Bridging Immersion Levels in Spatial Data Exploration: Visualization, Interaction, and Computing

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Lixiang Zhao

November 2025

UNIVERSITY OF LIVERPOOL

PGR DECLARATION OF ACADEMIC HONESTY

NAME	Lixiang Zhao
STUDENT NUMBER	201593126
SCHOOL/INSTITUTE	Computer Science
TITLE OF WORK	Bridging Immersion Levels in Spatial Data Exploration: Visualization, Interaction, and Computing

This form should be completed by the student and appended to any piece of work that is submitted for examination. Submission by the student of the form by electronic means constitutes their confirmation of the terms of the declaration.

Students should familiarise themselves with Appendix 4 of the PGR Code of Practice: PGR Academic Integrity Policy, which provides the definitions of academic malpractice and the policies and procedures that apply to the investigation of alleged incidents.

Students found to have committed academic malpractice will receive penalties in accordance with the PGR Academic Integrity Policy and/ or the University Policy on Misconduct in Research. In severe cases this might include termination of studies.

STUDENT DECLARATION

I confirm that:

- I have read and understood the University's PGR Academic Integrity Policy and the University Policy on Misconduct in Research.
- I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.
- I have not copied material from another source nor committed plagiarism nor fabricated, falsified or embellished data when completing the attached material.
- I have not copied material from another source, nor colluded with any other student in the preparation and production of this material.
- If an allegation of suspected academic malpractice is made, I give permission to the University to use Turnitin or equivalent software to ensure that the submitted material does not breach academic integrity requirements.

SIGNATURE…Lixiang Zhao……

DATE…2025.11.28……

Abstract

Visualization enables experts to interpret domain-specific data through cycles of exploration and decision-making. Immersive visualization—via VR, AR, and MR—has gained attention for spatial data analysis due to stereoscopic rendering and embodied 6DOF interaction. Despite progress in immersive analytics, spatial data exploration continues to present challenges across multiple levels of immersion. This thesis addresses the central question: How immersive environments should be designed to support spatial data exploration effectively and efficiently? First, we investigate imprecise mid-air input for spatial data selection and introduce target- and context-aware selection techniques for cosmological data that infer user intent from approximate pointing or stroke input combined with data properties such as density, generating accurate selection volumes despite occlusion, heterogeneous density, and complex geometry. A controlled study shows that these methods outperform region-based techniques in accuracy and intent alignment. Second, because our selection techniques depend on computationally expensive scalar fields, we present a GPU-accelerated adaptive kernel density estimation method that recomputes the field at finer resolution as users navigate to smaller scales, providing real-time evaluation of fine-grained structures without perceptible latency. Third, to support multi-scale visualization while preserving global context, we propose two interfaces that maintain users' awareness of scale, position, and orientation, enable smooth transitions across spatial scales, and support multi-user collaboration. Finally, recognizing that scientific workflows often require both 2D and 3D representations, we develop a cross-reality environment that unifies a 2D surface with a 3D AR workspace, allowing visualizations to be placed on the surface, in space, or both, while maintaining spatial perception and enabling flexible hybrid interaction. Together, these contributions provide visualization methods, interaction techniques, and high-performance computing solutions that collectively answer the thesis's central question and outline a path toward effective immersive spatial data exploration.

Acknowledgements

Completing this PhD has been a long journey filled with learning, challenges, and growth. I am deeply grateful to the many people who have supported and encouraged me throughout these years. Their guidance and companionship made this work possible, and I would like to express my sincere appreciation to all of them.

My deepest and most heartfelt thanks go to my supervisor, Dr. Lingyun Yu. Her mentorship has been central to every stage of my Ph.D. Lingyun is not only exceptionally strong in her professional expertise, but also holds herself-and her students-to high academic standards. Her rigor, responsibility, and dedication to research have shaped the way I think and work. Lingyun's clarity of thought, her ability to identify the essence of a problem, and her commitment to doing things the right way have continually inspired me. She pushed me to think more deeply, encouraged me to explore ideas with independence, and guided me to pursue excellence with confidence and integrity. I am especially grateful for the hope and support she gave me during the most difficult moments of my Ph.D. When I doubted myself the most, she remained patient and encouraging, helping me regain direction and reminding me of my potential. I sincerely appreciate her insight, her patience, and her unwavering belief in me. The lessons I learned from her—about research, about thinking, and about responsibility—will stay with me throughout my career and beyond. It has truly been a privilege to be mentored by her.

I am also grateful to my co-supervisor, Dr. Tobias Isenberg. Tobias has been an exceptional mentor whose clarity of thought, deep knowledge of visualization. His careful feedback and thoughtful discussions pushed me to refine my ideas and broaden my perspective. I have benefited enormously from his expertise and patience, and I am thankful for the many things he taught me.

I also wish to thank the members of my defense committee for their time and constructive

evaluation of my work. Their insightful comments, critical perspectives, and thoughtful suggestions have helped strengthen this dissertation in both clarity and depth. I am deeply grateful for the care with which they reviewed my research, the challenging questions that pushed me to think more rigorously, and the guidance they provided for future directions. Their expertise and generosity have made a meaningful contribution to the refinement of this thesis.

My sincerest thanks go to my collaborators and colleagues. Their contributions, discussions, and shared enthusiasm made the research process both enjoyable and intellectually stimulating. I deeply appreciate their support and the many opportunities to learn from them. A special thanks goes to Dr. Konstantinos Efstathiou for his excellent mathematical guidance and his careful verification of parts of the paper. His expertise and generosity have been invaluable throughout this work.

I am very fortunate to have friends who encouraged me, listened to me, and stood by me in moments of stress and self-doubt. Their presence made the most challenging periods easier and the good moments even brighter.

Above all, I want to express my heartfelt gratitude to my family. Their unconditional support, trust, and love have always been my foundation. They encouraged me to pursue what I truly cared about and gave me the freedom and confidence to follow this path. This achievement would not have been possible without them.

To everyone who has supported, inspired, and accompanied me along the way: thank you. I will always remember your kindness and generosity.

List of Publications

Full Papers—Publications as the Leading Author

The following journal and conference papers form the basis of the dissertation chapters.

1. **L. Zhao**, T. Isenberg, F. Xie, H.-N. Liang and L. Yu, "MeTACAST: Target- and Context-Aware Spatial Selection in VR," *IEEE Trans Vis Comput Graph*, vol. 30, no. 1, pp. 480-494, Jan. 2024, doi:10.1109/TVCG.2023.3326517.
   (**CCF A**, **JCR Q1**, **First author**, refer to **Chapter 3**)

2. **L. Zhao**, F. Xie, T. Isenberg, H.-N. Liang and L. Yu, "ScaleFree: Dynamic KDE for Multiscale Point Cloud Exploration in VR," in *Proc. VR*, IEEE Comp. Soc., Los Alamitos, pp. 358-368, Mar. 2026, doi:10.1109/VR67842.2026.00057.
   (**CCF A**, **First author**, refer to **Chapter 4**)

3. **L. Zhao**, N. Cao, S. He, H.-N. Liang and L. Yu, "L-WiM: Collaborative Exploration in Immersive Environments," in *Proc. ISMAR*, IEEE Comp. Soc., Los Alamitos, pp. 118-123, Oct. 2022, doi:10.1109/ISMAR-Adjunct57072.2022.00031.
   (**CCF B**, **First author**, refer to **Chapter 5**)

4. J. Yang, **L. Zhao**, F. Xie, T. Isenberg, Y. Li, H.-N. Liang and L. Yu, "Where Should We Look Together? Designing Overview+Detail for Collaborative Multiscale Exploration in VR," *submitted to IEEE ISMAR2027*.
   (**Submitted and under-review**, **CCF B**, **Co-first author**, refer to **Chapter 5**)

5. **L. Zhao**, T. Isenberg, F. Xie, H.-N. Liang and L. Yu, "SpatialTouch: Exploring Spatial Data Visualizations in Cross-reality," *IEEE Trans Vis Comput Graph*, vol. 31, no. 1, pp. 897-907, Oct. 2025, doi:10.1109/TVCG.2024.3456368.
   (**CCF A**, **JCR Q1**, **First author**, refer to **Chapter 6**)

Full Papers—Publications as the Second Author

The candidate's following journal and conference papers are not included in this thesis.

6. W. Xu, **L. Zhao**, H. Song, X. Song, Z. Lu, Y. Liu, M. Chen, E. G. Lim, and L. Yu. Mozualization: Crafting Music and Visual Representation with Multimodal AI. In Proc. CHI EA. ACM, New York, Article no. 407, 7 pages, Apr. 2025. doi:10.1145/3706599.3719686

7. H. Yao, **L. Zhao**, B. Chen, K. Li, H.-N. Liang, L. Yu, "3DStoryline: immersive visual storytelling," *Journal of Visualization*, vol. 28, no. 2, pp. 463-480, Apr. 2025, doi:10.1007/s12650-025-01058-5.

8. H. Yao, **L. Zhao**, H.-N. Liang, Y. Liu, Y. Li and L. Yu, "Exploring Embodied Asymmetric Two-Handed Interactions for Immersive Data Exploration," in *CHI EA*, ACM, New York, article no. 140, 10 pages, May 2024, doi:10.1145/3613905.3650777.

9. J. Li, **L. Zhao**, H.-N. Liang and L. Yu, "ImmerView: Adaptive Multi-View Layout for Immersive Situated Visualizations," in *Proc. ISMAR Adjunct*, IEEE Comp. Soc., Los Alamitos, pp. 108–112, Oct. 2023, doi:10.1109/ISMAR-Adjunct60411.2023.00030.

Contents

List of Tables

List of Figures

Abbreviations and nomenclature

AF Acceleration Factor.

AR Augmented Reality.

ASL Adaptive Smoothing Length.

CI Confidence Interval.

CR Cross Reality.

DR Dynamic Resolution.

FDE Final Density Estimation.

FN False Negatives; Missed Target Particles.

FP False Positives; Incorrectly Selected Particles.

FOV Field of View.

HCM Hierarchical Computational Method.

HMD Head-Mounted Display.

KDE Kernel Density Estimation.

MBE Modified Breiman Kernel Density Estimation Method.

MCC Matthews Correlation Coefficient.

MR Mixed Reality.

N Number of Particles.

PDE Pilot Density Estimation.

PM Precomputed Multi-Resolution.

PS Precomputed Single-Resolution.

ROI Region of Interest.

TN True Negatives; Correctly Unselected Particles.

TP True Positives; Correctly Selected Particles.

VR Virtual Reality.

WIM World in Miniature.

XR Extended Reality.

B Minimum Box that Covers the Whole Data.

l_k Smoothing Lengths Along k Direction.

$l_k^{(j)}$ Adaptive Smoothing Lengths Along k Direction.

$m^{(c)}$ 2D Screen Virtual Camera Projection Matrix.

n Index of the Grid Node.

P MaxLine.

$r^{(bl)}$ Screen Surface Bottom-Left Corner Position w.r.t. $r^{(c)}$.

$r^{(c)}$ 2D Screen Virtual Camera Position.

$r^{(h)}$ HMD Virtual Camera Position.

$r^{(sc)}$ Screen Surface Center Position.

$r^{(tr)}$ Screen Surface Top-Right Corner Position w.r.t. $r^{(c)}$.

res Grid Resolution.

s_k Distance Between Adjacent Nodes.

V_{init} Initial Volume of Interest.

$\mathbf{r}^{(m)}$ Local Maximum Position of the Scalar Field.

$\mathbf{r}^{(n)}$ Grid Node Position.

$\mathbf{r}^{(s)}$ Input Position.

∇f Gradient of the Scalar Field f.

$\rho(x)$ Density Value at the Position of x.

ρ_0 Initial Density Threshold.

ρ_s Adjusted Density Threshold.

CHAPTER 1

Introduction

Visualization plays a crucial role in helping experts interpret and understand the complex phenomena behind the domain-specific datasets [1]. Modern datasets in science and engineering—whether collected through advanced sensing instruments or generated by large-scale simulations—contain massive volumes of numerical information that are far too abstract for direct human interpretation. Visualization addresses this challenge by transforming numerical values into perceptually accessible representations, revealing structures and patterns that would otherwise remain hidden. Visualization for data from scientific computing, in particular, focuses on information derived from scientific measurements or computational simulations. For example, in the medical field, CT and MRI scanners produce volumetric datasets encoding intensity values throughout the body. Visualization techniques such as volume rendering convert this data into anatomical structure representations that clinicians can inspect, slice, and manipulate to support diagnosis and treatment planning. In astronomy, large-scale N-body simulations generate time-varying particle data describing galactic evolution. Rendering these particles allows researchers to explore spatial organization, examine gravitational interactions, and select regions of interest.

These examples illustrate that effective visualization depends not only on how information is presented, but also on how users are supported in interpreting and exploring the underlying features. For decades, traditional 2D displays—such as desktop monitors and large projection environments—has been the primary medium for examining spatial data. More recently, immersive environments such as virtual reality (VR), augmented reality (AR), and mixed reality (MR) have introduced new opportunities for visualizing and interacting with

complex datasets (shown in Fig. 1.1), attracting growing attention within the visualization community. A growing body of work over the past decades has demonstrated the value of immersive environments for spatial data exploration in scientific contexts. In recent years, advances in simulation techniques and sensing instruments have continued to increase the size, complexity, and scale of modern scientific datasets. Such datasets are often large-scale, multi-scale, unstructured, and computationally demanding, placing increasing demands on visualization, interaction, and computational techniques that help researchers explore, interpret, and analyze complex spatial data structures. These developments raise a critical design question for immersive spatial data exploration: **How should immersive environments be designed to support spatial data exploration effectively and efficiently?** This question forms the central question of this thesis. Understanding when and how immersive environments effectively support scientific exploration is essential for realizing the immersive environments' potential and avoiding inappropriate or ineffective use. To investigate this question, we begin by analyzing the challenges that arise when exploring inherently three-dimensional spatial data in immersive environments, taking into account both the characteristics of the data and the properties of the environment. Such considerations are essential, as they encourage us to re-examine common assumptions, for example, whether all 3D data truly benefits from a 3D immersive environment, and what specific challenges or limitations immersive systems introduce. Understanding these factors provides a foundation for selecting or designing appropriate immersive environments, visualization methods, and interaction techniques that effectively support scientific analysis.

