

TOUCHING 3D DATA

INTERACTIVE VISUALIZATION OF COSMOLOGICAL SIMULATIONS

LINGYUN YU

CONTENTS

1	INTRODUCTION	1
2	RESEARCH MOTIVATION AND DIRECTIONS	7
2.1	Research Questionnaire	7
2.1.1	Astronomical Data	7
2.1.2	Analysis Process	8
2.2	Challenges for Scientific Visualization	9
2.2.1	Navigation in 3D space	9
2.2.2	Selection in 3D space	10
2.2.3	High-dimensional datasets	12
2.3	Research Directions	13
3	FI3D: DIRECT-TOUCH INTERACTION FOR THE EXPLORATION OF 3D SCIENTIFIC VISUALIZATION SPACES	15
3.1	Introduction	16
3.2	Related Work	17
3.2.1	Interactive Visualization	17
3.2.2	Direct-Touch Interaction with 3D Environments	19
3.3	FI3D: A New Technique for Interaction with 3D Spaces	21
3.3.1	From 2D Planes to 3D Space Manipulation	22
3.3.2	Frame Interaction for 3D Visualization	24
3.4	User Study	27
3.5	Results	31
3.6	Discussion and Lessons Learned	34
3.7	Case Study: Brain Anatomy Exploration	36
3.7.1	Adaptation of Interaction Mappings	36
3.7.2	Manipulation of Cutting Planes	37
3.7.3	Selection of Fiber Tracks	38
3.7.4	Informal Evaluation	39
3.8	Conclusion	40
3.9	From 3D Space Navigation Interaction To 3D Data Selection	41
4	STRUCTURE-AWARE SELECTION	43
4.1	Introduction	43
4.2	Related Work	45
4.3	TeddySelection	47
4.3.1	Input Polygon Triangulation	47
4.3.2	Particle Mapping to Triangles	48
4.3.3	Construction of the Selection Mesh	48

4.3.4	Example Results	50
4.3.5	Performance	51
4.3.6	User Feedback on TeddySelection	51
4.4	CloudLasso	53
4.4.1	Density Estimation	53
4.4.2	Surface Extraction	56
4.4.3	Threshold Adjustment	57
4.4.4	Example Results	57
4.4.5	Performance	58
4.5	User Study	58
4.5.1	Study Description	59
4.5.2	Results	62
4.6	Discussion	65
4.6.1	Comparison of the Selection Techniques	65
4.6.2	Limitations	67
4.6.3	Other Applications and Possible Modifications	67
4.6.4	Further Selection Processing	69
4.7	Conclusion	69
4.8	From Interaction Techniques To A Visual Analytics Tool	70
5	A VISUAL ANALYTICS TOOL FOR DATA EXPLORATION ON A LARGE, TOUCH-SENSITIVE DISPLAY	71
5.1	Introduction	71
5.2	Related Work	73
5.2.1	Visualization of High-Dimensional Data	73
5.2.2	Interactive Exploration Tools	74
5.3	A Visual Analytics Tool for Data Exploration	75
5.3.1	Design Requirements	75
5.3.2	Design and Interactions	77
5.4	Evaluation	79
5.5	Summary and Future Plans	83
6	CONCLUSIONS & FUTURE WORK	85
6.1	Navigation in 3D Space	85
6.2	Structure-Aware Selection	86
6.3	Integrated Visual Analytics System	86
6.4	Future Work	87
6.4.1	Improved Interaction Techniques	87
6.4.2	Collaboration	88
6.4.3	Demonstration	89
6.4.4	Visualization of Dynamical Evolution	89
6.5	Conclusion	90
	BIBLIOGRAPHY	93

PUBLICATIONS 105

INTRODUCTION

BABIES start interacting with the world by opening their eyes and looking around. Their eyes wander until a face or a brightly colored toy attracts their attention. They use their hands and fingers to touch objects; feel whether they are smooth or hard, cold or warm. When this is not enough they put things into their mouth, chew them, taste them, and, as every parent knows, finally spit them out (Figure 1.1).

Scientists, when confronted with the task of making sense of their data, face similar problems as babies. They have to make sense of a new, abstract, often high-dimensional world using their already developed senses, senses that evolved to deal with very different environments. Nevertheless, these senses, together with their imagination, is all that a scientist has that can serve as a window to this abstract world made up of data. Scientists, much like babies, start by first trying to see their data. Then they focus on the parts that they find more interesting. They try to see their data from different points of view and often have to change focus. When something attracts their interest it has to be filtered and analyzed independently. At every step of this analysis, the scientist must make decisions on how to proceed, decisions that are informed by previous results and experiences.



Figure 1.1: A baby puts a toy into his mouth.



Figure 1.2: A scientist trying to make sense of data.

Scientific visualization aims to augment scientists' senses and imagination in order to help them better understand their data. This is done by assisting in two ways. First, by offering methods of visualizing the data that are intuitive and can easily reveal sought out structures. Second, by improving the iterative, interactive procedure of filtering and analysis. Scientific visualization is thus not only about visualization *per se*, but also about user interaction, and about methods for identifying and analyzing interesting parts of data.

Data visualization is today inextricable from, even unimaginable without, the use of computers. These provide users not only with static images, but also with animations and with virtual 3D environments simulating the physical world. Furthermore, they provide users the ability to interact with their data which they can select, filter, and analyze, often in real time. Interactivity has proved its importance in data exploration as interactive visualizations help users to 'play' with the data and receive immediate feedback. This aspect of visualization significantly improves the ability to understand the presented data and to generate new insights.

The research on scientific visualization tries to improve the visualization and interaction methods that exist today but also to imagine completely new ones. As the amount and detail of data produced from a sensor or in a typical numerical simulation increases more challenges are presented and have to be overcome. Note also that different scientific domains call for different visualization techniques and different types of interaction.

Large displays with touch-based input present new opportunities and challenges for scientific visualization. “Momentum recently seems to be increasing” toward using large displays to explore data in scientific visualizations, according to Keefe [2010]. On the one hand, large-size, high-resolution displays not only enable users to feel more immersed in the presented data, but also support demonstrations to wide audiences and sharing innovative thoughts with other cooperators. These displays often come in the form of tabletops or wall displays. On the other hand, direct-touch techniques have already made a significant impact in user interaction paradigms and are now the dominant interaction method for small portable devices. Their main benefit is that they delight users by giving them the feeling of “having the data under their finger tip”. Touch-based interfaces provide new data analysis platforms that can encourage alternative forms of scientific data exploration and promote the use of scientific visualization techniques even by non-experts. Direct-touch interaction techniques, for example, navigation and manipulation techniques in 3D space, have been widely explored in recent years. They have also been the focus of previous research projects in the visualization context, but much remains to be learned about applying direct-touch interaction to scientific visualization.

This thesis aims to study the challenges for natural interactions in data exploration tasks in scientific visualization domains that concern themselves with data which is inherently represented three-dimensionally. We present two intuitive and efficient interaction techniques for data exploration in 3D space. The first technique helps users to navigate 3D spaces and the second technique provides users the ability of easily selecting a subset of particle data. Finally, we incorporate these techniques into a visual analytics system in order to help scientists extract potentially useful information and gain insight. Our techniques can be applied in various scientific domains. The type of data central to this thesis as an exemplary application is astronomical simulations—3D point cloud data resulting from numerical simulations, such as simulations of galactic dynamics or high-dimensional information recorded in the form of particle datasets. In the rest of this chapter we briefly discuss the contents of the chapters that follow.

In order to uncover the main requirements of astronomical visualizations we created a research questionnaire addressed to practitioners in the field. The questionnaire investigated the main results that astronomers try to obtain by exploring their data, what types of data they deal with, what the current visualization, interaction, and analysis techniques that they use are, and finally what difficulties exist with their current analysis procedures. We used the results of this questionnaire, presented in Chapter 2, as motivation and roadmap for the present work.

One of the first problems pointed out by this questionnaire is navigation of 3D datasets. Chapter 3 introduces the design of *FI3D* (Figure 1.3),

a direct-touch interaction technique for the exploration of 3D scientific visualization spaces. The main idea behind FI3D is to map the 2-DOF input provided by touch displays to the 7-DOF (translation, rotation, and zoom) necessary for the navigation and visual exploration of 3D data. We conducted a study to compare the technique to conventional mouse-based interaction. In addition, we present a second case study, adapting our interaction technique to the illustrative visualization of brain fiber tracts. FI3D provides both large-scale and precise navigation capabilities for exploring 3D scientific data. This is of great assistance especially for understanding the structure of discrete point-based datasets, such as 3D point cloud data which consist of thousands or millions of particles and which are common in astronomy.

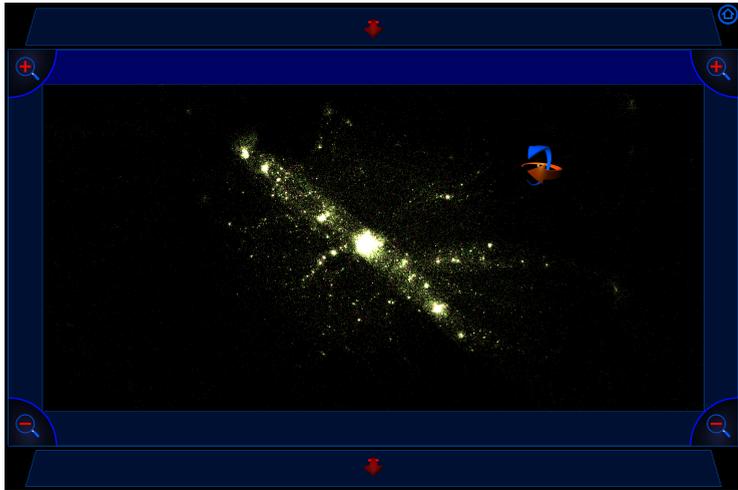


Figure 1.3: FI3D: A frame based, direct-touch interaction technique.

After navigation, data selection is also essential in data exploration because it serves as a prerequisite to many follow-up interactions. [Chapter 4](#) focuses on selection techniques for 3D point cloud visualizations. Selecting a subset from a 3D particle dataset is especially challenging since the user cannot select particles one-by-one. The two structure-aware selection techniques we present here, TeddySelection ([Figure 1.4a](#)) and CloudLasso ([Figure 1.4b](#)), intuitively select the target particles without the need of drawing a precise lasso. These methods exploit the structure of the particle dataset in order to infer a selection that matches the user's intention. Furthermore, these techniques help to uncover the structure of the dataset by focusing on the most dense parts. We compare these structure-aware techniques to the standard cylinder-based selection technique and we find that they outperform the latter in terms of speed, accuracy, and ease of use. In addition, we

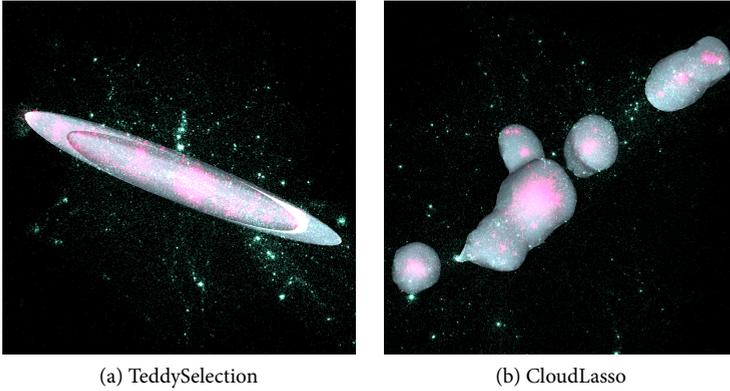


Figure 1.4: Two structure-aware selection techniques.

show that our methods can easily be applied to the analysis of abstract, high-dimensional datasets.

Then, in [Chapter 5](#) we introduce an integrated visual analytics tool on large, touch-sensitive displays ([Figure 1.5](#)). The purpose of the tool is to facilitate the analysis of high-dimensional datasets. In particular, the tool offers a visual, interactive interface for selecting low-dimensional subspaces from a high-dimensional dataset in such a way that the most important physical information is revealed [[Ferdosi et al., 2010a](#)]. In [Chapter 5](#), the detailed design of this tool is presented. The tool incorporates the FI3D navigation technique ([Chapter 3](#)) and the CloudLasso selection technique ([Chapter 4](#)). In addition, an observational study was carried out to evaluate our visual analytics tool in the context of astronomical data exploration and its results are presented.

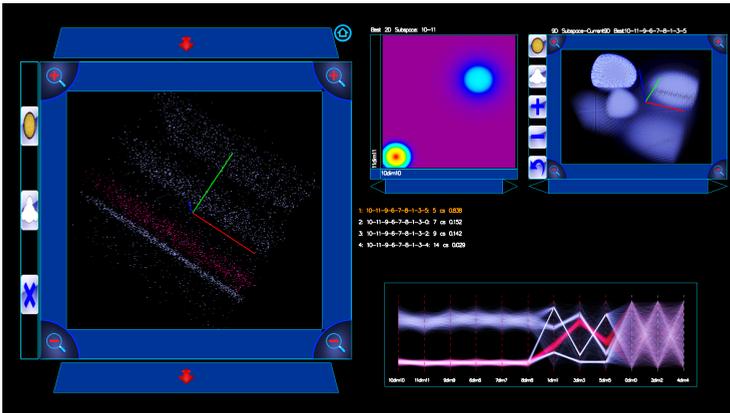


Figure 1.5: An integrated visual analytics tool.

Finally, [Chapter 6](#) provides a summary and discussion with general conclusions followed by an outlook on future work.

INTERACTIVE visualization is an iterative process that involves information representation, interactive exploration, and decision making. The goal is, through this iterative process, to gain insight into the problem or data until it has been understood. A high level of interactivity in visual exploration is essential for efficiently achieving this goal since it allows the user to try out new ideas, receive immediate feedback, and accordingly adjust the exploration.

To better understand the scientists' requirements for interactive visualization in their application domain we created, at the beginning of our research, a questionnaire. The questionnaire was directed to astronomy experts. The replies provided by these experts highlighted several problems in current visualization tools and interaction practices and assisted us in directing our research. We chose the domain of astronomy as our research target because it contains typical challenges which also exist in other scientific fields. Furthermore, the close collaboration with the Kapteyn Astronomical Institute in Groningen gave us the opportunity to have direct access to astronomers. Thus we were able to discuss with them about their requirements and the main challenges they face in their research in terms of visualization and data analysis.

2.1 RESEARCH QUESTIONNAIRE

Ten astronomy experts (four female, six male) from the Kapteyn Astronomical Institute of the University of Groningen participated in the survey. On average, they had $3\frac{1}{2}$ years of experience in astronomy after obtaining their Master's degree. The participants were asked to complete a written questionnaire and we collected and summarized their answers. The experts were asked in the questionnaire what kind of data they work with and what kind of information they try to extract from the data. Furthermore, they were asked to describe their analysis process, the difficulties that they meet during the analysis, and the tools and techniques that they use together with their shortcomings. Below we provide a summary of the participants' answers.

2.1.1 *Astronomical Data*

All of the participants reported that they have experience of working with various types of astronomical data, including both observations and simulations. The astronomical datasets usually contain millions

of objects, and each object contains hundreds of associated parameters such as spatial positions of stars or galaxies, velocities, densities, luminosities, and so on. All participants reported that they are used to working with multi-dimensional abstract data whose dimensionality varies from 3 to 25.

The exploration of such high-dimensional astronomical datasets raises several challenges for astronomers. First, it is not possible to simultaneously visualize all data in all dimensions. What the astronomers usually do is to use their experience to select a subset of the data / dimensions which they believe contain relevant astronomical information. However, the participants also reported that, by only relying on their intuition or guessing, interesting but unexpected information can be missed.

Second, researchers are often interested in physical properties of whole regions instead of just those of one single particle. However, it is challenging to pick a location spatially and specify the range of interest in 3D space.

Third, astronomical spaces span large spatial scales but most of the space is either empty or is occupied by low-density regions. The most dense regions, which also have the greatest astronomical significance, occupy comparatively small volume and are often occluded by lower density regions. This makes it difficult for the user to locate such dense regions and find an appropriate view direction.

2.1.2 *Analysis Process*

We now describe the individual steps that the astronomers perform during the analysis process.

The first step of the analysis is to visualize the data. Nine out of the 10 participants reported that 3D scatter plots are their most common visualization technique. In some cases, for the visualization of such high dimensional data the astronomers use Principal Components Analysis (PCA) [Jolliffe, 1986] or Parallel Coordinate Plots (PCP) [Inselberg, 2009].

The second analysis step is to observe the data from different perspectives. This does not only mean to rotate or scale 3D data in order to choose an appropriate viewing direction. It also means to select the right combination of dimensions to visualize, or to select a physically relevant subset of the data. The appropriate interactions that can be used for this purpose usually depend on the different exploration purposes. For example, the usual techniques for obtaining a clear visualization of 3D data include, but are not restricted to, translation, rotation, and scaling in the 3D space. This was corroborated by survey participants who reported that they first observe the whole dataset and then rotate,

translate, and zoom in the 3D scatter plot in order to find an interesting region.

Furthermore, as we mentioned before, it is challenging to present all available information simultaneously. Therefore, the high dimensionality of the data forces the scientists to select a small number of dimensions (2 or 3) to visualize. This is also related to the fact that another goal of the data analysis process is to find correlations, clusters, outliers, and linked (physical) properties from various attributes of galaxies. Thus, it is important for the astronomers to be able to identify the essential dimensions from multi-dimensional data, i. e., those dimensions whose visualization can provide the most information. Normally, astronomers use their experience to make a judgement on which dimensions are essential based on known relations between different physical parameters. Nevertheless, they find it difficult to uncover relations between different dimensions when they do not know *a priori* that such relations exist.

The next step is to focus on an identified interesting region. The purpose here is to select objects of interest from the large-scale environment in order to further study their physical properties and analyze their structure and context. The selection can be based on 3D spatial positions or other parameters that are part of the dataset. This step might be repeated several times until a good selection is obtained. However, astronomers are also used to selecting interesting information from data tables and then plotting the information to the screen. The disadvantage of this approach is that it does not provide immediate visual feedback and can be tedious.

2.2 CHALLENGES FOR SCIENTIFIC VISUALIZATION

The analysis of astronomical data, as presented in the previous section, presents a lot of challenges related to visualization and user interaction. Identifying and addressing these challenges is essential for providing astronomers with powerful and intuitive tools to assist in astronomical data analysis and has therefore been the basis of our research program. In the remainder of this section we identify the main problems and give a brief overview of previous work to address them.

2.2.1 *Navigation in 3D space*

The most basic type of visualization used in astronomy are scatter plots of the positions of different objects, such as stars or galaxies, in 3D space. Such visualizations are useful in order to uncover the spatial structure of the dataset and to associate other physical parameters to positions in space. Efficient and intuitive navigation in 3D space is therefore a main requirement. The challenge here is how to map the 2D input

provided by devices, such as mice and touch-screens, to the multitude of interactions that are required for efficient 3D space navigation. Such mapping should be easily learned and remembered and, also, it should provide precise 3D navigational control.

The main problem is that, on the one hand, navigation in 3D space typically needs at least seven interactions: three for translation, three for rotation, and one for (uniform) scaling. On the other hand, a single touch on a screen provides only two degrees of freedom. This is enough for translation in 2D space but it is clearly inadequate for the full range of navigation interactions necessary in 3D space. One way to bridge the gap is by using additional menus or extra buttons to switch from one type of interaction to another, for example, switching from translation to rotation. This is the solution that is most commonly employed in desktop applications working with traditional input devices such as keyboard and mouse. For example, a specific interaction mode can be activated by clicking an on-screen button. This approach, although it can be also used on touch-screens, makes it harder for users to remember which interaction type is active and to switch to the appropriate type every time. Another approach, in principle suitable for touch-screens, is to introduce gestural interactions, such as the one-touch rotation-and-translation (RNT) technique [Kruger et al. \[2005\]](#). By “gestural” interactions we refer here to gestures, postures, and quasi-postures; see [\[Isenberg and Hancock, 2012\]](#) for precise definitions and a thorough discussion of their respective differences. Nevertheless, there does not seem to be a set of natural 2D gestural interactions on a 2D touch-screen that covers all possible interactions in 3D space. This leaves the user with the difficult task of learning and remembering mappings that do not have a real-world correspondence. An approach that is commonly used in desktop applications is to activate different spring-loaded interaction modes [\[Buxton, 1986; Sellen et al., 1992\]](#) by keeping pressed a keyboard modifier key. A similar approach on touch-screens is to use spring-loaded modes that are active only while a button is being pressed. Such approach avoids the disadvantages of other methods. On the one hand, the user does not need to remember the active mode because the latter remains active only while the user presses a button. On the other hand, the buttons to activate the spring-loaded modes can be labeled so that the corresponding mode is clear. Therefore, this latter approach appears to be the most promising.

2.2.2 Selection in 3D space

After navigation, the next interaction technique that is necessary for data exploration is selection since it provides the basis for subsequent analysis steps. In astronomical datasets filtering based on physical properties may not work because often one does not know what to filter for

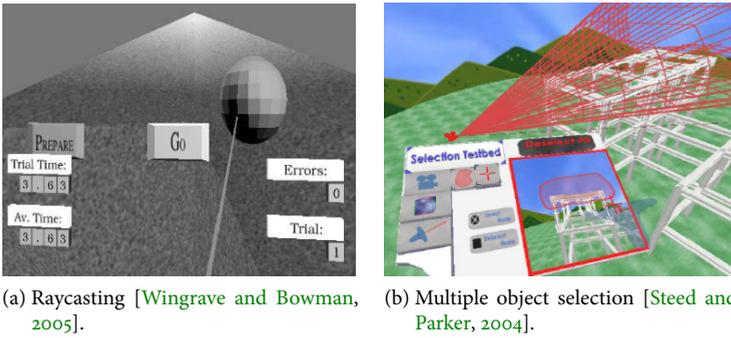


Figure 2.1: Simple selection techniques.

