

INTERACTIVE MULTIVIEW INFORMATION
VISUALIZATION IN TABLET-SIZED TOUCH
DEVICES FOR SCIENTIFIC DATA

Candemir Döger

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

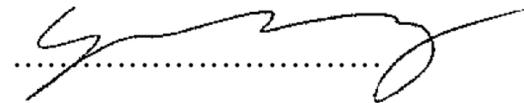
Sabanci University

January, 2014

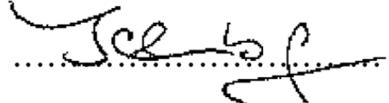
INTERACTIVE MULTIVIEW INFORMATION
VISUALIZATION IN TABLET-SIZED TOUCH DEVICES
FOR SCIENTIFIC DATA

APPROVED BY:

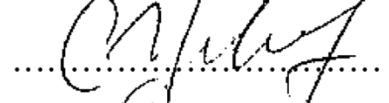
Assoc. Prof. Dr. Selim Balcısoy (Thesis Advisor)



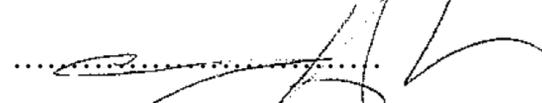
Senior Research Scientist. Dr. Tobias Isenberg (Thesis Co-Advisor)



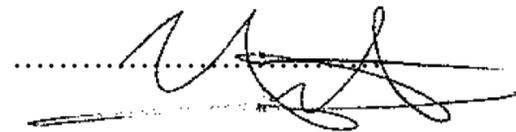
Asst. Prof. Dr. Cemal Yılmaz



Faculty Member Elif Ayter



Assoc. Prof. Dr. Yücel Saygın



DATE OF APPROVAL: 05/01/2015

© Candemir Döger 2014
All Rights Reserved

INTERACTIVE MULTIVIEW INFORMATION VISUALIZATION IN TABLET-SIZED TOUCH DEVICES FOR SCIENTIFIC DATA

Candemir Döger

Computer Science and Engineering, Master's Thesis, 2014

Thesis Advisor: Assoc. Prof. Dr. Selim Balcısoy

Thesis Co-Advisor: Senior Research Scientist Dr. Tobias Isenberg

Abstract

In this thesis we describe an exploratory visualization system for scientific data using multiple views in tablet-sized touch devices. The increasing ubiquity of mobile devices and vast amount of data generation create a need for such exploration environment. We make the data analysis available anywhere at anytime with our approach. This thesis includes an interaction framework for scientific data and a spatial selection technique based on transfer functions. We then combine the interaction and selection with multiple information visualization views to make the data exploration possible. The proposed approach is one of the first multiview exploratory visualizations in tablet devices.

TABLET BOYUTUNDAKİ DOKUNMATİK CİHAZLAR İÇİN ÇOK GÖRÜNÜMLÜ ETKİLEŞİMLİ BİLİMSEL VERİ GÖRSELLEŞTİRMESİ

Candemir Döğer

Bilgisayar Bilimleri ve Mühendisliği, Yüksek Lisans, 2014

Tez Danışmanı: Doçent Dr. Selim Balcısoy

Yardımcı Tez Danışmanı: Dr. Tobias Isenberg

Özet

Bu tez tablet büyüklüğündeki dokunmatik cihazlarda çok sayıda görünüm kullanılarak yaratılmış bir keşif görselleştirmesini konu almaktadır. Son zamanlarda mobil cihazların artması ve yüksek miktarda veri üretimi böyle bir sistem gereksinimini doğurmuştur. Burada sunduğumuz sistem veri analizinin mobil olarak yapılabilmesine olanak sağlamaktadır. Tez kapsamında; mobil cihazlarda bilimsel veriler için bir etkileşim yaklaşımı, geçiş fonksiyonlarına dayalı bir seçim tekniği ve son olarak bu ikisinin veri görselleştirmeleri üzerinde birleşimi anlatılmaktadır. Burada anlatılan yaklaşım mobil cihazlar üzerindeki çok görünümlü veri görselleştirmelerinde ilklere biridir.

Acknowledgements

I would like to thank Selim Balcisoy for his great support and guidance not only in my Master years but also in my earlier education life. I also thank him for motivating me for my graduate study.

I would like to thank Tobias Isenberg for his extensive support too. His supervision and detailed comments guided me through my research.

I have been honored to have Cemal Yilmaz, Elif Ayiter and Yucel Saygin as members of my thesis committee.

I thank all of my friends especially Yusuf Külah, Ceren Kayalar and Tolga Eren for supporting me and helping me through the way. I especially thank Selçuk Sümengen for his help with the implementation and the comments on the final application.

Finally, I would like to thank my family for their love and support.

Contents

1	Introduction	1
1.1	Motivation and Contributions	1
1.2	Outline	3
2	Related Work	4
2.1	Volume Rendering	4
2.1.1	Volume Ray Casting	7
2.1.2	Volume Rendering in Mobile Devices	8
2.1.3	Interacting with Volumetric Data	10
2.2	Multidimensional Multivariate Visualization	12
2.3	Multiview Visualization	15
2.4	Interactive Visualization on Mobile Touch Devices	17
3	Concept	21
3.1	Navigation	21
3.2	Selection	23
3.3	Multiview Visualization	26
4	Implementation	31
4.1	Platform and Dataset Selection	31
4.2	Volume Rendering on Mobile Devices	34
4.3	FI3D Direct-Touch Interaction	37
4.4	CloudLasso Selection	41
4.5	Multiview Visualization	42
5	Results and Discussion	45
6	Conclusion	48

List of Figures

1	Initial screen of our system with transfer function editor, selection tools and interaction frame.	2
2	Volume rendering examples of measurement and simulation. CT of a human head (left) (rendered using Exposure Render [41]) and rendering of computed turbulence (right) [8].	5
3	Two of the common techniques. Surface fitting (left) and direct volume rendering (right) [26]	6
4	Steps of ray casting: (1) Ray Casting, (2) Sampling, (3) Shading, (4) Compositing. [71]	7
5	A volume rendered by Lamberti's [43] technique on a PDA.	9
6	Two graphics showing the same data; relationship of actual rates of registration to predicted rates for voting. These examples shows how data-ink ratio affect the readability of graphics. Left one shows a lower data-ink ratio while right one shows a good ratio(0.7). Images are taken from Tufte's [72] book.	13
7	Visualization of 5D data of 400 automobiles with scatterplot(a) and parallel coordinates(b) [9]	14
8	Exploration of cardiac data with multiple views. 3D rendering of a heart is accompanied by a scatterplot and a histogram. Image is taken from WEAVE [20].	16
9	Early examples of information visualization research on mobile devices as shown in Chittaro's work [12]. These pictures shows the possible cluttering(a) or empty space(b) problems and a solution; adding color encoding to create a better visual mapping(c).	18
10	Kinetica's filtering approach; users create a physical barrier to filter out desired data points, the points that match the criteria are build up against the barrier.	19
11	Particle data initial view (left) vs. zoom-in view(right)	22

12	CloudLasso selection for volumetric data: a-c shows the first stage binning where near and far planes for the selection is found. Then the selection is finalized according to the desired opacity range. Different opacity ranges gives us different parts of the data while (e) shows the eye of the hurricane while (f) shows its surrounding.	25
13	Scrolling method used for handling multiple views. Horizontal scrolling(a) is used for switching between abstract and volumetric visualizations while vertical scrolling(b) is used for switching between view types.	28
14	Demonstration of brushing and linking; (a) shows the initial data set without any selection and its corresponding InfoVis views, while (b) shows the particles that are selected(brushed) and the selection is mirrored in its corresponding InfoVis view	30
15	First rendering with(a) and without(b) opacity transfer function	32
16	Rendering of vorticity field of a fluid(left) and the distribution of scalar values for this data(right). As it can be seen from the distribution, the data is mostly around zero. In addition to the poor distribution, resampling of this data to make it available for tablet device results in a poorer rendering.	33
17	Rendering of resampled hurricane data.	34
18	Multitouch interaction with transfer function editor.	36
19	Transfer function editor of ParaView	36
20	Different colormaps: HeatedObject(a), Linearized Grayscale(b), Optimal(c), Rainbow(d)	37
21	Implementation of FI3D on the tablet(a) and its original design(b) . . .	38
22	Detailed overview of FI3D interaction. Images are taken from Yu et al.'s original work [82]. Only (f) is edited to indicate our implementation difference.	40
23	The active regions in the screen where the user should start the gesture to initiate scrolling. Left one shows the horizontal scrolling area while the right one shows vertical scrolling area.	42

24	Implemented information visualization techniques; scatter plot(left) and bar chars(right)	43
25	Central data object which shares all the changes among views.	43
26	A use case scenario for our exploratory visualization system.	46

List of Tables

1	Summary of rules used while designing multiview visualizations and their impact on utility [76].	27
---	--	----

1 Introduction

1.1 Motivation and Contributions

Visualization help people to understand data better by helping their cognition with visual representations rather than just letters and numbers. Different visualization techniques are used for different data types. A car's mileage versus gas ratio can be visualized by a bar chart whereas the connections between the parts of the car can be visualized as a graph. Thus, multiple techniques are used in combination to make the exploration environment better. In addition to using multiple techniques, multiple visualization views are often used together. Multiple views help distributing the cognitive load and the data can be understood better or faster [31]. In the light of this knowledge, we decided to create a similar exploration framework for volumetric data. We used tablet devices to realize this framework. The reason for using mobile devices lies within the ubiquity of mobile devices. Mobile devices becoming increasingly widespread, new interaction and visualization techniques are being explored by researchers. Since these devices replace the PCs in some areas we believe that in the future there will be a strong need for mobile data exploration.

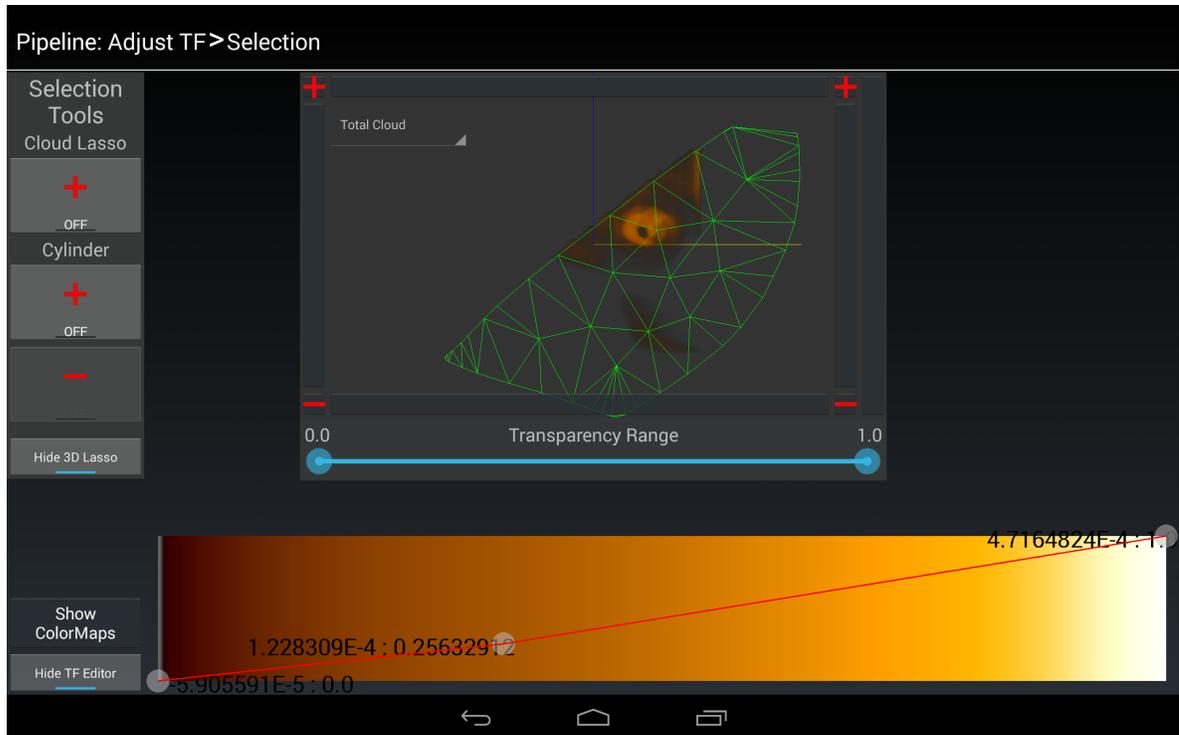


Figure 1: Initial screen of our system with transfer function editor, selection tools and interaction frame.

In this thesis we describe our exploration of possible interaction techniques with multiple information visualization views for scientific data. Figure 1 shows the initial screen of our system. This work is important because as the mobile devices become ubiquitous so the data and its analysis. We make the data analysis available anywhere at anytime by creating an exploration framework along with different selection and navigation techniques for volumetric data on tablet-sized touch devices. In summary our contributions are:

- An extension of the FI3D(Frame Interaction with 3D spaces) navigation on tablet devices,
- Application of CloudLasso spatial selection concept to selection within volumetric datasets based on transfer functions
- A multiview interaction framework which combines scientific visualization and information visualizations and that provides fluid exploration on tablet-sized touch devices,

- Realization of this framework on a tablet-sized touch input devices despite the constraints of the tablet.

1.2 Outline

This thesis proposes a data interaction framework for scientific data implemented as a mobile application, focusing on multi-view information visualization and the interaction.

First, we summarize the related work in Chapter 2. Key concepts and the current approaches associated with our work is also presented in this chapter. In Chapter 3 we describe our approach to the exploration pipeline. We discuss the interaction design for combining the analysis of SciVis data with InfoVis aspects. We explain related concepts such as brushing and linking, and how these needs to be realized both for the SciVis and the InfoVis parts. We also explain how we extended CloudLasso [81] to make it work for volume data. Then we talk about the exploration pipeline approach that guides our design. We clarify the needs like switching between views. Possible implications of using a large surface vs. a smaller tablet is also discussed here.

In Chapter 4 we explain how we realized the pipeline as a mobile application. This part includes more technical data then theoretical guidelines. We examine previously created frameworks/libraries for 3D rendering on mobile platforms. We also discuss how we implemented the volume ray casting method for our data and its limitations. We describe the usage of FI3D and the restrictions on the tablets . We also explore how to create a reliable rendering environment and development setup.

In Chapter 6 we present the results of our approach and how our pipeline could be extended for general purpose. We also present some feedback gathered from potential users.

2 Related Work

This section discusses the existing techniques and studies related to this thesis. This section is composed of three parts. In Section 2.1 we explain the common volume rendering techniques and present the brief history of volume rendering. We also include the detailed description of ray casting and a subsection for volume rendering on mobile devices. In Chapter 2.2 we talk about multidimensional multivariate visualization. Finally, in Chapter 2.4 we talk about the studies for data exploration and visualization on touch devices.

2.1 Volume Rendering

Volume visualization is used to visualize three-dimensional phenomena for the purpose of providing insights. Data for volume visualization is usually produced by measurements or simulations. An example for measurement is computed tomography(CT) which is heavily used by medical imaging. For the simulation, we can give the example of numerical simulations such as fluid dynamics or weather simulations. The measured data or the results of the simulations are represented as parallel 2D images. Then, with various techniques, these images are used to reconstruct the same 3D phenomenon. Figure 2 shows two different volume visualizations. The picture on the left in Figure 2 shows a CT scan of a human head(measurement) while the right one shows a rendering of computed turbulence(simulation).