A natural starting point is the inherent properties of spatial datasets, which already make efficient exploration difficult. Many spatial datasets contain millions or even billions of elements, placing heavy computational demands on processing and rendering pipelines. Their structures are often multi-scale, with meaningful features distributed across widely varying spatial resolutions. In addition, spatial data frequently exhibits complex spatial organization, such as heterogeneous density distributions, irregular geometries, and strong occlusion, which challenge users' ability to perceive relationships and identify patterns. Comprehensive analysis typically requires navigating between visual representations or levels of abstraction, which increases cognitive load and demands interaction support.

Given these complexities, it is helpful to reflect on the long-standing role of traditional 2D displays in visualizing spatial data. These systems offer clear advantages: large, high-resolution screens support detailed inspection, and mouse or touch-based input

Figure 1.1: A domain expert is exploring the volume data in the immersive environment [2].

enables familiar, direct interaction. Yet, because 2D displays project three-dimensional structures onto flat surfaces, users must mentally infer depth and spatial relationships from perspective cues. Interaction is also constrained to the screen plane, limiting the ways users can physically relate to or navigate within the data. Immersive environments, by contrast, provide a fundamentally different set of affordances. By placing users inside a stereoscopic, three-dimensional space, they align more closely with natural human spatial perception. With six degrees of freedom (6DOF), users can walk around structures, change viewpoints through natural head movement, and interact with data using mid-air gestures or tangible controllers. Such embodied interactions can enhance spatial awareness and make depth, orientation, and structural relationships more perceivable than on 2D displays.

However, immersive environments also introduce significant challenges for spatial data exploration. First, 6DOF input modalities lack physical support, making precise or fine-grained operations difficult to perform in mid-air. Second, immersive systems require much

higher rendering performance—often above 60 frames per second—to maintain visual stability and prevent motion sickness. These constraints place substantial pressure on computational pipelines and interaction design, particularly when working with large-scale or complex spatial datasets. Addressing these challenges requires an integrated perspective that connects three tightly coupled aspects: **visualization**, **interaction**, and **computing**. Visualization determines how complex spatial structures are represented and perceived; interaction determines how users specify intentions, navigate scales, and manipulate data; and computing determines whether these visualization and interaction can be supported responsively under the data and latency constraints of immersive environments. This thesis focuses on visualization, interaction, and computing because these aspects are necessary foundations for representing spatial structures, supporting user intent and action, and sustaining responsive performance in immersive environments. Together, these aspects provide the basis for formulating the research questions addressed in this thesis.

1.1 Visualization

Visualization is the first and most fundamental step in any data exploration workflow. Before performing analytical tasks, users must perceive, observe, and cognitively interpret visual representations of the underlying data. Effective visualization provides the essential entry point for understanding the spatial structure and organization of complex spatial datasets. In immersive environments, this role becomes even more critical: stereoscopic rendering and embodied viewpoints allow users to inspect structures from arbitrary angles and perceive spatial relationships in ways that traditional 2D displays cannot support [3].

In this thesis, we investigate how three core visualization challenges can be addressed in immersive environments: (1) presenting complex 3D structures, including how to encode spatial organization in ways that make use of immersion-specific depth cues; (2) visualizing multi-scale patterns, exploring how users can transition between scales while maintaining orientation, continuity, and a coherent mental model; and (3) integrating 2D and 3D representations, examining how different immersive levels and display modalities can be combined to support multi-view analysis without disrupting users' understanding of the data. These considerations aim to clarify how immersive environments can be designed to present spatial datasets in ways that preserve spatial understanding, support multi-scale reasoning, and maintain mental continuity as users move across immersive levels.

1.2 Interaction

Once users gain an initial perceptual understanding through visualization, interaction becomes essential for transforming perception into meaningful analysis. In the context of three-dimensional spatial datasets, interaction is not merely a mechanism for manipulating geometry; rather, it is a core component of sense-making. Unlike interactions applied to predefined geometric objects, interaction techniques for spatial data must support operations on complex, data-driven structures whose boundaries, shapes, and scales are not known in advance. Users must be able to adjust viewpoints, select fine-grained regions of interest in 3D space, and manipulate visualization widgets to reveal internal properties or structural relationships within the data. Interaction, therefore, plays a critical role in enabling users to construct analytical understanding-bridging the gap between visual perception and deeper scientific insight. However, interaction in immersive 3D environments introduces additional challenges. While 6DOF input affords flexible, embodied manipulation, its lack of physical support leads to imprecise or unstable actions, making it difficult for users to express fine-grained intentions or perform targeted operations.

In our scenario, we explore the possibilities of spatial data analysis in immersive environments where both visualization of high-volume spatial data and precise interaction are required. To address these dual challenges, we examine how users act and reason while engaging with spatial datasets in immersive 3D spaces. Building on these observations, we develop context-aware and target-aware interaction techniques that account for both users' intentions and intrinsic characteristics of the data. These techniques are designed to clarify how precise and meaningful operations can be supported in immersive environments—even when user input is imprecise—thereby enabling natural and intuitive scientific exploration.

1.3 Computing

Spatial datasets are often massive, high-resolution, and computationally intensive to process. They can contain millions of elements, require complex filtering or aggregation, and frequently involve multi-scale structures and multiple representations that demand adaptive level-of-detail management. Interactive visualization makes this even more challenging, as each user action may trigger expensive view-dependent or data-dependent computations that must update immediately to support analytical continuity. In immersive

environments, these computational demands become even more pronounced. Rendering must remain smooth, latency must be minimal, and system responsiveness must be preserved to maintain visual comfort and a stable sense of presence.

In this thesis, we investigate how computational techniques can support smooth, responsive, and scientifically meaningful interaction with large-scale datasets in immersive environments. We explore GPU-accelerated data structures and real-time algorithms that reduce latency, adapt rendering fidelity to users' visual focus, and ensure that interaction remains fluid even under substantial computational load. Together, these techniques clarify how immersive systems can remain responsive while handling the complexity of modern scientific spatial data, which is a critical prerequisite for evaluating whether immersive environments are suitable for scientific exploration.

1.4 Thesis Statement

Scientific spatial datasets pose substantial challenges for immersive exploration due to both their inherent properties and the limitations of immersive systems. These challenges motivate four research questions presented below.

First, although 6DOF input in immersive 3D environments offers flexible navigation and manipulation, it lacks physical support, reducing the precision required for fine-grained actions. This limitation raises important concerns about how interaction techniques should be designed to support the detailed, targeted operations often required in scientific exploration. This leads to our first research question: **(RQ1) How can immersive environments support precise, fine-grained scientific spatial data exploration despite the inherent imprecision of 6DOF input modalities?**

Second, immersive environments place high demands on system performance, as smooth and stable rendering is essential for maintaining visual comfort and a consistent sense of presence. At the same time, the large-scale nature of modern scientific spatial datasets imposes substantial computational burdens on rendering and interaction systems, making it challenging to sustain the responsiveness required for exploration. This tension leads to our second research question: **(RQ2) How can immersive environments maintain responsiveness when exploring computationally demanding spatial datasets?**

Third, many spatial datasets exhibit inherently multi-scale spatial structures, with meaningful features distributed across several orders of magnitude. Effective analysis requires

seamless transitions between large-scale overviews and fine-grained details, yet immersive environments can make it difficult for users to maintain awareness of position, orientation, and scale during such transitions. This leads to our third research question: **(RQ3) How can immersive environments support effective navigation across multiple spatial scales in spatial datasets?**

Fourth, spatial datasets often require the examination of multiple data features, which are commonly visualized through a combination of 2D projections and 3D spatial representations. Exploring such heterogeneous representations frequently requires switching between or integrating across multiple views, which can disrupt visual continuity and increase cognitive load. Immersive environments offer opportunities to spatially arrange these representations around the user, or even merge views within the surrounding 3D space. However, it remains unclear how these affordances can be leveraged to meaningfully support multi-feature scientific exploration. This leads to our fourth research question: **(RQ4) How can immersive environments support the effective presentation and integration of spatial datasets that rely on multiple 2D and 3D visual representations?**

This thesis contributes to:

- three target- and context-aware selection techniques for scalar-field-based spatial data in immersive environments, addressing **RQ1** by enabling accurate interpretation of user intent and supporting precise, fine-grained operations despite the inherent imprecision of 6DOF input (refer to **Chapter 3**);
- a fast, GPU-based kernel density estimation (KDE) method for real-time density computation, addressing **RQ2** by sustaining responsiveness during the exploration of large-scale and computationally demanding spatial datasets (refer to **Chapter 4**);
- a hierarchical WiM-based interface that visualizes the multi-scale structure of spatial data, addressing **RQ3** by helping users understand hierarchical relationships and navigate across different spatial scales in immersive environments (refer to **Chapter 5**);
- a set of overview+detail layouts that organize global context and local structures across multiple views, addressing **RQ3** by enabling users to access both overview and fine-scale details and navigate complex multi-scale spatial data (refer to **Chapter 5**);
- a cross-reality environment that seamlessly integrates monoscopic 2D and stereoscopic 3D spaces, addressing **RQ4** by providing a unified platform for presenting and combining multiple 2D and 3D visual representations, thereby supporting coherent multi-feature spatial data exploration (refer to **Chapter 6**).

1.5 Thesis Overview

This thesis is organized into eight chapters, each summarized below:

- **Chapter 2: Background and Overview of Related Work.** This chapter introduces the general properties of scientific spatial datasets and illustrates their structural and computational characteristics through examples including cosmological simulations, medical CT/MRI volumes, and biomolecular data. It then reviews visualization, interaction, and high-performance computing methods in immersive environments, situating our research questions within existing work and identifying the gaps that motivate this thesis.
- **Chapter 3: MeTACAST: Target- and Context-aware Spatial Selection in VR.** This chapter addresses **RQ1** by introducing three spatial selection techniques for scalar-field datasets in VR, each designed to support different user intents. We evaluate the techniques in a controlled user study and derive guidelines for choosing appropriate 3D selection strategies.
- **Chapter 4: ScaleFree: Dynamic KDE for Multiscale Point Cloud Exploration in VR.** This chapter addresses **RQ2** by proposing a GPU-accelerated adaptive KDE method enabling real-time density computation for multiscale point cloud exploration. We incorporate this method into adaptive selection and progressive navigation, and assess performance and usability through a controlled study.
- **Chapter 5: Collaborative Multi-scale Spatial Data Exploration in VR.** This chapter addresses **RQ3** by introducing two interfaces: one linking World-in-Miniature representations for multiscale visualization and navigation, and the other one using an overview+detail paradigm to organize global context and fine-scale structures.
- **Chapter 6: SpatialTouch: Exploring Spatial Data Visualizations in Cross-Reality.** This chapter addresses **RQ4** by presenting a cross-reality system integrating a 2D interactive surface with an AR-based 3D visualization space. We introduce three domain-specific prototypes and evaluate the system with domain experts.
- **Chapter 7: Discussion and Conclusion.** This chapter answers the four research questions and the central thesis question. We then summarizes the key findings and outlines the future research directions.

CHAPTER 2

Background and Overview of Related Work

2.1 Background Features of Spatial Data

Spatial datasets in the scientific domain possess structural, computational, and perceptual characteristics that make effective visualization particularly challenging in the immersive environment. To ground the challenges introduced in Chapter 1, which arise from the properties of spatial data, in this section we introduce three representative types of spatial data in scientific domain that are used throughout this thesis: cosmological simulations (Sec. 2.1.1), medical CT/MRI volumetric scans (Sec. 2.1.2) and biomolecular structural data (Sec. 2.1.3). These datasets were selected not only because they serve as the primary data sources for our technique designs and studies, but also because, together, they span a broad spectrum of data characteristics commonly found in scientific domains, as summarized in Sec. 2.1.4.

2.1.1 Cosmological Simulations

Cosmological simulations (shown in Fig. 2.1) model the formation and evolution of the universe by numerically evolving large ensembles of particles under gravity. These simulations capture the emergence of the cosmic web—filaments, sheets, clusters, and voids—that defines the universe's large-scale structure [5]. The resulting particle datasets span spatial extents from megaparsec to gigaparsec scales and include structures ranging from diffuse filaments to dense halos and subhalos. The spatial distribution is heterogeneous: com-

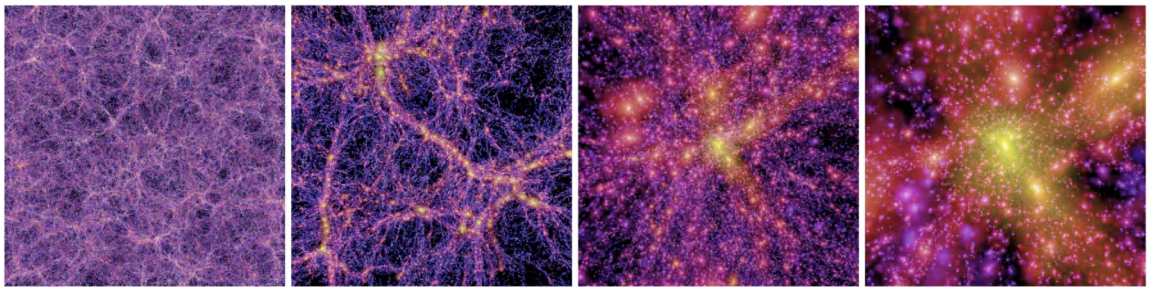


Figure 2.1: The renderings of the Millennium simulation in various scales [4].

pact, high-density regions are embedded within low-density voids, resulting in frequent occlusions. Because meaningful features arise across several orders of magnitude, these datasets exhibit multiscale behavior, with global web morphology and halo substructure coexisting within the same volume. These properties characterize cosmological simulations as large-scale, multiscale, and structurally complex particle datasets with heterogeneous and unstructured spatial organization.

