

7.3 Selection of Fiber Tracks

To be able to effectively explore fiber tract data in the context of the brain's anatomy it is often important to identify the subsection of all fiber tracts that connect two different regions of the brain. Traditionally, this is done by placing regions of interest by means of mouse and keyboard [2, 4], but mouse gestures have also been explored [1]. We support this interaction by making use of two simultaneous touches inside the exploration widget (which we previously had mapped to RST interaction). These touch locations are used to define two independent locations in 3D space through picking. Because this approach would normally only allow us to specify locations on the surface of the brain, we combine it with the previously described cutting plane interaction. This means that we determine the intersection of each picking ray with the surface of the *visible* brain section (Fig. 12), either a cutting plane intersection or the outer surface of the brain. These 3D locations are used to query the list of fiber tracts and only the ones that pass through the neighborhood of both locations are shown. When the fingers are lifted again, the set of fiber tracts that was selected last continues to be shown, so that further interaction with the cutting planes can be used

to reveal further spatial relationships of fibers to brain anatomy.

As an alternative to the two-touch fiber selection we also experimented with a dedicated interaction with the regions that select the fibers. Here we use relatively small axis-aligned boxes to select fibers of interest. To re-position these boxes users touch one of the sides of the box facing the viewer, specifying one of the dataset's main axes. The selected side is highlighted and 2D motions of the touching finger can now be mapped to translation of the box along the specified axis. This technique has the advantage that it is highly accurate and each box is independent. This means that one can specify more than one region of interest to show fibers that are passing through them. Nevertheless, it takes careful positioning to place a box in 3D space as intended.

7.4 Informal Evaluation

We conducted an informal evaluation to understand the usefulness of touch and frame interaction in this second context, in particular also compared to software packages that are normally used for brain and fiber exploration (e.g., TrackVis [51]). For this purpose we invited a neuroscientist to our lab who has experience working with tools like TrackVis. We started by explaining the frame interaction technique and then asked him to try it himself. While he was working with the tool, we asked for comments and observed how fast he was able to learn to use the interaction techniques.

The first thing the participant remarked was that he was missing two-point RST interaction, pointing to the screen with his two arms and suggesting a rotation and zooming motion. This is interesting and only reinforces our earlier point of including RST for general frame interaction. Despite the expected RST interaction not being available in this tool, however, the neuroscientist was able to interact with the fiber tract visualization immediately and did not need to be reminded of the mappings. He commented that he liked the idea of using the frame and that it was very easy to use. The neuroscientist particularly liked the way of manipulating the cutting planes and compared this technique to TrackVis, saying that TrackVis has three small viewports with extra sliders to control the slices and that he appreciated that our technique does not require such extra windows. He also enjoyed the possibility to explore fiber tracts that connect two regions by selecting them with two simultaneous touches. However, he disliked our second technique of moving the selection boxes by touching their sides and 'pulling' or 'pushing' them. He compared this technique to the software packages he is used to which often have spheres that can be dragged parallel to the view plane. He said that this view plane motion is easier and more intuitive to use than dragging the boxes one axis at a time.

In summary, he said that the interaction techniques are intuitive to use and that he appreciates the ability to work with data on a single large viewport without much clutter while all important interactions are possible. Also, he suggested to investigate collaborative scenarios since scientists usually work together when analyzing fiber tract data so that touch interaction on a wall display as we use it would be beneficial. While the results from one participant certainly cannot be generalized, it still gives some evidence for the applicability of frame-based direct-touch interaction with visualizations of medical 3D datasets.