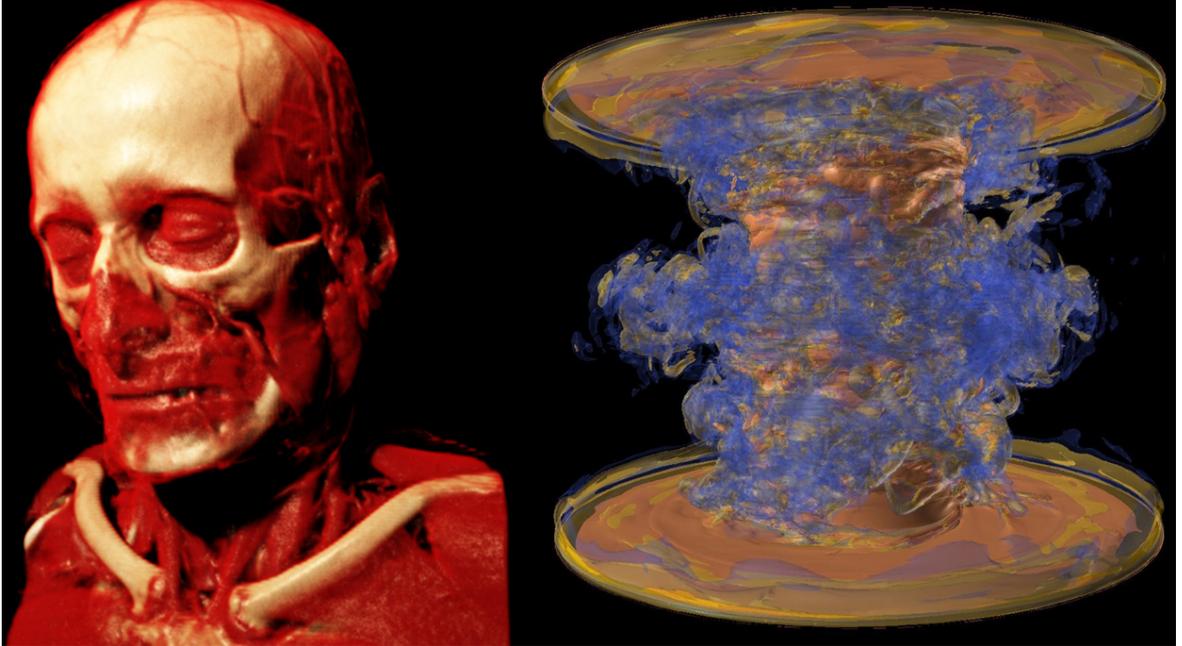


Figure 2: Volume rendering examples of measurement and simulation. CT of a human head (left) (rendered using Exposure Render [41]) and rendering of computed turbulence (right) [8].

Volume visualization is a large topic and has many branches. Thus, even survey papers on the subject [4, 16, 67] do not usually have a complete coverage of the topic but rather focus on some branches. In this thesis, we talk about some fundamental volume rendering techniques rather than a full coverage of volume visualization.

We can fit volume rendering algorithms into two categories as direct volume rendering (DVR) and surface-fitting as explained in Elvins work [16]. Figure 3 shows the same data with both of these methods. DVR methods are usually good for visualizing datasets which do not have a clearly defined shape, for example fluid data. Thus, these methods do not use any intermediate geometric primitives but rather use the elements directly. This group of methods assumes that the data is translucent and material properties are assigned according to the data values. Then, a transfer function is chosen to adjust the final image [4]. The main problem with direct volume rendering is the difficulty of choosing a good transfer function both for opacity and coloring. This process is usually done manually. However there are studies to make this process semi-automatic [35] or based on semantics [19, 56, 57].

The second category, surface-fitting, has also a broad application area. It has its own advantages and disadvantages likewise DVR. First of all, since algorithms in this group create a surface after a pass over the data it is easy to render the final image since we have a compact surface model in our hands. However, as a drawback, we might lose some important data after the surface fitting since not all the data participates in the final image. We will not consider such surface-fitting methods in this thesis. Further information about the techniques that are based on marching cubes and others can be found in Elvins' [16] survey.

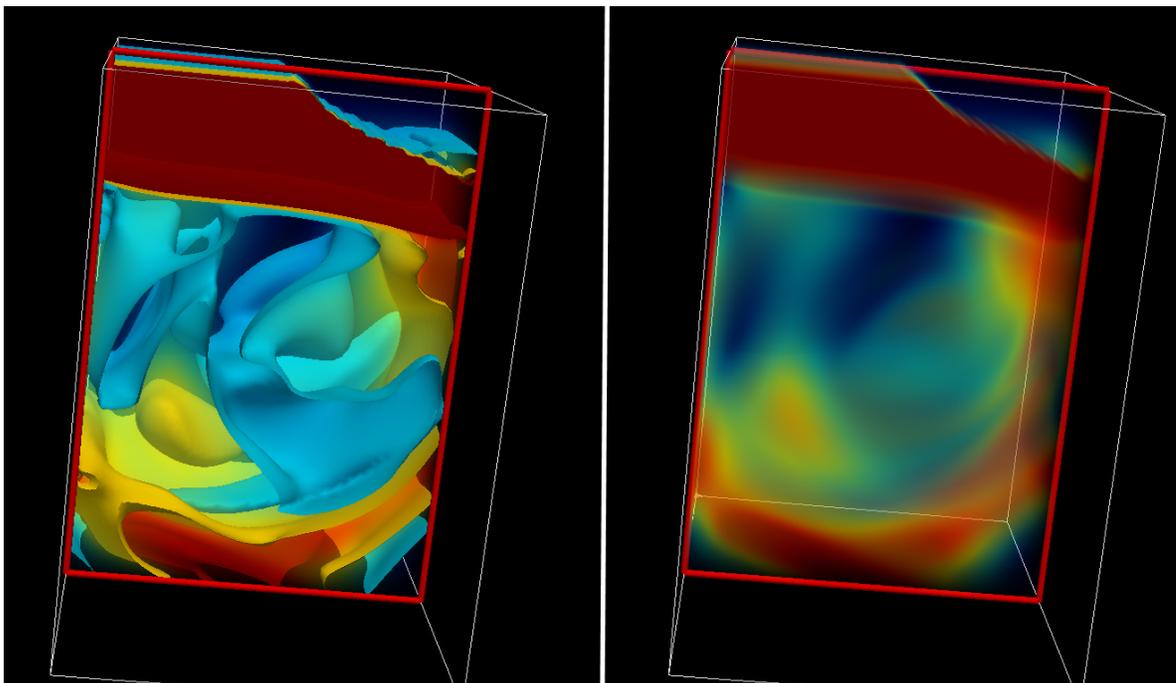


Figure 3: Two of the common techniques. Surface fitting (left) and direct volume rendering (right) [26]

We presented a general overview of volume rendering techniques. Now we can go deeper to explain some techniques in detail for DVR. We can divide the direct volume rendering techniques into two: image order approaches and object order approaches. In image order approaches, the process moves from image plane to the volume plane. In object order approaches, this process is reverse [4]. One of the widely used object order rendering techniques is splatting. Splatting was first introduced by Westover [78] in 1990. Splatting generate volumetric renderings via creating splats which are actually

projected voxels on the image plane. These projected voxels are called splats because the contribution of voxels are higher in the center of projection and lower at the corners. Projecting the voxels can be thought as throwing a snowball like explained in Elvins' [16] work. For further object-order techniques and improvements on current techniques like V-buffer, better splatting, or shear warp rendering please refer to Brodlie's [4] and Elvins' [16] surveys. Along with the object order techniques, image order techniques also have a big place. Especially with the improvements on new hardwares, the performance of image order techniques are highly improved. One of the most popular image order techniques, which is also used in this thesis, is ray casting. Ray casting is first mentioned in Levoy's paper in 1988 [46]. We explain the ray casting method in detail in Section 2.1.1.

2.1.1 Volume Ray Casting

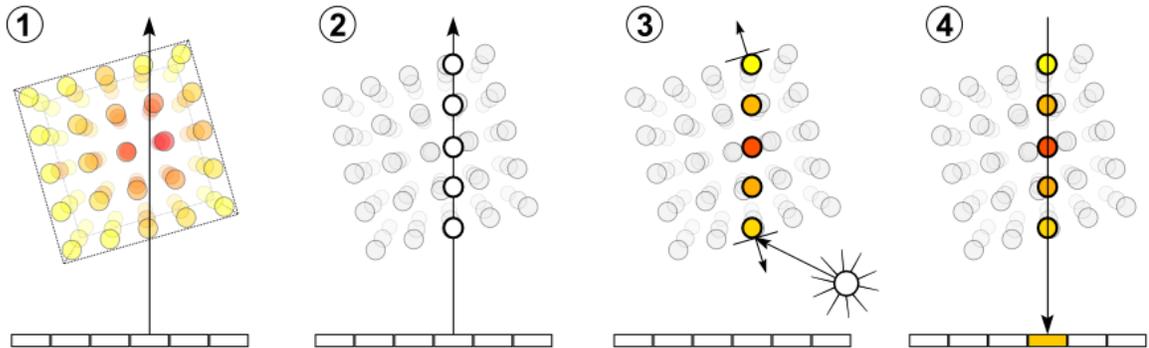


Figure 4: Steps of ray casting: (1) Ray Casting, (2) Sampling, (3) Shading, (4) Compositing. [71]

Ray casting is one of the most basic and flexible volume rendering algorithms. It is usually thought as the most direct implementation of volume rendering integral [22]. Volume rendering integral models the volume as a cloud of particles which both emit and absorb light [4]. Equation 1 shows the basic form of this integral. Here, t denotes the distance from the eye, $c(t)$ denotes emission, and $\varepsilon^{-\tau(0,t)}$ denotes the integral over the absorption coefficients. This integral gives the total amount of radiant energy C reaching the eye. For ray casting, this integral is evaluated numerically through front-

to-back compositing [22].

$$C = \int_0^{\infty} c(t) \cdot \varepsilon^{-\tau(0,t)} dt \quad (1)$$

In volume ray casting, a single ray is cast for every pixel on the image through the data. Then, at regular intervals, samples are gathered along the ray, at each sample location the value is usually calculated by tri-linear interpolation from 8 neighboring voxels. Afterwards, these samples are classified according to a transfer function and the final value is computed by front-to-back or reverse order composition of all the samples. The process is illustrated in Figure 4. There are several other papers on ray casting, focusing in different subjects like empty space skipping, caching, or faster implementation on GPUs. For detailed information, papers from Marsalek et al. [49], Kruger and Westerman [42] or Zhang et al. [85] can be read.

2.1.2 Volume Rendering in Mobile Devices

Volume visualization is a CPU/GPU-intensive process. It is thus difficult to accomplish good results in mobile devices in terms of FPS and interactivity because of their low processing power. With the emergence of mobile devices, research on volume visualization gained popularity. Even though the current devices are powerful they are not still feasible for CPU/GPU intensive operations. Thus, researchers try to find new approaches for volume visualization on mobile devices.

The early examples usually included a powerful server which was responsible for the calculations and a mobile device which is responsible for showing the pre-computed images. This client-server based approach for visualization tasks is not new, some of the early examples include Vizserver (a commercial application introduced in 2002) [52] for remote access to high-end graphics resources and Visapult (a non-commercial application) [2] for parallel volume rendering applications. In 2002, Stegmaier et al. [68] proposed a general framework for this kind of visualization architecture. One year later, a similar framework was proposed by Lamberti [44] which used a cluster-

based distributed rendering engine in the background that is responsible for feeding the mobile clients with images. A sample from this approach can be seen in Figure 5. In the following years, Zhou et al. [86] used the same client-server approach but rather than sending the images to the mobile clients they sent compressed iso-surfaces to utilize the client more. In 2007, Jeong and Kaufman presented a virtual colon navigation system [30]. They used a hybrid approach to use the processing power of the client for some parts of the rendering along with a powerful server. Even though client-server based approaches are good for rendering in high frame rates they have the disadvantage of low quality rendering and compression/decompression overheads. Since current mobile devices are highly powerful compared to old PDAs we do not borrow a client-server approach.



Figure 5: A volume rendered by Lamberti's [43] technique on a PDA.

Apart from the client-server approaches there are also studies which tried to create native rendering of volumes on mobile devices. Moser [50] created an interactive volume visualization based on 2D texture-slicing method. Their approach adapts itself by changing the resolution for better image quality or better interactivity. There is

also one similar application called ImageVis3D¹ which changes the sampling rate upon interactivity to help the user to navigate the data. There is also a framework called VES built upon VTK [65]. However, it is in an early stage of development and still a work in progress. We used VES in a previous project [47], it is very good in terms of interactivity and rendering quality with several scientific rendering techniques. However, it does not have a direct volume rendering method implemented. Thus we decided to implement our own both to see possibilities in mobile devices and have more control on our development environment.

2.1.3 Interacting with Volumetric Data

Volumetric datasets are usually difficult to examine through static images. Interactivity for the exploration of the volumetric data is essential nowadays with the emergence of high-performance hardware. Exploration occurs in many forms; segmentation, selection, transfer functions, and navigation are some examples of interactivity that we can add to volume rendering. Along with the variety of exploration techniques it is also important how these techniques are realized; different input devices or approaches might be used in combination of different exploration techniques; direct manipulation, widgets, or physical devices are some ways to apply the exploration techniques. New research continues for exploration techniques of volumetric data since we have the power to render more complex data we need more complex exploration techniques.

Transfer functions are one of the most popular exploration techniques. They map the values of data points to RGBA components to make the final image more meaningful. However, a lot of user interaction is needed to create a meaningful transfer function. There are several works to make this process more automated. Kniss et al. [37] provides a good overview of previous research on transfer functions, along with their own technique for easily creating transfer functions. Selection and segmentation are also essential for an interactive volume rendering if detailed examination of the dataset is needed. Segmentation usually needs little user attention and interaction. However,

¹ImageVis3D:<https://itunes.apple.com/us/app/imagevis3d-mobile-universal/id378071694?mt=8>

selection needs dedicated user input. Even though segmentation needs less user interaction than selection, it is still not fully automated. Different users might need different segmentations or different type of tasks might require different approaches. For example, assume that a doctor have a CT scan of kidneys and the vessels around them. It might be needed to see the vessels or kidneys only, or it might be desired to see the left kidney and the vessels of it at the same time.

Tzeng et al. [74] proposed a stroke-based volume classification model which allows users to draw directly on the 2D slices of data. This input is then used to train classifiers to classify voxels. Another stroke-based technique was presented by Yuan et al. [83] to cut out parts of volumetric data. They improved the automatic segmentation by including the direct user interaction in the process. A similar technique is also realized by Chen et al. [10] who use strokes to select regions of volumetric data by the help of seeded region growing technique. Since transfer functions also facilitate the segmentation of volumetric data, similar stroke-based approaches are also applied for transfer function designs. Ropinski et al. [60] proposed a system where strokes are drawn on the data and their algorithm performs a histogram analysis to adjust the transfer function to make the guessed segments of the data visible. Kohlmann et al. [38] presented a technique to identify interest points on 2D slices from picked points in 3D. To make the segmentation easier and more automated, Owada et al. [53] tried to keep the user input minimal and segment the desired part of the dataset automatically. The segmentation starts with a simple 2D stroke, then the depth of the selected region is automatically calculated and the selected part is shown to the user.

Selection is especially difficult in direct volume rendering techniques since there is not any precise polygons which represents the shape of the data. As the direct volume rendering techniques evolved and availability of the powerful hardware increased; different techniques for selection were developed. Wiebel et al. [79] proposed a similar technique to ray casting techniques for polygon selection. After a ray is cast in the scene, the usual compositing scheme is used to accumulate the opacity, and when a threshold is reached a possible surface is found. Another difficulty in selection is the dimension of the input. Since the input device is usually a mouse which is a 2D input

device; guiding the input in 3D space is difficult. Unfortunately, there are not much research on spatial selection for volumetric data. There are some techniques which are not directly intended but can be applied to the volumetric data. Yu et al.'s [81] CloudLasso is a good example which uses a grid in the scene to estimate the densities of the data and adjust the selection accordingly.

The related works presented in this section shows that there is a tendency toward making the interaction with volumetric data automated to make rapid exploration possible. Minimizing the human factor in tasks like selection or segmentation makes it possible to spend more time on the analysis of the data rather than exhaustive operations like selecting an area. In this thesis, we also examine automated selection methods to put more emphasis on the analysis.

2.2 Multidimensional Multivariate Visualization

Multidimensional multivariate visualization, shortly MDMV, is an old idea which was studied by other professions before computer scientists [80]. At the beginning of MDMV research, the data was not produced in vast amounts and it was relatively easy to visualize them. The problem was discovering new techniques or establishing principals for already existing ones. Tufte's [72] book is one of the best resources that explains these principals and how should the graphics/visualizations must be designed. Tufte presents important concepts like data-ink ratio. Data-ink is the non-erasable part of a graphic which represents the data and data-ink ratio is the ratio of data-ink to the total ink in the graphic. A graphic is considered good if this ratio is close to 1.0. This concept suggests that unnecessary visual objects should not be included in a good graphic. Figure 6 shows how data-ink ratio affect the representation of data.