2.1.2 Medical CT/MRI Volumetric Data

Figure 2.2: Volume renderings of the tooth dataset using different transfer functions [6].

Medical imaging technologies such as computed tomography (CT) and magnetic resonance imaging (MRI) generate volumetric datasets that capture anatomical structures as dense 3D scalar fields (shown in Fig. 2.2). Each voxel stores an intensity value derived from underlying tissue properties, producing millions of samples per scan and forming a continuous volume without explicit geometric boundaries. Anatomical structures are distributed throughout these volumes, resulting in substantial internal occlusion and making it challenging to isolate or interpret specific regions without appropriate visualization techniques. Because scalar intensities do not directly correspond to distinct anatomical entities, visual

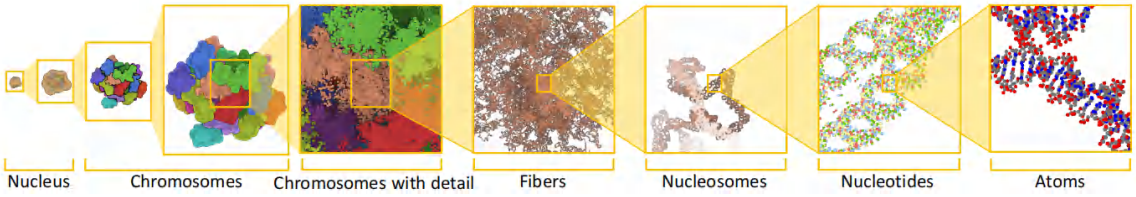


Figure 2.3: Multi-scale visualization of biological genome data [10].

meaning is established through volume rendering techniques [7, 8] and transfer functions [9], which map intensities to color and opacity to reveal tissues, boundaries, and internal structures. In clinical and research practice, clinicians examine these datasets through a combination of orthogonal 2D slice views and 3D volume renderings, each emphasizing different spatial or quantitative aspects of the same data. These properties characterize medical imaging volumes as dense, continuous scalar fields with high internal complexity and a natural need for complementary 2D/3D representations.

2.1.3 Biological Molecular Data

Biological molecular data encompasses a wide range of information describing the composition, structure, dynamics, and interactions of biomolecules such as proteins, nucleic acids, and large molecular complexes (shown in Fig. 2.3). These datasets store atomic coordinates and bonding relationships for thousands to millions of atoms, capturing spatial arrangements from global molecular folds to local secondary structures and atom-level interactions. Molecular organization is inherently multiscale: global folds and domain arrangements coexist with local secondary structures and atomic-level interactions. Spatial density is strongly heterogeneous—compact protein cores, helices, and binding pockets contrast with flexible loops or extended chains—resulting in substantial geometric intricacy and frequent self-occlusion. To manage this complexity, molecular visualization commonly employs multiple representational abstractions [11], such as space-fill, licorice, backbone, or ribbon views, each revealing different structural or functional aspects. Analysts often switch between or combine these representations to understand both global organization and fine-grained atomic relationships. Together, these properties characterize biomolecular datasets as dense, highly detailed, and multiscale spatial structures that require layered abstraction and flexible representation for effective interpretation.

2.1.4 Data Features

Spatial datasets across cosmology, medical imaging, and molecular biology share several intrinsic characteristics that make interactive visualization, particularly in immersive environments, especially challenging. Although these datasets differ in origin and representation, they exhibit common structural, spatial, and computational properties of spatial data, which directly motivate the research questions of this thesis.

Large-scale Data Magnitude

Spatial datasets often involve a large number of spatial samples and a broad spatial extent. In this thesis, data magnitude refers to the number of particles, voxels, atoms, or geometric primitives that must be stored, processed, and rendered during exploration. For example, cosmological simulations can track millions to trillions of particles; medical CT/MRI scans produce volumetric grids with millions of voxels; and molecular models may contain thousands to millions of atoms. These data characteristics place demands on memory capacity, data transfer, filtering and rendering. During exploration, view-dependent structures often need to be updated or recomputed, which further increases computational load. These requirements become even more demanding in immersive environments, where real-time responsiveness and stable high-frame-rate rendering are critical for maintaining visual comfort and a consistent sense of presence. As a result, the magnitude of large-scale data becomes a central bottleneck for effective visualization in immersive systems.

Multiscale Structural Complexity

A defining characteristic of spatial data is that essential structures and phenomena are distributed across multiple, nested spatial scales, with each level revealing qualitatively different patterns. This multiscale nature is evident across domains: cosmological simulations contain hierarchical structures from large filaments to small subhalos; medical volumes encode features ranging from entire organs to fine tissue boundaries; and molecular systems include global folds, local secondary structures, and atomic-level details. In practice, experts must explore features across these heterogeneous levels of detail to derive meaningful insight. However, visualizing such structurally diverse scales within a unified immersive environment is challenging. Fine-scale structures often require high-resolution sampling or on-the-fly recomputation, while larger-scale context must remain

visible to preserve a coherent understanding of the data. Immersive systems therefore must reveal detailed structures without losing global context and support smooth transitions across widely different scales, all while maintaining user orientation, visual continuity, and interpretability. These issues represent a central challenge for working with multiscale spatial data in immersive environments.

Reliance on Multiple 2D and 3D Representations

Across scientific domains, effective analysis often relies on a combination of 2D and 3D visual representations, as no single view can capture all relevant structural or quantitative information. This requirement is evident in the three dataset types considered in this thesis. Cosmological simulations benefit from 3D spatial views to understand the distribution of particle-based structures, yet 2D projections and statistical summaries remain essential for examining density fields or comparing large-scale distributions. Medical CT/MRI scans offer volumetric insight through 3D renderings, while 2D slice views are essential for inspecting anatomical boundaries or performing precise clinical measurements. Molecular datasets rely on 3D structural depictions to reveal global folds and spatial interactions, complemented by 2D schematics or projected abstractions that highlight functional regions or sequence-related attributes. Because these representations reveal different but complementary aspects of the same data, scientists routinely move between them or use them in tandem to form a coherent understanding. This dependence places analytical demands on visualization systems: spatial relationships must remain consistent across views; links between 2D and 3D representations must preserve users' mental models; and interactions within one representation must trigger updates in others in a way that maintains visual continuity and interpretability.

2.1.5 Summary

Across cosmology, medical imaging, and molecular biology, spatial datasets share three characteristics that are central to this thesis: many spatial samples, structures distributed across multiple scales, and reliance on multiple 2D and 3D representations. While immersive environments offer multiple levels of immersion and flexible spatial presentation, how to leverage these capabilities to present heterogeneous representations coherently—and to help users maintain a stable mental model across them—remains an important design problem for immersive visualization. These considerations motivate us to rethink how

visualization, interaction, and computation strategies should be designed to support spatial data exploration in the immersive environment.

2.2 Overview of Related Work

Creating effective exploration methods for large-scale spatial datasets in immersive environments requires addressing the distinct challenges arising from both the characteristics of spatial data and the constraints of immersive systems. As indicated in Chapter 1, it is crucial to start fundamentally and adopt an integrated perspective—spanning visualization, interaction, and computation—to address challenges in immersive spatial data exploration. In this chapter, we present an overview of related work including three aspects in the immersive environment: immersive visualization (Sec. 2.2.1), spatial interaction techniques (Sec. 2.2.2) and high-performance computing (Sec. 2.2.3).

2.2.1 Immersive Visualization

Immersive visualization has long been researched and employed in domain studies and applications. To contextualize immersive technologies within the broader spectrum of user–environment relationships, Milgram et al. proposed the well-known reality–virtuality continuum [12]. This model describes a continuous range of environments spanning from the real world on the far left to fully virtual environments on the far right. On the left side of the continuum, users directly perceive the physical environment with no computer-mediated content. Moving slightly toward the virtual side, traditional 2D displays (such as desktop monitors or large projection screens) introduce virtual imagery, but only through a framed, non-immersive “window-on-the-world” view. Further along the continuum, AR systems primarily present the real environment while overlaying virtual information onto it, allowing users to see and interact with both real and virtual elements simultaneously. The right endpoint corresponds to a fully VR environment, which creates a completely enclosed environment for the user that is independent of the physical world, providing a dedicated space for data visualization and interaction in which users cannot see the real world through the virtual display. The intermediate region spanning AR to VR is referred to as MR, encompassing technologies that blend the real and virtual worlds.

Immersive visualization experienced via VR, AR and MR techniques has emerged as a

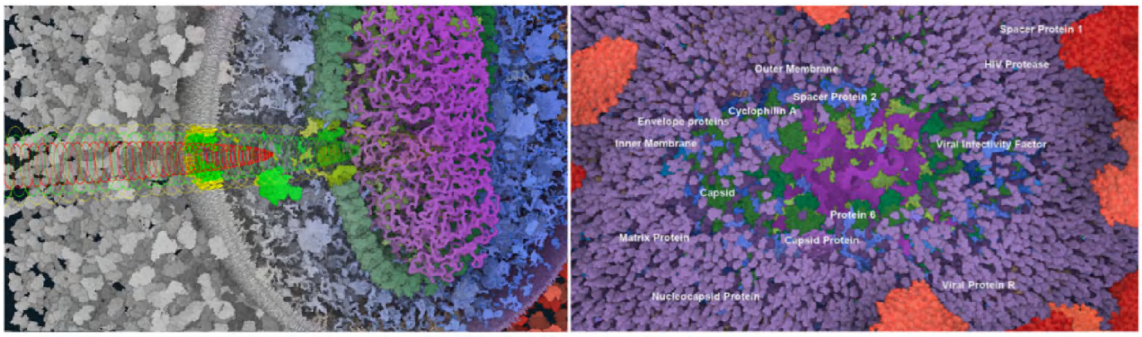


Figure 2.4: The illustration [15] shows the Nanotilus system exploring the HIV model in VR and the corresponding first-person view.

medium for spatial data exploration due to its ability to simulate real-world visual perception within computer-generated environments. By combining stereoscopic rendering with 6DOF motion tracking, the immersive environment enables users to view and manipulate spatial data from arbitrary perspectives [13]. This ability significantly enhances spatial understanding [14] and facilitates the exploration of complex spatial datasets that consist of millions or trillions of intricately detailed components.

One challenge for spatial data visualization in immersive environments is presenting multi-scale patterns of spatial data and enabling users to transition between different spatial scales. Previous work commonly employs techniques such as Worlds in Miniature (WIM) [16] and teleportation [17] to transition between spatial locations and levels of scale. Immersive systems also support pan-and-zoom interactions to let users focus on fine-scale structures; however, as Yang et al. [18] observed, repeated zooming and frequent perspective shifts can easily disorient users, emphasizing the importance of maintaining a global context. To address this point, Kuťak et al. [19] introduced Vivern , a multi-scale molecular visualization technique for DNA nanostructures that uses a lens: fine-grained nucleotide details are revealed within a spherical lens region, while the surrounding single- and double-strand structures are shown at more abstract scales, enabling seamless focus+context exploration across multiple semantic levels. Another strategy is overview+detail, which separates broad structural information from fine-grained details across distinct views, preserving visual continuity. This technique enables simultaneous access to global and local data representations, supporting user orientation and analysis. Prior work has applied overview+detail techniques to point cloud [18] and volumetric medi-

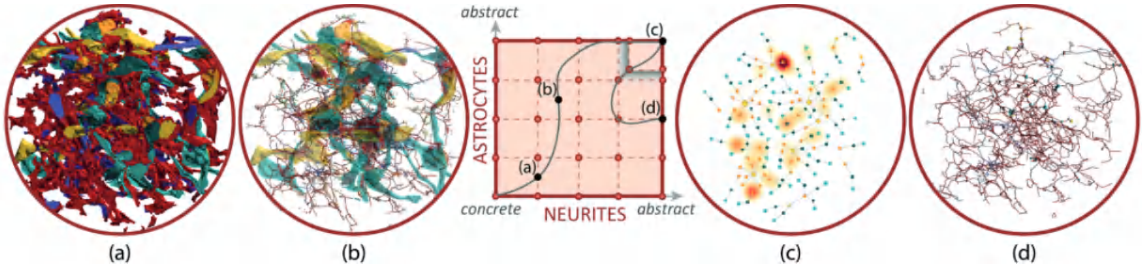


Figure 2.5: Abstractocyte [20] visualizes neurons and astrocytes with different levels of abstraction from left to right.

cal data [2] within single-user VR environments. To summarize, although prior work has proposed various techniques for scale transitions and multi-scale visualization in immersive environments, the problem remains underexplored—particularly for large, unstructured spatial datasets where users must maintain spatial, orientation, and scale awareness while navigating across multiple orders of magnitude. In this thesis, we introduce two interface designs that support multi-scale data exploration in immersive settings. The first, L-WiM, extends the classic World-in-Miniature concept into a linked representation, enabling users to maintain awareness of scale, position, orientation, and nearby structures through linked miniature views. The second approach adopts an overview+detail design. It includes four layouts that integrate global context with local structure, providing a complementary viewing strategy for multi-scale exploration of complex spatial datasets.