8 CONCLUSION

In this paper we presented FI3D, a design study for enabling direct-touch interaction with three-dimensional scientific visualization spaces. FI3D is based on using the frame around a centered visualization and spring-loaded modes to enable users to navigate the 3D space in 7 DOF and requires in its basic form only single-touch interaction (goals G1, G4, G7). It differs from previous techniques in that it allows interaction with the space itself and does not require large objects to be present to constrain the mapping (goals G2 and G3). If more simultaneous touches are available, however, we demonstrated that the technique can also support constrained interactions (G6). We discussed the application of the technique to two different scientific visualization domains to demonstrate its generality and extensibility (G5 and G8). In addition, we reported on a user study comparing FI3D to common mouse-and-keyboard interaction. This study showed that FI3D was competitive for translation and integrated RST interaction (G10)

while being slower for rotation and zoom. The latter effect, however, we attribute largely to technical issues with the specific touch sensing hardware we used. Moreover, our study also showed a clear preference of participants to use touch interaction for the exploration task, e.g., because of the immersion and control it provides and that the technique was easy to learn and use (G9).

With a more reliable touch sensing such as FTIR [20] or IFSR [43], we believe, we can also be competitive in the domains where mouse showed to perform significantly better than touch in our study. In the future, we therefore would like to test our interaction on such devices which would also allow us to provide several simultaneously usable widgets to explore co-located collaboration applications or other interaction mappings. However, we believe that our simple set of interactions makes the interface easier to master because less gesture configurations have to be remembered. On the other hand, more concurrent touch points or the recognition of the shape of the touch would also make it possible to extend the interaction vocabulary. For example, we could use a more sophisticated scale interaction than the simple zooming that we currently use (e.g., Fu et al.'s [18] powers-of-10 ladder) or provide means to select subsets of the dataset. It may also be interesting to explore how touch interaction with 3D spaces, using FI3D or other techniques, can be supported with tilted rather than completely vertical or horizontal surfaces. This would allow us to address the fatigue issues that arise from wall interaction (in our study, when asked about if touch made them feel tired, 3 participants said yes and 8 said somewhat). Finally, we would like to investigate the assumed improved understanding that resulted from participants feeling more immersed in the data using direct-touch interaction.

ACKNOWLEDGMENTS

From the University of Groningen, we would like to thank Amina Helmi for the cosmological simulation dataset from the Aquarius Project [47], Frans van Hoesel for the galaxy collision simulation, and Leonardo Cerliani for the brain dataset.

REFERENCES

- [1] D. Akers. Wizard of Oz for Participatory Design: Inventing a Gestural Interface for 3D Selection of Neural Pathway Estimates. In *CHI Extended Abstracts*, pp. 454–459, New York, 2006. ACM. doi: 10.1145/1125451.1125552
- [2] D. Akers, A. Sherbondy, R. Mackenzie, R. Dougherty, and B. Wandell. Exploration of the Brain's White Matter Pathways with Dynamic Queries. In *Proc. Visualization*, pp. 377–384, Los Alamitos, 2004. IEEE Computer Society. doi: 10.1109/VISUAL.2004.30
- [3] R. Bade, F. Ritter, and B. Preim. Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3D Rotation. In *Proc. Smart Graphics*, pp. 138–150, Berlin/Heidelberg, 2005. Springer Verlag. doi: 10.1007/11536482_12
- [4] J. Blaas, C. P. Botha, B. Peters, F. M. Vos, and F. H. Post. Fast and Reproducible Fiber Bundle Selection in DTI Visualization. In *Proc. Visualization*, pp. 59–64, Los Alamitos, 2005. IEEE Computer Society. doi: 10.1109/VIS.2005.40
- [5] R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. In *Proc. CHI*, pp. 519–526, New York, 2004. ACM. doi: 10.1145/985692.985758
- [6] D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, Boston, 2005.
- [7] S. Bryson. Virtual Reality in Scientific Visualization. *Communications of the ACM*, 39(5):62–71, May 1996. doi: 10.1145/229459.229467
- [8] M. Clifton and A. Pang. Cutting Planes and Beyond. *Computers & Graphics*, 21(5):563–575, May 1997. doi: 10.1016/S0097-8493(97)00036-8
- [9] L. D. Cutler, B. Fröhlich, and P. Hanrahan. Two-Handed Direct Manipulation on the Responsive Workbench. In *Proc. SIG3D*, pp. 107–114, New York, 1997. ACM. doi: 10.1145/253284.253315
- [10] J.-B. de la Rivière, C. Kervégant, E. Orvain, and N. Dittlo. CubTile: A Multi-Touch Cubic Interface. In *Proc. VRST*, pp. 69–72, New York, 2008. ACM. doi: 10.1145/1450579.1450593
- [11] J. Edelmann, S. Fleck, and A. Schilling. The DabR – A Multitouch System for Intuitive 3D Scene Navigation. In *Proc. 3DTV*, Piscataway, NJ, USA, 2009. IEEE. doi: 10.1109/3DTV.2009.5069671