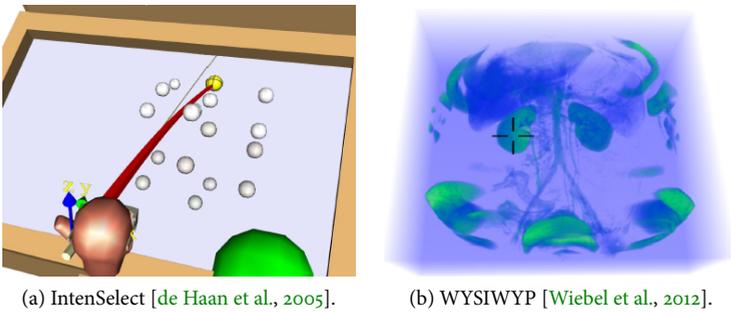


Figure 2.2: Advanced selection techniques

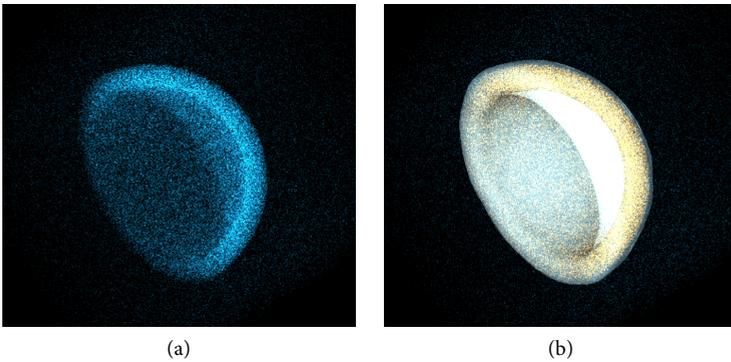


Figure 2.3: Identifying the structure of a particle cloud using the CloudLasso selection method [Yu et al., 2012].

or the necessary data for filtering may not be available. In such cases, *spatial* selection of particles may be the only possibility.

Selecting 2D objects on a 2D surface is fairly straightforward since all 2D objects are fully visible and accessible for interaction. In 2D, objects can be selected by clicking on the target or enclosing the targets with a rectangle, circle, or freeform lasso.

The most straightforward selection techniques in 3D space only permit the selection of a single object. An example here is the raycasting technique which can be used to choose the closest object in a virtual environment, e. g., [Wingrave and Bowman, 2005], (Figure 2.1a). Although raycasting has been shown to be efficient and accurate for selecting single objects from a distance, more powerful selection methods are required for the selection of multiple targets. Multiple object selection techniques [Steed and Parker, 2004], (Figure 2.1b) have been developed to support the user in performing a selection by drawing an arbitrary shape on the screen and selecting all objects which intersect with the resulting projection volume. More advanced selection techniques, such as IntenSelect [de Haan et al., 2005], (Figure 2.2a), assist users in selecting small objects in occluded and cluttered environments. WYSIWYP [Wiebel et al., 2012], (Figure 2.2b) enables users to intuitively select spatial positions in volumetric rendering. Astronomical datasets, consisting of millions of discrete points, present a serious challenge for selection. Clearly, methods for selecting objects one-by-one are not suitable for such datasets. A method such as CylinderSelection [Lucas and Bowman, 2005], where all objects that project inside a hand-drawn lasso are selected, are better suited to this domain. Nevertheless, they often require the use of repeated Boolean operations and are not well suited for selecting structures in complex or occluded environments. Therefore, it is important to develop selection techniques that can assist the user to select a subset of the dataset according to problem-specific requirements.

2.2.3 *High-dimensional datasets*

Astronomical datasets, as described in Section 2.1.1, are high-dimensional: the participants in our survey regularly deal with datasets of dimension from 3 to 25. It is thus important to be able to visualize such high-dimensional datasets and the relations between different dimensions.

Although there are algorithms that assist astronomers in uncovering the essential dimensions in their datasets, these algorithms are often not integrated with visualization tools. The result is that astronomers need to use a variety of disparate tools for their data analysis. The challenge here is to provide astronomers with integrated analysis and visualization tools so that they can apply data analysis algorithms in an interactive

way, combined with 3D interaction such as navigation and selection. Furthermore, such tools should give the users the ability to visualize the results in real-time and to adapt accordingly their exploration approach.

A related problem, connected to the fact that astronomers often work with 3D spatial datasets, is that not all information contained in a 3D dataset is immediately visible in a visualization. Therefore, the analysis of 3D astronomical datasets requires higher level exploration methods for uncovering the most important structures. We give here two specific examples of difficulties that astronomers meet in their study which are related to working with 3D datasets.

First, astronomical datasets span exceptionally large spatial scales occupied mostly by empty space, and this makes it difficult for the user to comprehend the spatial structure during exploratory navigation. The latter task is further obstructed by “noise” in the data leading to situations where particles in low-density regions occlude more interesting structures. For example, the halo of a galaxy can occlude the galactic core. In such cases it is important to be able to provide a way to focus on the important structures.

Second, in [Figure 2.3a](#) we present an example of the difficulty of understanding the structure of a 3D particle cloud. It is difficult to see in this picture that particles are arranged on the surface of a hemisphere. Nevertheless, this becomes clear with the help of a 3D density isosurface, shown in [Figure 2.3b](#), and created using the CloudLasso selection method ([Chapter 4](#)). Therefore, we need to develop methods that will assist the astronomer to find such structures that may not be easily visible in a scatter plot.

2.3 RESEARCH DIRECTIONS

Based on this analysis of the needs of astronomers and scientists in general, we can now formulate our research challenges for interactive scientific visualization, and our solutions to these questions.

For navigation in 3D space we concluded that using spring-loaded navigation modes appears to be a promising direction. Furthermore, in order to facilitate collaborative work in front of a large screen we decided to focus our efforts on touch-screens. Thus we designed a new navigation technique for 3D spaces, called FI3D. This is a touch-based navigation method using spring-loaded navigation modes. It maps the 2 DOF input from the touch surface to 7 DOF output (translation along three axes, rotation around three axes, and scaling) in 3D space. FI3D makes use of a frame around the main viewing area to invoke different interactions. Thus different spring-loaded interaction modes are activated depending on which part of the frame is initially touched and at which direction the touch starts. At the same time, FI3D leaves the main area available for visualizing the dataset and for interacting

once a mode has been determined. The method is described in detail in [Chapter 3](#).

After designing an appropriate navigation method we turned our attention into the problem of selection in 3D particle datasets. Our aim here was to develop methods that can assist astronomers to select the most interesting parts of a 3D dataset in an efficient and intuitive way. Such methods should take automatically into account the structure of the dataset. We designed two such methods, presented in [Chapter 4](#). Using these methods, called TeddySelection and CloudLasso, the user is able to draw on the 2D screen a lasso around the intended selection. Then, the methods based on problem-adapted heuristics compute a 3D volume that captures the intended selection. Furthermore, this kind of selection reveals the spatial structure of the dataset and provides better insights to the problem.

The last direction we pursued was integrating FI3D and CloudLasso into a visual analytics tool for subspace selection. This integrated visual analytics tool, presented in [Chapter 5](#), gives astronomers the ability to interactively find the essential dimensions of their datasets. At the same time it allows them to visualize the results of the analysis, locate clusters in the dataset, and associate other physical parameters to spatial positions.

In [Chapter 6](#) we discuss remaining open problems and possible future research directions related to the remaining challenges presented in [Section 2.2](#), such as collaborative analysis.

3

FI₃D: DIRECT-TOUCH INTERACTION FOR THE EXPLORATION OF 3D SCIENTIFIC VISUALIZATION SPACES



(a)



(b)

Figure 3.1: 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.

3.1 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 [Keefe, 2010]. 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 [Kin et al., 2009], to facilitate awareness in collaborative settings [Hornecker et al., 2008], and to provide somesthetic information and feedback that is beneficial for effective interaction both in real and virtual environments [Robles-De-La-Torre, 2006].

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 promote the use of scientific visualization techniques even by non-experts. Touch interaction has been the focus of previous research projects in the visualization context [Forlines and Shen, 2005; Frisch et al., 2009; Isenberg et al., 2008; North et al., 2009], 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 [Steinicke et al., 2008] 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 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.

3.2 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.

3.2.1 Interactive Visualization

In the field of information visualization, the challenges of interacting with data have been analyzed in several survey articles [Kosara et al.,

2003; Pike et al., 2009; Yi et al., 2007]. These surveys focus on specific data interaction techniques such as selection, exploration, change of representation, or filtering. Pike et al. [2009] 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 [Keefe, 2010]. 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 preconceived 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 [Kosara et al., 2003].

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 [Shoemaker, 1992] or similar virtual trackballs [Bade et al., 2005], 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 [Bowman et al., 2005] such as virtual reality environments like the Responsive Workbench [Krüger and Fröhlich, 1994] or the CAVE [Bryson, 1996]. One important advantage of these virtual environments is that they afford direct manipulation with the 3D worlds [Bowman et al., 2005; Cutler et al., 1997] because both stereoscopic projection of the virtual world and its manipulation (through tracking or by using haptic devices [Srinivasan and Basdogan, 1997]) 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 [Isenberg et al., 2008] or sketching [Schroeder et al., 2010]. Another example is Forlines and Shen's [2005] DTLens system. 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 [Forlines et al., 2006]. 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 [Robles-De-La-Torre, 2006] 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.

3.2.2 *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) [Hancock et al., 2006]. The rotation-and-translation (RNT) technique [Kruger et al., 2005] 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 [Liu et al., 2006], 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 [Hancock et al., 2006] (also named rotate-scale-translate, RST [Reisman et al., 2009]) 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 [Steinicke et al., 2008]. 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. [2007] 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. [2009a,b] 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 [Hancock et al., 2009].

Hancock et al.'s [2007; 2009] extension of the concept of direct manipulation from 2D interaction to 3D provides users with precise con-

trol 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. \[2008\]](#) 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 [[Wilson, 2009](#)]. 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. \[2009\]](#) and [Jung et al. \[2008\]](#) use dedicated definitions of gestures to navigate in 3D, [Reisman et al. \[2009\]](#) 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. \[2008\]](#) 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 scientific visualization, two previous gesture-based interaction approaches permit interaction with aspects of space itself: [Fu et al. \[2010\]](#) 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 and Lilien \[2008\]](#) 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. \[2008\]](#) 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.

3.3 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., [Edelmann et al., 2009; Jung et al., 2008; Reisman et al., 2009]). We also strive for our 3D interactions to be intuitive extensions (*G9*) of 2D RST direct touch manipulations [Hancock et al., 2009; Reisman et al., 2009]. 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’ [Hancock et al., 2009; Reisman et al., 2009]. 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 [Nacenta et al., 2009]. 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.

3.3.1 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 view-plane located at a certain distance from the viewer, depending on the control/display ratio [Blanch et al., 2004] or control gain [MacKenzie and Riddersma, 1994]. 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 [Hancock et al., 2007].

Rotation around x - and y -axes: In traditional mouse-based interfaces, this type of rotation is often achieved with a trackball/arcball metaphor [Bade et al., 2005; Shoemake, 1992]. 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 [Hancock et al., 2006; Reisman et al., 2009] 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 [Hancock et al., 2006]. 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 4 DOF dual-touch technique for x -/ y -translation, rotation around the z -axis, and zooming can easily be combined with the 2 DOF 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 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 6 DOF 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 (G_1) while also allowing single-touch-only interaction (G_7). This is described next.

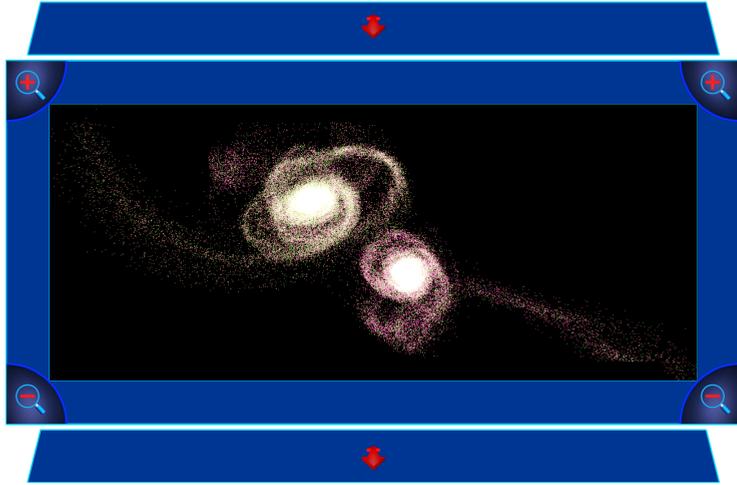


Figure 3.2: Screenshot of frame-based 3D visualization exploration widget.

3.3.2 Frame Interaction for 3D Visualization

The techniques described in the previous section allow us to control up to 4 DOF simultaneously by directly interacting on the visualization display space. To support full 7 DOF interaction (G₁) and address our remaining goals (G₂–G₁₀), we introduce a *frame-based interaction* technique. This method is inspired by previous interfaces [Nijboer et al., 2010; Zeleznik and Forsberg, 1999] that used widget borders to control aspects of the interaction. For instance, Zeleznik and Forsberg’s [1999] 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. [2010] 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 (Figure 3.2). Similar to Nijboer et al.’s [2010] 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 (Figure 3.3a). 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 [Kruger et al., 2005] or 3D [Hancock et al., 2007] shapes. Also, for 3D exploration of visualizations this has the

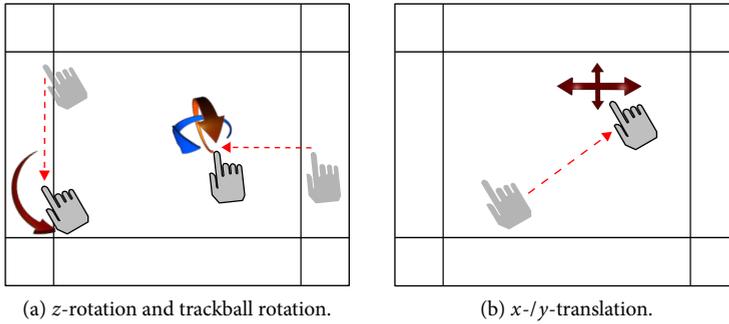


Figure 3.3: General frame interaction technique.

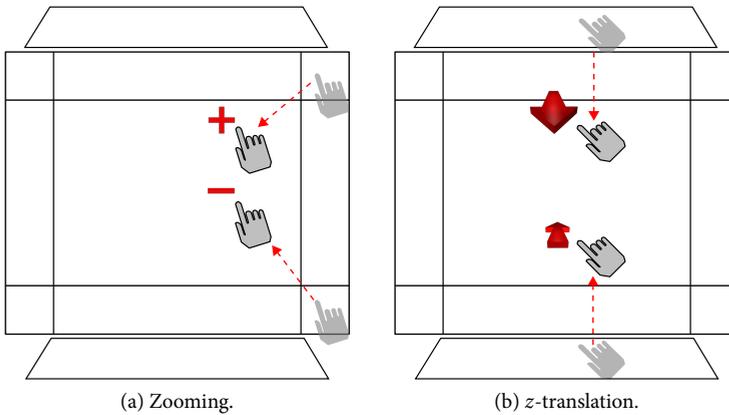


Figure 3.4: Zooming and z-translation with extra frame regions.

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 [1999] or mapping them to x -/ y -translation like Nijboer et al. [2010]. We experimented with both and chose to map to x -/ y -rotation (Figure 3.3a) 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 (Figure 3.3b).

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 (Figure 3.4a). 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

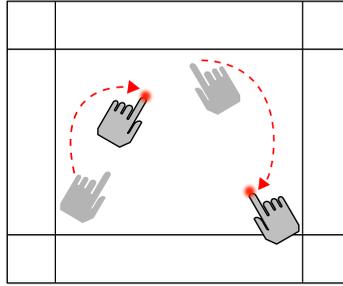


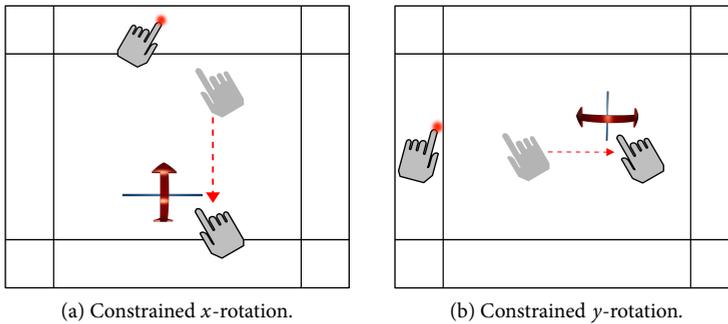
Figure 3.5: Two-touch RST interaction: translation, z -rotation, and zoom.

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 (Figure 3.4b). Here, the perspective shape of the extra frame elements suggests the direction of movement. Starting motions from the 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 (G_4). To further support usability, we always display icons associated to the chosen actions (as shown in Figure 3.3 and 3.4) 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 7 DOF necessary for the exploration of 3D visualization spaces (G_2) in one integrated paradigm (G_1), without requiring dedicated objects to be present (G_3). Compared to the integrated 6 DOF single-/dual-touch technique discussed in Section 3.3.1, however, the frame-based interactions allow us to separate out the interactions and, consequently, permit users to control the exploration more precisely (G_6). 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 7 DOF with only a single touch (G_7) 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 (G_8) and we provide RST interaction (Figure 3.5) 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

(a) Constrained x -rotation.(b) Constrained y -rotation.Figure 3.6: Constrained rotation techniques for the x - and y -axes.

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 (Figure 3.6). For example, for rotating 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.

3.4 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 techniques 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 [Forlines et al.,

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

2007; Hornecker et al., 2008], 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 (1920×1080 pixels, $115.4 \text{ cm} \times 64.5 \text{ cm}$). The display was equipped with a DVIT overlay [Smart Technologies Inc., 2003] 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 575×575 pixels ($34.5 \text{ cm} \times 34.5 \text{ cm}$) throughout the study so that we could show information relevant to the task side-by-side. The setup for the frame interaction was described in Section 3.3 (Figure 3.7a). We removed the interaction frame for the mouse condition (Figure 3.7b). 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.



(a)



(b)

Figure 3.7: Example setup for travel task, (a) touch and (b) mouse condition.

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

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



Figure 3.8: A person performing the wayfinding task.

and may include planning, higher-level thinking, or decision-making [Bowman et al., 2005]. 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 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 Figure 3.7a and 3.7b). 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 (Figure 3.8). 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.

3.5 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

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

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-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

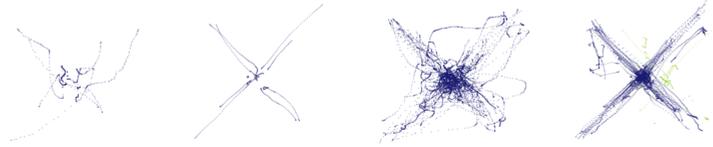


Figure 3.9: Translation for mouse and touch for a single participant (left two images, resp.) and for all participants overlaid (right two images, resp.).

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.

3.6 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' 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 (Figure 3.9) 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 [Smart Technologies Inc., 2003] 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



Figure 3.10: Screenshot of the medical illustrative visualization application.

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.

3.7 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 [Svetachov et al., 2010] and discuss some adaptations specific to this domain.

3.7.1 *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 [Svetachov et al., 2010] (see Figure 3.10). 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 [2008] physics-based interaction with a cube, as opposed to what would happen using 3D shallow-depth RNT [Hancock et al., 2007]. 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.

3.7.2 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., [Clifton and Pang, 1997; Fröhlich et al., 2000; Meyer and Globus, 1993]). 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 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

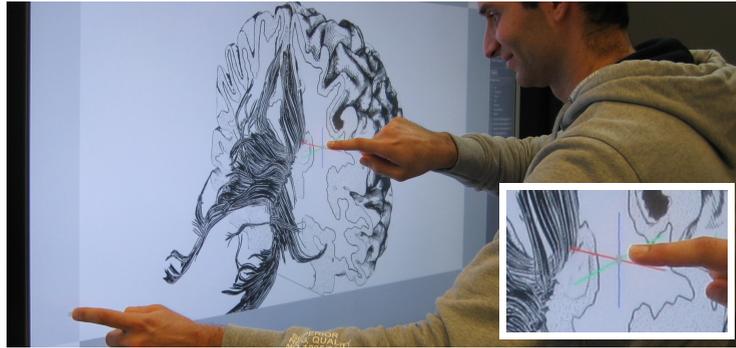


Figure 3.11: Coordinate axis selection (detail) for cutting plane interaction.

interaction. As soon as this finger touches the center part, three colored crossing lines are shown (Figure 3.11), 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.

3.7.3 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 [Akers et al., 2004; Blaas et al., 2005], but mouse gestures have also been explored [Akers, 2006]. 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



Figure 3.12: Selecting fibers that pass through two regions 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 (Figure 3.12), 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.

3.7.4 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 [Wang et al., 2007]). 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.

3.8 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 FI_{3D} to common mouse-and-keyboard interaction. This study showed that FI_{3D} 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 [Han, 2005] or IFSR [Rosenberg and Perlin, 2009], 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 [2010] 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 FI_{3D} 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.

3.9 FROM 3D SPACE NAVIGATION INTERACTION TO 3D DATA SELECTION

According to the information seeking mantra, “overview first, zoom and filter, then detail on demand”, before studying detail information, it is essential to first filter the data according to some physical properties. In this case, selection techniques would be needed for interactive visualization and later data exploration. A smart selection technique does not require users to carefully specify the target through several selection editing steps but can get exactly what intended to be selected. Therefore, in next chapter we will introduce two efficient structure-aware

selection techniques to support the selection of subsets in large particle 3D datasets in an interactive and visually intuitive manner.

ACKNOWLEDGMENTS

From the University of Groningen, we would like to thank Amina Helmi for the cosmological simulation dataset from the Aquarius Project [Springel et al., 2008], Frans van Hoesel for the galaxy collision simulation (originally from <http://www.galaxydynamics.org/>), and Leonardo Cerliani for the brain dataset.

This chapter is based on: Lingyun Yu, Pjotr Svetachov, Petra Isenberg, Maarten H. Everts, and Tobias Isenberg. FI3D: Direct-Touch Interaction for the Exploration of 3D Scientific Visualization Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613–1622, November/December 2010.