Besides presenting multi-scale patterns, another aspect we focus on in this thesis is how to present and integrate multiple visual representations in immersive environments. When exploring spatial datasets, experts frequently need to examine different forms of representation and switch between or combine them to form a complete understanding of the data. For instance, Abstractocyte [20] (shown in Fig. 2.5) enables users to transition between the detailed 3D structure of neurites and abstract 2D node–link diagrams; the two visualizations share a common data foundation and replace one another in the main view, offering complementary perspectives on neuronal morphology. Mota et al. 's 3De Lens [21] introduces a unified focus+context technique that combines volumetric 3D lenses with decal-style surface lenses, enabling users in VR to selectively reveal, inspect, and manipulate multiple geometric representations—such as surfaces and streamlines—within an integrated visualization (shown in Fig. 2.6). Lee et al. [22] proposed a comprehensive design space that characterizes the possible states of data visualizations in mixed reality and

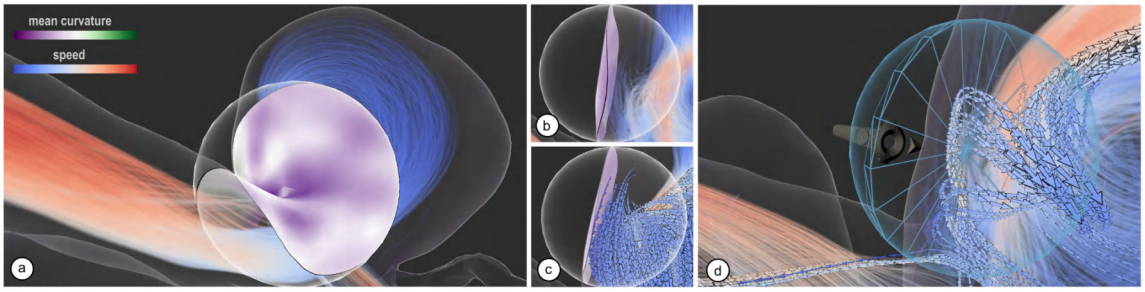


Figure 2.6: The illustration [21] shows a VR focus+context lens that integrates surface and streamline representations to support interactive exploration of aneurysm vessel geometry and blood-flow patterns.

the transformations that enable fluid transitions between 2D surface-based and 3D spatial representations. Coffey et al. [2] introduced Slice WIM, a hybrid tabletop-VR interface that enables interactive exploration of medical volumes by combining a floating miniature 3D view with touch-driven slice navigation on a 2D surface (shown in Fig. 1.1). This work demonstrated how tightly coupled 2D and 3D views can help users navigate volumetric data while maintaining spatial context, and it provides an important inspiration for later hybrid visualization systems. Building on this line of work, this thesis further investigates how 2D and 3D representations can be coordinated in a cross-reality setting, where a conventional 2D display and an AR workspace are used together for spatial data exploration. Our goal is to support flexible placement, transition, and interaction across physical screens and situated 3D space. This setting raises design questions about how different representations can be hosted across devices, how users can move visualizations between 2D and 3D contexts, and how spatial relationships can be preserved during such transitions. To explore these questions, we proposes a CR environment (**Chapter 6**) that combines traditional 2D displays with an AR workspace to jointly support the visualization and exploration of spatial data. Our environment unifies 2D display and 3D immersive environments into a single cross-reality workspace, allowing spatial data to be visualized and interacted with at any location across the reality–virtual continuum while ensuring that users always maintain a correct understanding of the spatial data's underlying three-dimensional structure.

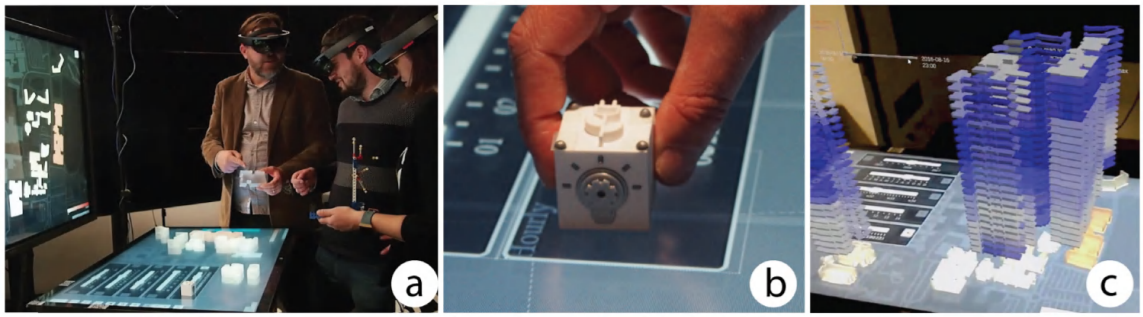


Figure 2.7: Uplift [31] integrates (a) a shared tabletop display for group discussion, (b) tangible widgets for intuitive parameter control, and (c) AR visualizations that extend data into 3D space above the table, together supporting casual collaborative visual analytics.

2.2.2 Spatial Interaction Techniques

Interactive data visualization [23] and embodied user interaction [24] are essential for enabling experts to engage with complex spatial data in the scientific domain, allowing gradually understand the data through direct manipulation and experiential learning. Previous surveys [25, 26] have categorized 3D spatial interaction based on tasks such as selection, manipulation, navigation, and system control. Recent taxonomy proposed by Besançon et al. [27] classified spatial interaction paradigms into the tactile interface, mid-air gestures, tangible interaction, and hybrid interaction.

Tactile interaction. Tactile interaction is well-suited for direct interaction scenarios [28], allowing users to place their fingers directly on the 2D or 3D representation of the data they want to manipulate. This provides a sense of immersion and enables intuitive data manipulation. For instance, FI3D [29] enables direct tactile interaction for exploring data in three-dimensional visualization environments. Recent projection-based AR visualization systems [30] have shown that combining stereoscopic 2D displays with precise screen-based interactions can significantly improve depth perception and support accurate exploration of complex 3D structures. However, tactile interaction can present challenges when the finger size is larger than the content displayed on the relatively small touchscreen, making it challenging to interact accurately with the intended elements.

Tangible interaction. Tangible interaction enables users to perform intricate 3D manipulations using intuitive gestures inspired by the real world [32]. For instance, Uplift [31] (shown in Fig. 2.7) demonstrates the effectiveness of tangible interaction in collaborative

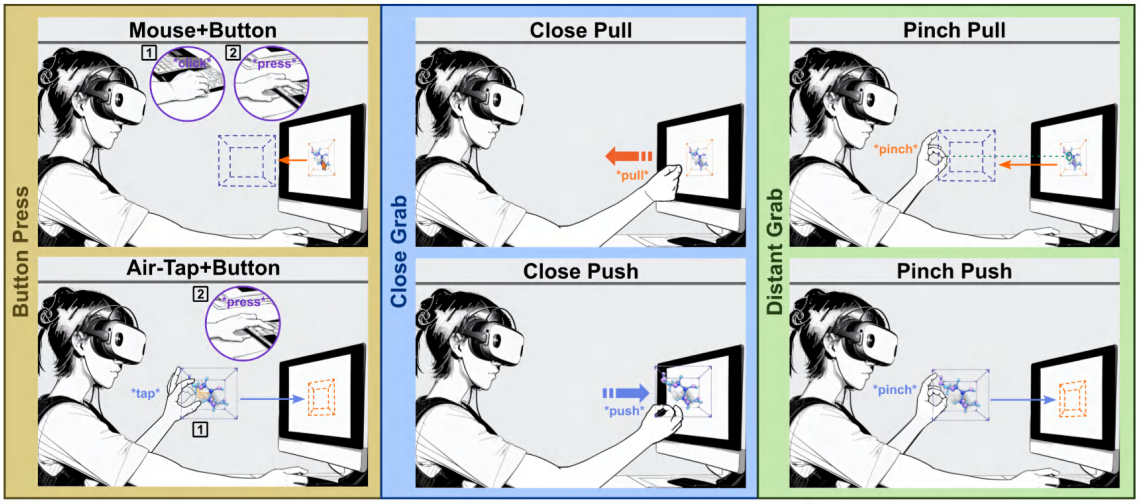


Figure 2.8: A set of techniques for transitioning 3D objects between desktop and augmented reality environments [37].

visual analytics by using physical widgets on a shared tabletop to let stakeholders intuitively manipulate temporal parameters and explore complex data. Embodied Axes [33] is designed to facilitate spatial data selection in augmented reality for 3D images and data visualizations. This innovative device acts as a tangible representation of the 3D data space, where its three orthogonal arms correspond to distinct data axes or specific domain reference frames. Jackson et al. [34] presented a tangible 3D interface that enables the selection of fibers with a matching orientation to the tangible prop. Sereno et al. [35] proposed a hybrid interface that leverage tangible touch tablets to perform spatial 3D selection tasks within the AR environment. Overall, tangible interaction, as demonstrated by studies such as Besançon et al. [36], effectively mimics the familiar interactions we engage in with the physical world on a daily basis. This inherent replication of real-world interactions grants tangible interaction a greater degree of flexibility when compared to other interaction paradigms.

Mid-air gestures. Similar to tangible interaction, mid-air gesture interaction replicates the physical actions we perform in the real world [38]. This paradigm focuses solely on input performed in the air without holding an object. Such gestures can be tracked using wearable technologies such as gloves or optical methods. For instance, Rau et al. [37] (shown in Fig. 2.8) designed three canonical transition techniques centered on mid-air gestures to

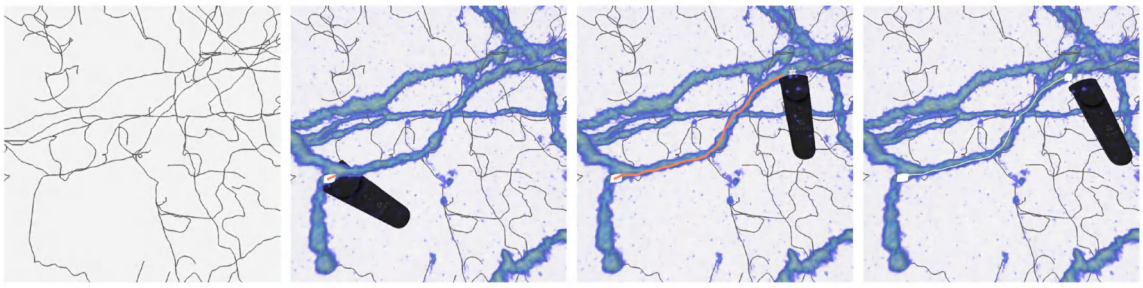


Figure 2.9: An illustration of the neuron tracing tool [43]. From left to right: a connected graph of neurons is reconstructed; users trace a neuron by drawing a path with the controller; and they can choose to add the traced path to the graph.

move and manipulate 3D objects between desktop and AR environments, thereby supporting seamless workflows across dual realities. There are also contributions to improving the accuracy of gesture tracking and the precision of control. Frees et al. [38] introduced PRISM, a technique that dynamically adjusts the control–display ratio to compensate for hand instability in immersive environments, enabling more accurate object translation, rotation, and selection while preserving fast, unconstrained movement. Hinckley et al. [39] proposed a set of one-handed and two-handed 3D manipulation techniques for immersive VR, showing through comparative evaluation that two-handed control enables more precise and efficient position and viewpoint adjustments than one-handed methods, while combining both does not necessarily yield further usability benefits.

Hybrid interaction. amalgamates the advantages of multiple spatial interaction paradigms, for instance, mid-air gesture and tangible interaction [33], tactile interaction and tangible interaction [40, 41], mid-air gesture interaction and tactile interaction [42], etc. Hybrid interaction, by combining the strengths of different interaction paradigms, offers significant potential in handling complex data and exploration tasks. However, it requires more thoughtful design considerations. For example, when dealing with a complex exploration task, careful consideration must be given to how to combine different interaction paradigms and determine the appropriate timing for employing each interaction paradigm.

A major challenge for interaction in immersive environments is the intrinsic imprecision of mid-air input, which results from the absence of physical support. This limitation has been well documented in previous literature. McDonald et al. [43], for example, found that it is difficult for users to trace a well-aligned path along filament-like neurons using VR controllers.

Arora et al. [44] also reported that drawing mid-air curves that lie precisely on the surface of a virtual 3D object in AR/VR is inherently difficult. Yao et al. [45] systematically examined how different levels of embodiment affect asymmetric two-handed interactions in VR and showed that varying embodiment conditions significantly influence users' input precision during immersive data exploration. The input imprecision makes it challenging to interpret user intent directly from raw mid-air input, especially when users attempt fine-grained operations on complex spatial structures. As a result, semi-automatic algorithms become essential for assisting users in performing precise or intent-aligned actions. To address this issue, McDonald et al. [43] developed a semi-automatic method that leverages topological features to guide users during neuron tracing (shown in Fig. 2.9). There are also approaches that leverage traditional 2D displays to perform precise input in the immersive environment. For example, DesignAR [46] combines a high-precision pen-and-touch interactive surface with stereoscopic AR to form a unified 2D–3D design workspace, enabling precise sketching, manipulation, and placement of 3D models beyond screen boundaries through coordinated surface input and spatial feedback (shown in Fig. 2.10).