- [12] C. Forlines, A. Esenther, C. Shen, D. Wigdor, and K. Ryall. Multi-User, Multi-Display Interaction with a Single-User, Single-Display Geospatial Application. In *Proc. UIST*, pp. 273–276, New York, 2006. ACM. doi: 10.1145/1166253.1166296
- [13] C. Forlines and R. Lilien. Adapting a Single-User, Single-Display Molecular Visualization Application for Use in a Multi-User, Multi-Display Environment. In *Proc. AVI*, pp. 367–371, New York, 2008. ACM. doi: 10.1145/1385569.1385635
- [14] C. Forlines and C. Shen. DTLens: Multi-User Tabletop Spatial Data Exploration. In *Proc. UIST*, pp. 119–122, New York, 2005. ACM. doi: 10.1145/1095034.1095055
- [15] C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-Touch vs. Mouse Input for Tabletop Displays. In *Proc. CHI*, pp. 647–656, New York, 2007. ACM. doi: 10.1145/1240624.1240726
- [16] M. Frisch, J. Heydekorn, and R. Dachsel. Investigating Multi-Touch and Pen Gestures for Diagram Editing on Interactive Surfaces. In *Proc. ITS*, pp. 149–156, New York, 2009. ACM. doi: 10.1145/1731903.1731933
- [17] B. Fröhlich, J. Plate, J. Wind, G. Wesche, and M. Göbel. Cubic-Mouse-Based Interaction in Virtual Environments. *IEEE Computer Graphics and Applications*, 20(4):12–15, July/Aug. 2000. doi: 10.1109/38.851743
- [18] C.-W. Fu, W.-B. Goh, and J. A. Ng. Multi-Touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations. In *Proc. CHI*, pp. 2213–2222, New York, 2010. ACM. doi: 10.1145/1753326.1753661
- [19] M. Hachet, F. Declé, S. Knödel, and P. Guittou. Navidget for Easy 3D Camera Positioning from 2D Inputs. In *Proc. 3DUI*, pp. 83–89, Los Alamitos, 2008. IEEE Computer Society. doi: 10.1109/3DUI.2008.4476596
- [20] J. Y. Han. Low-Cost Multi-Touch Sensing Through Frustrated Total Internal Reflection. In *Proc. UIST*, pp. 115–118, New York, 2005. ACM. doi: 10.1145/1095034.1095054
- [21] M. Hancock, S. Carpendale, and A. Cockburn. Shallow-Depth 3D Interaction: Design and Evaluation of One-, Two- and Three-Touch Techniques. In *Proc. CHI*, pp. 1147–1156, New York, 2007. ACM. doi: 10.1145/1240624.1240798
- [22] M. Hancock, T. ten Cate, and S. Carpendale. Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables. In *Proc. ITS*, pp. 145–152, New York, 2009. ACM. doi: 10.1145/1731903.1731930
- [23] M. S. Hancock, S. Carpendale, F. D. Vernier, D. Wigdor, and C. Shen. Rotation and Translation Mechanisms for Tabletop Interaction. In *Proc. Tabletop*, pp. 79–88, Los Alamitos, 2006. IEEE Computer Society. doi: 10.1109/TABLETOP.2006.26
- [24] E. Hornecker, P. Marshall, N. S. Dalton, and Y. Rogers. Collaboration and Interference: Awareness with Mice or Touch Input. In *Proc. CSCW*, pp. 167–176, New York, 2008. ACM. doi: 10.1145/1460563.1460589
- [25] T. Isenberg, M. Everts, J. Grubert, and S. Carpendale. Interactive Exploratory Visualization of 2D Vector Fields. *Computer Graphics Forum*, 27(3):983–990, May 2008. doi: 10.1111/j.1467-8659.2008.01233.x