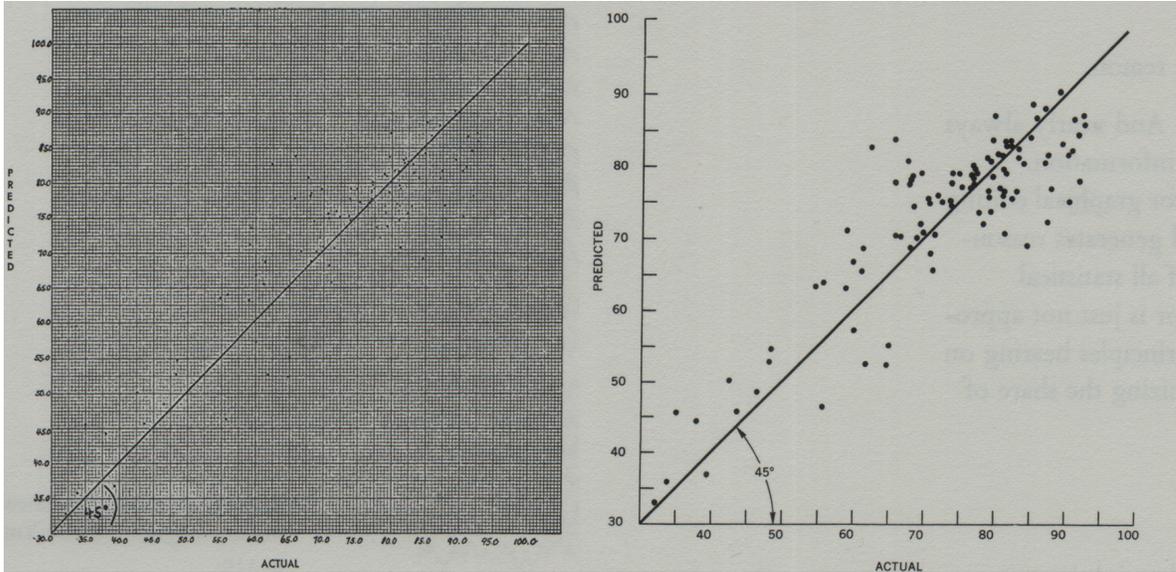


Figure 6: Two graphics showing the same data; relationship of actual rates of registration to predicted rates for voting. These examples show how data-ink ratio affects the readability of graphics. Left one shows a lower data-ink ratio while right one shows a good ratio(0.7). Images are taken from Tufte’s [72] book.

As seen in Figure 6 even relatively small number of data points are hard to represent with a poor visualization. In the modern world, a vast amount of data is generated across all domains and this data requires new skills and approaches along with principles like Tufte presented. Without going into the details of visualization principles or how big data is handled, we will give a brief history overview of MDMV visualization.

The purpose of data visualization is to transform data into visual encodings and create a perceivable image to help the human cognition to understand the data better. When the data has less than three dimensions, it is easy to use traditional line and point plots and as the dimension of the data increases other primitives get into the scene; colors, shapes, sizes are some of them [80]. However, it is usually not enough only to use one plot with multiple primitives for higher dimension data. New techniques emerge as the dimension of the data increases; multiview visualization, visualization in different axes and different 3D renderings are just some of the techniques. Since there are a lot of approaches and most of them are data-specific it is hard to categorize multivariate visualization techniques. There is one categorization which is proposed by Keim [33] which puts techniques into six categories: geometric, icon-based, pixel-

oriented, hierarchical, graph-based and hybrid techniques. We present some well-known techniques and the ones especially used in this thesis. Detailed information about the techniques can be found in the surveys by Chi [11], Keim [33], and Wong [80].

Some of the geometric projection techniques are scatterplot matrices, parallel coordinates, circular parallel coordinates, radial coordinates, hyperslice and hyperbox. Among these, there are two well-known techniques: scatter plot matrices and parallel coordinates. Scatterplots [18] map two-dimensional data to cartesian coordinates. For multidimensional data exploration, scatter plot matrices which is an extension of scatter plots is usually used. These matrices includes the 2-combination of every attribute in the data. Figure 7(a) shows an example of scatterplot matrix. They are useful for showing patterns and relations between pairs of attributes. One downside of scatterplots is that as the number of datapoints increases the readability of the scatterplot decreases. Parallel coordinates [25] can also be used to find correlations and patterns as well as functional dependencies in the data. In this technique, attributes are mapped to vertical axes which are parallel to each other. Then, for one data point, the value is marked on these axes and they are connected with straight lines. Figure 7(b) shows an example of parallel coordinates. Likewise the scatterplots when there are a lot of data points, parallel coordinates get cluttered. There is also the problem of space when the number of axes increases.

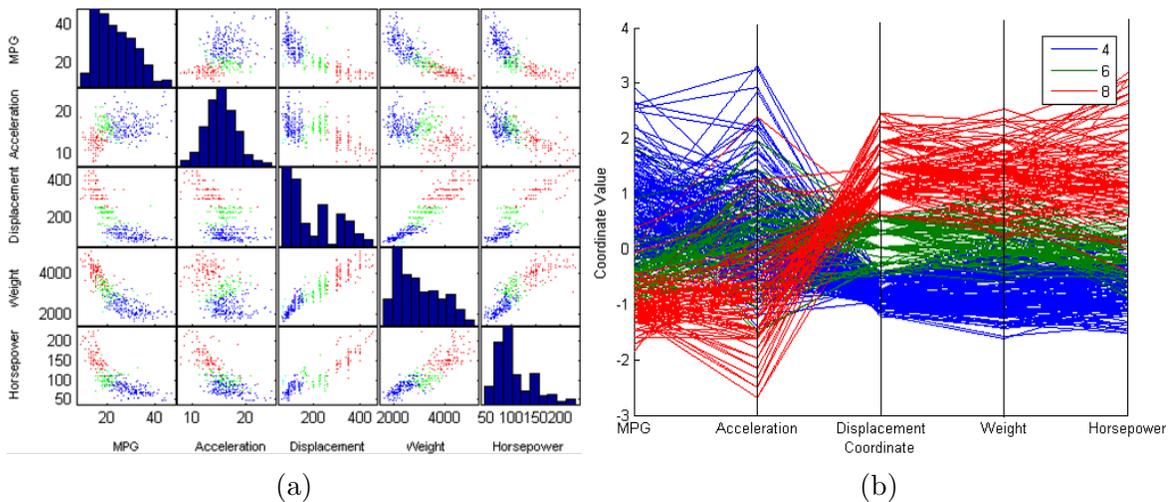


Figure 7: Visualization of 5D data of 400 automobiles with scatterplot(a) and parallel coordinates(b) [9]

Related works presented here is just a small glance into the information visualization field. New techniques and increased data generation on several domains are opening new research opportunities. In this thesis, we discuss information visualization not by oneself but as a helper for scientific visualization. Related works on the helping functionality of information visualization and how information visualization is applied in touch devices is presented in Section 2.3 and 2.4

2.3 Multiview Visualization

Multiple views help complex data to be explored easily by dividing the cognitive load. Creating multiple, easy to understand views with different aspects of the data instead of one cluttered view with a lot of variables is more effective [5]. Multiple views improve the user performance on certain tasks and helps the discovery of unforeseen relations [76]. Some applications make use of multi-view visualization in terms of abstract data with just 2D representations; XmdvTool [77] and VisDB [34] are two examples of such applications. On the other hand, there are multi-view visualization tools which use both 3D and 2D representations, these tools are especially useful for exploration of scientific data. In this thesis, we are closer to the second group.

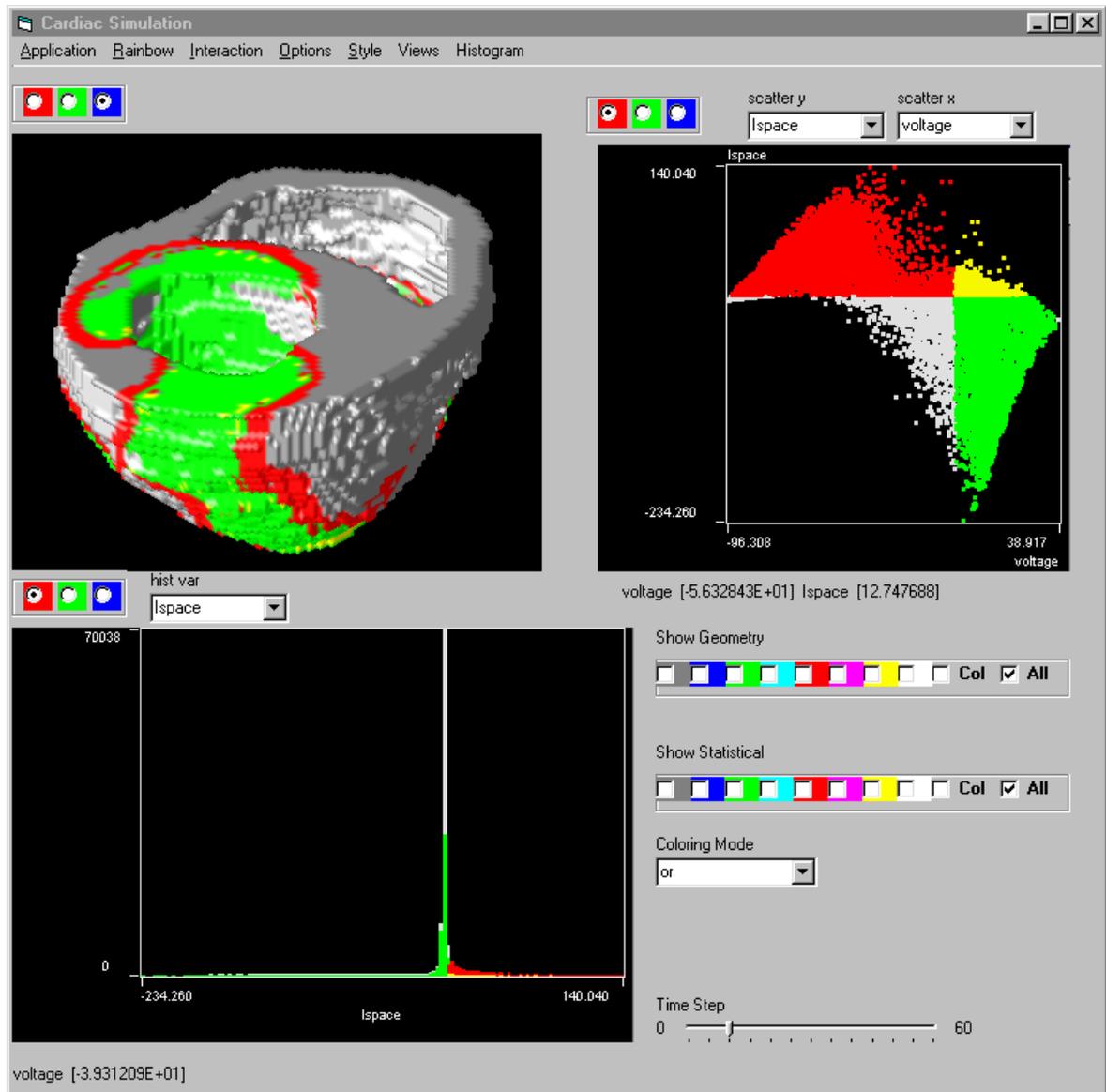


Figure 8: Exploration of cardiac data with multiple views. 3D rendering of a heart is accompanied by a scatterplot and a histogram. Image is taken from WEAVE [20].

Two earliest examples of multi-view visualization for scientific data are LinkWinds by Jacobson et al. [29] and Walts by Roberts [59]. There are also some more recent examples like WEAVE [20](shown in Figure 8), PointCloudXplore [62]. These recent examples are more data specific than their earlier counterparts. WEAVE is for cardiac simulation data while PointCloudXplore is for exploration gene expressions. The common point of all these systems that they provide an exploration environment in which brushing and linking is highly used. The results from these works suggest combin-

ing 3D scientific visualizations with 2D abstract visualizations can ease the navigation through complex data [31]. Thus, we take the multi-view visualization approach for our exploratory visualization system to make the users' job easier. Multi-view systems are also important for us since it is more useful to create multiple simple views for a small screen rather than complex visualizations.

2.4 Interactive Visualization on Mobile Touch Devices

Information visualization in mobile devices domain is a relatively new subject. The term mobile devices here both correspond to tablets and phones. There are several studies about information visualization on phones. However, most of the earlier works usually focus on limited screen space and limited hardware and how well these can be exploited in terms of information visualization.

Robbins et al. [58] divides the screen of a mobile phone to provide easy navigation on maps for the users. Chittaro et al. [12] focus on the presentation problem caused by small screen sizes and also uses maps to present their approach since they are understandable by a large audience. Figure 9 shows their approach on mobile devices. Pinheiro et al. [55] presents a method which uses treemaps and coordinates views for tourism information. These papers mostly focus on the maps thus they are not directly related to our work but they provide but good examples on how to use small screen space. There is also an earlier study of small screens with scatter plots by Waldeck and Balfanz [75]. They propose liquid browsing for overcoming the overlap issue on scatter plots. This is related to our work with the use of selection drawing(lasso) on the items. They provide selection on scatter plots which is also applicable to the scatter plots in our system to provide linking between 3D representation and InfoVis views. There are other early works on information visualization on mobile devices, especially small ones. However, they are not well related to visualization of scientific data. They are mostly for representation of abstract data on specific applications like calendars or maps. Further information can be found in works like [1, 7, 12, 17, 23, 55, 61].

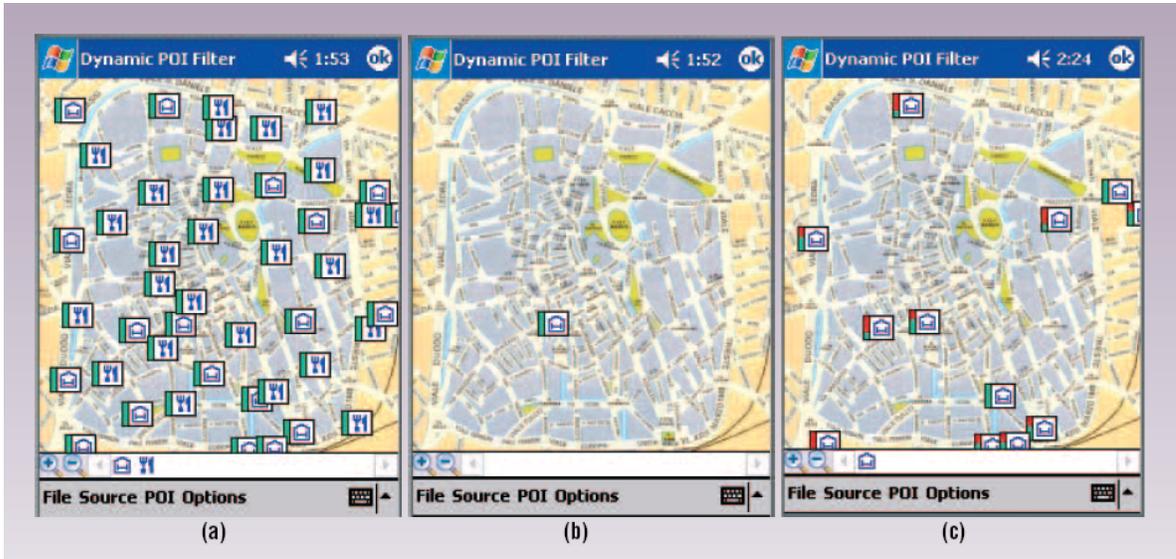


Figure 9: Early examples of information visualization research on mobile devices as shown in Chittaro’s work [12]. These pictures shows the possible cluttering(a) or empty space(b) problems and a solution; adding color encoding to create a better visual mapping(c).