In this thesis, we use data selection as a representative scenario for studying precise interaction in immersive environments (**Chapter 3**). Data selection is a fundamental step in data exploration and has been studied for decades [47, 27]. In spatial datasets, selection often requires users to specify regions of interest in 3D space based on point locations, local structures, density distributions, or other data-derived features. A wide range of techniques has been proposed for spatial selection in 3D environments. These include raycasting techniques for object or point selection [48, 49, 50], volume-based techniques that use predefined geometric shapes [51, 52], freeform lasso techniques that specify regions through strokes or sketches [53, 54, 55]. Other approaches further combine tangible devices, touch surfaces, or hybrid interaction setups to improve users' control over 3D selection volumes [56, 33]. However, precise spatial selection remains challenging in immersive environments, especially when the target regions are small, occluded, embedded in dense backgrounds, or do not correspond to explicit object boundaries. Prior work has shown the potential of incorporating user intention, local density, and contextual data features into selection processes [57, 58, 59]. Building on this direction, we develop target- and context-aware selection techniques that leverage users' intentions together with the structural properties of spatial datasets to enable precise and meaningful operations in immersive environments. By interpreting mid-air input in relation to spatial context, density patterns,

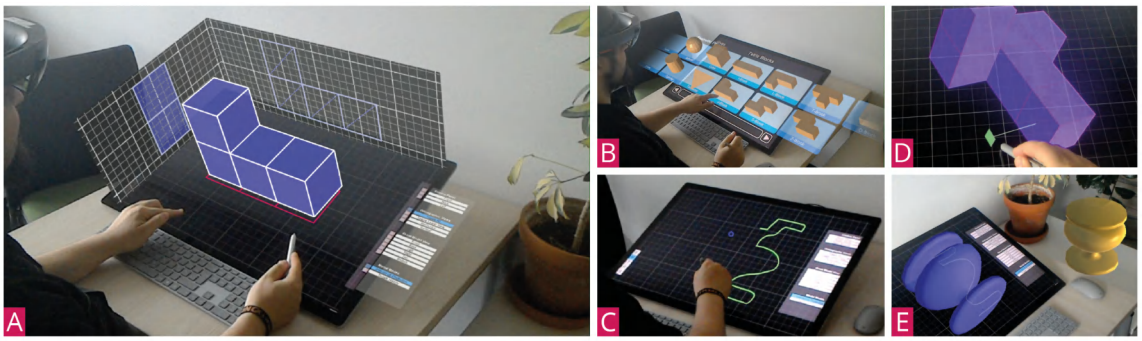


Figure 2.10: An illustration of a general overview of DesignAR [46], showing (a) the augmented workstation setup, (b) the AR object browser, (c) contour sketching, (d) surface construction for modeling, and (e) creating rotational solids and AR instances.

and salient features within the data, our techniques mitigate the inherent imprecision of 6DOF interaction and allow users to effectively explore complex structures.

2.2.3 High-Performance Computing

High-performance computing (HPC) has become a foundational technology in both scientific research and industrial applications. It leverages powerful, large-scale computing systems to perform complex numerical simulations, computationally intensive tasks, and massive data processing workloads. In recent years, rapid advances in parallel architectures, accelerator technologies, and programming models have significantly expanded the capabilities of HPC, driving breakthroughs across numerous domains, including molecular biology [60], astrophysics [61], climate modeling [62] and artificial intelligence [63] and other interdisciplinary sciences.

HPC plays a crucial role in enabling interactive visualization of large-scale spatial datasets. Recent advances in high-resolution imaging, volume acquisition, and large-scale simulation have dramatically increased data sizes. For example, high-throughput electron microscopy can acquire brain-tissue volumes at rates exceeding 10–40 megapixels per second [64], producing nearly a terabyte of raw data per day. While the Square Kilometer Array (SKA) [65] is expected to generate raw data streams of roughly 2 PB/s, reduced to about 600 PB per year even after hierarchical beam-forming. These extreme data volumes far exceed the memory and computational limits of conventional rendering pipelines, presenting challenges for interactive processing, manipulation, and visualization.

To sustain real-time responsiveness, visualization systems must efficiently manage massive data movement, execute computation-intensive operations, and maintain high-throughput rendering under the strict latency constraints (typically$>$60 FPS) imposed by immersive environments. As a result, HPC techniques—such as GPU parallelism, multi-threaded data streaming, out-of-core data management, and multi-resolution representations—have become essential for supporting interactive exploration of large, complex spatial datasets. These strategies constitute the computational backbone required for interactive large-scale visualization. Prior work has further shown that HPC-driven designs make it possible to achieve real-time performance even when interacting with extremely large and complex spatial datasets. Usher et al. [66] developed a VR-based neuron tracing system that achieves real-time exploration of terabyte-scale microscopy data through a tightly budgeted 90 FPS GPU pipeline, multi-threaded asynchronous streaming, and a two-level CPU-GPU caching architecture using sparse 3D textures and PBO-based parallel uploads. FiberClay [67] uses GP-GPU parallelism to render thousands of 3D trajectories simultaneously in immersive environments, enabling interactive selection and comparison of large trajectory sets. Mota et al. [21] further demonstrated that VR lens techniques depend on optimized GPU pipelines to support heterogeneous geometric primitives and update focus+context visualizations in real time. VR-GS [68] introduces a real-time, physics-aware Gaussian Splatting framework for VR, combining a highly efficient two-level embedding strategy with deformable-body simulation to enable realistic dynamic responses and intuitive 3D content manipulation. VR-Doh [69] enables intuitive sculpting and editing of elastoplastic objects in VR by combining a real-time material point method simulation with optimized Gaussian Splatting rendering, allowing natural contact- and gesture-based interactions through localized computation and particle-level collision handling . Zhang et al. [70] propose a perception-driven NeRF representation that reduces neural rendering workloads according to gaze-dependent acuity, yielding up to orders of magnitude latency reduction and achieving high-FOV, stereo VR view synthesis in real time.

Beyond rendering, computational support is also required when derived data representations must be updated during interaction. KDE [71, 72] is widely used to transform discrete point samples into continuous density fields, which can reveal clusters, density variations, and structural patterns in spatial datasets. Such fields have been used for feature detection [73, 74, 75, 76] and for density- or context-aware spatial selection [58, 59, 77]. However, adaptive KDE is computationally expensive because the bandwidth varies with

local point density [78, 79]. As a result, prior density-based interaction techniques often rely on precomputed density fields, which limits their ability to respond to changes in scale, viewpoint, or region of interest during immersive exploration. This issue directly motivates the studies presented in this thesis. In **Chapter 3**, we use scalar-field representations to make spatial selection more target- and context-aware. In **Chapter 4**, we further support multiscale exploration by dynamically recomputing density fields as users move between global context and local detail. These studies motivate our use of GPU acceleration and parallel spatial queries to support on-the-fly density estimation during interaction, instead of relying solely on precomputed single-scale or multi-resolution fields.

2.2.4 Summary

In summary, this section review prior work across three key areas that underpin immersive spatial data exploration: immersive visualization, spatial interaction techniques, and high-performance computing. First, we examined how VR, AR, and MR systems have been used to enhance spatial understanding, engagement, and awareness, and how existing approaches address challenges such as multi-scale visualization and transitions across heterogeneous representations. Second, we reviewed spatial interaction paradigms ranging from tactile and tangible interfaces to mid-air gestures and hybrid techniques, highlighting both their expressive potential and the inherent limitations of mid-air input, especially for precise operations on complex 3D structures. Third, we discussed the critical role of high-performance computing in supporting real-time interaction with large-scale spatial datasets, emphasizing how GPU parallelism, multi-threaded streaming, out-of-core data management, and optimized rendering pipelines enable responsive visualization in immersive environments.

Across these three domains, the challenges of multi-scale visualization, precise spatial interaction, and high-performance data processing represent fundamental barriers for immersive spatial data analysis. Prior work has proposed promising techniques in each area, underscoring the importance of solving these challenges. However, when applied to complex, large-scale, and inherently multi-scale spatial datasets, these challenges remain not fully addressed. The following four chapters are organized around the four challenge-driven research questions introduced in Sec. 1.4. Each chapter begins by stating its connection to the research question, followed by the project that addresses it.

CHAPTER 3

MeTACAST: Target- and Context-aware Spatial Selection in VR

In the first project, we address **RQ1: How can immersive environments support precise, fine-grained spatial data exploration despite the inherent imprecision of 6DOF input modalities?** Among the tasks involved in handling spatial data, we focus on spatial data selection, one of the most fundamental and precision-demanding interactions. Selection requires users to accurately specify regions of interest that reflect their analytical intent, often at fine spatial granularity and within complex 3D structures. However, achieving such precision with 6DOF controllers is inherently challenging due to limited hand stability and the absence of physical support in mid-air operations. Thus, spatial selection provides an ideal task for examining whether interaction techniques can enable precise, fine-grained operations even when user input is inherently imprecise in the immersive environment.

In this chapter, we develop three target- and context-aware selection techniques for particle-based spatial data in VR. Each technique is tailored to a specific selection inten—dense regions, filament-like structures, and spatial clusters—and supports flexible point- or path-based input as well as interactive threshold refinement. These methods allow users to select complex 3D structures accurately, even under occlusion, heterogeneous density, and irregular geometry. We evaluate techniques in a controlled user study against a region-based 3D painting baseline. Results show our methods support more accurate and intent-aligned selections across diverse scenarios. We further synthesize design guidelines for choosing spatial selection strategies according to data features and interaction needs.

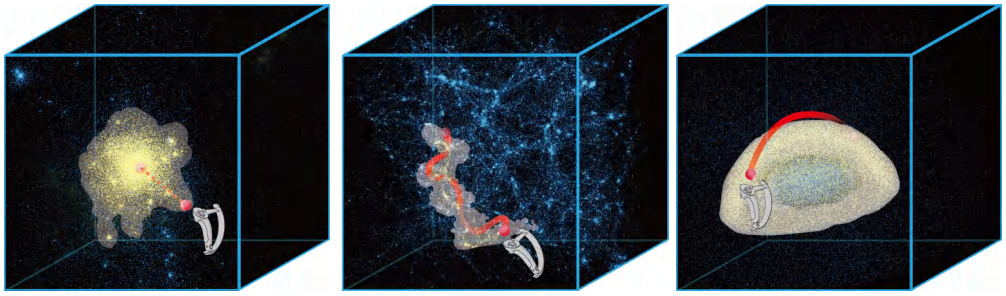


Figure 3.1: MeTACAST: Target- and context-aware spatial selection techniques in VR.

3.1 Introduction

Immersive environments experienced via virtual, augmented, and mixed reality HMDs are increasingly receiving attention from visualization researchers who aim to improve ways of understanding and exploring complex data. HMD-based immersive visualization is now an important research field with applications in natural sciences, in contexts that require users' comprehension and exploration of three-dimensional spatial data.

Likewise and due to the rapid development of scientific simulation and computing technologies, the size, scale, and complexity of datasets in the natural sciences have grown exceptionally. This development substantially increases the challenges of spatial data exploration in immersive environments. Astronomical datasets today, e. g., usually consist of multiple billions of spatial points. According to the cold dark matter paradigm [5, 80], cosmological simulations predict that a cosmic web is formed of the matter in the universe and that filaments transport matter to dense centers of clusters (formed by galaxies). To comprehend such vast amounts of spatial data, researchers need to *observe* the 3D space to gain an overview of the data, *zoom* in and out of the data space to find a clear view of the structures, and *select* and *explore* important subsets of clusters such as those linked by cosmic filaments (Shneiderman's mantra [81]). Dealing with multiple billions of data points, however, is a significant challenge in exploratory visualization on a 2D interface, for two reasons. First, a 2D planar surface cannot portray well the 3D shape or spatial positions of the points without constant animation. Second, it may be difficult for scientists to identify a specific target, define a precise location, and select particular parts with 2D input. These challenges motivate us to enhance the users' ability to explore massive 3D point cloud datasets in immersive contexts.

Data selection is a fundamental and prerequisite step of data exploration and has been explored for many years [47, 27]. Spatial selection, by selecting sub-regions in point cloud data in 3D space, is often based on the locations, structures, and density distribution of the 3D points. The goal of our paper is to *investigate effective techniques for selecting large-scale point cloud data in a VR environment*. In contrast to interactions on 2D monitor screens, immersive 3D spatial interactions are not limited by a static and planar screen for selecting objects—so interactive selection has the potential to be more expressive. In an immersive 3D space, however, users have not only the ability but also the need for six degrees of freedom for a single point of input, leading to substantial differences in their spatial interaction and selection strategies compared to those on 2D monitor screens. Nonetheless, we can take inspiration from 2D selection strategies, in particular those that use a structure- or context-aware approach [58, 59] as this technique allows us to limit the input complexity and adjust it to people's mental abilities and expectations. These strategies then have to be adapted, in particular, to unstructured point cloud data that contain vast amounts of detail distributed across multiple levels with unknown spatial distributions.