EFFICIENT STRUCTURE-AWARE SELECTION TECHNIQUES FOR 3D POINT CLOUD VISUALIZATIONS WITH 2DOF INPUT

ABSTRACT: *Data selection is a fundamental task in visualization because it serves as a pre-requisite to many follow-up interactions. Efficient spatial selection in 3D point cloud datasets consisting of thousands or millions of particles can be particularly challenging. We present two new techniques, TeddySelection and CloudLasso, that support the selection of subsets in large particle 3D datasets in an interactive and visually intuitive manner. Specifically, we describe how to spatially select a subset of a 3D particle cloud by simply encircling the target particles on screen using either the mouse or direct-touch input. Based on the drawn lasso, our techniques automatically determine a bounding selection surface around the encircled particles based on their density. This kind of selection technique can be applied to particle datasets in several application domains. TeddySelection and CloudLasso reduce, and in some cases even eliminate, the need for complex multi-step selection processes involving Boolean operations. This was confirmed in a formal, controlled user study in which we compared the more flexible CloudLasso technique to the standard cylinder-based selection technique. This study showed that the former is consistently more efficient than the latter—in several cases the CloudLasso selection time was half that of the corresponding cylinder-based selection.*

4.1 INTRODUCTION

IN scientific visualization, researchers are often interested in various physical properties of objects/regions to analyze their structure and context. To complete such analysis tasks it is essential to first select the objects of interest from their environment. While this is reasonably easy if one faces only a few objects which are also relatively large, the selection becomes a challenge if the dataset consists of thousands or millions of particles. While it is sometimes possible to filter particles according to some known data properties besides particle position, this may not always be the case because such properties may not yet be known or may not even exist. In such cases the only accessible possibility may be the interactive *spatial* selection of subsets of the particles by specifying a region in space in which the targeted particles are located. Therefore, interactive spatial selection is essential for many visualization applications and domains. A spatial selection also permits people to employ Boolean operations that involve several selection regions. However, when dealing with data in a three-dimensional domain, such a spatial selection often becomes a tedious multi-step process because

it is difficult for people to specify the correct enclosing 3D surface for the spatial region of interest.

Our goal was thus to design new structure-aware selection techniques that facilitate the selection of multiple objects at a time by specifying *spatially* which regions are important, without a complicated multi-step process. A major design requirement for a new selection technique is that it should be as easy to use as freeform lassos for 2D particle selection or as raycasting-based methods for selecting single objects in 3D. It is important to realize that raycasting based techniques are unsuitable for our problem due to the impracticality of selecting thousands of particles in a serial manner. Instead, we are inspired by lasso-based methods such as [Lucas and Bowman's \[2005\]](#) Tablet Free-hand Lasso which selects 3D points within the lasso-specified generalized cylinder and we, therefore, refer to it as CylinderSelection. Such techniques are better suited for the selection of a large number of particles but have to be combined with subsequent selections from different angles through Boolean operations.

To facilitate an easy and intuitive selection in 3D particle datasets we present two new methods, TeddySelection and CloudLasso. For both techniques, the interaction is based on a 2D lasso that the user draws onto the 2D projection of the 3D space. The methods then derive an appropriate 3D selection geometry, taking the full spatial structure of the dataset into account. TeddySelection is inspired by interaction techniques from sketch-based modeling [[Igarashi et al., 1999](#)]. Based on a heuristic that takes into account the local particle density, the area enclosed by the user-drawn 2D lasso is inflated and fitted to the region of space that the user intended to select. CloudLasso uses the seminal Marching Cubes method [[Lorensen and Cline, 1987](#); [Wyvill et al., 1986](#)] to identify and select the regions inside the lasso where the density or the value of another scalar property is above a threshold. Both our techniques can be employed not only in traditional mouse-based interaction but are also, in particular, suitable for direct-touch visualization environments. In fact, our work was motivated by a direct need for an enhanced spatial selection mechanism using direct-touch input in the domain of astronomy for particle data such as numerical simulations of the gravitational processes of stars or galaxies. Nevertheless, the new techniques presented here are applicable to any 3D point cloud dataset such as 3D scatter plots in information visualizations or particle flow simulations in physics, and can also be extended to allow selection based on other scalar properties besides density. The created 3D selection surfaces can also be used in an off-line process to enable the processing of datasets whose sizes do not fit into main memory.

In the remainder of the paper we first review related work in [Section 4.2](#). Then, we describe our own selection techniques in detail: TeddySelection in [Section 4.3](#) and CloudLasso in [Section 4.4](#). Next, we present the results of a formal, controlled user study in [Section 4.5](#)

where we compared CloudLasso to CylinderSelection. In [Section 4.6](#) we compare all three methods, discuss possible extensions of TeddySelection and CloudLasso, and consider applications of these methods in general visualization contexts. We conclude the paper in [Section 4.7](#).

4.2 RELATED WORK

Several techniques for selecting objects in 2D environments have been developed in the past. These include selections by clicking on the target (picking), brushing where users pass over several target objects, spatial selections by clicking and dragging a lasso to create a selection area, or selections based on property masks such as color. Two-dimensional selection is considerably easier than selection in 3D because the 2D space is fully visible and accessible for interaction. Nevertheless, structure-aware selection that exploits the perceptual grouping of objects may outperform standard selection techniques even in 2D [[Dehmeshki and Stuerzlinger, 2008, 2009](#)].

For 3D data, raycasting selection techniques (e. g., [[Argelaguet and Andujar, 2009](#); [Lee et al., 2003](#); [Pierce et al., 1997](#); [Wingrave et al., 2005](#)]) are common in virtual environments that range from desktop VR to CAVEs; for true volumetric displays similar techniques are used [[Grossman and Balakrishnan, 2006](#)]. Raycasting is frequently employed because it can be operated at a distance, is fast, and is easily understandable [[Lucas and Bowman, 2005](#)]. Cone selection [[Liang and Green, 1994](#)] is a related technique that selects the object whose location is closest to the cone center. To improve accuracy, shadow cone selection [[Steed and Parker, 2004](#)] selects groups of objects that lie within a cone projected from the interacting hand for the period of the selection interaction. This means that people need to move the input device to select a single object. While techniques generally based on ray casting are efficient for selecting single and large objects, they are less well suited for selecting small or occluded objects in cluttered environments. Thus, de Haan et al. designed IntenSelect [[de Haan et al., 2005](#)] to assist users in selecting potentially moving objects in occluded and cluttered VR environments. Here, a constantly updated scoring function is calculated for all objects that fall within a user-controlled, conic selection volume. While interacting, a bending ray remains snapped to the highest ranking object to ease selection. To address the selection of small objects, [Kopper et al. \[2011\]](#) developed SQUAD as a technique that is based on progressive refinement. ? use 3D freehand gestures to select objects in occluded environments. They overcome the accuracy problems of such techniques by splitting the selection task into several connected low demand tasks. 3D streamlines or tracts can be selected by drawing a 3D lasso around the objects of interest [[Keefe et al., 2008](#); [Zhou et al., 2008](#)] or using haptic input [[Jackson et al., 2012](#)]. 3D picking also relates to 3D selection

and is possible, e. g., via raycasting for distinct objects—even though object transparency may cause challenges with respect to what object was intended to be selected [Mühler et al., 2010]. Without distinct objects, however, more advanced techniques like WYSIWYP are needed [Wiebel et al., 2011, 2012], similar to our lasso-based selection.

Of course, there also exist techniques that enable 3D object selection based directly on a 3D position. Such techniques rely on rate controls such as 3D mice, tracked input hardware such as gloves, or non-wired 3D tracking such as Sixense’s TrueMotion. Although they are easy to understand and manipulate, they are only feasible for object selection within the user’s reach, unless combined with other non-linear mapping interaction techniques like the Go-Go technique [Poupyrev et al., 1996].

While all these selection techniques—both based on raycasting and on direct 3D positions—are easy to understand and to operate, the time necessary for completing a selection increases with the number of targets. These methods are thus not suitable for datasets such as particle clouds where huge amounts of tiny objects often need to be selected. Selection-by-volume techniques like our own provide faster and more effective selection in these situations due to less repetition being necessary during the selection process. A related but sequential approach is taken by Elmqvist et al.’s [2008] ScatterDice visualization system which selects multi-dimensional data through successive lasso selections in 2D scatterplots of the data. Uliniski et al. [2007] use two-handed techniques for selecting the data in a subset of space. The two hands specify and manipulate a cuboidal selection volume. However, this technique may also include undesired objects because the desired structure in the 3D dataset typically does not have a cuboidal shape [Lucas and Bowman, 2005]. A more flexible technique is the Tablet Freehand Lasso method (CylinderSelection) by Lucas and Bowman [2005] that lets people draw a lasso on a tracked physical canvas to extend a conical selection shape into the 3D space, as seen from the camera whose view is shown on the canvas. While this approach may still require complex multi-step Boolean operations to come to a final selection, it still serves as the foundation of our work: the lasso lets us take the dataset’s 2D structure into account as visible in the projection. In our approach we significantly improve the original idea by considering the data’s structure along the view direction. In the same spirit, Owada et al. [2005] introduced Volume Catcher, a technique for unsegmented volume data that, based on a 2D stroke, selects a region of interest by applying a segmentation algorithm. Owada et al.’s technique, however, requires a precisely drawn stroke and is not directly applicable to particle datasets.

Our selection techniques target, in particular, applications in a directly-manipulative context such as visualization exploration on touch-sensitive displays. One reason for employing a direct-touch input metaphor lies in its performance improvements over mouse interaction,

e. g., for target acquisition [Kin et al., 2009] and in form of more precise control over motions on the visualization display [Yu et al., 2010]. In this context also direct-touch interaction approaches with 3D environments play a role for which we refer to the overview by Isenberg and Hancock [Isenberg and Hancock, 2012].

4.3 TEDDYSELECTION

To facilitate a lasso-based selection in 3D particle datasets, we need to find techniques that permit the creation of a 3D selection volume from a shape drawn in 2D. For our first technique, TeddySelection, we are thus inspired by Igarashi et al.'s [1999] sketch-based Teddy modeling approach which, based on users' drawings of 3D shapes on a 2D surface, constructs a 3D mesh from the sketches. Teddy first triangulates the drawn shape and then extracts a *spine* in the triangulation's center. Then, the 2D shape is 'inflated' by determining the vertical depth of vertices on the spine from their distance to the drawn shape. Wide areas thus become fat, while narrow areas remain thin.

Relying on heuristics, we adjust the general Teddy approach to adapt the created mesh to the actual structure of the particles in the 3D point cloud dataset. More specifically, our selection algorithm consists of the following three main stages:

1. Input polygon triangulation: We compute a 2D triangulation between the spine and the drawn 2D outline, following Igarashi et al. [1999].
2. Particle mapping to triangles: We determine which particles are located inside the 2D selection polygon and map each of them to their corresponding triangle in the triangulated mesh.
3. Construction of the selection mesh: We inflate the mesh by adjusting the vertices' depth based on the particle distribution.

4.3.1 Input Polygon Triangulation

In the first step of our algorithm we follow Igarashi et al.'s [1999] lead. We first convert the user-drawn input lasso into a closed stroke by connecting its start and end points (Figure 4.2a). If the input stroke self-intersects we only consider its largest closed part, starting and ending at the intersection. To remove noise from the input device, we re-sample the input stroke with a uniform edge length before further processing which later-on also ensures a regular mesh polygon. Next, we determine the polygon's spine and create a complete 2D triangular mesh between the spine and the perimeter of the initial polygon.

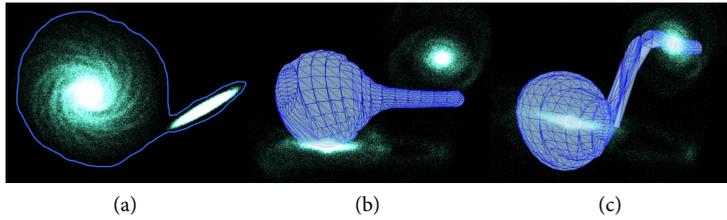


Figure 4.1: Problem with Teddy-like placement of a selection mesh in the 3D data space: (a) lasso in the original view direction, drawn from a ‘top-down’ view; (b) selection mesh after a 90° rotation, placed at constant depth—the mesh does not wrap around the data; (c) depth adjusted to local average particle depth—the mesh still does not properly wrap around the data due to only being parameterized by the drawn lasso.

4.3.2 Particle Mapping to Triangles

While the ‘inflating’ of the triangulated polygon in the original Teddy algorithm by Igarashi et al. [1999] is parameterized based on the local width of the drawn shape, we need to take both the location and distribution of the particles into account in this process. For this purpose we associate the particles with the generated 2D triangulation. Therefore, we render an ID-buffer of the triangulation and then check, for each particle’s screen position, its association to a specific mesh triangle.

4.3.3 Construction of the Selection Mesh

In the final step we need to inflate the 2D mesh and position it into the 3D space of the dataset. While Igarashi et al. [1999] can simply use one constant depth for the entire mesh, this does not work well for structured datasets (see Figure 4.1b). Also, a local adjustment of the depth of the mesh does not produce satisfying results. We can see in Figure 4.1c that narrow regions still remain thin and wide regions become fat regardless of the particle distribution since the inflation parametrization solely relies on the drawn shape (compare Figure 4.1a to Figure 4.1b). Therefore, we perform a structural analysis by triangle of the produced 2D mesh. Generally speaking, our goal is to identify the closest and the furthest distance to the camera per triangle such that all dense clusters associated to the triangle lie between these two points. For this purpose we use a two-stage binning process to identify and remove sparsely populated regions.

The first binning stage (Figure 4.2b) examines the whole generalized cone volume (i. e., only the part that is covered by the dataset) that is specified by the 2D lasso selection and splits it, at regular intervals in

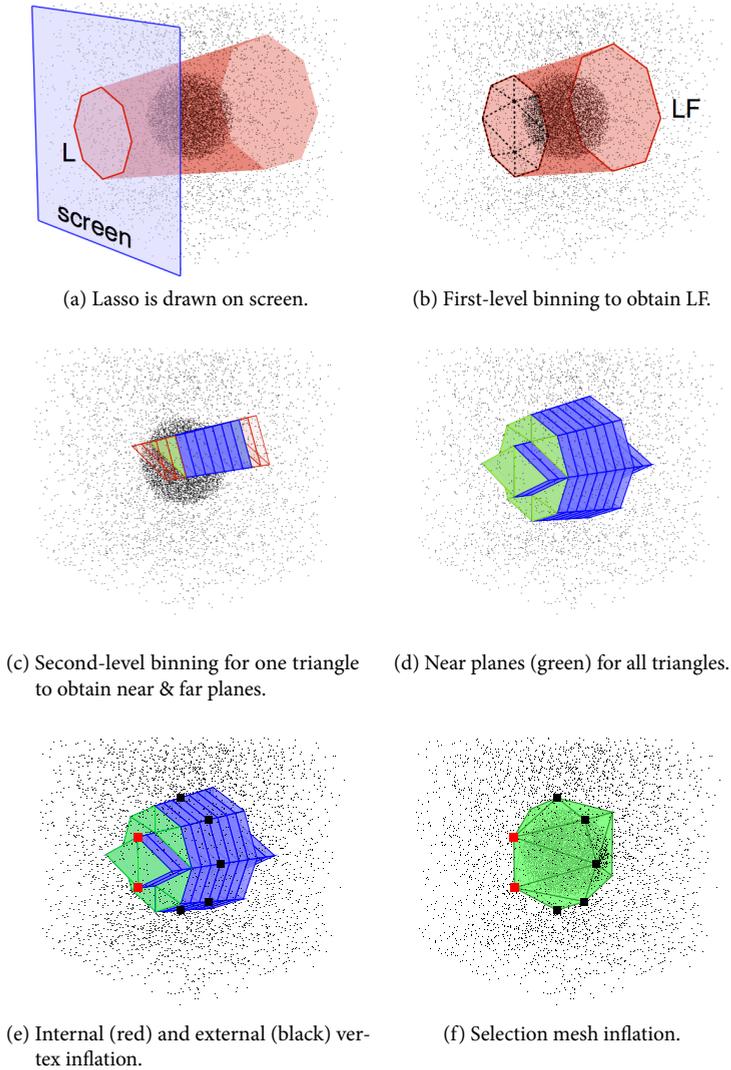


Figure 4.2: Main steps of the TeddySelection algorithm.

depth, into a certain number of bins (we use 100). We then calculate the number of particles within each of these bins. Next we threshold the bins and determine the closest and farthest bin with more than a given number of particles per volume unit. The specific particle number threshold depends on the dataset and can be adjusted. It is important to note that bins have different volumes, with bins closer to the viewer having a smaller volume. We call the section of the generalized selection

cone between the front of the closest non-empty bin and the back of the farthest non-empty bin the *lasso frustum* (LF).

Then we perform a second binning (Figure 4.2c), but this time by triangle of the 2D mesh to extract the local 3D structure of the dataset. Similar to the first binning, we split each generalized cone segment as defined by a mesh triangle into a number of bins (we use 60) and locate the closest and farthest *dense* bin. A bin is considered to be dense if its particle count is larger than a user-adjustable percentage of the expected particle count (we use a default of 400%, i. e., 4× the density average) if all particles were evenly distributed in the entire LF. This results in a depth range per triangle that includes all large clusters of particles between the front of the closest z_f and the back of the farthest z_b non-empty bin. An illustration of this result is shown in Figure 4.2d.

We now employ this second-level binning information to inflate the 2D mesh. First, we need to determine the exact z -depth of each 2D mesh vertex, both for the front and for the back part of the inflated mesh. Here we need to distinguish between internal vertices (the ones on the spine) and external vertices (the ones on the resampled sketched lines). For inflating the internal vertices we examined two approaches: using the average or using the most extreme ‘non-empty’ depth (front or back) of all adjacent triangles for a vertex. We experimented with both and found that both have advantages and disadvantages. By using the average depths of the adjacent triangles we obtain smoother 3D shapes but introduce errors in the form of some smaller clusters no longer being included in the selection volume due to the ‘contraction’ that happens because of the averaging. In contrast, the use of the most extreme depth values of all adjacent triangles results in a 3D shape that is not as smooth as the one based on averages but does ensure to include smaller clusters. We thus inflate the selection shape by moving all internal vertices to the extreme depths, while for each external vertex we average z_f and z_b of all its adjacent triangles to determine its location (Figure 4.2e). Finally, the vertices are connected, resulting in a polygonal selection mesh that selects particles from the dataset spatially (Figure 4.2f). In order to produce a smoother polygonal mesh we follow the original Teddy algorithm by Igarashi et al. [1999] and further subdivide fan triangles (triangles that connect the spine and the outside edge) to produce a gradual transition between spine and outside edge.

4.3.4 Example Results

Figure 4.3 shows examples of applying TeddySelection to two astronomical datasets, a galaxy collision simulation in Figure 4.3(a) and an N-body mass simulation in Figure 4.3(b,c). The top row in Figure 4.3 shows the dataset before the selection, the middle row shows the selec-

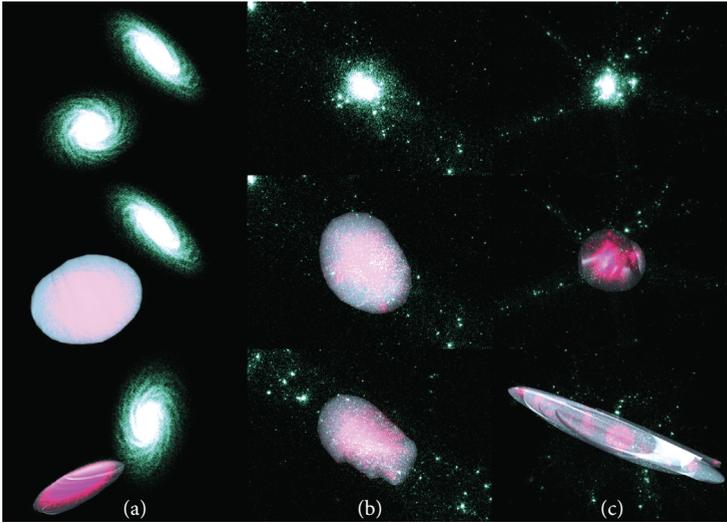


Figure 4.3: Different dataset configurations (top), TeddySelections (middle), and selections viewed from another angle (bottom).

tion applied for this view, and the bottom row shows a different view of the selection to illustrate how the TeddySelection takes the 3D structure of the dataset into account. In [Figure 4.3\(a\)](#) one of the galaxies was selected, resulting in a flat disk selection shape. [Figure 4.3\(b\)](#) shows the selection of an almost spherical particle cluster. In the case of [Figure 4.3\(c\)](#) it becomes clear after rotating the dataset (bottom), that the particle cluster is elongated and that the selection has taken this fact into account, creating a connected selection shape from the front to the back of the cluster. Note that in all cases the projection of the selection on screen is disk-shaped but TeddySelection takes into account the 3D structure of the selection, producing an appropriate selection geometry.

4.3.5 Performance

TeddySelection's processing time depends on the size of the dataset, the 2D size of the selection (number of triangles), and the particle count in the first-level selection volume. The selection on a 3.16 GHz Intel[®] Core[™] 2 Duo CPU using a $1.6 \cdot 10^5$ particle dataset ([Figure 4.3\(a\)](#)) takes 0.36 s. Selection from the $2.0 \cdot 10^5$ particle dataset for small ([Figure 4.3\(b\)](#)) and larger clusters ([Figure 4.3\(c\)](#)) takes 0.17 s and 0.21 s, resp.

4.3.6 User Feedback on TeddySelection

Our work was motivated by a direct need for an enhanced selection mechanism using direct-touch input in the domain of astronomy. To

elicit feedback on our technique from expert users we interviewed three astronomers who regularly work with particle data such as numerical simulations of the gravitational processes of stars or galaxies. We demonstrated TeddySelection in individual sessions to the experts and discussed its benefits and drawbacks. We used both datasets shown in Figure 4.3 (galaxy collision and cosmological N-body simulation) and asked the astronomers to experiment with the interactive selection.

For this purpose we presented the particle datasets on a $1920 \times 1080 \times 52''$ touch-sensitive display. Our interface allowed the experts to explore the data by navigating through the 3D space [Yu et al., 2010] and facilitated fluid switching between navigation and selection.