Research on information visualization on mobile devices gain speed as these devices become more ubiquitous. There are several tools and applications on mobile touch devices but they are usually platform dependent. Nascimento et al. [14] presented PRISMA Mobile which is an efficient information visualization tool for Android tablets. Drucker et al. [15] made a user study on gestural interaction on data visualizations for tablet devices, which shows that gestural interaction performs better over WIMP interfaces for some tasks. These results also motivate us for our work on tablets. Burigat and Chittaro [6] also showed the effectiveness of overview+details visualization on mobile devices. One extensive work on mobile data visualization is Guttormsen’s [21] thesis which provides several components and a navigation system on tablet devices for data exploration. This work also provides a good overview on how to harness the direct interaction. Since direct interaction is not only related with mobile devices but also relatively big devices like tabletop displays. Isenberg and Isenberg [27] provides a research overview for the data visualization on interactive surfaces. In their work, interactive surfaces are grouped according to physical properties; smartphones and tablets are two of the groups which are directly related to this thesis. Their findings show that mobile

interaction and mobile visualization are underexplored research areas with small number of publications. There are also other studies which examines the touch interaction, like PanoromicData by Zraggen et al. [84]. PanoromicData provides a flexible visual language by using a whiteboard metaphor and shows how fluid exploration of data can be implemented for touch input. Finally, which we thought really shows the possibilities of data exploration on touch devices is Kinetica. Kinetica is a proof of concept application developed by Rzeszotarski and Kittur [63]. This application uses physics-based affordances which are easy to learn. Figure 10 shows an example of natural interaction on Kinetica. Their user study shows the potential of natural interaction on data as most of the users prefer physics-based tools.

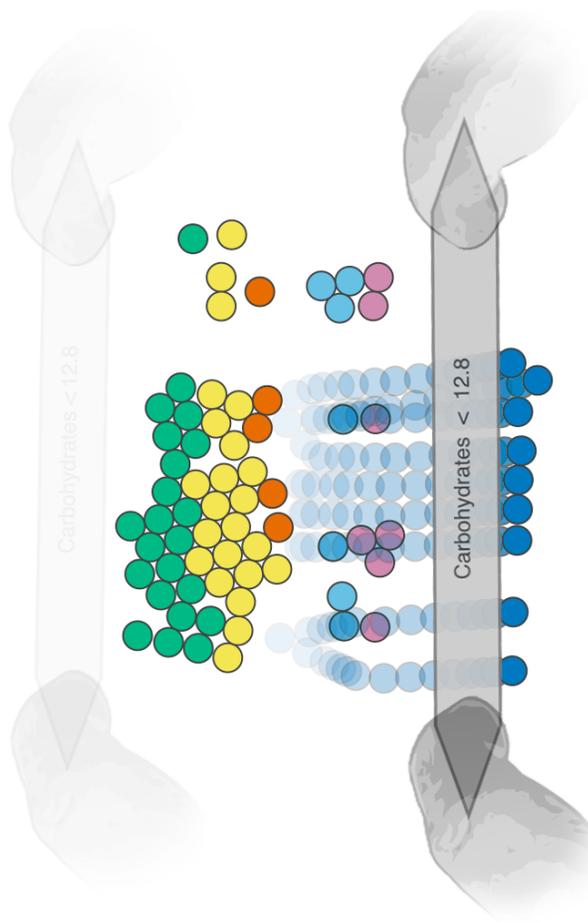


Figure 10: Kinetica’s filtering approach; users create a physical barrier to filter out desired data points, the points that match the criteria are build up against the barrier.

As shown in the related works, mobile platforms have many constraints and infor-

mation visualization on mobile devices is still an underexplored area. However despite these constraints, rising ubiquity of interactive surfaces and mobile devices promises more research subjects on the way. Mobile devices has the potential to affect how we explore data.

3 Concept

In this thesis, we explore a possible solution for exploratory visualization with linked scientific and information visualizations. As explained in Section 2.3 there are several works addressing the same problem. While addressing this question, we also aim to produce our solution for tablet devices. Because the ubiquity of mobile devices suggests that exploratory visualizations anywhere at anytime will become essential. Multi-view visualizations have issues like linking the views and coordination between visualizations. In addition to these issues, scientific visualizations have their own problems like spatial selection, segmentation or navigation. We aim to examine a model addressing these questions despite the limitations of tablet devices. We divide our concept development in 3 stages; navigation, selection and multi-view visualization.

3.1 Navigation

Navigation is essential for interactive 3D visualizations, it is needed to explore the data, to focus on some parts and to look at it from different viewpoints. Scientific data is usually complex and difficult to examine from a static point of view. For example, one could think of a cosmological simulation visualization where thousands of particles are presented. It is difficult to gain insight into the specific details of this dataset without any navigation. As shown in Figure 11, the user has interactively zoomed into the scene and the pattern in the center of the particles is revealed.

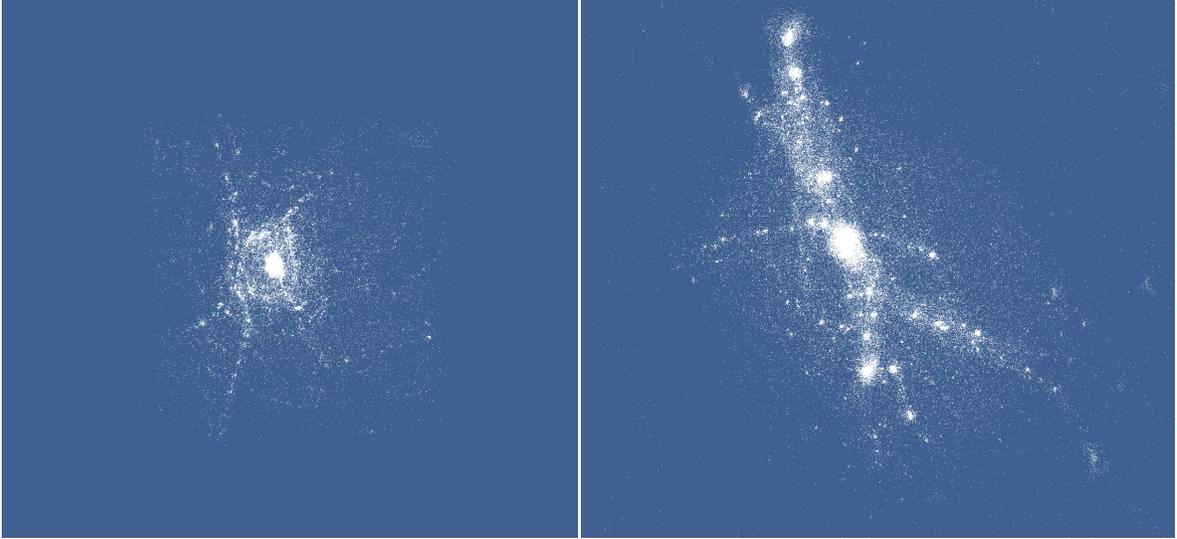


Figure 11: Particle data initial view (left) vs. zoom-in view(right)

Navigation for visualizations is a subset of broader research field; interactive visualization. There are several surveys which studies the challenges of interactive visualizations [39, 54]. Even though most exploration activities are same for 3D datasets there are special constraints for scientific data. Firstly, scientific data is in use by a highly motivated user population who need complex analysis features, and secondly scientific data is complex than most of the datasets [32]. Interactive scientific visualizations are usually more common in PCs or virtual reality environments. Even though these environments has advantages they have s like feeling of "empty space" interaction [28]. Touch interaction does not suffer from this problem since it is a direct interaction method and it is shown that somesthetic feedback is important for interaction [28].

There are several 3D interaction approaches for scientific datasets. They are usually divided into two groups according to which type of data they are addressing; general and specific. Yu et al.'s [82] FI3D and Coffey et al.'s Slice WIM [13] are examples for general interaction techniques. On the other hand, Sultanum et al.'s [69] oil deposit exploration techniques are specialized just for the domain with specific tools like peeling or tangible assisted views. Scientific visualizations have many constraints in terms of interaction. They need precise control, dedicated visualizations for different types of data and special actions beyond navigation and 3D manipulation [28]. In our work, we only deal with the navigation problem for the sake of concentration on the bigger

picture. Thus, we did not invest on developing a new technique for navigation and we followed the same goals from Yu et al.'s [82] work. These guidelines are useful for designing scientific visualization interactions. FI3D provides interaction in terms of manipulating the space rather than object which is useful for us since we use volumetric datasets for our work.

3.2 Selection

In scientific visualization researchers often want to examine a certain part of the data at some point. As mentioned above, navigation helps the researchers to realize some part of this activity. Another major helping technique is spatial selection. Since scientific visualization represents a 3D phenomena which has spatial properties it is beneficial for a scientist to examine certain spatial regions in a model. Selecting an object in the scene is important if there are too many objects in the space. In the past people tried just to interact with one object at a time. But now we usually have more complex scenes with thousands or even millions of objects in one scene.

Most of the work on volume visualization interaction usually deal with segmentation rather than selection as mentioned in Chapter 2.1.3. Selection is usually hard due to the nature of volumetric data; they do not expose specific geometric properties. In our work, we implemented segmentation by means of transfer functions. We also needed the selection for brushing the data spatially to help the users on data exploration. However, we had some challenges as we make the selection in 2D but our data is in 3D. That is also why selection needs more user interaction than segmentation, the precise selection are should be drawn, manipulated, shrunk or expanded manually. These problems makes the need for structure-aware techniques more important. Structure-aware techniques aims to minimize the human interaction during the process. There are several structure-aware, automated or half-automated techniques for segmentation as explained in Chapter 2.1.3. However, there is not any structure aware spatial selection technique for volumetric data which we are aware of. In this thesis, we propose a method for spatial selection of volumetric data based on Yu et al.'s [81] CloudLasso.

CloudLasso selection is a structure-aware spatial selection method which is based

on Marching Cubes algorithm [48]. It is mainly designed for particle data to make the separate selection of dense regions easier. We adjusted CloudLasso to work with volumetric data. The main difference from the particle selection is that we do not make any density estimation instead we tune our selection according to the voxel opacities supplied by the transfer function. Other than this adaption we have a similar implementation. There exists a first stage binning to find the near and far planes of the selection region as seen in Figure 12(a) to (d). After the first stage binning is done, we have the roughly selected shape. Then user can adjust the threshold for the opacity, deciding which voxels to include in the selection. This adjustment creates the structure aware part of our selection tool. Since the transfer function map different opacities for different scalar values we can filter out important or unnecessary parts of the data as shown in Figure 12(e) and 12(f).

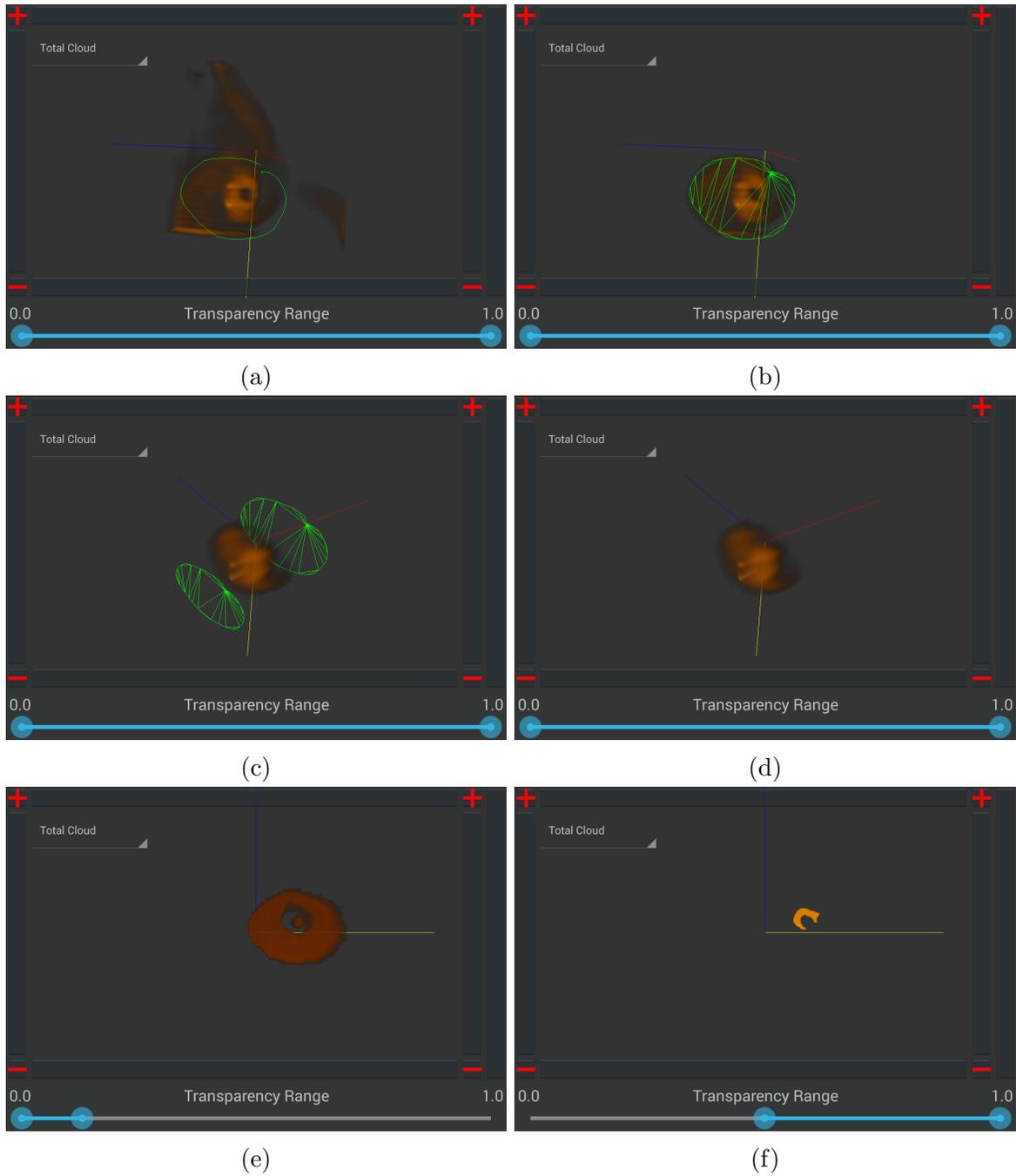


Figure 12: CloudLasso selection for volumetric data: a-c shows the first stage binning where near and far planes for the selection is found. Then the selection is finalized according to the desired opacity range. Different opacity ranges gives us different parts of the data while (e) shows the eye of the hurricane while (f) shows its surrounding.

This selection should ideally implemented with boolean operations (addition, intersection, deletion) to enhance the data exploration as stated in user studies by Yu et al. [82]. Because the users usually need to fine tune their selections if they are not

satisfactory at the first attempt. Even though we miss the boolean operations at this phase we have a tool to make a structure-aware selection which can make the selection job easier. This technique is also easy to use by touch input since drawing is an intuitive task accomplish with fingers.

3.3 Multiview Visualization

Multiview visualizations present different or integral views of the data. Multiple views in a visualization system usually brings some constraints and challenges. Table 1 gives a brief overview of these challenges which is proposed by Wang Baldonado et al. [76]. This table is like a guide for creating multi-view visualizations. Some of these challenges are also platform-dependent like space/time resource optimization. In this thesis, since we use a tablet-sized touch devices; in addition to this optimization problem we also have the challenge of touch based interaction since touch based interaction is relatively new to the exploratory visualization systems. Most of the visualization systems are still using WIMP interfaces (which are based on windows, icons, menus and a pointer) and post-WIMP interfaces are still underexplored [45]. In this thesis, we aim to explore the post-WIMP interaction possibilities with a multiview exploratory visualization.

Rule	Summary	Major Positive Impacts on Utility	Major Negative Impacts on Utility
Diversity	<i>Use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, or genres.</i>	memory	learning computational overhead display space overhead
Complementarity	<i>Use multiple views when different views bring out correlations and/or disparities.</i>	memory comparison context switching	learning computational overhead display space overhead
Decomposition	<i>Partition complex data into multiple views to create manageable chunks and to provide insight into the interaction among different dimensions.</i>	memory comparison	learning computational overhead display space overhead
Parsimony	<i>Use multiple views minimally.</i>	learning computational overhead display space overhead	memory comparison context switching
Space/Time Resource Optimization	<i>Balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views.</i>	comparison computational overhead display space overhead	
Self-Evidence	<i>Use perceptual cues to make relationships among multiple views more apparent to the user.</i>	learning comparison	computational overhead
Consistency	<i>Make the interfaces for multiple views consistent, and make the states of multiple views consistent.</i>	learning comparison	computational overhead
Attention Management	<i>Use perceptual techniques to focus the user's attention on the right view at the right time.</i>	memory context switching	computational overhead

Table 1: Summary of rules used while designing multiview visualizations and their impact on utility [76].

As mentioned above, multiview exploratory visualization systems has many challenges and since most of these systems are data-specific it is hard to define general models which can be applied to arbitrary datasets. There are some guidelines and models proposed before; further information about these models and guidelines can be found in Wang Baldonado et al.'s [76] and Boukhelifa et al.'s [3] work. These papers are explained in detail in Section 2.3. Here in the light of these previous studies we propose a use-case for multiview visualizations on tablet-sized touch devices.