To understand these constraints and ultimately develop a suitable approach, we conducted a think-aloud user study to observe users' perception, interpretation, and actions when they were asked to select point cloud data in VR. The study was conducted with local university students, as we were specifically interested in how selection strategies might differ when users were asked to select parts of unknown and unstructured point cloud data. Based on the findings, we developed MeTAPoint, MeTABrush, MeTAPaint, three context- and target-aware selection techniques for point cloud data rendered in an immersive VR environment (shown in Fig. 3.1). All three methods analyze the density distribution of the local area where users interact and as such, users can select the crucial features without precise input. MeTAPoint selects the cluster that is closest to the location where the user is pointing, and then changes its selections as the user continues to provide input. MeTABrush examines the user's input stroke and uses this information to infer their intention by identifying the primary filament that is closest to the path. Finally, MeTAPaint selects targets based on the user's perception of the structure and how well they can distinguish it from the surrounding context. To achieve real-time feedback, we compute the density distribution and generate the selection volume on the GPU. Thus, immediate results are shown, and based on these, users can further adjust the density threshold. We then conducted a controlled experiment to test the efficiency, accuracy, and user preferences

for our three selection techniques in an immersive environment, which demonstrated their effectiveness and suitability for different data scenarios.

In summary, we make the following four main contributions:

- a study on users' behavior in selecting point cloud data in an immersive environment;
- a toolbox of three context-aware and target-aware selection techniques that allow users to precisely select regions of space they intend to further explore with simple and approximate actions;
- techniques to compute density fields and generate selection volumes on the GPU to facilitate fast exploratory visualization; and
- guidelines for designing and selecting appropriate 3D spatial selection techniques in diverse environments and contexts.

3.2 Related Work

Several methods exist for selecting data points in visualizations, which differ based on the data type, user requirements, and the environment and platform used for data visualization. Our primary focus in this work is spatial selection, which entails defining a ROI in 3D space to specify a 3D subspace. We are specifically interested in selections in unstructured datasets that do not feature explicit objects (e. g., particle or volumetric data) based on their unique properties (e. g., density, spatial distribution, or given scalar fields). Below we review relevant work on spatial selection in general as well as on techniques specifically designed for immersive environments.

3.2.1 Spatial Selection Techniques

Raycasting techniques are commonly used for selecting single objects in 3D space (in both immersive and projected settings) and are found in many applications [48, 49, 50, 82, 83]. The concept relies on 3D pointing: the user selects a specific pre-defined shape or object in 3D space by pointing at it, and a 3D ray is cast from the pointer to the object. The target object is identified by finding the first shape or object that intersects the ray. Adjusted techniques for selection in dense environments exist as well [84]. Raycasting techniques can operate at a distance, are fast, and are easy to understand. Precise input, however, is critical for selecting the correct object, in particular when dealing with small objects that

are possibly occluded. To address this occlusion issue, Kopper et al. [85] designed a progressive refinement technique to reduce the required precision of the selection task. Many 3D selection approaches have been proposed to provide precise input for selection tasks, such as 3D picking [86] and 3D lasso [53]. The use of raycasting can also facilitate the identification of a point of interest (POI) in spatial data, such as in scalar fields. Wiebel et al. [87], e. g., devised WYSIWYP to analyze the scalar values along the selection ray to assist users in pinpointing a POI in medical volume data.

Raycasting approaches, however, are often not ideal for handling complex data distributions such as a 3D space with millions of particles. In such scenarios, selecting a single particle is often not meaningful because users are more interested in particle groups than in single data items. Selection of multiple targets or regions of interest (ROI) is often accomplished using various volume-based selection techniques that involve the use of pre-defined basic geometric shapes. These shapes can easily be manipulated in three-dimensional space, such as the use of cones to select multiple data points [51] or the use of cubes to select brain fibers [52]. Nonetheless, these techniques' ability to select target objects far from a pre-defined shape is restricted. Therefore, freeform lasso methods are proposed to offer more flexible input to select objects or ROI in 3D space based on the 2D lasso or stroke. Lazy Selection [55], e. g., allows users to quickly select one or more desired shape elements with a novel scribble-based tool by roughly sketching through them. The Tablet Freehand Lasso (Cylinder Selection) [88] and Volume Catcher [54] allow users to flexibly select an ROI in 3D space based on a 2D lasso or a stroke. Besançon et al. [56] proposed a hybrid approach that emphasizes the level of user control in defining the selection volume in 3D space. They introduced Tangible Brush, a technique that provides manual control over the final selection volume and which combines 2D touch lasso input with 6-DOF 3D tangible brushing to allow users to perform 3D selections in volumetric data.

Compared to the use of pre-defined shapes, freeform lasso selection offers greater adaptability and flexibility, in particular when it comes to manual control over the final selection volume in 3D space. This approach allows users to customize their ROIs based on the local data characteristics—which is why we also target a freeform technique. These fully manual techniques, however, may not be efficient for complex selection tasks such as identifying small, dense, or complex clusters in noisy backgrounds that may be difficult to visually recognize. Several approaches have been proposed to address this limitation. Chen et al. proposed LassoNet [57], a deep learning-based approach to lasso selection

for 3D point clouds. It aims to learn a latent mapping from viewpoint and lasso to point cloud regions, enabling users to select particles that best match their selection intention. Structure- and context-aware selection techniques [58, 59] have also been designed to take data features such as the local density into account. With these techniques, users draw a 2D lasso around the target, and only clusters with densities above a certain threshold are selected. While context-aware approaches have been effective, they are not designed for immersive environments—which we target with our paper—, where users do not have a dedicated 2D surface for drawing the lasso.

3.2.2 Spatial Selection in Immersive Environments

In immersive visualization settings, researchers can gain a clearer understanding of the data, not only through interaction and exploration but also simply through the stereoscopic projection. Studies have also demonstrated [89] that users can perform better in cluster identification tasks when using immersive scatterplot visualizations compared to traditional 3D-to-2D projected representations.

Also in immersive VR, raycasting is a common technique to help determine where a user is looking or pointing at. It is frequently used for various interactions such as object selection, manipulation, and action triggering. Hurter et al. [67] introduced FiberClay, a raycasting-based approach to brush paths using two VR controllers to select paths that connect both targeted points. Selecting multiple objects in VR, however, can be challenging and imprecise, especially when the objects are small, moving, occluded, or complex. To improve interaction precision, researchers have proposed various approaches. De Haan et al. [90] developed IntenSelect to aid users in selecting objects in motion within cluttered and occluded VR environments with a scoring function. Baloup et al. [91] introduced a controllable cursor positioned on top of a ray, allowing users to select nearby or occluded objects. Maslych et al. [92] showed how to select fully occluded objects in dense environments. Their two approaches maintain the spatial relations between minimized versions of the original objects by displaying the objects that collide with a cone-shaped selection volume in a flat or cylindrical mini-map. Wei et al. [93] recently proposed a user intention model for gaze-based selection techniques based on eye and head endpoint distributions to estimate the intended targets. Adaptive pointing [94] also enhances the accuracy of pointing devices. Finally, Stenholt [95] designed a magic wand, a novel multiple-object selection technique based on local proximity, enabling users to select hundred or thousands of objects. We

also aim for precise selection, yet one in which the input imprecision is mediated by context-aware selection processing as well as one in which the user can easily adjust the selection after the initial specification—like in the projected 3D settings we reviewed above.

Compared to interaction on a 2D surface, selecting within immersive VR is conceptually more straightforward and is more preferred by users than using traditional 2D monitors [96], as users can view the data in stereoscopic 3D and have manual control over the final selection. To facilitate 3D input, various techniques have been developed, such as user-defined box virtual tool [97, 98] and cone-based selection volumes [99]. Spatio-Data Coordination [100] and the Embodied Axes [33] proposed by Cordeil et al. enable users to accurately determine the selection region of the spatial data using sliders on the tangible axes. These techniques rely on predefined 3D selection volumes such as boxes, cones, and cylinders to select multiple objects within the space. Several approaches have also been proposed to select within complex datasets using freeform methods. Lubos et al. [101] introduced a bi-manual user interface for selecting point cloud data. Their approach involves tracking the users' hands and enabling them to touch and select a 3D point cloud in an immersive environment. Similarly, Gemoz et al. [102] proposed a contacting method with a spherical input to select neural fiber tracts. In addition, various hybrid approaches have been proposed to improve selection accuracy by using hybrid approaches. They leverage different technologies and techniques to make the selection process more efficient and intuitive for users. Sereno et al. [35], e. g., integrated an AR device with a tablet that acts as a tangible device for 3D selection. Montano-Murillo et al. [103] proposed a hybrid VR selection technique that allows users to place a selection volume at a specified position in the VR environment. The volume is initially attached to a virtual tablet and can be adjusted for size and thickness to better fit the target object. Researchers have also focused on incorporating data features or geometric features into the 3D selection process. For example, Malmberg et al. [104] used surface generation based on a 3D live-Wire drawn by the user to segment and select within volumetric images. This approach enables users to select objects based on their volumetric value as well as geometric properties. Torin et al. [43] introduced a semi-automatic neuron tracing method based on the Morse-Smale complex in an immersive environment. This approach enables users to select the desired region even with inaccurate trace input, reducing fatigue during the exploratory task. Jackson et al. [105] proposed Force Brushes, a haptic-based selection technique for multivariate flow visualization that enables users to snap to streamlines and progressively

filter regions of interest based on underlying data features. Jackson et al. [34], finally, proposed a tangible 3D interface that is capable to select the fibers that own the same orientation as the tangible prop.

All these spatial selection techniques for immersive environments can select single objects or regions based on the data's location and/or distribution. They employ tangible and hybrid interaction techniques to improve the selection accuracy for complex data. In our work, we aim for selection directly within the HMD-based VR environment, without any additional required hardware, and for an interaction design that is as flexible as the traditional context-aware techniques, but without any need for a planar interaction surface. We work toward this goal by studying users' strategies in various situations to be able to incorporate data features and user intention into the selection process.

3.3 Think-aloud Elicitation Study

To gain a deeper understanding of how users adopt selection strategies within VR in various contexts and for various targets, we began by conducting a think-aloud study with six participants from our local university. Here we present our findings and discuss the resulting design considerations related to selection context, target, action, and strategy. The think-aloud study received approval from the University Ethics Committee.

3.3.1 Study Design

Various factors such as the size of the visualization, the interaction environment, the characteristics of the data (e. g., context, data distribution, density), and input precision requirements can influence the selection strategies employed by users. To design effective selection techniques we need to understand the users' selection strategies in VR environments as they deal with these situations. In a think-aloud elicitation study we thus, instead of teaching them any specific selection methods, showed participants datasets and selection targets and first asked them to freely explore the 3D point cloud data. We then let them demonstrate selection strategies with a 3D drawing tool and asked them to explain why they thought it was the most appropriate and direct way to select crucial features or regions. We recorded all user actions and input trajectories for further analysis.

The **size of the visualization** is a crucial factor for users' selection strategies, especially

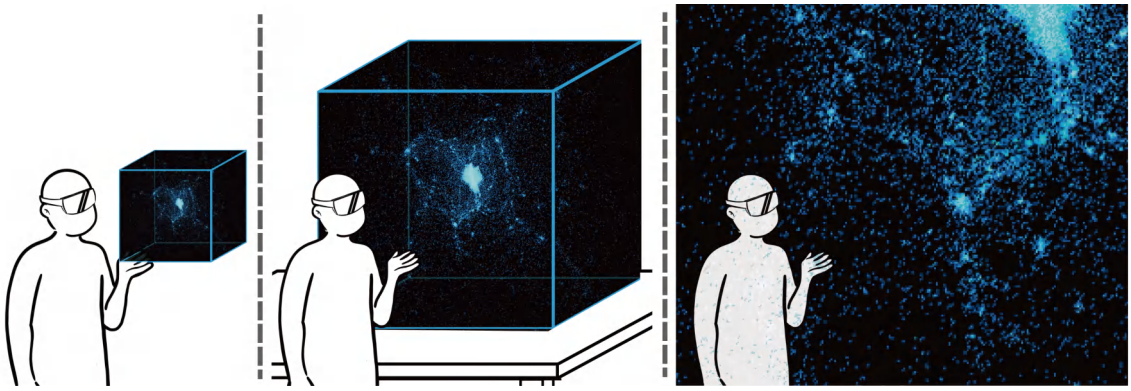


Figure 3.2: Three visualization sizes: (a) hand, (b) table, and (c) room size.

in VR. In the physical world, perspective distortion causes objects located closer to the human eye to appear larger than their actual size. This effect is even more obvious when viewing point clouds in VR due to the narrower field of view. As a result, gaps between 3D points that are closer to the user appear larger than those between points further away. This effect can make it challenging for users to accurately estimate the density distribution and, thus, to identify a dense cluster when they are "within" it. Kraus et al. [89] found cluster identification task performance differs between different visualizations (monitor screen for 2D/3D scatter plot, table-size VR, and room-size VR) in terms of accuracy, efficiency, memorability, sense of orientation, and user preference. To answer the question about whether users could find the entire cluster and further investigate how visualization size would impact the users' selection strategies, following Danyluk et al. [106] we investigated three different sizes (shown in Fig. 3.2): hand-size, table-size, and room-size:

***Hand-size**:* We adopted a hand-size scale of 30 cm for the data visualization—a suitable size to observe the detailed structures clearly in most cases. The data is controlled by the left and, its center is attached to the left controller with a 35 cm offset to prevent the left hand from obstructing interactions with the right hand in the data.

***Table-size**:* We placed the data statically 1 m above the floor with a base length of 64 cm; users can walk around and reach inside.

***Room-size**:* In this setting we also do not allow users to manipulate the data placement and for the selection, they have to rely on the picking/raycasting metaphor to reach far-away data locations.