- [26] Y. Jung, J. Keil, J. Behr, S. Webel, M. Zöllner, T. Engelke, H. Wuest, and M. Becker. Adapting X3D for Multi-Touch Environments. In *Proc. Web3D*, pp. 27–30, New York, 2008. ACM. doi: 10.1145/1394209.1394218
- [27] D. F. Keefe. Integrating Visualization and Interaction Research to Improve Scientific Workflows. *IEEE Computer Graphics and Applications*, 30(2):8–13, Mar./Apr. 2010. doi: 10.1109/MCG.2010.30
- [28] K. Kin, M. Agrawala, and T. DeRose. Determining the Benefits of Direct-Touch, Bimanual, and Multifinger Input on a Multitouch Workstation. In *Proc. Graphics Interface*, pp. 119–124, Toronto, 2009. CIPS.
- [29] R. Kosara, H. Hauser, and D. L. Gresh. An Interaction View on Information Visualization. In *Eurographics State-of-the-Art Reports*, pp. 123–137, Aire-la-Ville, Switzerland, 2003. Eurographics.
- [30] R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid Integration of Rotation and Translation. In *Proc. CHI*, pp. 601–610, New York, 2005. ACM. doi: 10.1145/1054972.1055055
- [31] W. Krüger and B. Fröhlich. The Responsive Workbench. *IEEE Computer Graphics and Applications*, 14(3):12–15, May 1994. doi: 10.1109/38.279036
- [32] J. Liu, D. Pinelle, S. Sallam, S. Subramanian, and C. Gutwin. TNT: Improved Rotation and Translation on Digital Tables. In *Proc. Graphics Interface*, pp. 25–32, Toronto, 2006. CIPS.
- [33] I. S. MacKenzie and S. Riddersma. Effects of Output Display and Control-Display Gain on Human Performance in Interactive Systems. *Behaviour & Information Technology*, 13(5):328–337, 1994. doi: 10.1080/01449299408914613
- [34] A. Martinet, G. Casiez, and L. Grisoni. 3D Positioning Techniques for Multi-Touch Displays. In *Proc. VRST*, pp. 227–228, New York, 2009. ACM. doi: 10.1145/1643928.1643978
- [35] A. Martinet, G. Casiez, and L. Grisoni. Design and Evaluation of 3D Positioning Techniques for Multi-touch Displays. Technical Report RR-7015, INRIA, France, 2009.
- [36] T. Meyer and A. Globus. Direct Manipulation of Isosurfaces and Cutting Planes in Virtual Environments. Technical Report CS-93-54, Brown University, Providence, RI, USA, Dec. 1993.
- [37] M. A. Nacenta, P. Baudisch, H. Benko, and A. Wilson. Separability of Spatial Manipulations in Multi-Touch Interfaces. In *Proc. Graphics Interface*, pp. 175–182, Toronto, 2009. CIPS.
- [38] M. Nijboer, M. Gerl, and T. Isenberg. Exploring Frame Gestures for Fluid Freehand Sketching. In *Proc. SBIM*, pp. 57–62, Aire-la-Ville, Switzerland, 2010. Eurographics. doi: 10.2312/SBM/SBM10/057-062
- [39] C. North, T. Dwyer, B. Lee, D. Fisher, P. Isenberg, K. Inkpen, and G. Robertson. Understanding Multi-touch Manipulation for Surface Computing. In *Proc. Interact*, pp. 236–249, Berlin/Heidelberg, 2009. Springer Verlag. doi: 10.1007/978-3-642-03658-3_31
- [40] W. A. Pike, J. Stasko, R. Chang, and T. A. O’Connell. The Science of Interaction. *Information Visualization*, 8(4):263–274, Winter 2009. doi: 10.1057/ivs.2009.22