Their feedback confirmed our motivation that spatial selections are essential for analyzing datasets, especially when properties about a subset of data such as the movement of stars in a particular region are of importance to the analysis. The astronomers' normal way of making selections is to rotate the dataset to find a good view, and then to use an unconstrained selection by means of a lasso or a selection rectangle. However, they reported that, generally, performing rotations to get a good view is tedious for them. Therefore, they much appreciated TeddySelection, commenting that our technique is helpful to find the prominent subset in depth without having to consider the view direction much. Similarly, the experts found it easy to select tails or arms of the collision galaxy dataset which was difficult to do previously. They also noted that the direct-touch interaction using the technique made it easy to draw precise lassos around the clusters of interest. As an addition they asked for a feature to make inverse selections (i. e., to select everything but the sketched regions) which is not yet possible with our prototype, but which can easily be included.

While the astronomers did not comment on further issues, TeddySelection has three limitations. One is that TeddySelection does not work well in generally sparse regions due to the noise contained therein. For example, if one zooms into a dataset too much the particles become sparsely distributed so that selections performed in such regions are less predictable. However, in such situations the structure of the data is less important so that cuboidal selections [Ulinski et al., 2007] work generally better. A second limitation is that the chosen view direction can have an effect if dense clusters lie visually behind the intended selection, and because regions between dense clusters in the front and in the back are always included. The need for structure in the data to constrain the selection—the third limitation—also means that the TeddySelection is not as useful in complex environments which contain many small, evenly distributed clusters. While the last two limitations can be addressed by a small change in view direction or by Boolean combinations of Freehand Lasso selections [Lucas and Bowman, 2005], the expert feedback indicated that a technique that addresses these

issues would be extremely useful. We therefore discuss a new selection technique next that addresses the latter two mentioned limitations.

4.4 CLOUDLASSO

Our second technique, CloudLasso, is based on the application of the Marching Cubes (MC) algorithm [Lorensen and Cline, 1987; Wyvill et al., 1986] to the identification of dense parts of a particle dataset inside a user-drawn lasso; i. e., CloudLasso is a lasso-constrained Marching Cubes method. The MC method allows us to spatially select dense clusters within a lasso region individually even if these lie visually one behind another—without including the lower-density space in-between. To be able to apply the MC method, however, we need to first compute a continuous scalar field approximating the particle density. Using the scalar field we can then select the threshold for the intended selection. The CloudLasso algorithm, therefore, comprises the following three main steps:

1. Density estimation: The particle density is estimated inside a volume that contains all particles that project inside the lasso.
2. Volume selection: The subset of the volume where the density exceeds a threshold is computed using Marching Cubes.
3. Threshold tuning: Interactively adjusting the density threshold.

4.4.1 Density Estimation

In the first step of CloudLasso, we convert the user-drawn input lasso L (Figure 4.4a) to a closed stroke, similar to TeddySelection, by connecting its start and end points, re-sampling it to remove noise, and addressing self-intersections. All subsequent computations and constructions in the algorithm are, unless otherwise noted, carried out in the view coordinate system. We, therefore, first transform all particle coordinates to view coordinates by applying the graphics system’s model-view transformation, while preserving volumes and densities. Then we perform the first-level binning stage as we have also used it for TeddySelection to obtain the lasso frustum (Figure 4.4b).

Next, we determine the minimal rectangular box B such that it completely encloses LF and that its edges are parallel to the view coordinate axes. We construct a uniform rectangular grid G inside B such that the latter is split into 2^{18} cubes of equal volume ($64 \times 64 \times 64$; Figure 4.4c). Then, we apply a kernel density estimation method in order to obtain a scalar density field from the particle data. Such methods effectively ‘smear’ each particle over a region and assign to each point in space a scalar value that approximates the local particle density. In our case, we compute a value for the scalar density field at each grid node using the modified Breiman kernel density estimation method (MBE)

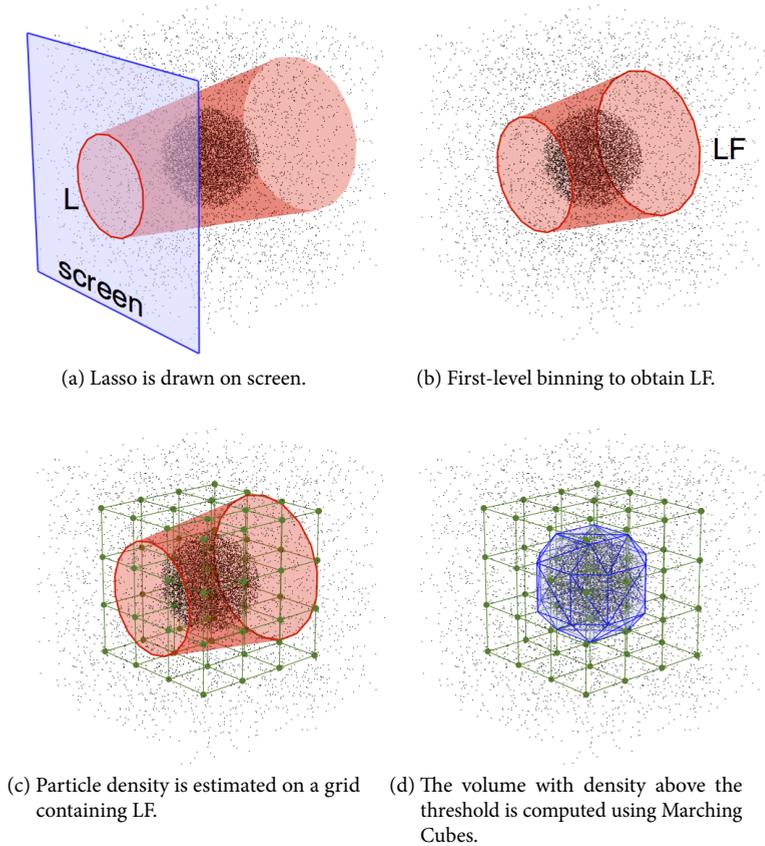


Figure 4.4: Main steps of the CloudLasso selection algorithm.

with a finite-support adaptive Epanechnikov kernel [Ferdosi et al., 2011; Wilkinson and Meijer, 1995]. Our reason for choosing this particular method is that it was shown by Ferdosi et al. [2011] to be optimal with respect to speed and reliability when compared to the k -nearest neighbors, adaptive Gaussian kernel, and Delaunay tessellation field methods. Moreover, kernel methods (incl. Gaussian kernel methods) have two practical advantages over other density estimation methods. First, the Marching Cubes method that we use in later steps requires a grid-based density estimation, and kernel methods always compute this information. Second, kernel methods compute a continuous density field and, thus, are more suitable for usage in combination with Marching Cubes. The main benefit of the Epanechnikov kernel compared to Gaussian kernels, however, is its reduced computational effort: the number of grid points that have to be considered per particle is limited due to the Epanechnikov kernel's finite support.

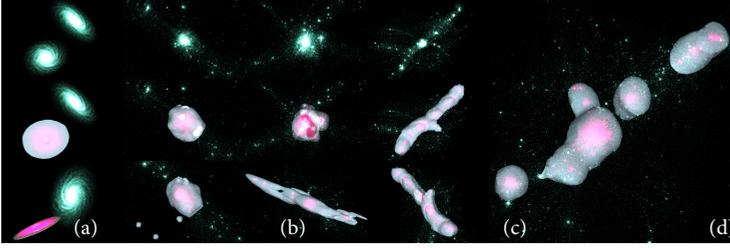


Figure 4.5: Interactive CloudLasso selection: (a)–(d) different dataset configurations (top), CloudLasso selection (middle), and the selection viewed from another angle (bottom); (e) close-up of the selection in (d) with an interactively adjusted (higher) density threshold.

We now give a brief description of the MBE method and we refer to [Ferdosi et al. \[2011\]](#) for more details. First, for each direction $k = x, y, z$ we compute a smoothing length as

$$\boxtimes_k = 2(P_k^{(80)} - P_k^{(20)})/\log N,$$

where N is the number of particles in the rectangular box B that encloses LF and $P_j^{(q)}$ is the q -th percentile value for coordinate k . For each data particle i at position $\mathbf{r}^{(i)}$ and for each node n at position $\mathbf{r}^{(n)}$ we define the vector $\mathbf{r}^{(i;n)}$, the k -th component of which is given by

$$\mathbf{r}_k^{(i;n)} = (\mathbf{r}_k^{(i)} - \mathbf{r}_k^{(n)})/\boxtimes_k.$$

Then, we compute the *pilot density* $\rho_{\text{pilot}}(\mathbf{r}^{(n)})$ at the node n as

$$\rho_{\text{pilot}}(\mathbf{r}^{(n)}) = \frac{15}{8\pi N} \frac{1}{\boxtimes_x \boxtimes_y \boxtimes_z} \sum_i 1 - \mathbf{r}^{(i;n)} \cdot \mathbf{r}^{(i;n)},$$

where only particles for which $\mathbf{r}^{(i;n)} \cdot \mathbf{r}^{(i;n)} \leq 1$ are considered in the sum. In other words, the only contribution to $\rho_{\text{pilot}}(\mathbf{r}^{(n)})$ comes from particles for which the node n is inside an ellipsoid with semi-axes $\boxtimes_x, \boxtimes_y, \boxtimes_z$ centered at the particle.

We compute the pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(i)})$ at the position of the i -th particle using multi-linear interpolation with respect to the nearby nodes. Next, we define the particle-specific smoothing lengths $\boxtimes_k^{(i)}$ with $k = x, y, z$ for each particle i as

$$\boxtimes_k^{(i)} = \min\{\boxtimes_k (m/\rho_{\text{pilot}}(i))^{1/3}, 10s_k\},$$

where m is the arithmetic mean of $\rho_{\text{pilot}}(\mathbf{r}^{(i)})$ over all particles in the rectangular box B , and s_k is the distance between adjacent grid

points in the k -th direction. Note that here we have introduced two modifications with respect to the standard method described by [Ferdosi et al. \[2011\]](#). First, we employ the arithmetic mean of the pilot densities instead of the geometric mean because the pilot density for a particle can often be zero, which would lead to a zero geometric mean. Second, we introduce a cut-off threshold for the value of $\mathbb{V}_k^{(i)}$ because particles with a small pilot density define large corresponding ellipsoids with semi-axes $\mathbb{V}_x^{(i)}$, $\mathbb{V}_y^{(i)}$, $\mathbb{V}_z^{(i)}$ and, thus, contribute to the final density of a large number of nodes. We found that due to this reason particles with large ellipsoids completely dominated the computational time and thus made the density estimation unsuitable for interactive applications. Now, we re-define the vectors $\mathbf{r}^{(i;n)}$ as

$$\mathbf{r}_k^{(i;n)} = (\mathbf{r}_k^{(i)} - \mathbf{r}_k^{(n)}) / \mathbb{V}_k^{(i)}.$$

Finally, we compute the density $\rho(\mathbf{r}^{(n)})$ at the node n as

$$\rho(\mathbf{r}^{(n)}) = \frac{15}{8\pi N} \sum_i \frac{1}{\mathbb{V}_x^{(i)} \mathbb{V}_y^{(i)} \mathbb{V}_z^{(i)}} (1 - \mathbf{r}^{(i;n)} \cdot \mathbf{r}^{(i;n)}),$$

where only particles with $\mathbf{r}^{(i;n)} \cdot \mathbf{r}^{(i;n)} \leq 1$ are included in the sum.

4.4.2 Surface Extraction

To extract the selection shape surface based on the density estimation, we start by computing the average density for nodes of the grid G that lie inside LF and setting this average density as our initial *selection threshold* ρ_0 . We could then naïvely apply the Marching Cubes algorithm to G to compute the selection region with $\rho \geq \rho_0$. However, we are actually interested in the region inside LF where $\rho \geq \rho_0$. Therefore, we first compute for each node n of G its projection n' on the screen and the distance $d(n')$ of n' from the lasso L . We define the corresponding *signed* distance $\delta(n)$ as $\delta(n) = d(n')$ if n' is inside L and as $\delta(n) = -d(n')$ if n' is outside L . This generalizes to arbitrary points: \mathbf{r} in view coordinates projects on screen inside L if and only if $\delta(\mathbf{r}) \geq 0$. Hence, the point \mathbf{r} is simultaneously inside the region $\rho \geq \rho_0$ and inside LF if

$$f(\mathbf{r}) = \min\{\rho(\mathbf{r}) - \rho_0, \delta(\mathbf{r})\} \geq 0.$$

Therefore, we apply the Marching Cubes method for the iso-surface $f(\mathbf{r}) = 0$ to obtain the bounding surface S of the required volume. The surface S might consist of more than one disconnected component. Furthermore, note that in order to ensure that the surface is closed we pad G with a layer of outer nodes where the value of f is defined to be $-DBL_MAX$. Also, after having constructed S , we mark particles inside this selection shape as being selected.

4.4.3 Threshold Adjustment

In the previous step we automatically set the density threshold to ρ_0 . In our experience this setting already yields good selections but adjusting the threshold can improve the result. We thus provide means to interactively adjust the threshold at runtime in the range $[\rho_0/16, 16\rho_0]$ by mapping a linear parameter $s \in [-4, 4]$ to the threshold value using $\rho_s = 2^s \rho_0$. When s is adjusted, we thus recompute the scalar $f(\mathbf{r}) = \min\{\rho(\mathbf{r}) - \rho_s, \delta(\mathbf{r})\}$ for all grid nodes and obtain the iso-surface $f(\mathbf{r}) = 0$ using Marching Cubes. It is important to note that we only need to recompute $f(\mathbf{r})$ in this case because $\rho(\mathbf{r})$ and $\delta(\mathbf{r})$ remain constant, therefore adjusting the threshold is computationally much less expensive than the previous step and can be done interactively.

Furthermore, note that if ρ_s becomes smaller than the minimum density inside LF then the whole LF is selected. This means that by setting the threshold low enough the result of the CloudLasso method becomes identical to that of CylinderSelection, i. e., lasso selection using generalized cylinders. One could thus see CylinderSelection as a special case of CloudLasso selection.

4.4.4 Example Results

Figure 4.5 shows a collection of results we generated by applying the CloudLasso to the same two datasets as discussed in Section 4.3.4 (colliding galaxies and cosmological N-body simulation). The top row shows the dataset before the selection, the middle row shows the result of selection applied for this view, and the bottom row shows a different view of the selection to illustrate how CloudLasso takes the 3D structure of the dataset into account. In Figure 4.5(a) a compact cluster was selected with a circular lasso, however, after rotating the dataset (bottom) we see that the galaxy's compact disk was selected. In Figure 4.5(b) an initially visually similar Figure 4.5(b, top) compact cluster was selected, but this selection reflects the compact rounded shape of the cluster. Figure 4.5(c) shows again a visually similar case where the cluster and consequently the selection also appear to be compact and rounded from the initial view Figure 4.5(c, top). However, after rotating the dataset (bottom) we can see that the cluster is elongated and that CloudLasso has taken this fact into account for the selection shape (including holes in the selection where appropriate). Figure 4.5(d) shows a more complex cluster where the 3D structure of the 'arms' is captured. Figure 4.5(e) shows a close-up view of Figure 4.5(d) after changing the density threshold. As can be seen, only high density clusters remain selected.

Table 4: CloudLasso performance. Times are in seconds.

Selection	Total Time	Density Estimation	Marching Cubes
Figure 4.5(a)	6.38	4.89	0.46
Figure 4.5(b)	2.13	0.83	0.39
Figure 4.5(c)	4.14	2.87	0.39
Figure 4.5(d)	4.50	1.95	0.39
Figure 4.5(e)	0.30	0	0.28

4.4.5 Performance

The computationally most expensive part of CloudLasso is the density estimation. The density estimation processing time increases with the number of particles inside the rectangular box B . The processing time also heavily depends on the distribution of the particles within B because, as outlined in Section 4.4.1, particles in sparse areas can dominate the computation. After density estimation, the second-most expensive part is the selection geometry construction with Marching Cubes.

We have measured the performance of CloudLasso on a 3.16 GHz Intel® Core™ 2 Duo CPU for the selections shown in Figure 4.5 and the results are summarized in Table 4. While the time required for density estimation is currently in the order of a few seconds, it can easily be parallelized using multiple parallel threads and cores (we discuss further possible improvements in Section 4.6.1). Moreover, changing the density threshold and recomputing the selection as in the selection of Figure 4.5(e) takes only little time since the density is not recomputed. Thus, the technique is well suited for interactive applications.

4.5 USER STUDY

To understand the user performance of and satisfaction with our selection technique we conducted a comparative quantitative evaluation (Figure 4.6). As the baseline we selected Lucas and Bowman’s Tablet Freehand Lasso (CylinderSelection) which, at this point, can be thought of as the standard selection method for point-based datasets. Due to the constraints of a controlled experiment we had to restrict ourselves to compare it to only one of our own techniques: CloudLasso or TeddySelection. We decided to use CloudLasso in the study because it is more flexible than TeddySelection due to the included threshold adjustment and the flexible construction of selection shapes in depth. Our comparison was based on both speed and accuracy as well as participants’ qualitative feedback for four tasks. Because CloudLasso is capable of cre-

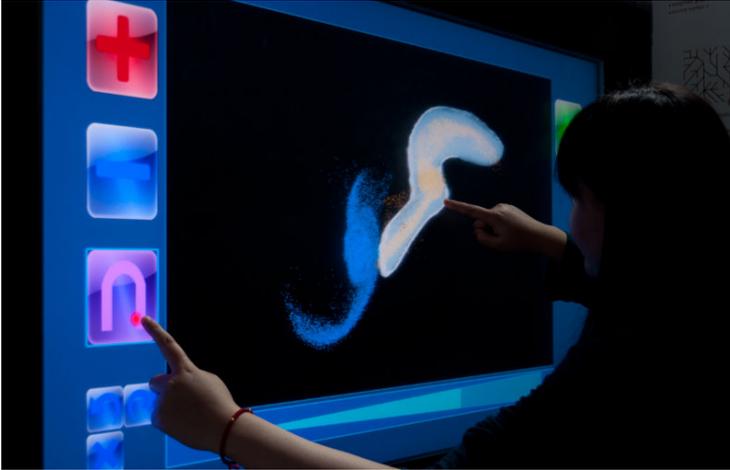


Figure 4.6: A selection being performed using our direct-touch interface.

ating selection shapes based on the intended selection's spatial structure we hypothesized that CloudLasso would outperform CylinderSelection for all tasks based on speed. Due to the flexible threshold-based adjustment of the selection possible with CloudLasso we also hypothesized that it would be at least as accurate as CylinderSelection. We further hypothesized that the CloudLasso would score higher on questions related to how efficient participants perceived each method to be and would be generally preferred.

4.5.1 Study Description

Participants. Twelve people (8 male, 4 female) participated in the study. Eight participants were students from different disciplines and four non-students. All of them had at least a Bachelor's degree. Ten participants reported prior experience with 3D computer games with playing games up to three times per week, with four participants reporting at least weekly experience. Ages ranged from 24 to 33 years ($M = 28.75$, $SD = 3.3$). Ten participants reported to be right-handed, while the remaining 2 people reported to be ambidextrous.

Apparatus. The experiment was performed on a 52" LCD screen with full HD resolution (1920×1080 pixels, $115.4 \text{ cm} \times 64.5 \text{ cm}$). The display was equipped with a DVIT overlay [Smart Technologies Inc., 2003] from Smart Technologies, capable of recognizing two independent inputs. The display was positioned so that its center was at a height of 1.47 m above the ground.

Tasks. Our study comprised four tasks. The dataset for each task (Figure 4.7) contained target particles (orange), interfering particles (blue), and noise particles (light blue). Participants were always asked to select

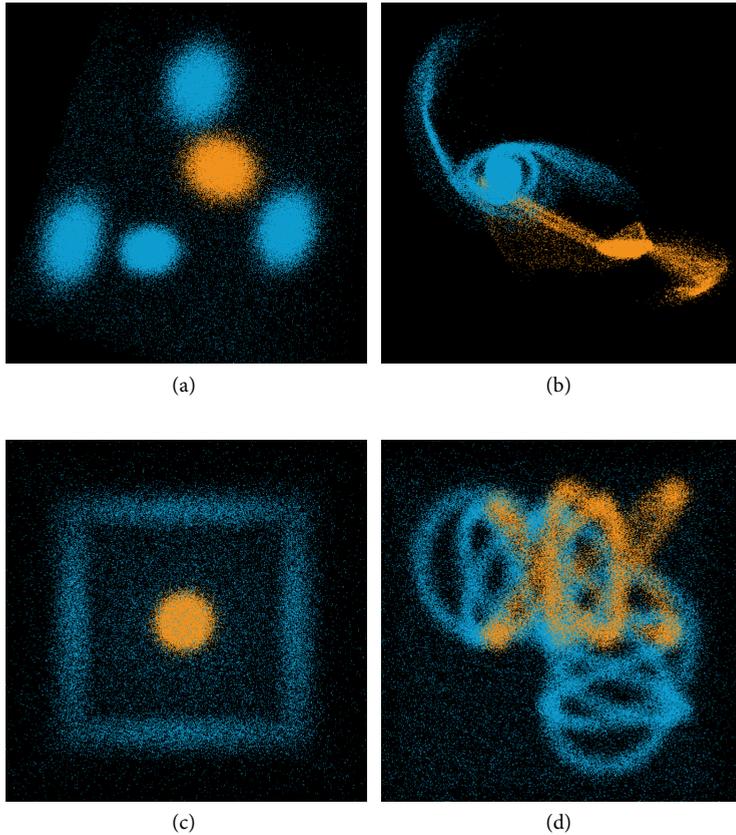


Figure 4.7: Four tasks: (a) five simple particle clusters, (b) two galaxies, (c) cubic shell and central core, and (d) three intertwined figure-eight knots.

the orange target particles. To avoid measuring unnecessary time spent on navigation we only provided trackball rotation. A selection was activated through spring-loaded modes [Buxton, 1986; Sellen et al., 1992] that allowed the participants to adjust selections while keeping a ‘button’ pressed (Figure 4.6); otherwise the 2DOF input on the data was used for rotation. There were three possible selection modes corresponding to three Boolean operations (Figure 4.6): union (+), intersection (\cap), and subtraction (-).

Before the actual experiment, four additional practice tasks were provided for participants to get accustomed to the selection techniques. The dataset for each of these tasks consisted of a low density, cubic volume of noise particles depicted in light blue and a higher density volume of orange target particles with a simple geometric shape: a sphere, a pyramid, a cylinder, and a torus.