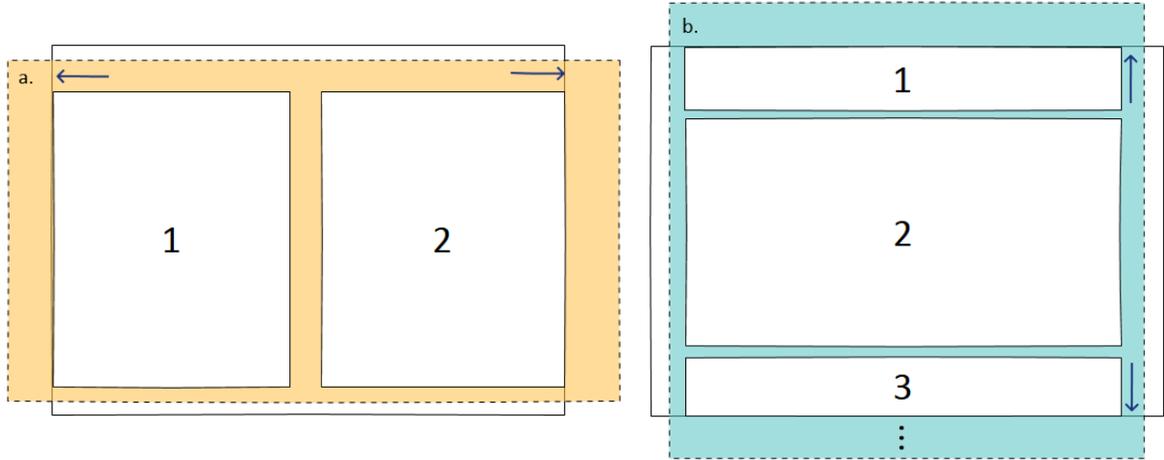


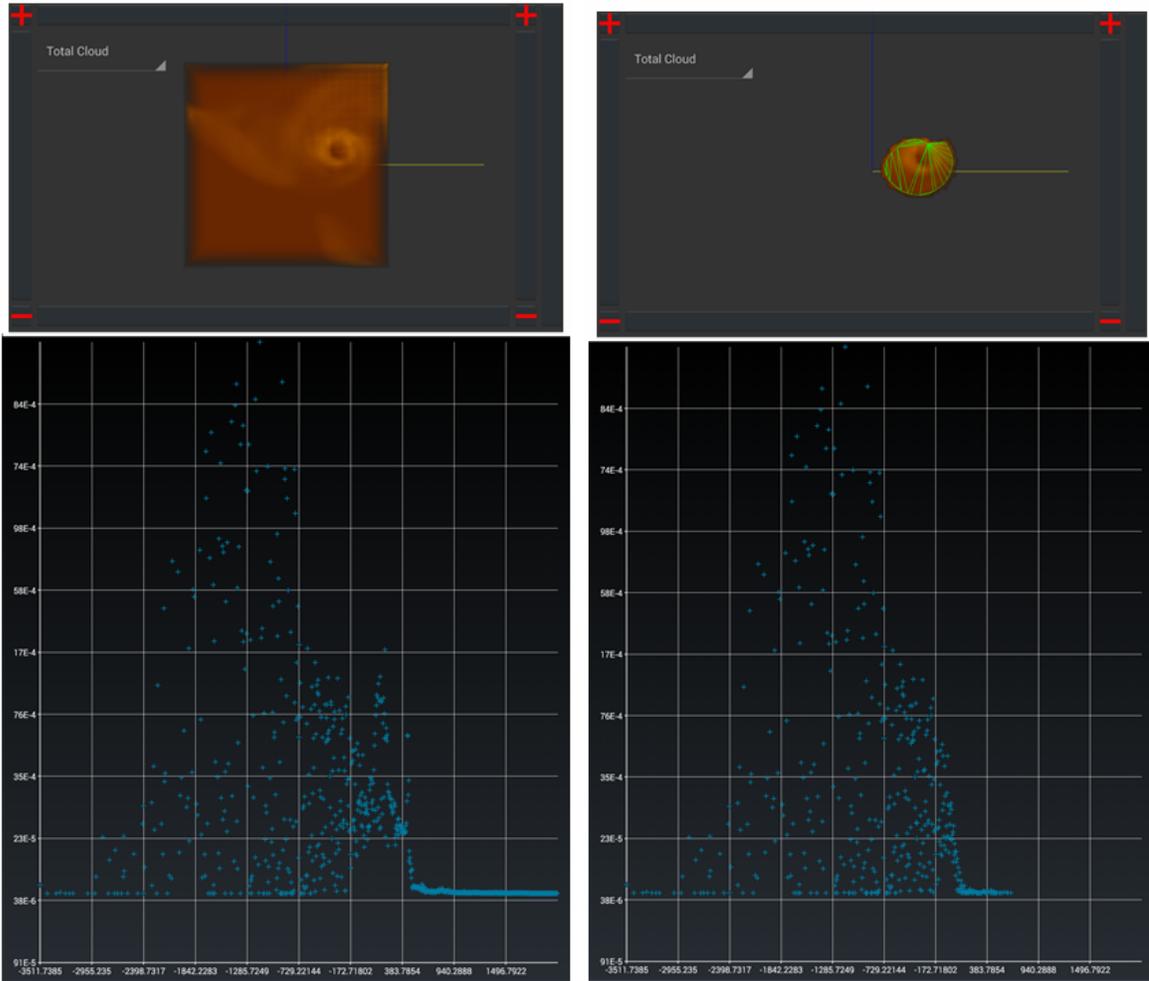
Figure 13: Scrolling method used for handling multiple views. Horizontal scrolling(a) is used for switching between abstract and volumetric visualizations while vertical scrolling(b) is used for switching between view types.

Since we have the space constraint of the tablet, we first decided on the position of the multiple views, for this we choose the scrolling views concept. This gives the user that the views are spatially near each other [76]. This is also a common interaction model among popular end-user applications like web browsers in mobile phones, for example Chrome Browser. Users can switch between different webviews by scrolling horizontally from the edge of the screen². In addition to the horizontal scrolling we also have the vertical scrolling which serves a different purpose. As shown in Figure 13, we have two views horizontally and arbitrary number of views vertically. Thus, if the user wants to switch between different representations of data, horizontal scrolling is used. Alternatively if more detailed views (like different volumetric representations) or different type of views (like a scatter plot vs. a bar chart), vertical scrolling is used. We used this multi-way scrolling method to provide flexible scrolling to the user. Flexible scrolling is needed for our exploratory visualization to provide quick insights to the users. Users can easily select a region on the data and see the selected regions distribution with just one swipe on the screen. One downside of the swiping motion is context switching, the users need to refocus to the new view after an old view is swiped. We try to minimize the refocusing time by making these views always

²Chrome Browser: <https://play.google.com/store/apps/details?id=com.android.chrome&hl=en>

work in coordination. For example, filtering in one affects all the other views. However, representational changes, like changes in transfer function, do not affect the other views.

Multiview visualizations are usually used together with brushing and linking. Brushing makes it possible to mark a specific part of a view [5] while linking help the user to understand which structures in one set of dimensions translate to which points in another [40]. We can achieve brushing by fading out all the non-selected data points from the view. Thus, selection performed in one view is mirrored in other views which creates a link between all the views, this is demonstrated in Figure 14 This way synchronization between views are achieved automatically. This is important to create a meaningful representation to the user and help them stay focused on the data. Linking can be applied unidirectional or bidirectional. When it is bidirectional, meaning that changes in the 3D representation is mirrored in the abstract visualization representations and vice versa, it should be carefully implemented since it can harm the exploration flow. Because selection in the 3D representation is a spatial selection which is directly seen by the user but the selection in InfoVis views is not the same, it is spatial in its own coordinate system which can be very different than the spatial representation of the data.



(a)

(b)

Figure 14: Demonstration of brushing and linking; (a) shows the initial data set without any selection and its corresponding InfoVis views, while (b) shows the particles that are selected (brushed) and the selection is mirrored in its corresponding InfoVis view

4 Implementation

In this chapter, we will talk about the realization of the concept we explained in Chapter 3 for a tablet-sized touch device. The realization, at least in a prototype level, is important to see the possible implications, test the possible use cases and explore the mobile development environment to see what is possible or not. We will talk about how we implemented ray casting in Section 4.2 and the problems we encountered on the road. Afterwards we explain how we adopted FI3D frame interaction for a small screen. Finally we will talk about the implementation of exploration pipeline approach with InfoVis views.

4.1 Platform and Dataset Selection

Before starting to develop our prototype we needed to decide on our development platform and dataset. We chose to use a Nexus 10 Android tablet since it was the most powerful end-user tablet at the time of the development. Also the flexibility of the Android platform helped us during the development. For the dataset, we decided on some constraints:

C1: Data should have 3D spatial attributes, namely x,y,z coordinates. Along with these spatial attributes, there should be at least 2 more variables to make the data multivariate.

C2: The data should be able to fit into the graphics memory of the tablet device.

Along with these constraints we reviewed possible dataset candidates. The first dataset we experimented with was a fluid dynamics dataset with the scalar field of Finite Time Lyapunov Exponent. (FTLE) field shows the rate of divergence of neighboring material particles [36]. We used this dataset to test our initial implementation of ray casting on the tablet device. However, since this dataset does not comply with C1 we dismissed it. The result of our first attempt can be seen in Figure 15a.

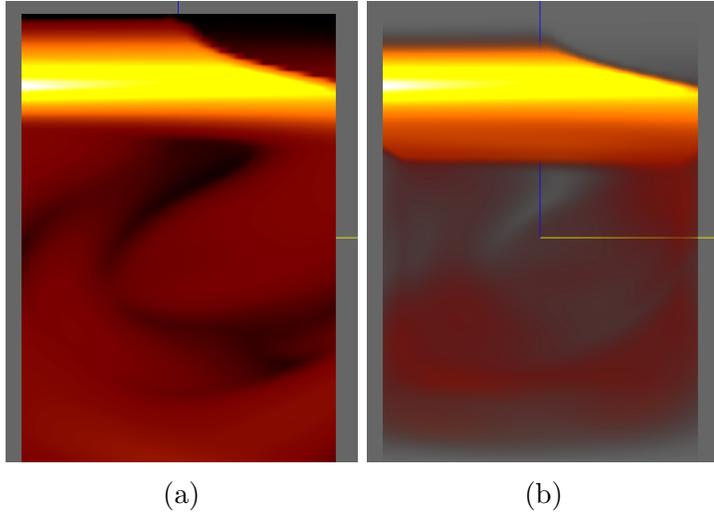


Figure 15: First rendering with(a) and without(b) opacity transfer function

For our second trial we decided to use another fluid dynamics dataset. In this data set we have 4 attributes along with spatial component so this well complies with C1. However, we had two problems with this dataset. First of all it does not comply with C2, one dimension of the data is about 60MB on average. This problem can be resolved by reducing the resolution of the data with resampling techniques. However, we also dismissed this data because it was not rich enough. Figure 16 shows a rendering and the distribution of the data.

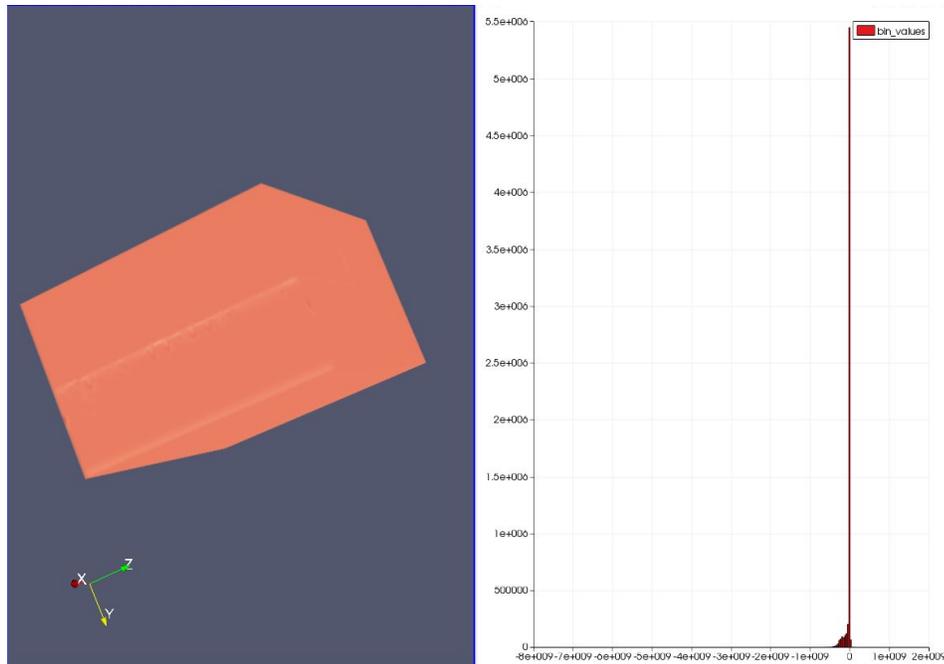


Figure 16: Rendering of vorticity field of a fluid(left) and the distribution of scalar values for this data(right). As it can be seen from the distribution, the data is mostly around zero. In addition to the poor distribution, resampling of this data to make it available for tablet device results in a poorer rendering.

Finally, after looking over several resources we decided to use IEEE Visualization 2004 Contest data which is a simulation of a hurricane from the National Center for Atmospheric Research in the United States³. The data has several scalar and vector variables. We used 9 of the scalar variables present in the dataset. This feature of the dataset complies with C1 pretty well. However, to make the data available for the tablet device and to make it compliant with C2 we needed to resample it. The data, for one variable, normally consist of 500 x 500 x 100 floating point scalars. Thus, with 9 variables this means 225 million scalar points. During the debugging and initial implementation phase we reduced the variables to 50 x 50 x 10 to be able to see the results fast. The resizing is done using a discrete cosine transform⁴. The final results can be seen in Figure 17. How this rendering is realized will be explained in detail in Chapter 4.2

³IEEE Visualization 2004 Contest:<http://sciviscontest.ieeevis.org/2004/index.html>

⁴Resize N-D arrays: <http://www.mathworks.com/matlabcentral/fileexchange/26385-resize-n-d-arrays-and-images>

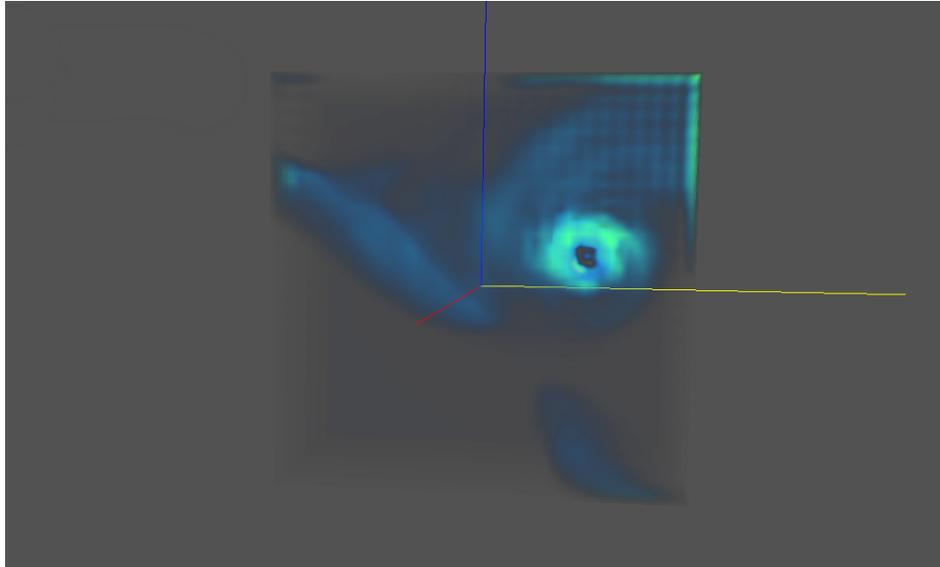


Figure 17: Rendering of resampled hurricane data.

4.2 Volume Rendering on Mobile Devices

As mentioned in Chapter 2.1.2, volume rendering on mobile devices have many constraints and it is usually realized with client-server approach to harness the power of powerful computers. In our case, we chose to implement the rendering natively on the mobile device itself because we do not aim to resolve the rendering issues but rather try to understand the possible interactions with the volume data. In addition to this, current tablets are powerful enough for interactive 3D renderings. Thus, we took a straight forward step to implement the classical ray-casting method natively in the tablet. However, it is not possible to directly port an existing ray-casting algorithm to the tablet, we did some modifications to overcome the constraints of the tablet.