- [41] J. L. Reisman, P. L. Davidson, and J. Y. Han. A Screen-Space Formulation for 2D and 3D Direct Manipulation. In *Proc. UIST*, pp. 69–78, New York, 2009. ACM. doi: 10.1145/1622176.1622190
- [42] G. Robles-De-La-Torre. The Importance of the Sense of Touch in Virtual and Real Environments. *IEEE MultiMedia*, 13(3):24–30, July–Sept. 2006. doi: 10.1109/MMUL.2006.69
- [43] I. Rosenberg and K. Perlin. The UnMousePad: An Interpolating Multi-Touch Force-Sensing Input Pad. *ACM Transactions on Graphics*, 28(3):65:1–65:9, Aug. 2009. doi: 10.1145/1531326.1531371
- [44] D. Schroeder, D. Coffey, and D. F. Keefe. Drawing with the Flow: A Sketch-Based Interface for Illustrative Visualization of 2D Vector Fields. In *Proc. SBIM*, pp. 49–56, Aire-la-Ville, Switzerland, 2010. Eurographics. doi: 10.2312/SBM/SBM10/049-056
- [45] K. Shoemake. ARCBALL: A User Interface for Specifying Three-Dimensional Orientation Using a Mouse. In *Proc. Graphics Interface*, pp. 151–156, San Francisco, 1992. Morgan Kaufmann Publishers Inc.
- [46] Smart Technologies Inc. Digital Vision Touch Technology. White paper, Feb. 2003.
- [47] V. Springel, J. Wang, M. Vogelsberger, A. Ludlow, A. Jenkins, A. Helmi, J. F. Navarro, C. S. Frenk, and S. D. M. White. The Aquarius Project: The Subhalos of Galactic Halos. *Monthly Notices of the Royal Astronomical Society*, 391(4):1685–1711, Dec. 2008. doi: 10.1111/j.1365-2966.2008.14066.x
- [48] M. A. Srinivasan and C. Basdogan. Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. *Computers & Graphics*, 21(4):393–404, July/Aug. 1997. doi: 10.1016/S0097-8493(97)00030-7
- [49] F. Steinicke, K. H. Hinrichs, J. Schöning, and A. Krüger. Multi-Touching 3D Data: Towards Direct Interaction in Stereoscopic Display Environments coupled with Mobile Devices. In *Proc. AVI Workshop on Designing Multi-Touch Interaction Techniques for Coupled Public and Private Displays*, pp. 46–49, 2008.
- [50] P. Svetachov, M. H. Everts, and T. Isenberg. DTI in Context: Illustrating Brain Fiber Tracts In Situ. *Computer Graphics Forum*, 29(3):1024–1032, June 2010. doi: 10.1111/j.1467-8659.2009.01692.x
- [51] R. Wang, T. Benner, A. G. Sorensen, and V. J. W. Wedeen. Diffusion Toolkit: A Software Package for Diffusion Imaging Data Processing and Tractography. In *Proc. ISMRM*, volume 15, p. 3720, 2007.
- [52] A. D. Wilson. Simulating Grasping Behavior on an Imaging Interactive Surface. In *Proc. ITS*, pp. 137–144, New York, 2009. ACM. doi: 10.1145/1731903.1731929
- [53] A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing Physics to the Surface. In *Proc. UIST*, pp. 67–76. ACM, 2008. doi: 10.1145/1449715.1449728
- [54] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, Nov./Dec. 2007. doi: 10.1109/TVCG.2007.70515
- [55] R. Zeleznik and A. Forsberg. UniCam—2D Gestural Camera Controls for 3D Environments. In *Proc. I3D*, pp. 169–173, New York, 1999. ACM. doi: 10.1145/300523.300546