The datasets used in the actual experiment are illustrated in [Figure 4.7](#). These tasks were designed to have different features and were ordered by difficulty. [Figure 4.7a](#) shows five randomly placed compact clusters of particles with equal uniform densities inside a low density noise environment, with one being set as the target cluster. [Figure 4.7b](#) is a simulation of two colliding galaxies which do not have uniform density. The participants were required to select one of the two galaxies. [Figure 4.7c](#) shows a spherical, high-density core of target particles surrounded by a medium-density, cubic shell of interfering particles. Both structures are inside a low-density noise environment. [Figure 4.7d](#) contains three intertwined particle ‘strings,’ each one shaped as a figure-eight knot, inside a low-density noise environment. Each ‘string’ had the same uniform density and participants were asked to select one of them.

At the start of each trial, the data space was oriented in a defined way (different orientations per trial). Participants were asked to finish the selection goal, i. e., to select the orange particles as quickly and accurately as possible. Participants thus needed to try their best to select, if possible, all target particles but to avoid selecting interfering or noise particles. They were asked in advance to find a balance between accuracy and speed and it was pointed out that a perfect selection was difficult or even impossible. We allowed participants to undo/redo the 5 most recent operations. Once participants felt that they accomplished the selection goal or that they were not able to improve the result, they could press a finish button to advance to the next trial. An additional density threshold slider was provided for CloudLasso trials ([Figure 4.6](#)).

Design. We used a repeated-measures design with the within-subject independent variable *selection method* (CylinderSelection, CloudLasso). Per method, each participant performed 4 tasks and per task 4 trials. For each trial we chose a unique dataset starting orientation. Tasks were always performed in the same order and the presentation order of the two methods was counterbalanced among participants.

In summary, the design consisted of 12 participants \times 2 methods \times 4 tasks \times 4 trials = 384 interactions in total. Participants moved from training to experiment after they reported to be able to perform a selection with the presented technique. After the experiment, participants were given a written questionnaire. They were asked to rate the usability of the techniques on a seven-step Likert scale with respect to ease of remembering, ease of use, efficiency, ease of drawing the lasso, and whether it was working as expected. Also, participants were asked to compare both techniques, comment on which technique they preferred and why, and whether the techniques allowed them to select the data as they desired. Finally, they filled in their demographic background information and were asked to provide additional verbal feedback on their experience of which the experimenters took notes.

4.5.2 Results

Completion times, errors, and selection volumes were recorded for the analysis. The density field for the CloudLasso method was pre-computed to compare only the real interaction times. With this optimization, the processing times for computing the selected particles for CylinderSelection and CloudLasso were both approximately the same (in the order of 0.5s). Time and error in our experiment did not follow a normal distribution. We thus used the non-parametric Wilcoxon Signed Ranks test to analyze the data. The first block of trials was removed from the analysis due to a strong learning effect being present between blocks 1 and 2, across participants. Thus, we analyzed three trial blocks per task and method for each participant. In addition, trials were marked as outliers when each metric (time, error) was beyond two standard deviations from the mean for a given task and method per participant. Outliers were replaced with the closest value two standard deviations from the mean for each participant according to standard procedure. The datasets used in the different tasks differed in their characteristics, so we analyzed the results of each task independently.

We used two different metrics from information retrieval to calculate the error of our results [Manning et al., 2008]. To compute these scores we used the response number of true positives (TP , correctly selected particles), false positives (FP , incorrectly selected particles), false negatives (FN , missing particles that had to be selected), and true negatives (TN , correctly unselected particles). From these, the precision (P)—the fraction of target particles of all the retrieved particles—is calculated as: $P = TP / (TP + FP)$ and the recall (R)—the fraction of the target particles that were selected—is calculated as $R = TP / (TP + FN)$.

The first metric, the F_1 score, calculates the harmonic mean of precision and recall and is often used in information retrieval to measure query classification performance. It is defined as $F_1 = P \cdot R / (P + R)$. A value of 1 for the F_1 score signifies a perfect result, while 0 is the worst possible result. Since the F_1 metric does not take the TN rate into account, we also used the Matthews correlation coefficient (MCC) as our second error metric which is often used in machine learning to assess the performance of a binary classifier. The MCC is calculated as:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Finally, we use a third error metric: the ratio of the selection volume and the target's real volume V_S/V_R , with values closer to 1 being better.

The task completion time analysis showed a significant effect with CloudLasso being significantly faster than CylinderSelection in all tasks except the task of the galaxies (Figure 4.8). Moreover, with the same exception of the galaxies task and the string task for MCC, the F_1 score

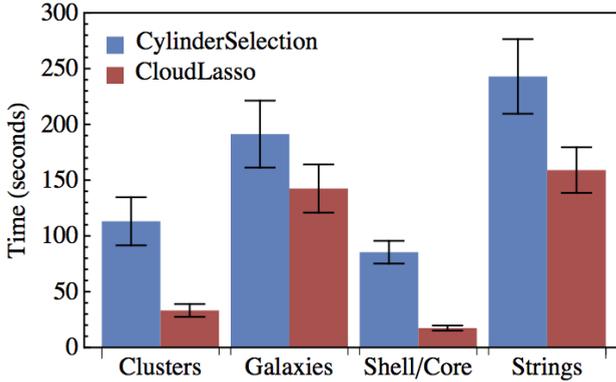


Figure 4.8: Mean completion times for the user study tasks. Error bars represent 95% confidence intervals.

and the MCC score also showed statistically significant differences between the two methods, with CloudLasso being more accurate than CylinderSelection. Table 5 shows the details of the statistical analysis.

Five clusters. Tests showed that CloudLasso was significantly faster than CylinderSelection (36 s vs. 119 s). Moreover, the F_1 and MCC scores showed statistically significant differences between the two methods, with CloudLasso being more accurate. While CloudLasso created smaller volumes, the V_S/V_R differences were not significant.

In the post-session questionnaire the participants were asked to choose the method they preferred for each task. All participants chose CloudLasso over CylinderSelection on this dataset. They reported that CloudLasso was easy and convenient to use and that they could always get the right result (8×), was faster (8×), and used less steps (3×). In fact, many participants finished the task with only one CloudLasso selection step while, at the same time, getting more accurate results than with CylinderSelection. Furthermore, CloudLasso did not require participants to precisely draw a close lasso around the target particles.

Two colliding galaxies. While CloudLasso, on average, was faster than CylinderSelection (141 s compared to 190 s) and more accurate according to all error metrics, we did not observe a significant effect in any of these measurements. In the post-session questionnaire, six participants preferred CloudLasso, four favored CylinderSelection, while the remaining two felt both methods were more or less the same.

A reason for these results may lay in the fact that this selection task required participants to select particles with varying average density, the density of the galaxy's core being high while the arms have much lower density. Because participants were asked to select the whole galaxy, multi-step selections were necessary for both CloudLasso and CylinderSelection. We had intentionally designed this task with these

Table 5: Mean completion time (in seconds), mean F_1 , mean MCC, and mean volume ratios for CloudLasso and CylinderSelection for the four user study tasks together with the corresponding significance scores.

		Clusters	Galaxies	Shell/Core	Strings
Mean Time (s)	CloudLasso	35.96	141.14	17.86	169.01
	CylinderSelection	118.94	189.62	88.01	248.68
	Z	3.06	1.41	3.06	2.28
	p	<.01	.16	<.01	.023
Mean F_1	CloudLasso	.9789	.9866	.9980	.7494
	CylinderSelection	.9759	.9855	.9960	.7303
	Z	2.67	0.71	3.06	2.04
	p	<.01	.48	<.01	.041
Mean MCC	CloudLasso	.9765	.9733	.9974	.6519
	CylinderSelection	.9731	.9712	.9948	.6305
	Z	2.75	0.78	3.06	1.89
	p	<.01	.43	<.01	.06
Mean V_S/V_R	CloudLasso	1.244	4.055	1.327	1.852
	CylinderSelection	1.303	5.855	1.360	2.691
	Z	0.94	1.49	1.57	2.98
	p	.347	.14	.875	<.01

characteristics because we hypothesized that it would be difficult for CloudLasso to select the whole galaxy in one step because it sets a density threshold based on the average density inside the lasso frustum. This threshold can only be varied inside a finite range because otherwise the interaction would become too imprecise. Since the average density in the galaxy is dominated by the very high density core, setting the threshold to its minimum possible value is still not enough to include the very low density arm edges. More than one selection step is thus necessary with CloudLasso to select all parts of the galaxy. CylinderSelection, in contrast, does not depend on the density distribution and always selects or deselects all particles in the lasso frustum. Therefore, CylinderSelection might have been more straightforward for some of the participants who all saw this dataset for the first time.

Cubic shell and core. CloudLasso was both faster than CylinderSelection (18 s vs. 88 s) and more accurate with respect to F_1 and MCC, with statistical significance. It also produced smaller volumes (smaller V_S/V_R) but this difference was not statistically significant. This clear advantage for CloudLasso is due to the fact that the noise and interfering

particles with a low density did not create any problem for selecting the high-density core in the center. With CloudLasso, participants could finish the selection task in just one step, while CylinderSelection required several Boolean operations for satisfying results. We consequently also received positive feedback about using CloudLasso for this dataset in the post-session questionnaire: all participants preferred CloudLasso over CylinderSelection. The main reason for this choice as reported by participants was that CloudLasso was faster (7×), easier to use (4×), more precise (2×), and needed less steps (2×).

Three figure-eight knot shaped particle strings. In this last task, CloudLasso was also significantly faster than CylinderSelection (169 s as opposed to 249 s) and significantly more accurate with respect to F_1 . It also showed a trend to be significantly more accurate on MCC and produced better (but not statistically significant) volume ratios.

We used this task at the end of the study because it is, by its very design, impossible to complete perfectly. Nevertheless, after the first CloudLasso selection step the selection already contained mostly target and interfering particles and not many noise particles. Thus, in subsequent steps, participants only needed to deselect the interfering particles. With CylinderSelection, in contrast, participants had to spend much time on ‘carving’ the particle strings. Many complained about fatigue using CylinderSelection for this task. In the questionnaire, eleven participants reported to prefer CloudLasso and one participant had no preference. All felt that both techniques were hard to use with this dataset, but also that the task was almost impossible with CylinderSelection while they could get better results with CloudLasso.

Overall Preferences. Participants were asked to compare both techniques in general. All of them named CloudLasso as their preferred technique. As their main reasons they reported CloudLasso to be easier to use (5×), faster (4×), more efficient (3×), and more precise (2×).

4.6 DISCUSSION

Based on the results from our study we now provide a comparing discussion between the three selection techniques mentioned in this paper, mention application domains other than the ones used so far, and discuss a further processing of the selections made with our techniques.

4.6.1 Comparison of the Selection Techniques

The TeddySelection and CloudLasso techniques introduced in this paper are two new *spatial* and *structure-aware* selection techniques. While we had previously already briefly touched upon their merits and limitations, we now provide an extended comparison between them and also include the CylinderSelection technique in this discussion.

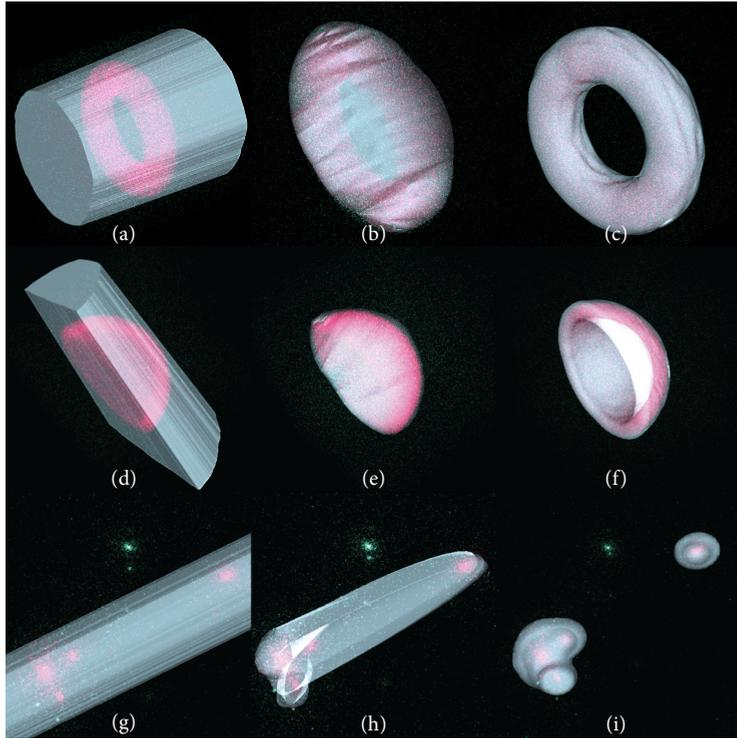


Figure 4.9: Comparison of CylinderSelection (left column), TeddySelection (middle column), and CloudLasso (right column) in three different examples. From top to bottom: selection of a torus-shaped structure, a concave hemispherical shell (i. e., an empty, thickly-walled bowl), and two clusters from an astronomical N-body simulation. The three selections in each example are made from the same viewpoint with respect to the particles and the result is viewed from roughly the same direction.

Both CloudLasso and TeddySelection are based on the same principle: a user-specified 2D lasso defines a region in space and particles inside this region are selected based on the particle cloud structure. Nevertheless, this structure is taken into account in different ways by both methods so their results typically differ. In contrast to our structure-aware techniques, CylinderSelection starts with the same 2D lasso but selects everything in the lasso frustum. Therefore, CylinderSelection usually selects much more space than the desired target and several Boolean operations are needed to achieve the intended selection, something that we saw confirmed in our user study. This property of CylinderSelection is clearly visible in the examples shown in Figure 4.9(a,d,g) where, in all cases, the selection geometry is very large after one step of the CylinderSelection method.

For roughly spherical shapes (not shown), both `TeddySelection` and `CloudLasso` give similar results and both methods are equally suitable because they deliver the intended selection in a single step. Nevertheless, even in such simple cases `CloudLasso`'s selection geometry is often smoother than that of `TeddySelection` due to `CloudLasso`'s density approximation. Furthermore, `CloudLasso` is generally more flexible than `TeddySelection` as the comparisons in [Figure 4.9](#) demonstrate.

In the selection of the torus and the hemispherical shell, `CloudLasso` perfectly fits the intended target ([Figure 4.9\(c,f\)](#)). On the other hand, `TeddySelection` provides a closed geometry that approximates the intended selections better than `CylinderSelection` but also selects the hole in the center of the torus ([Figure 4.9\(b\)](#)) and the bowl's cavity ([Figure 4.9\(e\)](#)). Finally, [Figure 4.9\(h\)](#) shows that `TeddySelection` selects not only the two clusters that lie behind each other but also the space in-between them. `CloudLasso` addresses this issue and correctly selects only the clusters, resulting in several disconnected parts ([Figure 4.9\(i\)](#)).

4.6.2 *Limitations*

`CloudLasso`'s most important limitation is its performance. In our performance analysis reported in [Section 4.4.5](#) we found that a selection typically required a few seconds. Results could vary, ranging from 2 s to more than 6 s in a way that can be unpredictable for the user. The main bottleneck is density estimation which can be addressed by parallelizing the respective computation as mentioned before. However, in the future we also want to investigate a GPU implementation of the kernel density estimation [[Daae Lampe and Hauser, 2011](#); [Srinivasan et al., 2010](#)] as well as of Marching Cubes. This would also address the second performance bottleneck of `CloudLasso` and take advantage of the fact that the data from the density estimation would already be available in GPU memory.

Both `TeddySelection` and `CloudLasso` are based on a number of parameter choices. For instance, `TeddySelection` uses two binning stages to stabilize the structure detection or clusters along the camera's z -direction. While `CloudLasso` also needs several parameters, most (e. g., the number of initial binning levels) do typically not have to be adjusted for different datasets. In fact, `CloudLasso` automatically adjusts to the density of an intended selection and provides a single parameter to adjust how closely a selection wraps around a cluster.

4.6.3 *Other Applications and Possible Modifications*

Our motivation for this work and the examples we have shown in the paper concern the selection of clusters in 3D astronomical particle datasets. Nevertheless, our methods can easily be applied for efficient

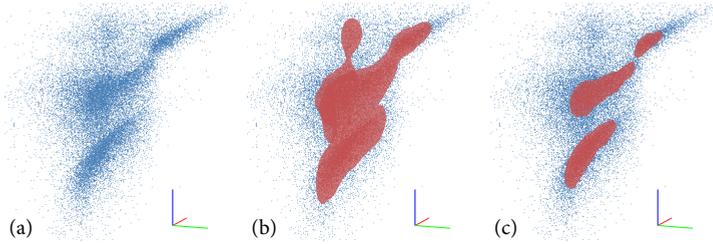


Figure 4.10: CloudLasso being applied to a 3D scatter plot; data from a subset of the milliMillennium dataset [De Lucia and Blaizot, 2007] (showing the dimensions virial radius, R-I color, and low X-ray luminosity); (a) original 3D scatter plot, (b) initial CloudLasso selection, (c) after interactive threshold adjustment.

selection in particle datasets that arise in other visualization domains as well as for any grid-based numerical data. For example, we recently applied CloudLasso as the selection technique in an application for the analysis of abstract, high-dimensional data. In the application the possible n -dimensional subspaces of a high-dimensional dataset are automatically ranked in terms of predefined quality measures, based on the number and prominence of clusters, to focus subsequent analysis on the highest-ranked subspaces. Three-dimensional subspaces are visualized directly and for larger-dimensional subspaces their respective three principal components are visualized. Based on such data or any other three-dimensional scatter plot, CloudLasso can be used for spatially selecting dense clusters as shown in Figure 4.10. The selections can then be further analyzed by means of brushing and linking.

So far we have only considered the selection of unstructured 3D particle data. In many applications such as simulations of 3D flows and medical imaging, however, it is common to have scalar data defined on a 3D grid. We can easily modify CloudLasso for such datasets. CloudLasso would need to select those grid points that project inside the drawn lasso and where the value of the scalar quantity is above an automatically or user-defined threshold. In this case we can avoid, in fact, the density estimation step of CloudLasso since we already know the value of the scalar data on a 3D grid. On the other hand, in the visualization for such applications the region of interest may be enclosed by a semi-transparent surface that the user wants to ignore but that may otherwise interfere with the selection. In such cases CloudLasso should be combined by automated methods that can detect and resolve such ambiguities (e. g., [Mühler et al., 2010; Wiebel et al., 2012, 2011]).

Finally, CloudLasso is suitable for particle selection based on scalar properties other than density. If a desired property is continuously defined in space we can evaluate it on a regular 3D grid. We can thus

use Marching Cubes to select the spatial region where the target value is above a given threshold as described in the previous paragraph.

4.6.4 *Further Selection Processing*

While our selection methods allow users to intuitively perform spatial selections based on particle density, users sometimes need to modify their current selections if they did not achieve their desired precision in one step. This issue can easily be addressed by allowing Boolean operations with consecutively specified selection volumes. In this Boolean processing we can also combine selection methods, so that some of them are done with CloudLasso or TeddySelection, while others are done with CylinderSelection. In fact, adding new selections works best for CloudLasso or TeddySelection, while subtractions or set intersections work most intuitively with CylinderSelection.

Moreover, particle datasets including the cosmological N-body simulation data we used as our examples often consist of tens of terabytes of particle data, all of which rarely fits into the computer's main memory. To enable selections for these situations we can perform our selection specification based on a well-defined sample of the whole data and then store the selection shapes along with the employed Boolean operations in a selection pipeline. We can then apply this pipeline to all particles of the large dataset in an off-line process.

4.7 CONCLUSION

We presented CloudLasso and TeddySelection—two spatial, structure-aware selection techniques that can be applied to 3D particle cloud datasets in a visualization context. These techniques only require that users draw a lasso around the 2D projection of what they consider to be important by means of 2DOF input such as a mouse/pen or a finger on a direct-touch display (Figure 4.6). With this input, our approaches allow users to select whole regions of space in a single step without having to rely on the selection of individual objects or having to refine a single selection repeatedly. In TeddySelection we married a technique from sketch-based modeling with generalized cone-based lasso selection and extended both to enable data selections that take 3D structural information into account. In CloudLasso we combined density estimation methods and a constrained version of Marching Cubes to uncover and select the most prominent 3D structures along the lasso direction.

Our techniques allow people to perform complex spatial selections in a wide range of applications and datasets, and the resulting selection shapes can be considered to be intuitive and efficient according to the feedback from the user study and from our collaborating experts.

Both methods solve issues with current selection techniques employed in data analysis tools, allow people to explore sub-regions of particle datasets without requiring prior knowledge about the structures within this data, and thus open up new ways for data exploration.

CloudLasso and TeddySelection are the first steps in the development of structure-aware selection techniques for 3D particle datasets. We are certain that further work can improve these methods and generate interesting new ideas. In this paper we have established the relevance of such techniques for 3D particle selection and showed that they are more efficient than the traditional selection techniques. We, thus, contribute to the integration of interaction and visualization research [Keefe, 2010] with the goal of improving scientific data analysis workflows.

4.8 FROM INTERACTION TECHNIQUES TO A VISUAL ANALYTICS TOOL

Together with Chapter 3, we introduced two essential interaction techniques for 3D data explorations which can be used in many fields. However, the concepts would need to be applied to the real scientific data explorations to see whether scientists can better understand the data in their domain by using the new techniques. Therefore, in next chapter we would introduce a visual analytic tool which integrated the interaction techniques on large, touch-sensitive displays. Furthermore, an observational study was carried out to evaluate the visual analytics tool in the context of astronomical data exploration.

ACKNOWLEDGMENTS

From the University of Groningen, we would like to thank Amina Helmi for the cosmological simulation dataset from the Aquarius Project [Springel et al., 2008], Frans van Hoesel for the galaxy collision simulation (originally from <http://www.galaxydynamics.org/>), and Michael Wilkinson for his advice and valuable comments on the paper and for his density estimation code from DATAPLOT [Wilkinson and Meijer, 1995].

This chapter is based on: Lingyun Yu, Konstantinos Efstathiou, Petra Isenberg, and Tobias 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, December 2012.