We used four distinct **datasets** (shown in Fig. 3.3). We rendered the target particles

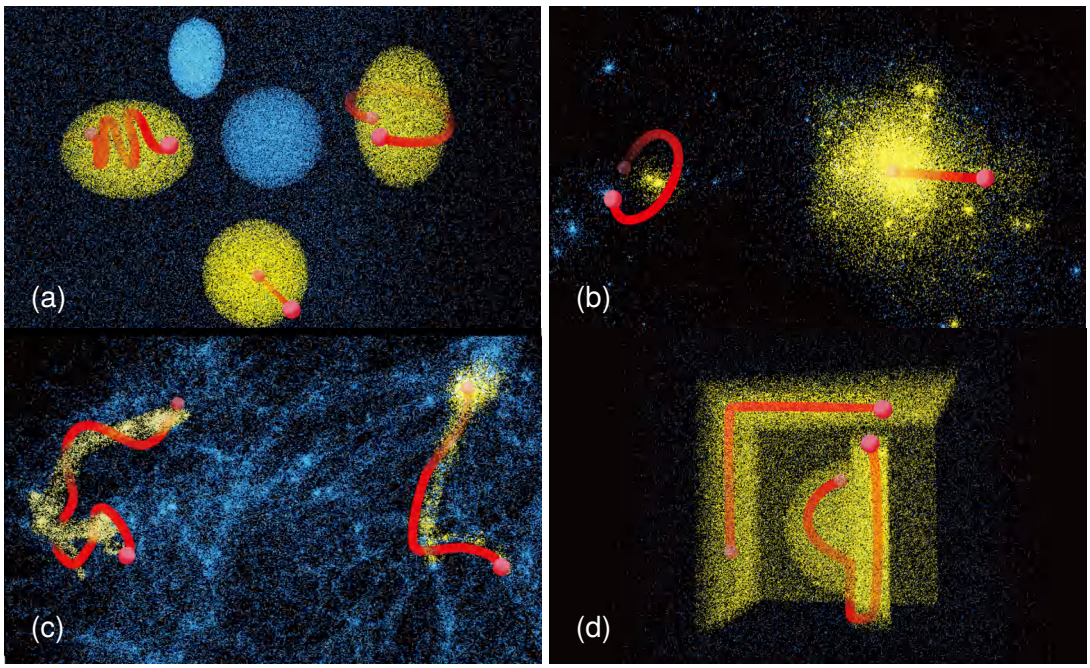


Figure 3.3: User input trajectories in selecting four point cloud datasets employed in the think-aloud study: (a) *Clusters*; (b) *N-body Simulation*; (c) *Filament*; (d) *Complex geometries*.

in yellow, while we presented the interfering particles in blue. We selected each dataset based on its unique features:

***Clusters**:* a synthetic dataset generated for the study, containing five uniformly dense ellipsoidal clusters in a noisy setting.

***N-body simulation**:* a cosmological N-body simulation that comprises a huge, extremely dense cluster at the center, which is surrounded by multiple smaller clusters (from [107]).

***Filament**:* a cosmic web simulation with thin filaments (from [108]).

***Complex geometries**:* a synthetic dataset, containing an empty half-box outside and a half-ball combined with a cuboid of particles located inside the half-box.

3.3.2 Findings

Preferred visualization size. All participants preferred hand-size and table-size visualization settings that allowed them to select data directly. Moreover, these settings also facilitated easier recognition of point cloud clusters and their boundaries by the participants. Thus, in the following, we focus on analyzing users' actions and selection strategies with hand- and table-size environments.

Target: Dedicated cluster. When we asked participants to select clusters that are visually clearly separable from the background (e. g., a separate ball-shape cluster or small, isolated clusters), most pointed at the target clusters directly and hoped that the target clusters could be selected with minimal actions. This observation aligns well with traditional 3D selection, which relies on picking and raycasting.

Target: Regular, simple shape. When the shape of the target particles is simple, many participants attempted to draw lassos in the air that followed the geometric features of the target, while a minority also mentioned that pointing was also possibly applicable if the selection range could be determined in some way.

Target: Filament-like structure. In this case some participants brushed the points following the dominant branch, while others tried to separate the target points from the noisy context through a helix in 3D space. Both strategies seem to be valid approaches.

Context: Unclear boundary. When participants were unable to distinguish the boundary of clusters (e.g., many small clusters surrounded by particle noise), they attempted to enclose the target cluster(s) by drawing a lasso around them as they would on a monitor screen. Most of them realized, however, that drawing a 1D linear lasso to enclose points in 3D space is insufficient. So they expressed the desire for a way to define a 3D region around the input location that would allow them to include the target points.

Context: Occlusion. We found that, for easy shapes, participants were able to estimate the 3D location and shape of target clusters even when the particles were fully or partially occluded. In the *complex geometries* dataset, e. g., participants placed a contact point (the interacting point in 3D) on the opposite side of the target without the need to rotate the data or selected occluded geometries by placing the contact point inside directly. This ability may allow us to design an effective 3D interaction for clear target clusters.

Accuracy. In comparison to screen-based selection, in VR participants were generally able to *mark* the targets *closer* to their actual locations. We observed, however, that

participants often had difficulties accurately clicking on, for instance, the exact position of an intended cluster. When attempting to select data by drawing lines directly in VR using controllers (6DOF input), participants also often failed to align their input precisely with target filaments. Even the input lasso looked as if it would precisely follow the filament. After rotating the data to observe it from another viewpoint, participants saw that the input lasso was at a distance from the target due to VR's imprecise depth perception—an issue that was also noted in related work [43].

3.3.3 Design Considerations

Based on these findings, we propose the following design considerations and requirements for the design of selection techniques in VR.

- *Target-aware selection.* The techniques should be capable of handling various target shapes, such as clusters, filaments, partially occluded structures, and structures intertwined with others.
- *Context-aware selection.* The techniques should be capable of handling various challenging situations, including non-uniform feature density, unclear boundaries, and occlusion. Users should not be limited or hindered by these interfering factors but, instead, should be able to focus on their selection tasks, including identifying the shape, location, and critical features of the data.
- *Accurate and intuitive selection.* The techniques should minimize a user's need to move and, instead, infer their selection intention from the approximate location and path of the input, precisely adjusting the selection to crucial data features such as geometric shape and region density. Users should be able to use simple, natural, and approximate input to select intended regions precisely in VR.
- *Exploratory selection and immediate results.* When dealing with a complex dataset that contains a large number of noisy points and unclear boundaries, it is crucial for users to visually examine the selection results and receive them without delay. This is important, e. g., in astronomical data exploration where numerous unknown features are concealed within the data and noise is prevalent.
- *Partial selection and threshold adjustment.* Enabling post-selection threshold adjustment is important for our selection tool to be flexible in handling various situations such as selecting complex data shapes, partial regions, and non-homogeneous density.

3.4 Selection Techniques

Based on these design considerations, we developed the MeTACAST family for selection in HMD-based VR environments, which consists of three spatial techniques, each tailored to a specific selection intent.

3.4.1 Interaction Metaphor

Our think-aloud study showed that participants preferred visualization tools that rely on direct interaction with data. We thus adopted the Worlds in Miniature (WIM) metaphor [16, 109, 110] and developed a hand-held miniature of the 3D point cloud data to facilitate data selection in a virtual environment. We attached the WIM to the VR controller of the non-dominant hand, providing users with an exocentric view of the 3D data and facilitating correct density comprehension of the point clouds. To select points, the user can use their dominant hand to control a red sphere. We positioned the sphere 1 cm above the top of the VR controller to prevent collisions between the two VR controllers. This red sphere follows the movement of the controller, allowing users to have full control in 3D space to point, brush, or paint within the data.

Similar to CloudLasso [58] and CAST [59], instead of relying on particle positions we use a continuous density field $\rho(\mathbf{r})$ that represents the particle density at position $\mathbf{r}$ in space. We compute the field at all nodes i of a regular 3D grid (box B) that covers the dataset as $\rho(\mathbf{r}^{(i)})$. We then derive the value $\rho(\mathbf{r})$ of the field at any other point $\mathbf{r}$ in space through linear interpolation from the values $\rho(\mathbf{r}^{(i)})$ at the grid-nodes that are closest to $\mathbf{r}$. This approach enables us to apply our method not only to particles but also to volumetric data that samples a scalar field representing any visually salient data aspect, rather than just density.

Based on this field concept we can now design interaction techniques that are adjusted to our VR environments. In contrast to projection-based approaches such as CloudLasso and CAST, we need to allow users to directly point, brush, or draw on the dataset in the 3D virtual space according to their preferred strategies. We aim to meet the design goals we formulated in Sec. 3.3.3 and developed three techniques—MeTAPoint, MeTABrush, and MeTAPaint—that we describe next.

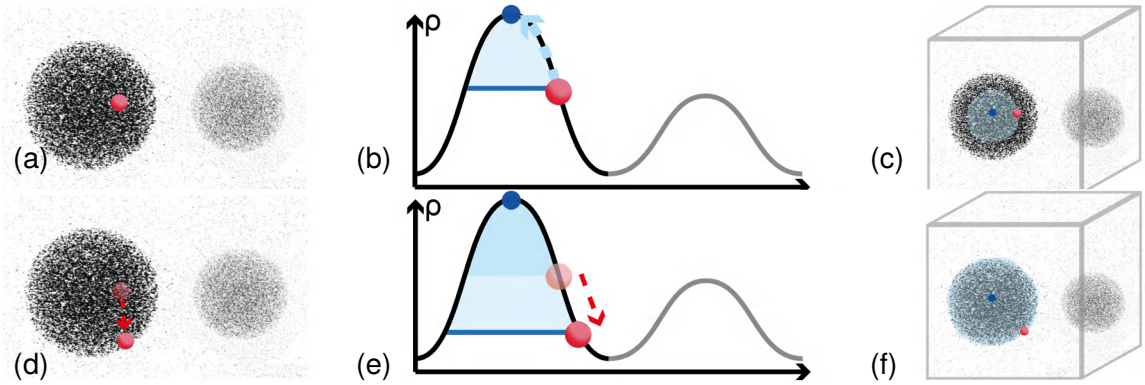


Figure 3.4: MeTAPoint: (a) the user points at the target cluster (red); (b) we derive the closest maximum point (blue) and density threshold (schematic representation); (c) we compute the selection volume; (d) the user drags the controller to adjust the density threshold; (e, f) we recompute the density threshold and selection volume.

3.4.2 MeTAPoint

The problems in VR with general free-hand input imprecision and placement with respect to a 3D structure (Sec. 3.3.2) lead to users wanting to test their initial selection results through interactive exploration to better understand the data distribution. Users attempted to draw a stroke on the target to explore the boundary of the target cluster. To address this need, we propose our first selection technique MeTAPoint. With it, the user begins by pointing at the target cluster, or near it, and the initial selection is based on this input position. To account for imprecision, we attempt to determine the intended target by analyzing the input position. Specifically, we follow the direction of the gradient from the input position $\mathbf{r}^{(s)}$ (red point in Fig. 3.4(a)) to find the local maximum $\mathbf{r}^{(m)}$ of the scalar field (blue point in Fig. 3.4(b)). We use this input adjustment in all MeTACAST techniques and discuss technical details in Sec. 3.4.5.

Next, we determine the selection volume V that surrounds the local maximum $\mathbf{r}^{(m)}$, based on the density field. To enable users to explore the edges of the target cluster, we derive the density threshold based on the density at the initial input position $\rho(\mathbf{r}^{(s)})$. Using the Marching Cubes algorithm [111, 112] we identify all volumes where the density ρ is above $\rho(\mathbf{r}^{(s)})$ in the whole space, and we pick the volume that encloses $\mathbf{r}^{(m)}$ as the initial selection volume V (blue area in Fig. 3.4(b, c)).

The user can continue to explore the target boundary by dragging the controller with

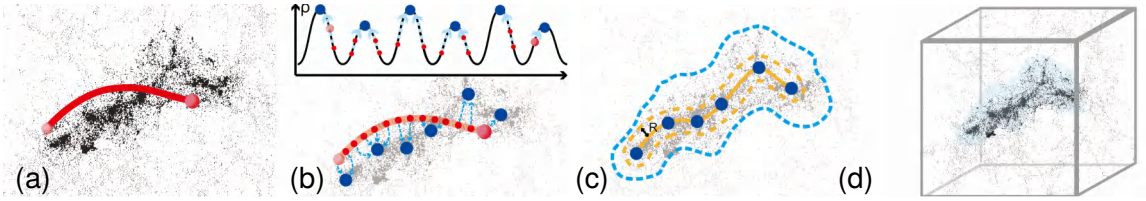


Figure 3.5: MeTABrush: (a) the user draws a 3D stroke (red); (b) the stroke points (red) are extracted; by following the direction of gradient (blue arrow), we identify the local maximum points (blue); (c) we construct a tunnel-like volume (yellow dotted region) based on the MaxLine with the radius R; V_{init} is derived (blue dotted line); (d) the final selection.

the trigger pressed. As they drag the red point, we adjust the density threshold accordingly (Fig. 3.4(d, e)) and compute a new selection volume (Fig. 3.4(f)). It is important to note that we designed MeTAPoint to select the single volume that is closest to $\mathbf{r}^{(s)}$. Therefore, if the user drags the red point close to another cluster, the selected target may switch. In this way, we provide the user with an immersive experience of testing the target volume and selection boundary through continuous interaction in the 3D environment.

To enhance exploratory visualization and accelerate iso-surface computation, we leveraged a GPU-based implementation of Marching Cubes. We assign a dedicated thread to each voxel, resulting in smooth threshold modification and efficient triangle generation.