Before implementing anything we did our research to see if we can use any existing frameworks or libraries to realize volume rendering on tablet devices. The first thing we found was the VTK OpenGL ES Rendering Toolkit called VES. VES is built on top of famous VTK [64]. We used VES in a previous study [47] to create a prototype of FI3D for tablet devices which will be explained in detail in Chapter 4.3. Since we found out that VES is not flexible enough and does not have the desired volume rendering techniques at that time we decided to look for other scientific visualization libraries.

However, than we only found frameworks which are not intended for scientific rendering like libGDX⁵ or volume rendering applications that do not expose their libraries/codes like CTvox⁶. After our research, we decided to implement our own volume renderer using ray casting.

As explained before volume rendering is divided into 2 main groups as object based approaches and image based approaches. Ray casting is one of the image based approaches and easy to implement with programmable pipeline of the GPU. For realizing ray casting on our tablet's GPU we adopted Movinia's [51] implementation. In this implementation, the data is loaded into textures and the shaders estimate the position of each texture in a unit cube. At this point, Android 4.0 was available for our tablet device and since they support OpenGL ES 3.0⁷ we were able to use 3D textures, previous versions of OpenGL ES do not support 3D textures. After implementing the first version of the approach we had a working prototype of raycasting without any color or opacity transfer functions which works around 30fps which is enough for interactivity.

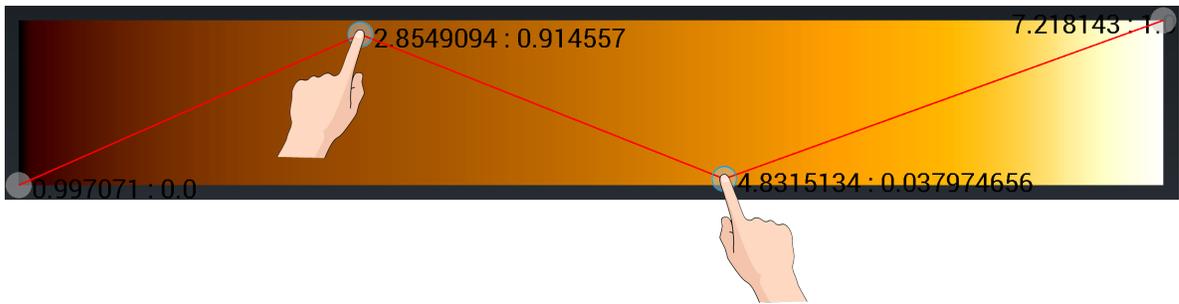
⁵libGDX: <http://libgdx.badlogicgames.com/>

⁶CTvox: https://play.google.com/store/apps/details?id=com.brucker_microct.android.ctvox

⁷OpenGL ES 3.0.4 Specification: https://www.khronos.org/registry/gles/specs/3.0/es_spec_3.0.4.pdf



(a)



(b)

Figure 18: Multitouch interaction with transfer function editor.

The next step was to create a transfer function to have a meaningful exploration environment in the tablet device. We used Paraview's [24] transfer function editor as basis. In this editor, the users chooses a color map then the function is displayed on this color map with 2 control points initially. The X axis represents the scalar points, the Y axis represents the opacity on the specific scalar point. Figure 19 shows the transfer function editor of paraview.

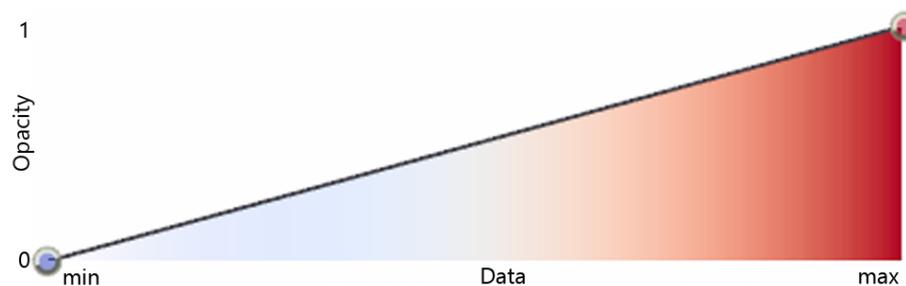


Figure 19: Transfer function editor of ParaView

After deciding on the transfer function editor we chose a set of colormaps to use in

it which can be seen in Figure 20. We chose these colormaps because they are common and suitable for most of the datasets. The main difference of our transfer function editor is the method of input. Most of the volume rendering application have a transfer function editor which is usually controlled by mouse input. However, in our scenario, we have the touch input which is a direct manipulation method. Thus, we implemented some improved features like multitouch manipulation of the transfer function which can be better seen in Figure 18. A user can add multiple control points to the editor and manipulate them at the same time. Finally, with the basic ray casting and the transfer function editor is finished. We integrated them to create the final rendering of our data. For this we manipulated the fragment shader code to fetch the necessary colors from colormap textures to map to the data. However, we found out as we change the opacity with the transfer function editor our rendering speed dropped dramatically. It dropped nearly to 6fps which not good for interaction. Thus, to solve this we decreased our resolution to half. Since Nexus 10 has a dense display with the resolution of 2560x1600 half of this resolution was enough for us in the development phase. However, this resolution might not be good enough for end users.

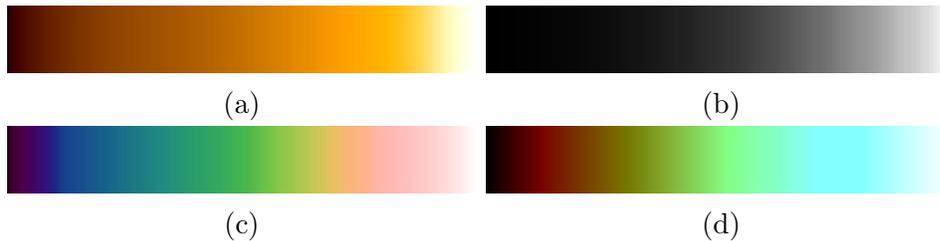
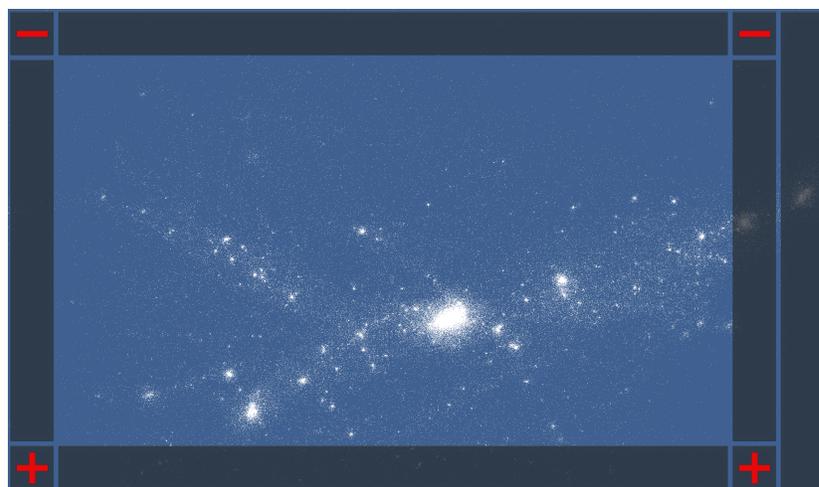


Figure 20: Different colormaps: HeatedObject(a), Linearized Grayscale(b), Optimal(c), Rainbow(d)

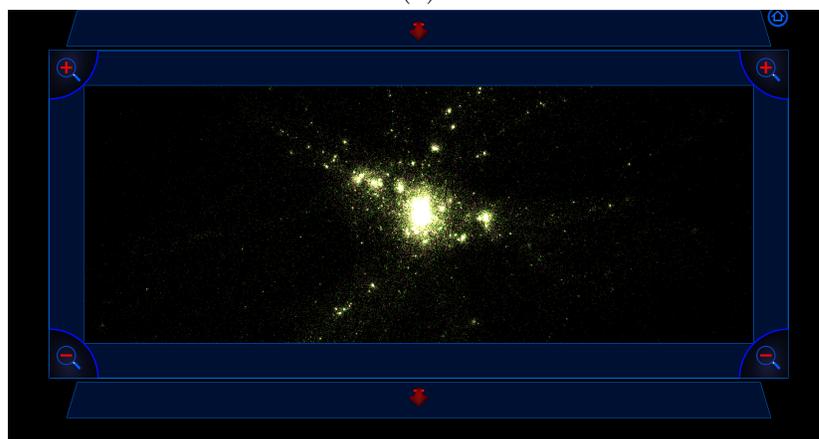
4.3 FI3D Direct-Touch Interaction

As explained in Chapter 3.1 navigation is essential for interactive 3D visualizations. We also needed a navigation technique to explore our data. We chose FI3D frame interaction from Yu et al.'s [82] work. FI3D basically provides a frame around the viewport for 7DOF navigation. It supports both large-scale and precise interaction with the data. It is intended for manipulation of the space rather than separate objects

thus it is good for volumetric data. We modified the FI3D to be able to use it on the small (relative to a big touch screen) screen of our tablet device. Especially the size of the screen put the biggest constraints for our modifications. We decreased the size of the each part of the frame and also made those parts transparent to exploit the screen area better. Originally FI3D has 2 parts (one at the top and one at the bottom) for translation in Z axis. We removed them and put a vertical piece on the right side of the screen. Figure 21 shows the difference of tablet implementation from the original design. We also used FI3D in our previous study and got positive user feedback [47] so it was another reason to use this interaction technique in this thesis.



(a)



(b)

Figure 21: Implementation of FI3D on the tablet(a) and its original design(b)

FI3D provides rotation around x-, y- and z- axis, translation parallel to the view

plane, translation in z, scaling and zooming and RST(Rotate-Scale-Translation) interaction. Translation parallel to the view plane and the translation part of RST interaction needs some precalculations before they happen. Because the interaction for these two is only sticky for a single plane located at a certain distance [82]. This distance is sometimes chosen by a interaction designer or automatically found by the center point of the dataset. In our implementation, likewise Yu et al.'s [82] implementation, this point lays at the half space interval of the visible part of the dataset. Thus, it creates the feeling of stickiness to the user during interaction.

Figure 22 gives the detailed overview of FI3D interaction. In our implementation, the parts at the top, bottom, left and right is used for initiating rotations; while perpendicular gesture on the bars initiate the trackball rotation, parallel gesture initiates rotation along the Z axis(Figure 22a). These parts also provide constrained rotations along x- and y-axes, holding one of the vertical bars and moving the finger horizontally in the viewport initiates the constrained y-rotation while holding one of the horizontal bars and moving the finger vertically in the viewport initiates the constrained x-rotation(Figure 22b-c). Moving the finger on the viewport is used for panning(Figure 22d). Square parts with plus/minus signs on them are used to zoom; movement into the center of the data initiates the action which is indicated by the sign of the bar, movement outwards the center initiates the opposite action(Figure 22e). The bar at the rightmost side is used for translation along the Z axis(Figure 22f). Different from zooming, translation along Z axis does not change the field of view of the camera but rather moves the camera together with its focal point along Z axis. Finally on the viewport itself we have the RST interaction with two finger input. Zooming can be done by pinching, translation is done by moving both the fingers and rotation is done by rotating the fingers(Figure 22g).

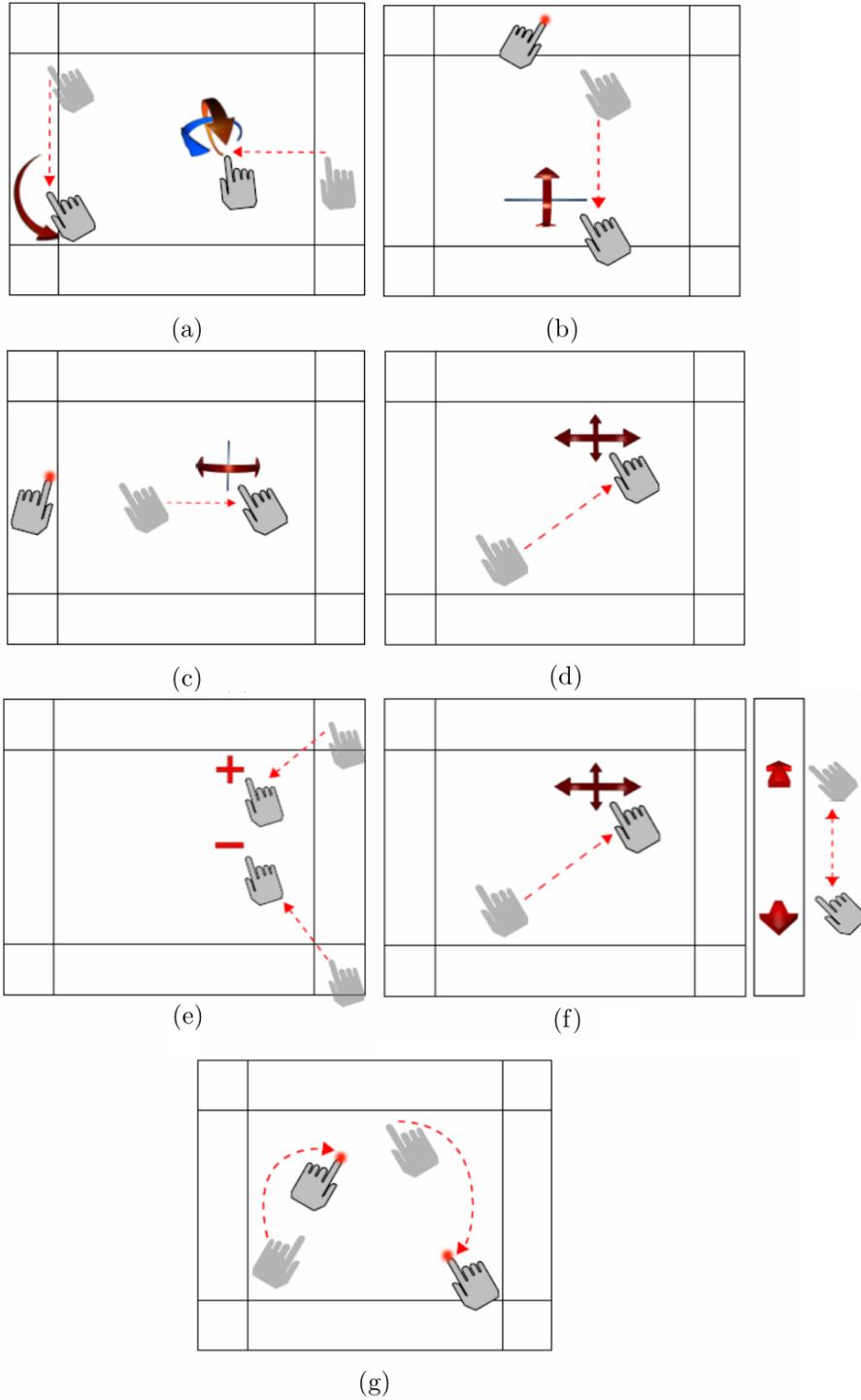


Figure 22: Detailed overview of FI3D interaction. Images are taken from Yu et al.'s original work [82]. Only (f) is edited to indicate our implementation difference.

4.4 CloudLasso Selection

The original CloudLasso selection is actually a computationally expensive technique since it uses Marching Cubes, triangulation techniques and density estimation techniques in the background. Even though we have implemented a much simpler version of cloud lasso based on transfer functions it still requires much processing time for a tablet device since these devices have less powerful hardware. That is one of the reasons we have worked with a 1000 times smaller dataset while developing our application. For the original dataset selection usually takes ≈ 10 seconds which really interrupts the user's focus. Because of this processing time we show a loading indicator while the user is waiting for the selection calculation.