A VISUAL ANALYTICS TOOL FOR DATA EXPLORATION ON A LARGE, TOUCH-SENSITIVE DISPLAY

ABSTRACT: *We designed an integrated visual analytics tool for subspace clustering analysis of astronomical datasets on large, touch-sensitive displays. Our tool integrates several interaction techniques in 3D space. The FI3D navigation technique provides users the ability to navigate through and to explore the 3D space. Moreover, the CloudLasso selection technique not only assists the user in selection in 3D space avoiding the need for a precise lasso and reducing the number of selection steps, but also reveals the structure of the selected data. We also use another selection technique, brushing, to select data from a parallel coordinates plot showing all dimensions of a high-dimensional dataset. Finally, in order to evaluate our integrated visual analytics tool, we carried out an observational study and received positive feedback from the participants.*

5.1 INTRODUCTION

DATASETS in many scientific areas are growing to enormous sizes. For example, modern astronomical surveys do not only provide image data but also catalogues of millions of objects (stars, galaxies), each object with hundreds of associated parameters. A main line of research within astronomy is geared toward investigating multi-dimensional and multiscale patterns in this kind of data. For example, an important task for data analysis in astronomical research is to explore the relation between galaxy morphology (i. e., the spatial distribution of objects) and the parameters associated to the objects which characterize the stellar environment.

While the exploration of high-dimensional information spaces poses a huge challenge, combining data mining approaches with visualization can enable users to explore such large datasets. Clustering is a well known data mining task that facilitates the discovery of natural structures in a dataset [Kriegel et al., 2009]. Full-dimensional clustering techniques are not particularly useful due to the exploratory nature of the task. Clusters may exist in different subspaces that may indicate different relations among particular subsets of dimensions. Therefore, subspace clustering is an approach that can be applied for this purpose. Subspace clustering is the process of finding clusters in subspaces of the full feature space, either directly [Agrawal et al., 1998] or by identifying relevant subspaces for (later) clustering based on some quality criteria [Baumgartner et al., 2004]. Ferdosi et al. [2010a] proposed to find relevant subspaces by an approach that is strongly tied to morphological properties of the distributions of objects.

Interactive visual analytics can be useful for the study of high-dimensional datasets that are not only large in size and dimension but also complex in nature. Visual analytics can make automatic analysis processes more transparent to the users because it supports visualization of the intermediate results. It also assists people in better understanding the obtained information because it allows the users to interact with the system and study the properties of the dataset from several perspectives. Several tools exist to facilitate the analysis of high dimensional data. Ferdosi et al. [2010a] listed several analytics tools in astronomical community to facilitate the analysis of astronomical data: MTdemo¹ which visualizes 3D density images, TOPCAT² which provides 1D/2D/3D scatter plots, and GGobi³ which has several information visualization techniques such as 1D/2D scatter plots, parallel coordinate plots, tours etc., with linked views. However, none of these tools provides any guidance for exploring high-dimensional datasets that have a large search space, like the work presented by Ferdosi et al. [2010a]. Furthermore, users need to switch among several separate visualization tools to study the properties of the data from different perspectives. Selecting interesting data from one interface and then checking their properties from another interface is even more difficult.

Thus, it is useful to have an integrated visual analytics tool for visualizing, analyzing, and exploring relevant information out of the flood of high-dimensional data. The interface should provide several different visualizations to facilitate data exploration from different perspectives. In addition, collaborative data exploration involves several people and combines different points of view to achieve the goals of the data analysis. Large, touch-sensitive displays are particularly suitable for collaboration. The observations by Klein et al. [2012], for example, indicate that collaborative scientific visualization is facilitated by multi-touch settings on large displays. Our user study in Chapter 3 also showed a clear preference of participants to use touch interactions on the large display for exploration tasks.

In this chapter we therefore present an integrated visual analytics tool for data exploration on large, touch-sensitive displays. Our tool integrates the FI3D navigation technique (Chapter 3), the CloudLasso selection technique (Chapter 4) and the subspace clustering approach by Ferdosi et al. [2010a]. It supports users in collaboratively analyzing high-dimensional datasets. An observational study was carried out to evaluate our visual analytics tool in the context of astronomical data exploration.

The remainder of this chapter is organized as follows. Related work is discussed in Section 5.2. A detailed design of the integrated visual analytics tool is presented in Section 5.3. Section 5.4 describes the ob-

¹ <http://www.cs.rug.nl/~michael/MTdemo>

² <http://www.star.bris.ac.uk/~mbt/topcat>

³ <http://www.ggobi.org>

servational study and its results which was carried out to evaluate this tool. [Section 5.5](#) gives a summary along with plans for future work.

5.2 RELATED WORK

Our visual analytics tool is designed to work with high-dimensional astronomical data and combines visual analytics, interactive visualization, and touch interactions. Therefore, in this section, we first talk about visual analytics tools for data exploration. Then we discuss visual analytics tools on large, touch-sensitive displays.

5.2.1 Visualization of High-Dimensional Data

The work by [Keim \[2002\]](#) presents a classification of information visualization and data mining techniques with respect to data type, visualization technique, and interaction & distortion technique, respectively. [Ferreira de Oliveira and Levkowitz \[2003\]](#) discuss general graphical and interaction techniques for visual data mining of large and/or high-dimensional datasets represented as tabular data. [Yang et al. \[2003\]](#) discuss interactive hierarchical displays as a general framework for visualization and exploration of large multivariate data sets; see also [\[Fua et al., 1999\]](#), [\[LeBlanc et al., 1990\]](#) for earlier work. [Heine and Scheuermann \[2007\]](#) discuss interaction techniques for manual clustering refinement using the soap bubble metaphor. This work is motivated by applications such as gene expression data analysis or natural language studies, where automatic clusterings need to be manually corrected. [Elmqvist and Fekete \[2010\]](#) present a model for visualizing and interacting with multiscale representations of information visualization techniques using hierarchical aggregation, in order to improve overview and scalability for large datasets. There exist several tools adapted for data exploration in the specific domain of astronomy; a detailed discussion of such tools was carried out by [\[Ferdosi, 2011\]](#).

All the previously mentioned tools serve as an inspiration for our approach. Nevertheless, they are all particularly useful when the analysts know where to look for certain information. However, the goal of the analysis is to find “unexpected” phenomena in the data for which by definition no *a priori* description is available, thus precluding the possibility of fully automated search. For this reason, we used the subspace clustering method for the analysis of high-dimensional sets in combination with interactive exploration.

Subspace clustering is an approach that can be applied for extracting the most important physical information. The main idea is to find clusters in subspaces of the full feature space, either directly [\[Agrawal et al., 1998\]](#) or by identifying relevant subspaces for (later) clustering based on some quality criteria [\[Baumgartner et al., 2004\]](#). [Ferdosi et al. \[2010a\]](#)

present a method for ranking subspaces in high-dimensional data in terms of their relevance for clustering. This approach provides not only a good quality criterion for subspace clusters but also offers visual support to the subspaces analysis. Therefore, it extracts and visually reveals the most important physical information of the high-dimensional data.

Our interactive analytics tool incorporates the subspace clustering approach of [Ferdosi et al. \[2010a\]](#). Moreover, it provides the ability to analyze and visualize the relations between the abstract information—subspace clusters—and the actual spatial information.

5.2.2 *Interactive Exploration Tools*

One crucial aspect of our visual analytics tool is its interactivity that assists users to explore the datasets and to refine the results given by the subspace clustering method. There is a large number of interactive exploration tools for specific domains. One example is VolumeShop [[Bruckner and Gröller, 2005](#)], an interface to specify objects of interest through three-dimensional volumetric painting; see also the work by [Gerl et al. \[2012\]](#) for an application of VolumeShop. Linked views appear frequently in analytics tools for complex data. [Chen et al. \[2009\]](#) present a novel interface for interactive exploration of DTI (Diffusion Tensor Imaging) fibers and provides widgets for manipulating the DTI fibers as both 3D curves and 2D embedded points. [Jianu et al. \[2009\]](#) introduce a visual exploration approach to exploring complex fiber tracts by combining 3D model viewing with 2D representations. [Turkay et al. \[2011\]](#) introduced a visual interactive analysis tool that helps the analysts to understand the underlying relations between items and dimensions of high-dimensional data, by using Brushing and Focus + Context. There also exist some visual analytics systems using large displays. One example is [Isenberg and Fisher's \[2009\]](#) Cambiera, a system for information foraging activities on a multi-touch tabletop display. It encourages analysts to collaboratively analyze large text document collections around the tabletop.

In this work we designed an interactive visual analytics tool for a large, touch-sensitive display that is particularly suited to the exploration of high-dimensional astronomical data. This tool combines existing interactive techniques, such as linked views and brushing, together with the navigation and selection interaction methods for 3D spaces, described in [Chapter 3](#) and [Chapter 4](#) respectively, in order to provide an integrated exploration environment.

5.3 A VISUAL ANALYTICS TOOL FOR DATA EXPLORATION ON A LARGE, TOUCH-SENSITIVE DISPLAY

Given the advantages of visual analytics tools and large, touch-sensitive displays, we designed an integrated visual analytics tool for analyzing astronomical datasets on touch-sensitive displays. The proposed system integrates into a single interface the required facilities and visualization techniques for the analysis of subspace clustering. Unlike other existing tools, our system

- considers the domain-specific needs of astronomical data such as the spatial and non-spatial visualization of astronomical objects, exploration of large-scale and precise interaction with the astronomical objects,
- enables users to find and study relevant subspaces from clustering this high-dimensional data,
- provides 7 DOF for spatial navigation in the physical space,
- supports users to select or brush in spatial or non-spatial visualization of astronomical datasets, and consequently link the selection to other views, and
- allows users to interact with the system to select and examine interesting particles / properties.

5.3.1 *Design Requirements*

To facilitate the interactive and iterative approach of parameter selection, the interactive data exploration system should provide a dedicated visualization window. However, there should be two such interfaces: one for 2D subspaces and another one for 3D subspaces, or for the first three principal components of the subspaces with dimensionality higher than three. In addition, the system should provide a way to navigate the ranked subspaces individually.

Secondly, the spatial positions of astronomical objects play an important role in the analysis process in astronomy. Astronomers are used to associate findings in parametric space with those in the spatial domain. Therefore, the system should also provide a dedicated interface for the exploration of spatial plots in combination with the parametric space plots.

Finally, the previously mentioned visualization requirements concern only 2D and 3D visualization techniques, with subspaces of dimensions higher than three being visualized after a transformation using PCA. However, such transformations can cause a reduction of information. Therefore, the visualization of these subspaces without

any transformation and in their original dimensionality can add value to the analysis. Thus, the tool should support high-dimensional data visualization as well.

To summarize the requirements stated above, the system should

- R1: support simultaneous interactive exploration of spatial and non-spatial data,
- R2: support visualization of data in 2D, 3D, and in dimensions higher than three,
- R3: support visual analytics to find and explore large parametric spaces for finding trends or clusters,
- R4: support linked views, and
- R5: allow both large-scale and precise interactions.

Interactive visual analytics tools to explore the data should be provided for all of the dimensions. This is a bottom-up process, ranging from one-dimensional subspaces to $(d-1)$ -dimensional subspaces, where d is the dimensionality of the dataset. However, considering that users need to have a clear insight for being able to find clusters in 2D subspaces, we start the subspace computation by determining all possible two-dimensional subspaces using the MBE method with automatic smoothing parameter [Wilkinson and Meijer, 1995; Ferdosi et al., 2011]. The procedure is different for 3D or >3D subspaces: the user sets what kind of highest-ranked subspace she or he wants to compute (highest-ranked 3D subspace, or 4D etc., up to $(d-1)$ D). Note that in our method not all possible combinations will be computed; instead, we use the highest ranked subspace from the previous subspace calculation, starting from 2D (which is the only dimension for which we compute all possible combinations) [see Ferdosi et al., 2010b].

Even though our tool aims to support visual analytics processes to find subspaces for clustering, scatter plot visualizations are not sufficient to observe the clustering of the subspaces. The visualization of density images with color mapping can help in identifying the relevant structures. That leads us to adding another requirement for our system which needs to:

- R6: provide a visualization technique that emphasizes the structures that are present in space.

The size and dimensionality of astronomical data can vary and can be very large, thus the tool should also

- R7: be scalable in terms of size and dimensionality of the dataset.

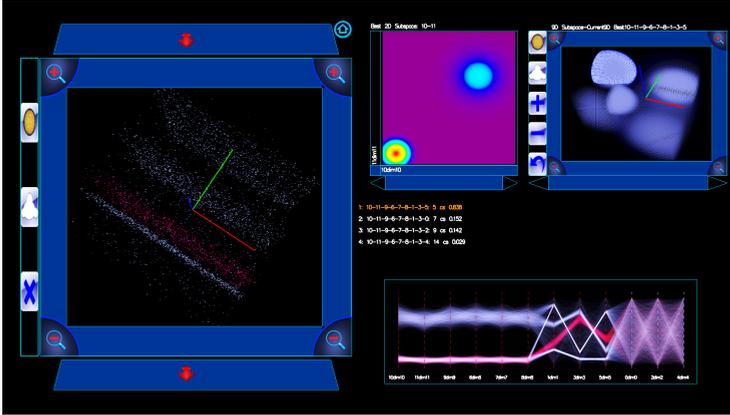


Figure 5.1: Interface of the proposed system. Left half: window I_1 for a 3D spatial plot. Right half: (top left) window I_2 for a 2D scatter plot / density plot; (top right): window I_3 for a 3D scatter plot / density plot; (bottom): window I_4 for a parallel coordinate plot of parametric space.

5.3.2 Design and Interactions

In astronomical data analysis such as the study of the large-scale structure of the universe, the interactions that allow both large-scale and precise interactions with the data are invaluable. Such interactions can be also useful for other types of dataset.

To enable such interactions, we describe the design of an interface that incorporates the subspace finding technique introduced in this paper. Thus we added four widgets (Figure 5.1):

- I_1 a spatial plot window (left half) which shows physical positions;
- I_2 a 2D subspaces window (right half, top left) which provides a visualization of the 2D data as a density plot;
- I_3 a 3D or $>3D$ subspaces window (right half, top right) shows a visualization of the high-dimensional data as a density plot; and
- I_4 a parallel coordinates plot window (right half, bottom), which shows all data dimensions.

SPATIAL PLOT WINDOW I_1 To support the users for interactively exploring spatial data (requirement R_1) we added the widget I_1 , which provides 7DOF (requirement R_5 , translation in x -, y -, and z -direction, orientation with respect to the 3D coordinate system, and uniform zoom) using only a single touch [Yu et al., 2010]. In addition, the frame interaction widget [Yu et al., 2010] also provides 2-touch interactions

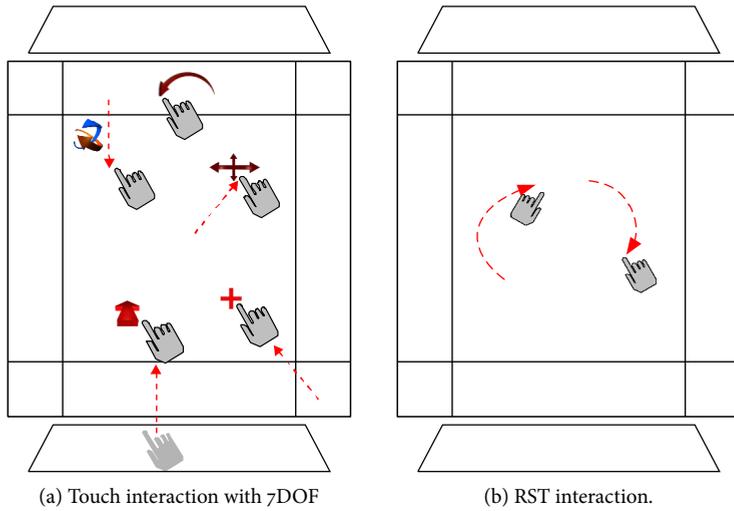


Figure 5.2: Spatial plot window I_1 with 7DOF using only a single touch, and RST (rotation, scaling, translation) by 2-touch interactions.

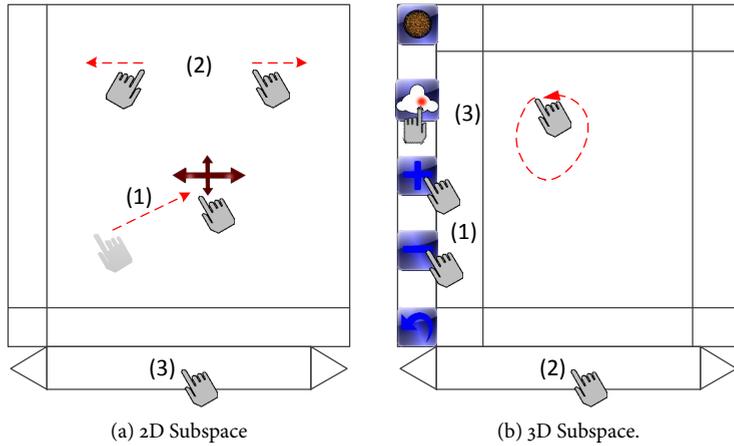


Figure 5.3: Subspace finding interfaces I_2 (left) and I_3 (right), for 2-D, and higher than 2-D, subspace computation, respectively.

for RST (combined z -Rotation, Scaling, and x/y - Translation), and for constrained $x/y/z$ -rotation; see Figure 5.2).

2D SUBSPACE WINDOW I_2 To support visual analytics for visualizing and exploring large parametric spaces (requirements R_1 , R_2 and R_3) we added the widgets I_2 and I_3 (Figure 5.3). By clicking the button

on the bottom of the widget I_2 (Figure 5.3a (3)), 2D subspaces are calculated and requirement R_3 is achieved. After computation, the ranking list shows the ranking result and, accordingly, the dimension ordering in the interface I_4 will be changed. Thus, the result from 2D subspace computation is visualized in I_2 . To emphasize the structures present in the data (R_6) we use a color mapping of the 2D density plot initiated by the user's demand. The user can click the left/right button to navigate the ranked subspaces. Additionally, the user can interact with I_2 with one touch (x/y -translation, Figure 5.3a (1)) and two touches (combined x/y -translation and scaling, Figure 5.3a (2)).

3D SUBSPACE WINDOW I_3 After the user selects an interesting 2D subspace, she or he can continue to compute higher-dimensional subspaces according to the selected 2D combination in window I_3 (Figure 5.3b). By clicking the “+” or “-” button (Figure 5.3b (1)), the dimension number can be changed according to the computation requirement and the computation is initiated after clicking the button on the bottom of the widget (Figure 5.3b (2)). For the 3D case, a color mapping is not useful as it hides the structures inside the volume, thus an X-ray rendering of the 3D density was chosen. Similar as in window I_1 , the user can interact with 7DOF using one touch and RST with two touches. Therefore, we satisfied the requirements R_1 and R_5 .

PARALLEL COORDINATES PLOT WINDOW I_4 To emphasize structures present in the data using parallel coordinate plots, we render the data using histogram-based color coding (lower-right window in Figure 5.1). The order of the dimensions is updated according to the computation results from I_2 and I_3 .

SELECTION AND BRUSHING We also provide the users with the ability of selecting and brushing. In windows I_1 and I_3 , the user can select particles using the CloudLasso technique; see Chapter 4. The user draws a lasso around the target particles with a bi-manual gesture, one finger to specify the selecting mode and the other finger (from the dominant hand) for drawing the lasso (Figure 5.3b (3)). In window I_4 the user can directly brush the data above the dimension axes. The selection is linked to the other windows. Therefore, we satisfied the requirement R_4 .

5.4 EVALUATION

In order to determine whether our design decisions satisfy the desired requirements we performed a user evaluation. Considering that the proposed tool is our first exploratory design, we were interested in studying how the astronomical experts would analyze their data by

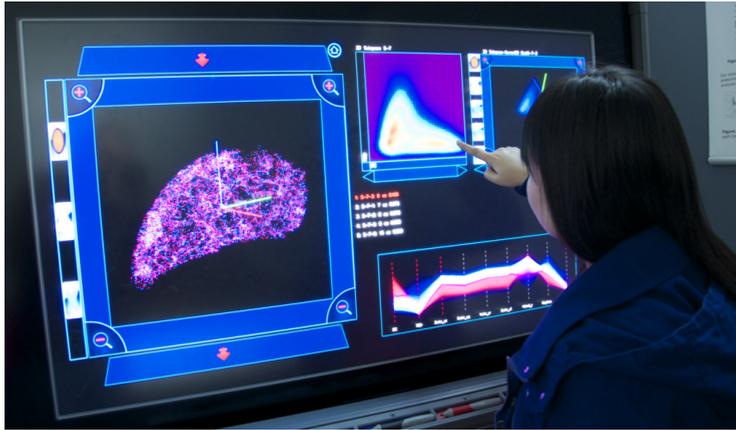


Figure 5.4: System demonstration on a 52" dual-touch wall display.

using our tool, in particular whether they would find interesting phenomena which they did not expect before. On the other hand, our tool integrated several techniques, such as subspace finding, navigation, selection, brushing, and linked views. Thus, we conducted the user evaluation as an observational study [Carpendale, 2008] combined with contextual interviews. During the evaluation, the participants could freely choose different techniques to explore the data and discuss their questions with us.

PARTICIPANTS Four astronomical domain experts (one female, three male) participated in the evaluation. Two of them had 19 years of experience each in astronomy and computer science after their Master's degree, while another two had six years of experience each. All of them reported that they work with various types of astronomical data, including 2D/3D-spatial positions, multi-dimensional abstract data (the dimensionality varies from six to fifty), and data from both observations and simulations. Their common concern with respect to data analysis is to find the correlations, clusters, outliers, and linked (physical) properties from various attributes of galaxies. Thus, it is important for them to be able to extract essential dimensions from multi-dimensional data. Normally, they would use their background knowledge about the astronomical data to make a judgement on which dimensions are essential. One participant described it as follows: "Sometimes certain dimensions should be close to essential because of approximate symmetries in the problem, but in real life things can be more complicated". They found it difficult to find relations between properties if they did not know that such relations already exist.

DATASET AND APPARATUS The datasets we used in the evaluation include one of the synthetic datasets (Ferdosi et al. [2010b], 12 dimensions) and the galaxy sample from SDSS (Ferdosi et al. [2010b], 15 dimensions). The evaluation was performed on a 52" LCD screen with full HD-resolution (1920 × 1080 pixels, 115.4 cm × 64.5 cm). The display was equipped with a DVIT overlay [Smart Technologies Inc., 2003] from Smart Technologies, capable of recognizing two independent inputs. The display was positioned so that its center was at a height of 1.47 m above floor level (Figure 5.4).

DESIGN The participants were asked to fill out a questionnaire about their demographic background, the datasets they work with, their current analysis tools, as well as the information they try to obtain from the analysis. After that, we used an example (synthetic) dataset to show the participants how to use our proposed system. Considering that expert knowledge is very important for making decisions during the evaluation, we did not only show them how to use the system but also explained the underlying method, including subspace finding, selection, and linked views. After the participants fully understood how the system works, we started the real evaluation with the astronomical dataset (the galaxy sample from SDSS). During the exploration, according to their own needs, the participants could compute 2D, 3D, or > 3D subspaces, or select and brush the data using different widgets. Meanwhile, we discussed with them about interesting new phenomena that they detected during their exploration. Finally, the participants were asked to evaluate our system, comment on what they found, and give suggestions about the proposed system.

FEEDBACK The feedback of the participants was positive in general. They quickly understood the interface and were able to use it after a short instruction. The subspace finding method was useful to extract essential dimensions of the multi-dimensional data. The linked views supported them to explore the relationships present in the data, and the interactivity and immediate feedback further assisted them in the data exploration. They commented that, with the visualization and the interactions we provided, it was easy and flexible for them to compare the ranked subspaces and get an immediate qualitative estimate of the properties of each subspace, such as clustering. In addition, they reported that they could easily choose an interesting combination of dimensions and compare the result with their intuitive expectations.

Participants particularly liked the immediate visual feedback on the large touch interface. They commented that the large touch screen meant a significant advantage to them because it was able to show the whole set of properties in multiple views, which helped them to better interact with the data, and to understand the overall structure and relations between the information shown in the different windows.

One of the participants stated that, with the large touch screen, it is much easier to explore and try new things and follow where the data takes you. She stated that it is a lot better than only relying on her own intuition or guessing which dimension will be important. Another participant stated that he could imagine employing a system like ours for discussing astronomical datasets with others.

Moreover, participants felt that the linking between the abstract and the spatial visualizations provided by our system was very powerful. It gave insight into where different types of objects, as determined by their (non-spatial) attributes, are located in the spatial domain, or the other way around. Furthermore, they even learned about some interesting but unexpected correlations and uncovered physical information that they would have missed in the usual and less flexible approach (without the subspace finding technique and without the linked spatial and abstract visualization). In addition, the interactions provided by our interface such as rotation, translation, zoom, and selection all helped participants to understand the global and detailed properties of the datasets. The ability to interact with data in this way revealed new insights into the underlying properties of the data.

However, there are also some issues with our tool. The main issue is that, for some datasets (in our example the SDSS dataset), it is difficult to recognize clusters from the visualization in the window I_3 , which affects the ability to understand the underlying properties of the data. Furthermore, it would be helpful if we could list which three principal components are shown in the visualization, and how they are related to the given physical properties. Moreover, sometimes it is cumbersome to scroll through the whole ranking list to find an interesting combination, especially in the case of the 2D subspace finding process when all 2D combinations are computed.

Furthermore, the participants also gave us some suggestions about improving the proposed system. Although the current strategy, in which all dimensions are included into the subspace computation, helps the users to discover unknown phenomena from their datasets, sometimes the users' expert knowledge may also be useful to efficiently get more accurate and meaningful results. For example, they would like the system to allow them to manually select the dimensions they want to be included in the computation. Besides that, in case the datasets do not have spatial information, or the spatial positions are not important, they would like the system to provide the ability to select which dimensions would be visualized in the window I_1 . We are considering these suggestions for future work.

5.5 SUMMARY AND FUTURE PLANS

In order to facilitate exploration and analysis of datasets with a large number of dimensions, we designed a visual analytics tool on a large, touch-sensitive display. The tool supports the users to visually analyze the data and explore large parametric spaces for finding clusters. More specifically, the FI3D navigation technique provides the users the ability to examine the physical position of the astronomical particles in a 3D representation with 7 DOF navigation. Furthermore, selection serves as a prerequisite for further studying the properties / relations for some interesting parts. Here we use CloudLasso to select the region of interest in both the spatial positions window and the abstract information window. Moreover, CloudLasso also helps the users to identify the structure of the data / clusters in the 3D space. In the parallel coordinates plot window we use the brushing selection technique to select data directly on dimension axes. After that, the selection is linked to all other windows. We performed a user evaluation and the feedback from the participants was positive in general. One thing that was clear from the user feedback was that automatic data analysis is not sufficient and users asked for the possibility to take advantage of their domain expertise by being able to manually affect the analysis process.

In the future, we plan to refine the design of our visual analytics tool on the large, touch-sensitive display according to the feedback from the user evaluation. We plan to adapt our navigation and selection technique to other types of datasets. Furthermore, cooperative work is very important in visual analytics systems in the sense that users can more easily share their ideas with each other. However, it also presents a lot of challenges. For example, it is difficult to distinguish touches from different people. Therefore, in order to complete our design, we would like to also develop techniques for cooperative work for multiple users.

ACKNOWLEDGMENTS

We thank Prof. Amina Helmi of the Kapteyn Astronomical Institute for the Galactic stellar halo dataset used in this chapter.

This chapter is based on: Bilkis J. Ferdosi, Lingyun Yu, Hugo Buddelmeijer, Scott Trager, Michael H.F. Wilkinson, Tobias Isenberg, and Jos B.T.M. Roerdink. Finding and Visualizing Relevant Subspaces for Clustering High-Dimensional Data Using Connected Morphological Operators. Submitted.

In this thesis we discussed a number of novel interaction techniques for 3D scientific visualizations: FI3D, a navigation technique for 3D spaces, as well as TeddySelection and CloudLasso, two selection techniques for 3D particle datasets. These techniques assist users at each step as they explore their datasets. First, users see the data from different points of view, and then, they focus on the most interesting parts by selecting and analyzing them.

The development of these techniques was motivated by discussions with astronomers concerning the problems they face in the analysis of their experiments and data. As we described in [Chapter 2](#), we asked several astronomers to fill in a questionnaire that asked for the types of datasets that they study and their established analysis methods. This work led us to identify a number of challenges for the domain of scientific visualization: navigation and selection in 3D space, uncovering structures in 3D datasets, studying high-dimensional datasets, and collaborative analysis.

In this chapter, we first summarize how we addressed these challenges in our work and the contributions of this thesis. Then we take a step back and look at the wider picture. We discuss what else can be done to address the identified challenges and present a plan for future work.

6.1 NAVIGATION IN 3D SPACE

In [Chapter 3](#) we introduced FI3D, a navigation technique for 3D scientific visualizations. This technique maps the 2 DOF input from the touch screen to 7 DOF interactions in the 3D space (translation along the x -, y -, and z -axes, rotation around the x -, y -, and z -axes, and uniform scaling). FI3D uses a frame around the main viewing area to activate interactions in the 3D space leaving the central part of the interface available for visualizing the data. This design allows the users to concentrate on the data exploration and provides an intuitive mapping from the 2D touch surface to the 3D space.

In order to see whether our design can help users better understand the data, we did a user study to compare FI3D on a large touch-screen display to traditional input devices—mouse and keyboard. In general, FI3D was competitive to the more familiar devices. More specifically, we found that familiar devices are suitable for the time-pressured tasks, while the FI3D technique is more suitable for open-ended tasks. Fur-

thermore, we also found that the study participants enjoyed with FI3D the feeling of “having the data under their finger tip”.

In addition, we also adapted our technique to explore other kinds of datasets—for example, an illustrative medical visualization tool [Svetachov et al., 2010]. Note that for applying the technique to different domains, some mappings might need to be changed. For example, it may not be necessary to travel through the space (therefore, there is no need for z -translation) but it may be useful to be able to have interactions that progressively reveal the inner structure of an object. Our FI3D frame interaction technique can be easily modified to satisfy any such different exploration requirements.

6.2 STRUCTURE-AWARE SELECTION

In Chapter 4 we addressed the next important data exploration step in scientific visualizations—selection. We presented two spatial selection techniques, TeddySelection and CloudLasso, for particle-based data such as can be commonly found in astronomical simulations. These two intuitive methods only need the users to draw a 2D lasso around the interesting particles on a 2D surface and automatically select the intended particles in the 3D space. Moreover, our methods also identify the structure of the selected particles by constructing a selection volume according to particle density.

We did a user study to compare the structure-aware selection technique CloudLasso with the traditional selection technique CylinderSelection. During the study, completion times, errors, and selection volumes were recorded for the analysis. The results showed that CloudLasso was significantly faster than CylinderSelection. Moreover, CloudLasso created smaller and more accurate selection volumes.

6.3 INTEGRATED VISUAL ANALYTICS SYSTEM

After we designed the navigation and selection techniques in Chapter 3 and Chapter 4 respectively, we aimed to integrate them into a system for analyzing astronomical datasets on touch-sensitive displays. In Chapter 5 we introduced a visual analytics system which assists the user to find out the most important parameters from a high-dimensional dataset using the subspace clustering approach of Ferdosi et al. [2010a].

Our design of the interface integrates many interaction techniques: FI3D, a navigation technique for 3D scientific visualizations; CloudLasso, a structure-aware selection technique for selecting particle-based data; and brushing: a selection technique for selecting data from a parallel coordinates plot. These interactions allow astronomers to more easily understand the global and detailed properties of their datasets. The effectiveness of our technique was supported by positive comments

in an informal evaluation by astronomy experts who particularly noted the possibilities in showing relationships between spatial positions and other physical parameters of the data. The participants especially liked CloudLasso for selection, and the linked views which supported them to further explore the relationships between the abstract and the spatial visualizations.

6.4 FUTURE WORK

We close this chapter, and the thesis, by discussing some further possibilities for future work.

6.4.1 *Improved Interaction Techniques*

A natural extension of our work is to continue in the direction of developing other intuitive and natural interaction techniques for data exploration tasks. As we already mentioned, it is not possible to have a completely natural mapping from the 2DOF input of touch-screens to the 7DOF output necessary for 3D navigation. The natural postures that we use in real life to rotate an object in space or to bring it closer to us for inspection do not have obvious 2D counterparts. This situation presents us with two possible solutions. The first option is to find postures that, despite not being completely natural, are easy to learn and offer precise control. The second option is to move beyond the standard 2D touch-screen input to 3D input devices. Right now, we see such input devices becoming more common with the introduction of Microsoft's Kinect and Leap Motion's The Leap. Exploring the possibilities that these devices offer for navigating 3D spaces with natural postures is a very exciting prospect. Furthermore, such devices are readily available and can be easily integrated into existing systems and workflows.

In another direction we would like to consider the combination of 2D touch interaction techniques for navigation and selection in 3D spaces with stereoscopic displays. In such settings it is important to project the touch plane on the 3D stereoscopic display in order to avoid touch-through issues, see [Isenberg, 2011; Keefe and Isenberg, 2013].

We would also like to apply FI3D to different application domains. In this direction we would like to mention the work by Klein et al. [2012], who present an interaction design for fluid mechanics applications. Their design supports FI3D for navigation/zoom and integrates this with other interactions such as cutting plane interaction, drilling exploration, placement of seed particles in 3D space, and exploration of temporal data evolution.

Our spatial selection methods, TeddySelection and CloudLasso, provide a heuristics-based selection volume when the user draws a 2D lasso. One possibility here is to improve the heuristics. CloudLasso, for

example, uses a uniform density threshold for deciding the selection volume. One can consider having a density threshold that depends on the local particle density. Such modifications would offer both advantages and disadvantages. On the one hand, an appropriate selection of such density thresholds would allow for easier selection of structures, such as galaxies, where the density varies considerably throughout the structure. On the other hand, it would no longer be clear that the selection volume contains areas of similar density and therefore the selection would no longer offer direct insights into the dataset structure.

Another possibility is to consider selection techniques based not on density but on another scalar property, for example, temperature or luminosity. This would allow to more clearly see the relation between the positions of objects (stars or galaxies) and their other properties.

Furthermore, we should consider giving users the tools for manually refining the selection volume. Right now, with CloudLasso and Teddy-Selection, the user controls the initial 2D lasso and the density threshold. It is possible to add control points in the final selection surface that the user can move around in 3D space in order to change the shape of the surface. In such cases, selecting the intended control points on the surface and providing intuitive interactions for moving such point in 3D space is itself a challenging problem.

Finally, the next step is to adapt our techniques to other types of datasets from other domains. For example, we can consider selection in datasets coming from 3D surface scanners or 3D volumetric data.

6.4.2 Collaboration

One of the reasons that we have focused on implementing our techniques on large, touch-screen displays is that the latter provide a natural environment for collaboration. Collaboration is an essential part of scientific research. It makes it possible to combine ideas coming from different persons and to approach a problem from different perspectives. It is therefore crucial for scientific visualization to provide setups and methods to facilitate such collaborative work. A general discussion of the challenges presented by collaborative visualization is given by [Isenberg et al. \[2011\]](#).

Large direct-touch displays offer natural support for many users to simultaneously work on the same problem. For example, *Cambiera* [[Isenberg and Fisher, 2009](#)], a system for information foraging activities on multi-touch tabletop displays, encourages users to collaboratively analyze large text-document collections around the tabletop.

We have not explored how well suited our interaction techniques are for collaboration. A user study in this direction can reveal whether the FI3D navigation technique has to be adapted in order to provide efficient, collaborative 3D navigation through simultaneous interac-

tion or whether users are equally efficient by interacting one at a time. If simultaneous interaction is needed, then it should be possible to distinguish touch inputs coming from different users. An important challenge here is that current multi-touch displays are not able to distinguish between touches coming from different users. For example, a bi-manual gesture could be considered as two single spontaneous touch interactions. Furthermore, one should explore the possibilities for remote collaboration.

6.4.3 *Demonstration*

Another future work direction is the development of interaction techniques suitable for demonstration to a big audience using very large, touch-sensitive displays. Such a display is the curved 396" (10m or 33ft) wide touch-screen, with a resolution of 4900×1700 pixels and support for 100+ simultaneous touches, in the Donald Smits Center for Information Technology (CIT) of the University of Groningen. Our frame-based interaction technique FI3D has been adapted for such a display. In this case, because of the size of the display, it is clearly impossible to use a frame at the screen edge for interaction. Our solution that was implemented as part of a student project [Klein et al., 2012] uses two FI3D components on the surface: a large one is used for demonstration to the audience while a smaller one is used for the interaction. Then the audience can watch in the demonstration component the transformations and manipulations performed in the interaction component. This solution has not yet been evaluated in a user study and it is open whether an alternative approach would be more efficient.

6.4.4 *Visualization of Dynamical Evolution*

In this thesis we have treated astronomical datasets as static, i. e., as snapshots in time. Nevertheless, astronomical datasets often come from simulations of dynamical processes, for example, galaxy formation. This means that each dataset should not be considered in isolation but as part of a larger dataset consisting of several time snapshots. It is thus important to provide tools to visualize the time evolution of the dataset. The most basic affordance is a time-slider that moves between different snapshots and allows visualizing the evolution as an animation.

Furthermore, after having selected a part of the dataset one can visualize how the positions of the objects in the selection change in time by drawing the paths that they follow. Another possibility for visualizing the particle motion is to mark the selected particles by a different color and visually track them in different time snapshots of the simulation.

6.5 CONCLUSION

In this chapter we presented a summary of our results in this thesis. We also gave ideas and research questions for further extending our work in the future. Our list is not exhaustive but it sets forth a short-term program for research in interactions in 3D scientific visualizations. At the same time it shows that there is still a lot of work that can be done to offer scientists powerful and intuitive visualization tools to help them analyze their data.

The navigation and selection techniques for 3D interaction that we presented in this thesis, although essential, are not enough. Additional manipulation techniques (for example, cutting plane manipulation and parameter exploration) are also needed. Depending on the required manipulations it is necessary to design appropriate supporting interactions. The danger here is that integrating several different interactions into one user interface can make the latter complicated and difficult to learn. Therefore, the corresponding interactions should complement and naturally integrate with the established interactions for navigation and selection with which they should be integrated into a toolkit. At the same time, additional interactions should remain easy to learn and intuitive.

There is already research on integrated scientific exploration environments designed along these directions. Such research includes, for example, the works by [Sultanum et al. \[2010, 2011\]](#) for GeoVis, by [Fu et al. \[2010\]](#) for astrophysics, by [Klein et al. \[2012\]](#) for flow visualization, and by [Schroeder et al. \[2012\]](#) for surgical training. Nevertheless, our experience in designing and building the visual analytics system presented in [Chapter 5](#) suggests that programming such systems could become easier than it is now by using a suitable programming libraries and tools.

For example, we can imagine having different touch interactions as standard UI components of a comprehensive UI toolkit. Such touch interactions would then be connected to specific program actions. The emergence of this type of comprehensive touch UI toolkit first requires the establishment of a design language for touch user interfaces such as it already exists for desktop user interfaces. By this we mean that there is a need to establish a set of conventions for the expected meaning of each gesture and posture and how it interacts with more traditional UI widgets such as menus and buttons that are also often used in touch interfaces. Creating such a common language would mean that users do not have to learn the UI for each program separately but they could transfer their experience from one program to all others. Unfortunately, there are currently no such established conventions. On the contrary, new interaction techniques are being constantly proposed, evaluated, but only rarely catch on. This is indicative of the difficulty of creating a vocabulary of efficient, easy to learn, and intuitive touch interactions.

Having a comprehensive touch UI toolkit and, at a minimum, a touch interaction language would make it easier to explore more advanced techniques. At the same time it would make advanced visualization tools and techniques more easily accessible to scientists. Along this direction of giving scientists the ability to build their own scientific visualization and analysis tools we can imagine that we can also provide them with the possibility of designing their own gestures or postures as they see fit for their needs. Here it is natural to consider whether the intuitiveness and learnability of interactions, i.e., whether they can be called “natural”, depends on users’ specific scientific expertise or, more generally, on their cultural background.

Another problem of using gestures and postures for touch interfaces is that, unless these map naturally to real-life gestures, they are not easily discoverable. It is therefore necessary for the program to offer visual cues for what types of gestures and postures are available. Alternatively, one can imagine such touch interactions as “advanced” interaction methods that complement more traditional techniques based on a combination of buttons, menus, and real-life touch interactions.

Note that answers to these questions are not always clear and our intuition for what are the best approaches in the context of touch interfaces are often affected by our experience with desktop interfaces. For this reason we cannot stress enough the value of evaluation studies for designing useful interaction techniques.

Furthermore, it is important that research in scientific visualization stays informed by the requirements of domain experts, such as astronomers, medical researchers, and other scientists, and turns to them for validation of the proposed techniques. For this reason it is essential to establish permanent collaborations with scientific groups from different domains in order to learn their visualization needs and to satisfy them. Such collaborations will help us further guide our research and create new interaction techniques and scientific visualization tools. The final validation of these tools and techniques will be their practical adoption in scientific research.

BIBLIOGRAPHY

- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *ACM SIGMOD Record*, 27(2):94–105, June 1998. doi> 10.1145/276305.276314
- D. Akers. Wizard of Oz for Participatory Design: Inventing a Gestural Interface for 3D Selection of Neural Pathway Estimates. In *CHI Extended Abstracts*, pages 454–459, New York, 2006. ACM. doi> 10.1145/1125451.1125552
- 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*, pages 377–384, Los Alamitos, 2004. IEEE Computer Society. doi> 10.1109/VISUAL.2004.30
- F. Argelaguet and C. Andujar. Efficient 3D Pointing Selection in Cluttered Virtual Environments. *IEEE Computer Graphics and Applications*, 29:34–43, November/December 2009. doi> 10.1109/MCG.2009.117
- R. Bade, F. Ritter, and B. Preim. Usability Comparison of Mouse-Based Interaction Techniques for Predictable 3D Rotation. In *Proc. Smart Graphics*, pages 138–150, Berlin/Heidelberg, 2005. Springer Verlag. doi> 10.1007/11536482_12
- C. Baumgartner, C. Plant, K. Kailing, H. Kriegel, and P. Kröger. Subspace selection for clustering high-dimensional data. In *Proc. ICDM*, pages 11–18, Los Alamitos, 2004. IEEE Computer Society. doi> 10.1109/ICDM.2004.10112
- 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*, pages 59–64, Los Alamitos, 2005. IEEE Computer Society. doi> 10.1109/VIS.2005.40
- R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. Semantic Pointing: Improving Target Acquisition with Control-Display Ratio Adaptation. In *Proc. CHI*, pages 519–526, New York, 2004. ACM. doi> 10.1145/985692.985758
- D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley, Boston, 2005.