First step in implementing the CloudLasso selection for volumetric data was converting the voxels to something selectable. Since we use direct volume rendering in our application the data we present does not expose a solid geometrical shape. To solve this issue we attached invisible points to the center of every voxel. Thus selecting the voxels actually mean selecting the points in 3D space, this may cause some problems like jagged edges however for the purpose of our prototype this precision is good enough. When the user draws a lasso on the viewport it is first triangulated using delaunay triangulation to be able to define it with triangles. Then the triangulated polygon is projected along the viewing direction with the current MVP(model-view-projection) matrix. In the original CloudLasso, the Z planes where the near and far planes is placed are decided by separating the extruded cylinder to bins and checking the number of particles inside each bin. We do this by selecting the nearest and furthest voxel for volumes inside the extruded cylinder. After that, the points that are not between the near and far selection planes are discarded. Finally, all the 3D points are projected onto the near selection plane a point-in-polygon algorithm, which is ray-casting in our case [70], is used to determine which points are inside our drawn lasso. We use a bitmap to store which points are selected by switching 1 for selected points and 0 for non-selected ones. During the render the selection bitmap is checked in the fragment shader and necessary voxels are discarded during ray casting.

4.5 Multiview Visualization

After designing our concept as explained in Section 3.3 we examined the implementation possibilities. Since mobile platforms are more recent than PCs there is not a high availability of libraries and frameworks. This especially true for rendering as explained in Section 4.2. Hopefully we had more chances in terms of creating a multiple view flow with InfoVis views. We chose an open-source library to create our InfoVis views; snowdon. Snowdon is a graphing library which has basic visualizations implemented⁸. Then, for the multiple view flow we used Android’s internal paging API which is widely used for applications. We decided regions to initiate the scrolling as shown in Figure 23. This way user’s exploration is not interrupted during other interactions.

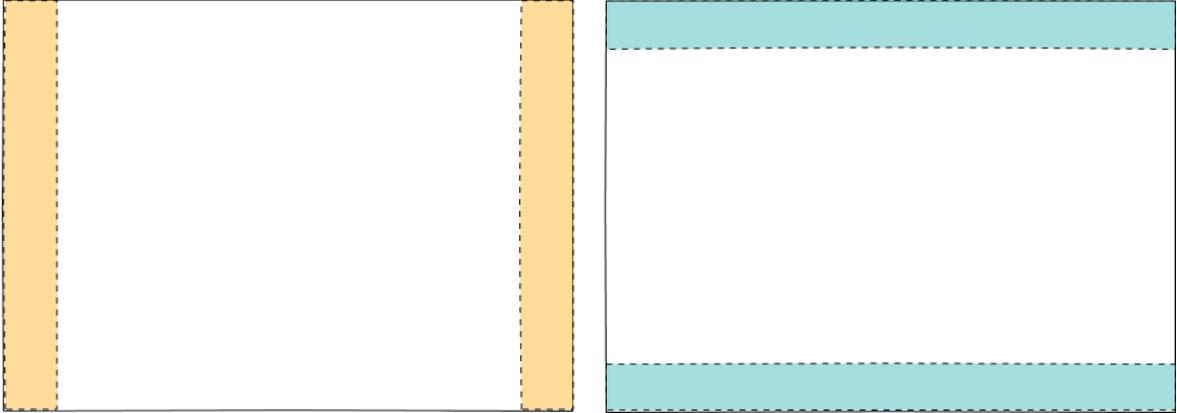


Figure 23: The active regions in the screen where the user should start the gesture to initiate scrolling. Left one shows the horizontal scrolling area while the right one shows vertical scrolling area.

Afterwards, we chose our InfoVis views among the available ones from snowdon library; scatter plot and bar chart. Although bar chart is not really suitable for representing high number of data points at time of implementation it was more feasible to choose among the available visualizations rather than implementing our own. Chosen visualization can be seen in Figure 24 Because we wanted to focus on the exploration pipeline rather than implementing InfoVis views in the tablet. We also implemented a histogram to see the data distribution for the selected parts.

⁸snowdon:<https://code.google.com/p/snowdon/>

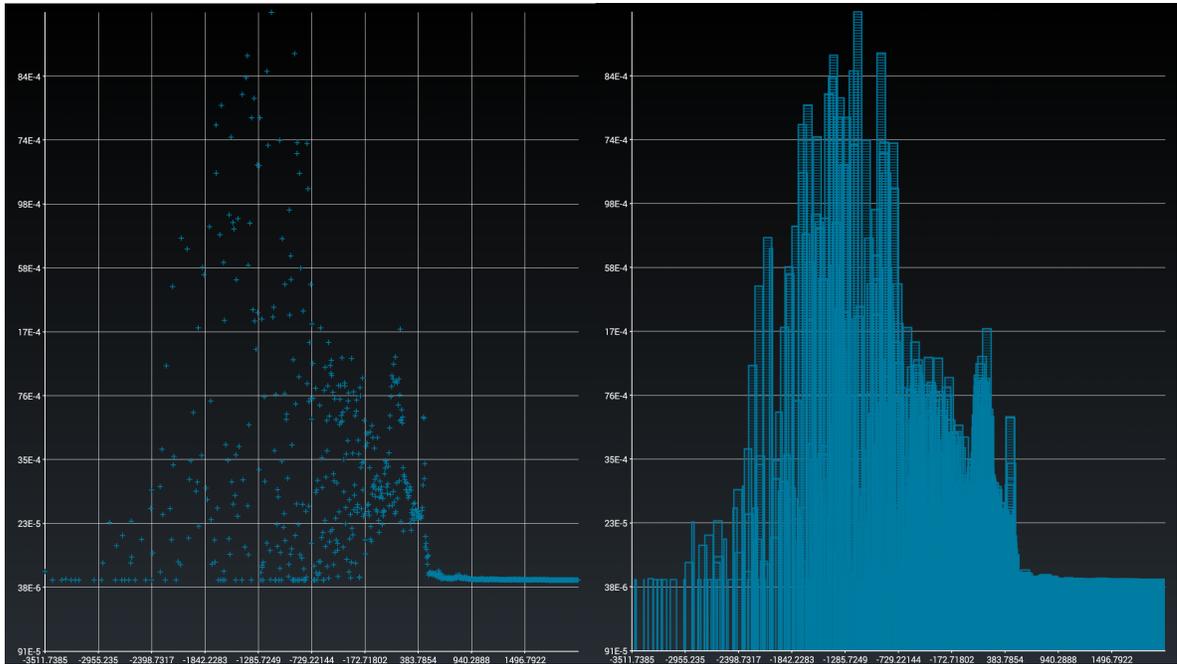


Figure 24: Implemented information visualization techniques; scatter plot(left) and bar chars(right)

Then we implemented linking among our views with CloudLasso selection. A selection in the 3D views is reflected among all the other views. We used a central data object to share the changes across all the views. The role of this object can be seen in Figure 25.

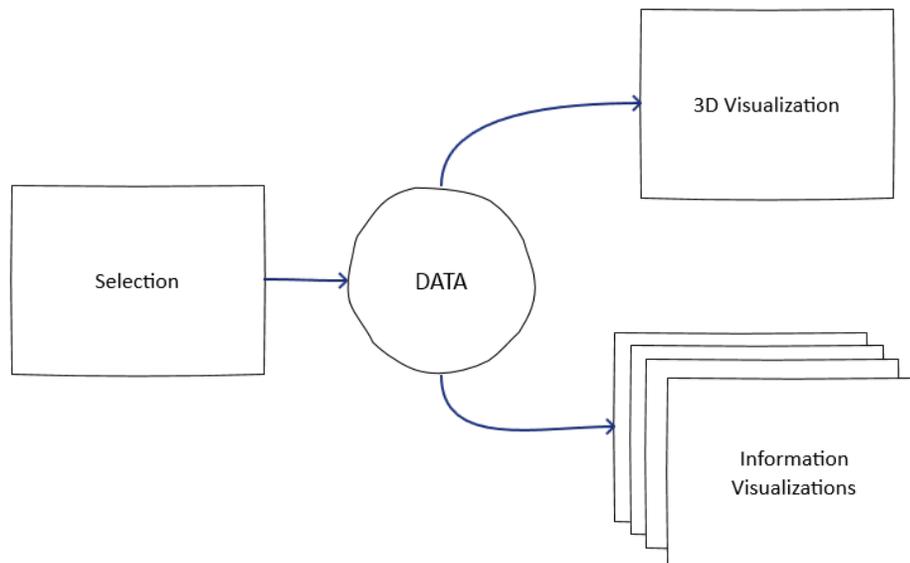


Figure 25: Central data object which shares all the changes among views.

Although 2D renderings are not as slow as volumetric rendering they still have some performance issues. When the user selects a region from the volumetric representation it usually takes 3-5 seconds render its InfoVis counterpart. This might be caused by the underlying library we used as well as the limitation of hardware. We will implement our own InfoVis views in the future to have better control over rendering performance and other parameters such as coloring or sizing of the charts.

5 Results and Discussion

In this thesis, we proposed a concept for scientific data exploration with multiple information visualization views on tablet-sized touch devices. Our system is proposed as an exploratory visualization system rather than a confirmatory one [73]. Thus, we did not look for an answer to a specific question while exploring our data but we provide a tool to the domain experts to uncover relations, correlations or features in the data. Our system is based on direct-touch interaction and open for multi-touch gestures. We believe that creating a touch-based system for exploratory visualization is useful. There are also studies which show that touch-based visualization systems are more preferable over WIMP interfaces for certain tasks [15].

The resulting system can be demonstrated by a use case scenario; a meteorology expert exploring a hurricane simulation. The steps in this scenario is as follows:

1. User chooses the data file data is loaded into the memory,
2. user fine-tunes the representation by transfer function editing,
3. an area of interest is selected by the user via CloudLasso,
4. user begins the exploration via help of InfoVis views,
5. user either goes back to 3 for new views or finishes the exploration.

The use case explained above can be seen in Figure 26

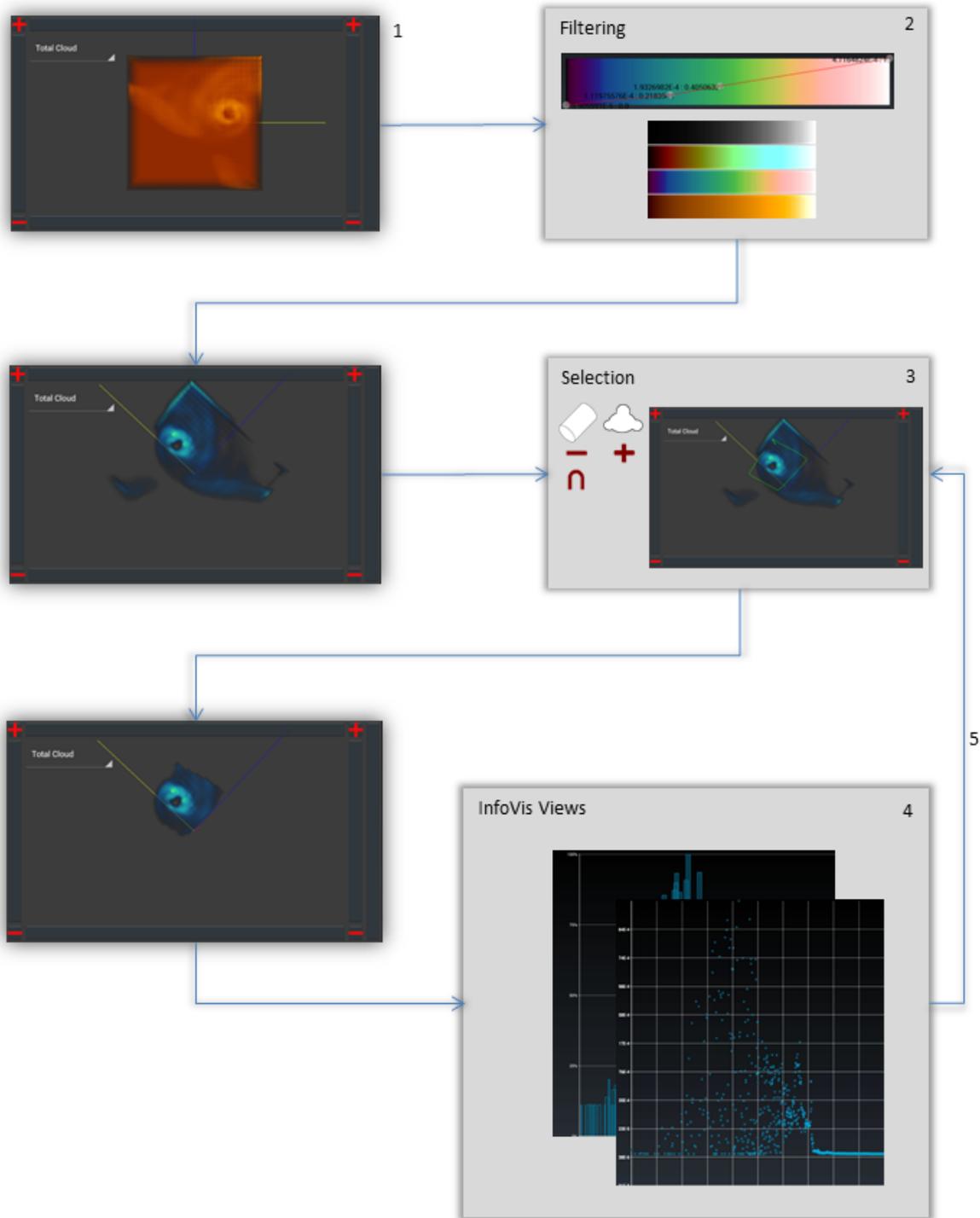


Figure 26: A use case scenario for our exploratory visualization system.

We gathered feedback from an expert by utilizing the use case scenario above to validate the functionality of our application. The feedback we gathered was from a

computer graphics expert who has prior experience with volumetric data. He found the application highly useful by stating that it can be very practical to use it in presentation or collaboration environments. In addition to this, he said that the FI3D navigation framework is easy to use and it is hard to get lost in the data but the movements lack acceleration so it is a little slow to use it. It is also stated that, gestural interaction would be nice with the transfer function editor like switching between curves and linear functions by two finger rotation or changing the color distribution of the color maps by pinching. Since our system is not a final product but a prototype some functionalities are missing. One of the missing functionalities is the indication of the selection area by a surface. The user stated that it is hard to modify the selection after it is done since all the unselected area is missing. In terms of selection together with the transfer functions, it is told that it would be good to have a grouping system where the user can create groups for selection and modify them separately with different transfer functions, this is also stated for InfoVis views. One final comment was that, selection in the InfoVis views and linking that selection to the volumetric part would be nice to have.

Even though the feedback we got was mostly positive the system should be formally evaluated possibly with a comparison of existing SciVis exploratory visualization system. Along with the positive comments and the novelty of our application we have significant problems to address. We aim to address these problems in our future work as explained in Chapter 6. Some significant ones of these problems are:

1. Even though touch interaction is intuitive there is still a learning curve for the application,
2. there are problems specific to the small screens like fat finger problem [66],
3. although current mobile devices are powerful they are not still as capable as PCs, so there exists rendering performance problems.

6 Conclusion

The vast generation of data across all domains and increased usage of mobile devices request data exploration anywhere at anytime. We worked in the scientific data exploration area to show the possibility of such tool on mobile devices. The exploratory visualization tool proposed in this thesis is a first step into the fertile field of scientific data exploration on mobile devices. We believe this is important because already at this time and to an even larger degree in the future, scientists will be carrying mobile devices which they want to use to show their data and analyses to colleagues, explore and analyze their data during travel time, and use mobile devices in the context of larger data exploration scenarios such as VR scenarios.

During our research we tackled some fundamental problems such as; view coordination, spatial selection and data segmentation as well as some additional ones; volume rendering in mobile devices, touch interaction in SciVis and multiple views in small screens. Even though we designed our multi-view visualization system with a high-level approach in our minds which emphasizes platform and device independency it is not possible to ignore the driving force of constraints in mobile devices. Together with constraints of scientific visualization and multi-view visualizations we designed a system with the following properties:

- Basic volume rendering engine in Android platform,
- touch-based transfer function editor,
- spatial selection technique for volumetric datasets,
- multi-view exploratory visualization for scientific data, specifically designed for touch-based control.

Our approach is different from the previous ones as it is specifically designed for tablet-sized touch devices. Since it is one of the first multi-view exploratory visualization tools in a mobile platform it has a lot of room for improvements. In the future, we plan to work on better brushing and linking between views, more robust spatial

selection with boolean operations and more variety in terms of InfoVis views. Then, our system needs to be formally evaluated with a user study to prove/disprove its effectiveness. Even though our system is data specific it should be extended for any forms of scientific data. We can also think our application in a larger context where, in the future, it can be used in CAVE environments or together with stereoscopic visualizations or maybe even with tangible controls. Since non-desktop platforms are founding their place for data visualization applications as the time goes on, we believe the area of exploratory visualization on mobile platforms will gain more importance in the future.