- S. Bruckner and M. E. Gröller. Volumeshop: An interactive system for direct volume illustration. In H. R. C. T. Silva, E. Gröller, editor, *Proceedings of IEEE Visualization 2005*, pages 671–678, October 2005. ISBN 0780394623.
- S. Bryson. Virtual Reality in Scientific Visualization. *Communications of the ACM*, 39(5):62–71, May 1996. doi> 10.1145/229459.229467
- W. Buxton. Chunking and Phrasing and the Design of Human-Computer Dialogues. In *Proc. IFIP World Computer Congress*, pages 475–480, 1986.
- S. Carpendale. Evaluating information visualizations information visualization. In *Information Visualization: Human-Centered Issues and Perspectives*, pages 19–45. Springer-Verlag, Berlin / Heidelberg, 2008. doi> 10.1007/978-3-540-70956-5_2
- W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A novel interface for interactive exploration of dti fibers. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1433–1440, November 2009. doi> 10.1109/TVCG.2009.112
- 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
- L. D. Cutler, B. Fröhlich, and P. Hanrahan. Two-Handed Direct Manipulation on the Responsive Workbench. In *Proc. SI3D*, pages 107–114, New York, 1997. ACM. doi> 10.1145/253284.253315
- O. Daae Lampe and H. Hauser. Interactive Visualization of Streaming Data with Kernel Density Estimation. In *Proc. IEEE Pacific Vis*, pages 171–178, Los Alamitos, 2011. IEEE Computer Society. ISBN 978-1-61284-935-5. doi> 10.1109/PACIFICVIS.2011.5742387
- G. de Haan, M. Koutek, and F. H. Post. IntenSelect: Using Dynamic Object Rating for Assisting 3D Object Selection. In *Proc. EGVE*, pages 201–209, Goslar, Germany, 2005. Eurographics Association. doi> 10.2312/EGVE/IPT_EGVE2005/201-209
- J.-B. de la Rivière, C. Kervégant, E. Orvain, and N. Dittlo. CubTile: A Multi-Touch Cubic Interface. In *Proc. VRST*, pages 69–72, New York, 2008. ACM. doi> 10.1145/1450579.1450593
- G. De Lucia and J. Blaizot. The Hierarchical Formation of the Brightest Cluster Galaxies. *Monthly Notices of the Royal Astronomical Society*, 375(1):2–14, February 2007. doi> 10.1111/J.1365-2966.2006.11287.X

- H. Dehmeshki and W. Stuerzlinger. Intelligent Mouse-Based Object Group Selection. In *Proc. Smart Graphics*, volume 5166 of *Lecture Notes in Computer Science*, pages 33–44. Springer-Verlag, Berlin / Heidelberg, 2008. ISBN 978-3-540-85410-4. doi> 10.1007/978-3-540-85412-8_4
- H. Dehmeshki and W. Stuerzlinger. GPSel: A Gestural Perceptual-Based Path Selection Technique. In *Proc. Smart Graphics*, volume 5531 of *Lecture Notes in Computer Science*, pages 243–252. Springer-Verlag, Berlin / Heidelberg, 2009. ISBN 978-3-642-02114-5. doi> 10.1007/978-3-642-02115-2_21
- J. Edlmann, 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
- N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, May/June 2010. doi> 10.1109/TVCG.2009.84
- 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):1141–1148, 2008. doi> 10.1109/TVCG.2008.153
- B. J. Ferdosi. *Scalable analysis and visualisation of high-dimensional, astronomical data sets*. PhD thesis, Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, The Netherlands, July 2011. 1st promotor: J. B. T. M. Roerdink, 2nd promotor: S. Trager, 3rd promotor: J. M. van der Hulst, copromotor: M. H. F. Wilkinson.
- B. J. Ferdosi, H. Buddelmeijer, S. Trager, M. H. Wilkinson, and J. B. Roerdink. Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology (IEEE VAST), October 2010*, pages 35–42, 2010a.
- B. J. Ferdosi, H. Buddelmeijer, S. Trager, M. H. Wilkinson, and J. B. Roerdink. Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators. In *Proc. VAST*, pages 35–42, Los Alamitos, 2010b. IEEE Computer Society. doi> 10.1109/VAST.2010.5652450
- B. J. Ferdosi, H. Buddelmeijer, S. C. Trager, M. H. F. Wilkinson, and J. B. T. M. Roerdink. Comparison of Density Estimation Methods

- for Astronomical Datasets. *Astronomy & Astrophysics*, 531:A114/1–16, July 2011. doi> 10.1051/0004-6361/201116878
- M. C. Ferreira de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394, July–September 2003. doi> 10.1109/TVCG.2003.1207445
- 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*, pages 367–371, New York, 2008. ACM. doi> 10.1145/1385569.1385635
- C. Forlines and C. Shen. DTLens: Multi-User Tabletop Spatial Data Exploration. In *Proc. UIST*, pages 119–122, New York, 2005. ACM. doi> 10.1145/1095034.1095055
- 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*, pages 273–276, New York, 2006. ACM. doi> 10.1145/1166253.1166296
- C. Forlines, D. Wigdor, C. Shen, and R. Balakrishnan. Direct-Touch vs. Mouse Input for Tabletop Displays. In *Proc. CHI*, pages 647–656, New York, 2007. ACM. doi> 10.1145/1240624.1240726
- M. Frisch, J. Heydekorn, and R. Dachsel. Investigating Multi-Touch and Pen Gestures for Diagram Editing on Interactive Surfaces. In *Proc. ITS*, pages 149–156, New York, 2009. ACM. doi> 10.1145/1731903.1731933
- 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/August 2000. doi> 10.1109/38.851743
- C.-W. Fu, W.-B. Goh, and J. A. Ng. Multi-Touch Techniques for Exploring Large-Scale 3D Astrophysical Simulations. In *Proc. CHI*, pages 2213–2222, New York, 2010. ACM. doi> 10.1145/1753326.1753661
- Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proc. IEEE Visualization*, pages 43–50, Los Alamitos, 1999. IEEE Computer Society. doi> 10.1109/VISUAL.1999.809866
- M. Gerl, P. Rautek, T. Isenberg, and M. E. Gröller. Semantics by analogy for illustrative volume visualization. *Computers & Graphics*, 36(2): 201–213, 2012. doi> 10.1016/j.cag.2011.10.006

- T. Grossman and R. Balakrishnan. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proc. UIST*, pages 3–12, New York, 2006. ACM. ISBN 1-59593-313-1. doi> 10.1145/1166253.1166257
- M. Hachet, F. Decle, S. Knödel, and P. Guitton. Navidget for Easy 3D Camera Positioning from 2D Inputs. In *Proc. 3DUI*, pages 83–89, Los Alamitos, 2008. IEEE Computer Society. doi> 10.1109/3DUI.2008.4476596
- J. Y. Han. Low-Cost Multi-Touch Sensing Through Frustrated Total Internal Reflection. In *Proc. UIST*, pages 115–118, New York, 2005. ACM. doi> 10.1145/1095034.1095054
- M. Hancock, S. Carpendale, and A. Cockburn. Shallow-Depth 3D Interaction: Design and Evaluation of One-, Two- and Three-Touch Techniques. In *Proc. CHI*, pages 1147–1156, New York, 2007. ACM. doi> 10.1145/1240624.1240798
- M. Hancock, T. ten Cate, and S. Carpendale. Sticky Tools: Full 6DOF Force-Based Interaction for Multi-Touch Tables. In *Proc. ITS*, pages 145–152, New York, 2009. ACM. doi> 10.1145/1731903.1731930
- M. S. Hancock, S. Carpendale, F. D. Vernier, D. Wigdor, and C. Shen. Rotation and Translation Mechanisms for Tabletop Interaction. In *Proc. Tabletop*, pages 79–88, Los Alamitos, 2006. IEEE Computer Society. doi> 10.1109/TABLETOP.2006.26
- C. Heine and G. Scheuermann. Manual clustering refinement using interaction with blobs. In *Proc. EuroVis*, pages 59–66, Goslar, Germany, 2007. Eurographics Association. doi> 10.2312/VISSYM/EUROVIS07/059-066
- E. Hornecker, P. Marshall, N. S. Dalton, and Y. Rogers. Collaboration and Interference: Awareness with Mice or Touch Input. In *Proc. CSCW*, pages 167–176, New York, 2008. ACM. doi> 10.1145/1460563.1460589
- T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. In *Proc. SIGGRAPH*, pages 409–416, New York, 1999. ACM. doi> 10.1145/311535.311602
- A. Inselberg. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer-Verlag, Berlin / Heidelberg, 2009.
- P. Isenberg and D. Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. *Computer Graphics Forum*, 28(3):1031–1038, June 2009. doi> 10.1111/1.1467-8659.2009.01444.X

- P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K.-L. Ma, and H. Hagen. Collaborative Visualization: Definition, Challenges, and Research Agenda. *Information Visualization*, 10(4):310–326, 2011. doi> 10.1177/1473871611412817
- T. Isenberg. Position Paper: Touch Interaction in Scientific Visualization. In P. Isenberg, S. Carpendale, T. Hesselmann, T. Isenberg, and B. Lee, editors, *Proceedings of the Workshop on Data Exploration for Interactive Surfaces (DEXIS 2011, November 13, 2011, Kobe, Japan)*, pages 24–27, Le Chesnay, France, 2011. DEXIS proceedings published as INRIA research report # RR-0421 in May 2012.
- T. Isenberg and M. Hancock. Gestures vs. Postures: ‘Gestural’ Touch Interaction in 3D Environments. In K. Anderson, L. Arhippainen, H. Benko, J.-B. de la Rivière, J. Häkkinen, A. Krüger, D. Keefe, M. Pakanen, and F. Steinicke, editors, *Proceedings of the CHI Workshop on “The 3rd Dimension of CHI: Touching and Designing 3D User Interfaces” (3DCHI 2012, May 5, 2012, Austin, TX, USA)*, pages 53–61, 2012.
- 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> 10.1111/1.1467-8659.2008.01233.x
- B. Jackson, D. Coffey, and D. F. Keefe. Force Brushes: Progressive Data-Driven Haptic Selection and Filtering for Multi-Variate Flow Visualizations. In *Short Paper Proc. EuroVis*, pages 7–11, Goslar, Germany, 2012. Eurographics Association. doi> 10.2312/PE/EUROVISSHORT/EUROVISSHORT2012/007-011
- R. Jianu, C. Demiralp, and D. Laidlaw. Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1449–1456, November 2009. doi> 10.1109/TVCG.2009.141
- I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
- 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*, pages 27–30, New York, 2008. ACM. doi> 10.1145/1394209.1394218
- D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Tech-note: Dynamic Dragging for Input of 3D Trajectories. In *Proc. 3DUI*, pages 51–54, Los Alamitos, 2008. IEEE Computer Society. ISBN 978-1-4244-2047-6. doi> 10.1109/3DUI.2008.4476591

- D. F. Keefe. Integrating visualization and interaction research to improve scientific workflows. *IEEE Computer Graphics and Applications*, 30:8–13, 2010. doi> 10.1109/MCG.2010.30
- D. F. Keefe and T. Isenberg. Re-imagining the Interaction Paradigm for Scientific Visualization. *IEEE Computer*, 46(5):1–6, 2013. To appear.
- D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, January–March 2002. doi> 10.1109/2945.981847
- 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*, pages 119–124, Toronto, 2009. CIPS.
- T. Klein, F. Guéniat, L. Pastur, F. Vernier, and T. Isenberg. A Design Study of Direct-Touch Interaction for Exploratory 3D Scientific Visualization. *Computer Graphics Forum*, 31(3):1225–1234, 2012. doi> 10.1111/J.1467-8659.2012.03115.X
- R. Kopper, F. Bacim, and D. A. Bowman. Rapid and Accurate 3D Selection by Progressive Refinement. In *Proc. 3DUI*, pages 67–74, Los Alamitos, 2011. IEEE Computer Society. doi> 10.1109/3DUI.2011.5759219
- R. Kosara, H. Hauser, and D. L. Gresh. An Interaction View on Information Visualization. In *Eurographics State-of-the-Art Reports*, pages 123–137, Aire-la-Ville, Switzerland, 2003. Eurographics.
- H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1/1–1/58, March 2009. doi> 10.1145/1497577.1497578
- R. Kruger, S. Carpendale, S. D. Scott, and A. Tang. Fluid Integration of Rotation and Translation. In *Proc. CHI*, pages 601–610, New York, 2005. ACM. doi> 10.1145/1054972.1055055
- W. Krüger and B. Fröhlich. The Responsive Workbench. *IEEE Computer Graphics and Applications*, 14(3):12–15, May 1994. doi> 10.1109/38.279036
- J. LeBlanc, M. O. Ward, and N. Wittels. Exploring n-dimensional databases. In *Proc. IEEE Visualization*, pages 230–237, Los Alamitos, 1990. IEEE Computer Society. doi> 10.1109/VISUAL.1990.146386
- S. Lee, J. Seo, G. J. Kim, and C.-M. Park. Evaluation of Pointing Techniques for Ray Casting Selection in Virtual Environments. In *Proc. 3rd Int. SPIE Conf. on Virtual Reality and Its Application in Industry*,

- pages 38–44, Bellingham, WA, USA, 2003. SPIE. doi> 10.1117/12.497665
- J. Liang and M. Green. JDCAD: A Highly Interactive 3D Modeling System. *Computers & Graphics*, 18(4):499–506, July/August 1994. doi> 10.1016/0097-8493(94)90062-0
- J. Liu, D. Pinelle, S. Sallam, S. Subramanian, and C. Gutwin. TNT: Improved Rotation and Translation on Digital Tables. In *Proc. Graphics Interface*, pages 25–32, Toronto, 2006. CIPS.
- W. E. Lorensen and H. E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, July 1987. doi> 10.1145/37402.37422
- J. F. Lucas and D. A. Bowman. Design and Evaluation of 3D Multiple Object Selection Techniques. Report, Virginia Polytechnic Institute and State University, USA, 2005.
- 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
- C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, 2008.
- A. Martinet, G. Casiez, and L. Grisoni. 3D Positioning Techniques for Multi-Touch Displays. In *Proc. VRST*, pages 227–228, New York, 2009a. ACM. doi> 10.1145/1643928.1643978
- A. Martinet, G. Casiez, and L. Grisoni. Design and Evaluation of 3D Positioning Techniques for Multi-touch Displays. Technical Report RR-7015, INRIA, France, 2009b.
- 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, December 1993.
- K. Mühler, C. Tietjen, F. Ritter, and B. Preim. The Medical Exploration Toolkit: An Efficient Support for Visual Computing in Surgical Planning and Training. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):133–146, January/February 2010. doi> 10.1109/TVCG.2009.58
- M. A. Nacenta, P. Baudisch, H. Benko, and A. Wilson. Separability of Spatial Manipulations in Multi-Touch Interfaces. In *Proc. Graphics Interface*, pages 175–182, Toronto, 2009. CIPS.

- M. Nijboer, M. Gerl, and T. Isenberg. Exploring Frame Gestures for Fluid Freehand Sketching. In *Proc. SBIM*, pages 57–62, Aire-la-Ville, Switzerland, 2010. Eurographics. doi> 10.2312/SBM/SBM10/057-062
- 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*, pages 236–249, Berlin/Heidelberg, 2009. Springer Verlag. doi> 10.1007/978-3-642-03658-3_31
- S. Owada, F. Nielsen, and T. Igarashi. Volume Catcher. In *Proc. I3D*, pages 111–116, New York, 2005. ACM. ISBN 1-59593-013-2. doi> 10.1145/1053427.1053445
- J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image Plane Interaction Techniques in 3D Immersive Environments. In *Proc. I3D*, pages 39–44, New York, 1997. ACM. doi> 10.1145/253284.253303
- 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
- I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. In *Proc. UIST*, pages 79–80, New York, 1996. ACM. ISBN 0-89791-798-7. doi> 10.1145/237091.237102
- J. L. Reisman, P. L. Davidson, and J. Y. Han. A Screen-Space Formulation for 2D and 3D Direct Manipulation. In *Proc. UIST*, pages 69–78, New York, 2009. ACM. doi> 10.1145/1622176.1622190
- G. Robles-De-La-Torre. The Importance of the Sense of Touch in Virtual and Real Environments. *IEEE MultiMedia*, 13(3):24–30, July–September 2006. doi> 10.1109/MMUL.2006.69
- 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, August 2009. doi> 10.1145/1531326.1531371
- 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*, pages 49–56, Goslar, Germany, 2010. Eurographics. doi> 10.2312/SBM/SBM10/049-056
- D. Schroeder, T. Kowalewski, L. White, J. Carlis, E. Santos, R. Sweet, T. S. Lendvay, T. Reihnsen, and D. F. Keefe. Visualizing Surgical Training Databases: Exploratory Visualization, Data Modeling, and Formative Feedback for Improving Skill Acquisition. *IEEE Computer Graphics and Applications*, 32(6):71–81, 2012.

- A. J. Sellen, G. P. Kurtenbach, and W. A. S. Buxton. The Prevention of Mode Errors Through Sensory Feedback. *Human Computer Interaction*, 7:141–164, June 1992. doi> 10.1207/s15327051HCIO702_1
- K. Shoemake. ARCBALL: A User Interface for Specifying Three-Dimensional Orientation Using a Mouse. In *Proc. Graphics Interface*, pages 151–156, San Francisco, 1992. Morgan Kaufmann Publishers Inc.
- Smart Technologies Inc. Digital Vision Touch Technology. White paper, February 2003.
- 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, December 2008. doi> 10.1111/1.1365-2966.2008.14066.x
- B. V. Srinivasan, Q. Hu, and R. Duraiswami. GPUML: Graphical Processors for Speeding up Kernel Machines. SIAM Data Mining 2010 Workshop on High Performance Analytics – Algorithms, Implementations, and Applications, 2010.
- M. A. Srinivasan and C. Basdogan. Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. *Computers & Graphics*, 21(4):393–404, July/August 1997. doi> 10.1016/s0097-8493(97)00030-7
- A. Steed and C. Parker. 3D Selection Strategies for Head Tracked and Non-Head Tracked Operation of Spatially Immersive Displays. In *Proc. 8th International Immersive Projection Technology Workshop*, pages 163–170, 2004.
- 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*, pages 46–49, 2008.
- N. Sultanum, E. Sharlin, M. C. Sousa, D. N. Miranda-Filho, and R. Eastick. Touching the Depths: Introducing Tabletop Interaction to Reservoir Engineering. In *ACM International Conference on Interactive Tabletops and Surfaces, ITS '10*, pages 105–108, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0399-6. doi> 10.1145/1936652.1936671
- N. Sultanum, S. Somanath, E. Sharlin, and M. C. Sousa. “Point it, Split it, Peel it, View it”: Techniques for Interactive Reservoir Visualization on Tabletops. In *Proceedings of the ACM International Conference*

- on *Interactive Tabletops and Surfaces*, ITS '11, pages 192–201, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0871-7. doi> 10.1145/2076354.2076390
- 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
- C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions—a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2591–2599, December 2011. doi> 10.1109/TVCG.2011.178
- A. Ulinski, C. Zanbaka, Z. Wartell, P. Goolkasian, and L. Hodges. Two Handed Selection Techniques for Volumetric Data. In *Proc. 3DUI*, pages 107–114, Los Alamitos, 2007. IEEE Computer Society. doi> 10.1109/3DUI.2007.340782
- 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, page 3720, 2007.
- A. Wiebel, F. M. Vos, and H.-C. Hege. Perception-Oriented Picking of Structures in Direct Volumetric Renderings. Technical Report 11–45, ZIB, Berlin, Germany, 2011.
- A. Wiebel, F. M. Vos, D. Foerster, and H.-C. Hege. WYSIWYP: What You See Is What You Pick. *IEEE Transactions on Visualization and Computer Graphics*, 18(12), December 2012. In this issue.
- M. H. F. Wilkinson and B. C. Meijer. DATAPLOT: A Graphical Display Package for Bacterial Morphometry and Fluorimetry Data. *Computer Methods and Programs Biomedicine*, 47(1):35–49, June 1995. doi> 10.1016/0169-2607(95)01628-7
- A. D. Wilson. Simulating Grasping Behavior on an Imaging Interactive Surface. In *Proc. ITS*, pages 137–144, New York, 2009. ACM. doi> 10.1145/1731903.1731929
- A. D. Wilson, S. Izadi, O. Hilliges, A. Garcia-Mendoza, and D. Kirk. Bringing Physics to the Surface. In *Proc. UIST*, pages 67–76. ACM, 2008. doi> 10.1145/1449715.1449728
- C. A. Wingrave and D. A. Bowman. Baseline Factors for Raycasting Selection. In *Proc. HCI International*. Lawrence Erlbaum Associates, Inc., 2005.
- C. A. Wingrave, R. Tintner, B. N. Walker, D. A. Bowman, and L. F. Hodges. Exploring Individual Differences in Raybased Selection: Strategies and Traits. In *Proc. IEEE VR*, pages 163–170, Los Alamitos, 2005. IEEE Computer Society. doi> 10.1109/VR.2005.1492770

- G. Wyvill, C. McPheeters, and B. Wyvill. Data Structure for Soft Objects. *The Visual Computer*, 2(4):227–234, August 1986. doi> 10.1007/BF01900346
- J. Yang, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical displays: A general framework for visualization and exploration of large multivariate data sets. *Computers & Graphics*, 27(2):265–283, April 2003. doi> 10.1016/S0097-8493(02)00283-2
- 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, November/December 2007. doi> 10.1109/TVCG.2007.70515
- L. Yu, P. Svetachov, P. Isenberg, M. H. Everts, and T. Isenberg. FI3D: Direct-Touch Interaction for the Exploration of 3D Scientific Visualization Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613–1622, November/December 2010. doi> 10.1109/TVCG.2010.157
- 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), November/December 2012. To appear.
- R. Zeleznik and A. Forsberg. UniCam—2D Gestural Camera Controls for 3D Environments. In *Proc. I3D*, pages 169–173, New York, 1999. ACM. doi> 10.1145/300523.300546
- W. Zhou, S. Correia, and D. H. Laidlaw. Haptics-Assisted 3D Lasso Drawing for Tracts-of-interest Selection in DTI Visualization. *IEEE Visualization 2008 Poster Compendium (Best Poster Nominee)*, 2008.

PUBLICATIONS

JOURNAL PAPERS

Lingyun Yu, Pjotr Svetachov, Petra Isenberg, Maarten H. Everts, and Tobias Isenberg. F13D: Direct-Touch Interaction for the Exploration of 3D Scientific Visualization Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613–1622, November/December 2010.

Lingyun Yu, Konstantinos Efstathiou, Petra Isenberg, and Tobias Isenberg. Efficient Structure-Aware Selection Techniques for 3D Point Cloud Visualizations with 2DOF Input. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1613–1622, November/December 2012.

Bilkis J. Ferdosi, Lingyun Yu, Hugo Buddelmeijer, Scott Trager, Michael H.F. Wilkinson, Tobias Isenberg, and Jos B.T.M. Roerdink. Finding and Visualizing Relevant Subspaces for Clustering High-Dimensional Data Using Connected Morphological Operators. Submitted.

WORKSHOP PAPERS AND POSTERS

Lingyun Yu and Tobias Isenberg. Exploring One- and Two-Touch Interaction for 3D Scientific Visualization Spaces. In *Mark Ashdown and Mark Hancock, eds., Posters of Interactive Tabletops and Surfaces (ITS 2009, November 23–25, 2009, Banff, Alberta, Canada)*. 2009. Extended abstract and poster.

Lingyun Yu and Tobias Isenberg. Interactive Visualization of Cosmological Simulations. In *SIREN: Scientific ICT Research Event Netherlands*, (November 5, 2009, University of Twente, The Netherlands). 2009. Poster.