References

- [1] Benjamin B. Bederson, Aaron Clamage, Mary P. Czerwinski, and George G. Robertson. Datelens: A fisheye calendar interface for pdas. *ACM Trans. Comput.-Hum. Interact.*, 11(1):90–119, March 2004.
- [2] W. Bethel. Visualization dot com. *Computer Graphics and Applications, IEEE*, 20(3):17–20, May 2000.
- [3] N. Boukhelifa, J.C. Roberts, and P.J. Rodgers. A coordination model for exploratory multiview visualization. In *Coordinated and Multiple Views in Exploratory Visualization, 2003. Proceedings. International Conference on*, pages 76–85, July 2003.
- [4] Ken Brodlie and Jason Wood. Recent advances in volume visualization. *Computer Graphics Forum*, 20(2):125–148, 2001.
- [5] Andreas Buja, J.A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Visualization, 1991. Visualization '91, Proceedings., IEEE Conference on*, pages 156–163, 419, Oct 1991.
- [6] Stefano Burigat and Luca Chittaro. On the effectiveness of overview+detail visualization on mobile devices. *Personal Ubiquitous Comput.*, 17(2):371–385, February 2013.
- [7] Maria Beatriz Carmo, Ana Paula Afonso, and Paulo Pombinho Matos. Visualization of geographic query results for small screen devices. In *Proceedings of the 4th ACM Workshop on Geographical Information Retrieval, GIR '07*, pages 63–64, New York, NY, USA, 2007. ACM.
- [8] Fausto Cattaneo. Image gallery, 2005.
- [9] Winnie Wing-Yi Chan. A survey on multivariate data visualization. 2006.

- [10] H. L. J. Chen, F. F. Samavati, M. C. Sousa, and J. R. Mitchell. Sketch-based volumetric seeded region growing. In *Proceedings of the Third Eurographics Conference on Sketch-Based Interfaces and Modeling, SBM'06*, pages 123–130, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [11] Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 69–75, 2000.
- [12] L. Chittaro. Visualizing information on mobile devices. *Computer*, 39(3):40–45, March 2006.
- [13] Dane Coffey, Nicholas Malbraaten, Trung Le, Iman Borazjani, Fotis Sotiropoulos, and Daniel F. Keefe. Slice wim: A multi-surface, multi-touch interface for overview+detail exploration of volume datasets in virtual reality. In *Symposium on Interactive 3D Graphics and Games, I3D '11*, pages 191–198, New York, NY, USA, 2011. ACM.
- [14] J. de Jesus Nascimento da Silva Junior, B.S. Meiguins, N.S. Carneiro, A.S.G. Meiguins, R.Y. da Silva Franco, and A.G.M. Soares. Prisma mobile: An information visualization tool for tablets. In *Information Visualisation (IV), 2012 16th International Conference on*, pages 182–187, July 2012.
- [15] Steven M. Drucker, Danyel Fisher, Ramik Sadana, Jessica Herron, and m.c. schraefel. Touchviz: A case study comparing two interfaces for data analytics on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13*, pages 2301–2310, New York, NY, USA, 2013. ACM.
- [16] T. Todd Elvins. A survey of algorithms for volume visualization. *SIGGRAPH Comput. Graph.*, 26(3):194–201, August 1992.
- [17] Björn Engdahl, Malin Köksal, and Gary Marsden. Using treemaps to visualize threaded discussion forums on pdas. In *CHI '05 Extended Abstracts on Human*

Factors in Computing Systems, CHI EA '05, pages 1355–1358, New York, NY, USA, 2005. ACM.

- [18] Michael Friendly and Daniel Denis. The early origins and development of the scatterplot. *Journal of the History of the Behavioral Sciences*, 41(2):103–130, 2005.
- [19] Moritz Gerl, Peter Rautek, Tobias Isenberg, and Eduard Grller. Semantics by analogy for illustrative volume visualization. *Computers & Graphics*, 36(3):201 – 213, 2012. Novel Applications of {VR}.
- [20] D.L. Gresh, B.E. Rogowitz, R.L. Winslow, D.F. Scollan, and C.K. Yung. Weave: a system for visually linking 3-d and statistical visualizations applied to cardiac simulation and measurement data. In *Visualization 2000. Proceedings*, pages 489–492, Oct 2000.
- [21] Tor Guttormsen. Visual calamity: Exploring interactive information visualization for tablet interfaces. Master’s thesis, Chalmers University of Technology, 2013.
- [22] Markus Hadwiger, Patric Ljung, Christof Rezk Salama, and Timo Ropinski. Advanced illumination techniques for gpu-based volume raycasting. In *ACM SIGGRAPH 2009 Courses*, SIGGRAPH '09, pages 2:1–2:166, New York, NY, USA, 2009. ACM.
- [23] Tomi Heimonen. Information visualization on small display devices. Master’s thesis, University of Tampere, 2002.
- [24] Amy Henderson, Jim Ahrens, and Charles Law. *The ParaView Guide*. Kitware Clifton Park, NY, 2004.
- [25] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.
- [26] P. Isenberg, T. Isenberg, T. Hesselmann, Bongshin Lee, U. von Zadow, and A. Tang. Data visualization on interactive surfaces: A research agenda. *Computer Graphics and Applications, IEEE*, 33(2):16–24, March 2013.

- [27] Petra Isenberg and Tobias Isenberg. Visualization on interactive surfaces: A research overview/visualisierung auf interaktiven oberflächen: Ein forschungüberblick. *i-com*, 12(3):10–17, 2013.
- [28] Tobias Isenberg. Position Paper: Touch Interaction in Scientific Visualization. In Petra Isenberg, Sheelagh Carpendale, Tobias Hesselmann, Tobias Isenberg, and Bongshin Lee, editors, *Proceedings of the Workshop on Data Exploration on Interactive Surfaces (DEXIS 2011, November 13, 2011, Kobe, Japan)*, pages 24–27, Le Chesnay, France, November 2011. INRIA. DEXIS proceedings published as INRIA research report RR-0421 in May 2012.
- [29] Allan S. Jacobson, Andrew L. Berkin, and Martin N. Orton. Linkwinds: Interactive scientific data analysis and visualization. *Commun. ACM*, 37(4):42–52, April 1994.
- [30] Seok-Jae Jeong and Arie E. Kaufman. Interactive wireless virtual colonoscopy. *Vis. Comput.*, 23(8):545–557, July 2007.
- [31] R. Jianu, C. Demiralp, and D.H. Laidlaw. Exploring 3d dti fiber tracts with linked 2d representations. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1449–1456, Nov 2009.
- [32] Daniel F. Keefe. Integrating visualization and interaction research to improve scientific workflows. *Computer Graphics and Applications, IEEE*, 30(2):8–13, March 2010.
- [33] D.A. Keim. Information visualization and visual data mining. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):1–8, Jan 2002.
- [34] D.A. Keim and H.-P. Kriegel. Visdb: database exploration using multidimensional visualization. *Computer Graphics and Applications, IEEE*, 14(5):40–49, Sept 1994.
- [35] Gordon Kindlmann and James W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the 1998 IEEE Symposium on Volume Visualization, VVS '98*, pages 79–86, New York, NY, USA, 1998. ACM.

- [36] Tijmen Klein, Florimond Guniat, Luc Pastur, Frdric Vernier, and Tobias Isenberg. A design study of direct-touch interaction for exploratory 3d scientific visualization. *Computer Graphics Forum*, 31(3pt3):1225–1234, 2012.
- [37] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 8(3):270–285, Jul 2002.
- [38] P Kohlmann, S Bruckner, A Kanitsar, and E Groller. Contextual picking of volumetric structures. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pages 185–192, 2009.
- [39] Robert Kosara, Helwig Hauser, and Donna L. Gresh. An interaction view on information visualization, 2003.
- [40] Robert Kosara, Gerald N. Sahling, and Helwig Hauser. Linking scientific and information visualization with interactive 3d scatterplots. In *In Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 133–140, 2004.
- [41] Thomas Kroes, Frits H. Post, and Charl P. Botha. Exposure render: An interactive photo-realistic volume rendering framework. *PLoS ONE*, 7(7):e38586, 07 2012.
- [42] J. Kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 38–, Washington, DC, USA, 2003. IEEE Computer Society.
- [43] F. Lamberti and A. Sanna. A streaming-based solution for remote visualization of 3d graphics on mobile devices. *Visualization and Computer Graphics, IEEE Transactions on*, 13(2):247–260, March 2007.
- [44] Fabrizio Lamberti, Claudio Zunino, Andrea Sanna, Antonino Fiume, and Marco Maniezzo. An accelerated remote graphics architecture for pdas. In *Proceedings of the Eighth International Conference on 3D Web Technology, Web3D '03*, pages 55–ff, New York, NY, USA, 2003. ACM.

- [45] Bongshin Lee, P. Isenberg, N.H. Riche, and S. Carpendale. Beyond mouse and keyboard: Expanding design considerations for information visualization interactions. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2689–2698, Dec 2012.
- [46] Marc Levoy. Display of surfaces from volume data. *Computer Graphics and Applications, IEEE*, 8(3):29–37, 1988.
- [47] David Lopez, Lora Oehlberg, Candemir Doger, Tobias Isenberg, et al. Tablet-based interaction for immersive 3d data exploration. In *Posters at the IEEE Conference on Visualization (IEEE VIS 2014, November 9–14, Paris, France)*, 2014.
- [48] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21(4):163–169, August 1987.
- [49] L. Marsalek, A. Hauber, and P. Slusallek. High-speed volume ray casting with cuda. In *Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on*, pages 185–185, Aug 2008.
- [50] Manuel Moser and Daniel Weiskopf. Interactive volume rendering on mobile devices. In *In Vision, Modeling, and Visualization VMV 08 Conference Proceedings*, pages 217–226, 2008.
- [51] Muhammad Mobeen Movania. *OpenGL Development Cookbook*. Packt Publishing Ltd, 2013.
- [52] C Ohazama. Opengl vizserver white paper. *White Paper*, 1999.
- [53] Shigeru Owada, Frank Nielsen, and Takeo Igarashi. Volume catcher. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games, I3D '05*, pages 111–116, New York, NY, USA, 2005. ACM.
- [54] William A. Pike, John Stasko, Remco Chang, and Theresa A. O’Connell. The science of interaction. *Information Visualization*, 8(4):263–274, 2009.

- [55] S. Pinheiro, B.S. Meiguins, A.S.G. Meiguins, and L.H. Almeida. A tourism information analysis tool for mobile devices. In *Information Visualisation, 2008. IV '08. 12th International Conference*, pages 264–269, July 2008.
- [56] P. Rautek, S. Bruckner, and E. Groller. Semantic layers for illustrative volume rendering. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1336–1343, Nov 2007.
- [57] Peter Rautek, Stefan Bruckner, and M. Eduard Groller. Interaction-dependent semantics for illustrative volume rendering. *Computer Graphics Forum*, 27(3):847–854, 2008.
- [58] Daniel C. Robbins, Edward Cutrell, Raman Sarin, and Eric Horvitz. Zonezoom: Map navigation for smartphones with recursive view segmentation. In *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '04*, pages 231–234, New York, NY, USA, 2004. ACM.
- [59] Jonathan C. Roberts. Waltz: an exploratory visualization tool for volume data using multiform abstract displays, 1998.
- [60] T. Ropinski, J. Prassni, F. Steinicke, and K. Hinrichs. Stroke-based transfer function design. In *Proceedings of the Fifth Eurographics / IEEE VGTC Conference on Point-Based Graphics, SPBG'08*, pages 41–48, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association.
- [61] Virpi Roto, Andrei Popescu, Antti Koivisto, and Elina Vartiainen. Minimap: A web page visualization method for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '06*, pages 35–44, New York, NY, USA, 2006. ACM.
- [62] O. Rübel, G. H. Weber, S. V. E. Keränen, C. C. Fowlkes, C. L. Luengo Hendriks, L. Simirenko, N. Y. Shah, M. B. Eisen, M. D. Biggin, H. Hagen, D. Sudar, J. Malik, D. W. Knowles, and B. Hamann. Pointcloudxplore: Visual analysis of 3d gene expression data using physical views and parallel coordinates. In *Proceedings of*

- the Eighth Joint Eurographics / IEEE VGTC Conference on Visualization*, EURO-VIS'06, pages 203–210, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.
- [63] Jeffrey M. Rzeszutarski and Aniket Kittur. Kinetica: Naturalistic multi-touch data visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 897–906, New York, NY, USA, 2014. ACM.
- [64] Will Schroeder, Ken Martin, and Bill Lorensen. *An Object-Oriented Approach To 3D Graphics*. Prentice hall, 1997.
- [65] William J Schroeder, KM Martin, LS Avila, and CC Law. The vtk users guide. kitware, 1998.
- [66] Katie A Siek, Yvonne Rogers, and Kay H Connelly. Fat finger worries: how older and younger users physically interact with pdas. In *Human-Computer Interaction-INTERACT 2005*, pages 267–280. Springer, 2005.
- [67] Cláudio Teixeira Silva, Joao Luiz Dihl Comba, Steven Paul Callahan, and Fabio Fedrizzi Bernardon. A survey of gpu-based volume rendering of unstructured grids. *Revista de informática teórica e aplicada. Porto Alegre, RS. Vol. 12, n. 2 (out. 2005), p. 9-29*, 2005.
- [68] Simon Stegmaier, Marcelo Magallón, and Thomas Ertl. A generic solution for hardware-accelerated remote visualization. In *Proceedings of the Symposium on Data Visualisation 2002*, VISSYM '02, pages 87–ff, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [69] Nicole Sultanum, Ehud Sharlin, Mario Costa Sousa, Daniel N. Miranda-Filho, and Rob 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.

- [70] Ivan E. Sutherland, Robert F. Sproull, and Robert A. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Comput. Surv.*, 6(1):1–55, March 1974.
- [71] Thetawave. Volume ray casting, January 2006.
- [72] Edward R Tufte and PR Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.
- [73] John W Tukey. Exploratory data analysis. 1977.
- [74] Fan-Yin Tzeng, Eric B. Lum, and Kwan-Liu Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 66–, Washington, DC, USA, 2003. IEEE Computer Society.
- [75] C. Waldeck and Dirk Balfanz. Mobile liquid 2d scatter space (ml2dss). In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 494–498, July 2004.
- [76] Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '00*, pages 110–119, New York, NY, USA, 2000. ACM.
- [77] M.O. Ward. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *Visualization, 1994., Visualization '94, Proceedings., IEEE Conference on*, pages 326–333, Oct 1994.
- [78] Lee Westover. Footprint evaluation for volume rendering. *SIGGRAPH Comput. Graph.*, 24(4):367–376, September 1990.
- [79] Alexander Wiebel, Frans M Vos, and Hans-Christian Hege. Perception-oriented picking of structures in direct volumetric renderings. Technical Report 11-45, ZIB, Takustr.7, 14195 Berlin, 2011.

- [80] Pak Chung Wong and R Daniel Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization*, pages 3–33, 1994.
- [81] Lingyun Yu, K Efstathiou, P Isenberg, and T Isenberg. Efficient structure-aware selection techniques for 3d point cloud visualizations with 2dof input. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2245–2254, 2012.
- [82] Lingyun Yu, P. Svetachov, P. Isenberg, M.H. Everts, and T. Isenberg. Fi3d: Direct-touch interaction for the exploration of 3d scientific visualization spaces. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1613–1622, Nov 2010.
- [83] Xiaoru Yuan, Nan Zhang, Minh X. Nguyen, and Baoquan Chen. Volume cutout. *The Visual Computer*, 21(8-10):745–754, 2005.
- [84] E. Zraggen, R. Zeleznik, and S.M. Drucker. Panoramicdata: Data analysis through pen & touch. *Visualization and Computer Graphics, IEEE Transactions on*, 20(12):2112–2121, Dec 2014.
- [85] Changgong Zhang, Ping Xi, and Chaoxin Zhang. Cuda-based volume ray-casting using cubic b-spline. In *Virtual Reality and Visualization (ICVRV), 2011 International Conference on*, pages 84–88, Nov 2011.
- [86] Hong Zhou, Huamin Qu, Yingcai Wu, and Ming-Yuen Chan. Volume visualization on mobile devices. In *14th Pacific conference on computer graphics and applications*, pages 76–84, 2006.