3.4.3 MeTABrush

A potential issue for MeTAPoint is that it requires users to carefully adjust the contact point—the input position has a direct impact on the selection result, including the target and threshold. Consequently, it may be challenging for MeTAPoint to select a part of a filament in a complex dataset, which is a common task, e. g., in astronomical exploration. In our elicitation study, we observed that users usually want to brush the points following the dominant branch or create a helix in 3D space to separate the target points from the noisy context (Fig. 3.3(c)). While their input may not be precise, their selection intent is clear from the input. To address this need, we propose our second selection technique, MeTABrush, which infers users' intention using the entire input path, adjusts the position of the input stroke, and then provides a precise selection—without being limited by imprecise input, non-homogeneous features, or complex structures.

With MeTABrush, the user can brush a stroke along the target filament particles in

3D space (Fig. 3.5(a)). We then take a sample of points $\mathbf{r}^{(s)} = \{\mathbf{r}^{(s_0)}, \mathbf{r}^{(s_1)}, ...\mathbf{r}^{(s_n)}\}$ on the input stroke and follow the direction of gradient from each point (Fig. 3.5(b)) like before. This process yields a set of destinations, that is, local maximum points $\mathbf{r}^{(m)} = \{\mathbf{r}^{(m_0)}, \mathbf{r}^{(m_1)}, ..., \mathbf{r}^{(m_t)}\}$ of the density field; note that n may not equal t. Next, we connect successive pairs of destinations $\mathbf{r}^{(m_i)}$ and $\mathbf{r}^{(m_{i+1})}$ to obtain the path P (indicated as MaxLine below) as follows:

$$\begin{aligned} &P : x \in \mathbb{R}^+ \mapsto \mathbb{R}^3, P(0) = \mathbf{r}^{(m_i)}, P(end) = \mathbf{r}^{(m_{i+1})}, \\ &P'(x) = \frac{\nabla f\left(P(x)\right)}{\|\nabla f\left(P(x)\right)\|} + 2\frac{(P(m_{i+1}) - P(x))}{\|(P(m_{i+1}) - P(x))\|}, \end{aligned} \tag{3.1}$$

With this process we connect all local maximum points following roughly the direction of the gradient, while also smoothing the MaxLine P. Based on P, we construct a tunnel-like shape T (region circled by the yellow dotted line in Fig. 3.5(c)) that extends along the MaxLine with a pre-defined radius of R (represented by the size of the input marker, which can be adjusted by the user later). We then identify all particles whose destinations, following the direction of the gradient, fall within T, using the algorithm we detail in Sec. 3.4.5. We thus identify the set of the potential target particles J, which we indicate by the blue dotted line in Fig. 3.5(c). Next, for each potential target particle $j \in J$, we consider the ellipsoid with semi-axes $\ell_x^{(j)}$, $\ell_y^{(j)}$, and $\ell_z^{(j)}$, which are the smoothing lengths along x, y, z of the j^{th} particle (Sec. 3.4.6 has more details on smoothing length and density estimation). We further combine these ellipsoids into an initial volume of interest V_{init} in the data box B that covers the whole data:

$$V_{\text{init}} = \{\mathbf{r} \,|\, \mathbf{r} \in B,\, \exists j \in J,\, \|\mathbf{r}^{(r;j)}\| \leq 1\}, \tag{3.2}$$

where $\mathbf{r}_k^{(r;j)} = (\mathbf{r}_k - \mathbf{r}_k^{(j)})/\ell_k^{(j)}$ along the k^{th} direction ($k = x, y, z$), and $\|\mathbf{r}\|$ denotes the Euclidean norm of the vector $\mathbf{r}$.

Next, we calculate the initial density threshold ρ_0 by as the arithmetic mean of the density of all the grid-nodes inside V_{init}. Through Marching Cubes we obtain the iso-surface inside V_{init}. We then obtain the volumes inside of the iso-surface and select the volumes that contain a segment of the MaxLine, which means that the selected volumes should contain at least one local maximum point $\mathbf{r}^{(m_i)}$ on the MaxLine. We regard these volumes as the initial selection volumes V In addition, users are able to adjust the density threshold post-selection

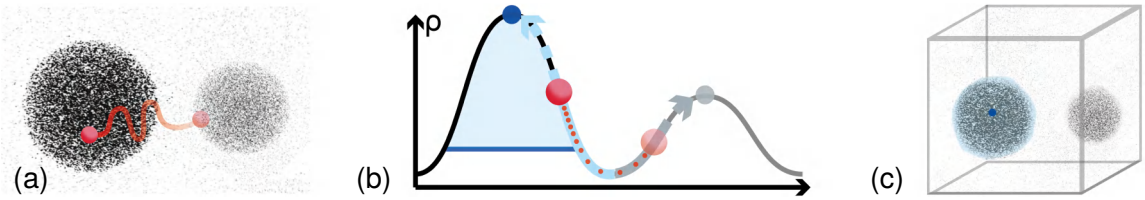


Figure 3.6: MeTAPaint: (a) the user draws a 3D stroke (red) near the black cluster, some parts of the input are located near the gray one; (b) the stroke is split into multiple points (red), which flow towards local density maxima (blue and gray points) along the direction of the gradient; (c) the black cluster is selected since it receives most seeds.

via the VR controller. They can modify the threshold in a range of $[\rho_0/16, 16\rho_0]$, mapping with the function $\rho_s = 2^s\rho_0$ with $s \in [-4, 4]$. When s is adjusted, we thus recompute the scalar quantity for all grid-nodes inside V_{init} with ρ_0 replaced by ρ_s and obtain the iso-surface using Marching Cubes.

3.4.4 MeTAPaint

In our elicitation study, we observed users trying to select regular points shapes. These strategies allowed users to depict the geometric features of their target accurately. The provided input, however, was inherently imprecise. To address this issue, we thus introduce MeTAPaint, a technique that interprets the 3D path of a drawn selection stroke and selects the candidate cluster that best fits the drawn stroke.

With MeTAPaint, the user brushes a target cluster with a 3D stroke (Fig. 3.6(a)). We first identify the initial density threshold ρ_0 by sampling the density of the grid-nodes surrounding the input stroke. We define a tunnel-like region T that extends along the input stroke with a radius of $(\ell_x + \ell_y + \ell_z)/3$, where ℓ_x, ℓ_y, ℓ_z are the smoothing lengths along x, y, z, respectively ([58, 59]). We then calculate the initial threshold ρ_0 as the arithmetic mean of the grid-nodes' density $\mathbf{r}^{(n)}$ within T, given by

$$\rho_0 = \frac{1}{N_T}\sum_{n=1}^{N_T}\rho(\mathbf{r}^{(n)}), \tag{3.3}$$

where N_T is the number of the nodes in T. Then we take a sample of points $\{\mathbf{r}^{(s_0)}, \mathbf{r}^{(s_1)}, ...\mathbf{r}^{(s_n)}\}$ on the input stroke and follow the direction of gradient from each point to obtain the desti-

nations $\{\mathbf{r}^{(m_0)}, \mathbf{r}^{(m_1)}, ..., \mathbf{r}^{(m_t)}\}$ (Fig. 3.6(b)), as we did in MeTABrush. We then count the number of $\mathbf{r}^{(s_i)}$ that contribute to each $\mathbf{r}^{(m_j)}$, and find $\mathbf{r}^{(m_{\max})}$, that is, the destination for the most stroke points $\mathbf{r}^{(s_i)}$. Using Marching Cubes, we obtain the iso-surface with density ρ_0, and the corresponding enclosed volumes. Finally, we regard the volume containing the destination $\mathbf{r}^{(m_{max})}$ as the initial selection volume V (Fig. 3.6(c)). The post-selection threshold adjustment follows the same procedures as we described for MeTABrush. Note that MeTAPaint selects the cluster that is most heavily brushed by the input stroke, meaning that users only need to brush on or around the geometric features of their target, and the intended single cluster gets selected.

3.4.5 Local Maximum Point Extraction

To increase input precision, in all MeTACAST methods, we adjust the positions of the input point/stroke by extracting local maxima. For a smooth density field $\rho : \mathbb{R}^3 \to \mathbb{R}$, a point $\mathbf{r}$ is a local maximum of ρ, if

$$\nabla\rho(\mathbf{r}) = 0 \text{ and } \lambda_1 < 0, \tag{3.4}$$

where λ_1 is the largest among all the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ of the Hessian of ρ at $\mathbf{r}$. To follow the user input point $\mathbf{r}^{(s)}$ along the direction of the gradient of ρ and find the local maximum point $\mathbf{r}^{(m)}$ for the density field, we define the path P as

$$P : t \in \mathbb{R}^+ \mapsto \mathbb{R}^3, \quad P(0) = \mathbf{r}^{(s)}, \quad P'(t) = \nabla\rho(P(t)). \tag{3.5}$$

We take $\mathbf{r}^{(m)}$ as the destination of the path P, that is, $\mathbf{r}^{(m)} = \mathrm{dest}(P) = \lim_{t\to\infty} P(t)$. According to Morse theory [113], for a smooth function with a non-degenerate Hessian matrix, the path P converges to a local maximum except in the case when $\mathbf{r}^{(s)}$ is located on the stable manifold of a saddle or is a minimum of the density field.

Given a position $P(t)$ along the path P, we compute $P(t + \delta t)$ along the direction of $\nabla\rho(P(t))$, which is calculated by linear interpolation from the values of ρ at the closest grid-nodes to $P(t)$, obtained in Sec. 3.4.6. Finally, we obtain the local maximum point $\mathbf{r}^m = \mathrm{dest}(P)$ at the end of the path P. In the case where the user's input is a stroke, we compute the destination points for all the sample points on the stroke with a parallel algorithm to increase computation speed.

3.4.6 Kernel Density Estimation

We used the same density estimation method as Yu et al. [58, 59]. For efficiency reasons we implemented it on the GPU. We define a box B covering the point cloud data and divide the space into a 100 × 100 × 100 grid. For each direction k = x,y,z, we define the smoothing length

$$\ell_k = 2(P_k^{(80)} - P_k^{(20)})/\log N \tag{3.6}$$

where N is the particle count in the box B and $P_j^{(q)}$ is coordinate $k's$ q-th percentile value. For the i^{th} grid-node at position $\mathbf{r}^{(i)}$, we compute the density $\rho(\mathbf{r}^{(i)})$ using the modified Breiman kernel density estimation with a finite-support adaptive Epanechnikov kernel [114], given by a

$$\rho(\mathbf{r}^{(i)}) = \frac{15}{8\pi N} \sum_j \frac{1}{\ell_x^{(j)} \ell_y^{(j)} \ell_z^{(j)}} E(\|\mathbf{r}^{(j;i)}\|), \tag{3.7}$$

with

$$\mathbf{r}_k^{(j;i)} = (\mathbf{r}_k^{(j)} - \mathbf{r}_k^{(i)})/\ell_k^{(j)} \tag{3.8}$$

where $\mathbf{r}^{(j)}$ is the position of the j^{th} particle and $\ell_k^{(j)}$ are the smoothing lengths of the j^{th} particle along the k^{th} direction ($k = x, y, z$) generated from the pilot density calculation [58, 59]. The Epanechnikov kernel $E(x)$ is given by

$$E(x) = \begin{cases} 1 - x^2, & |x| < 1, \\ 0, & |x| \geq 1. \end{cases} \tag{3.9}$$

The density field of the data is pre-computed offline and the selection geometry construction with Marching Cubes is performed on GPU.

3.4.7 System Performance

We have tested the performance of the MeTACAST with Unity on Intel Core™ i9, GeForce RTX3090. All experiments were conducted under identical conditions to ensure fair comparisons across datasets. The evaluation focuses on three major computational components of the system: density estimation, the selection algorithm, and geometry construction using marching cubes. Tab. 3.1 shows the performance measured by execution times. We report the average execution time over 10 runs for all tests.

Table 3.1: MeTACAST performance. Times are in seconds.

Dataset	Particle Size	Density Estimation (offline)	Selection Algorithm	Marching Cubes
Fig. 3.1(left)	76k	0.38	0.06	0.005
Fig. 3.1(mid)	442k	3.47	1.69	0.008
Fig. 3.1(right)	146k	0.80	0.17	0.009

3.5 User study

We conducted a controlled experiment to evaluate the performance of the MeTACAST methods in identifying various target structures with uniform or non-uniform density and shape. We compared our three MeTACAST methods to a conventional, region-based method, to which we refer as Baseline. We directly derived the Baseline method based on Touching the Cloud [101], a technique designed for selecting particles within the region where the user brushes using 3D input in a VR environment. We compared all methods based on the accuracy and time taken for particle selection. We pre-registered the user study plan, our analysis code, and our hypotheses at `osf.io/dvj9n`.The user study received approval from the University Ethics Committee.

3.5.1 Study Design

Participants. We recruited 32 unpaid participants (17 male, 15 female) from the local university, 18–33 years old (M=23.1, SD=2.8), all of whom reported to be right-handed. Among them, 15 use VR at least once per week, 14 at least once per year, and 3 had never used VR devices before. Furthermore, 18 participants had obtained a Bachelor's degree or higher. They all had normal or corrected-to-normal vision and were able to distinguish clearly the colors used in our application.

Apparatus. We used Valve Index [115], a PC-based VR HMD (1440 × 1600 resolution per eye, 108° field of view, 90 Hz refresh rate). We carried out the study on a PC (Intel Core™ i9, 3.5GHz, 64GB RAM, GeForce RTX3090, 24 GB video memory).

Datasets. All datasets in our study (Fig. 3.7) have target (yellow) and interfering particles (blue). We designed or selected these datasets to have different features that made the selection of targets challenging:

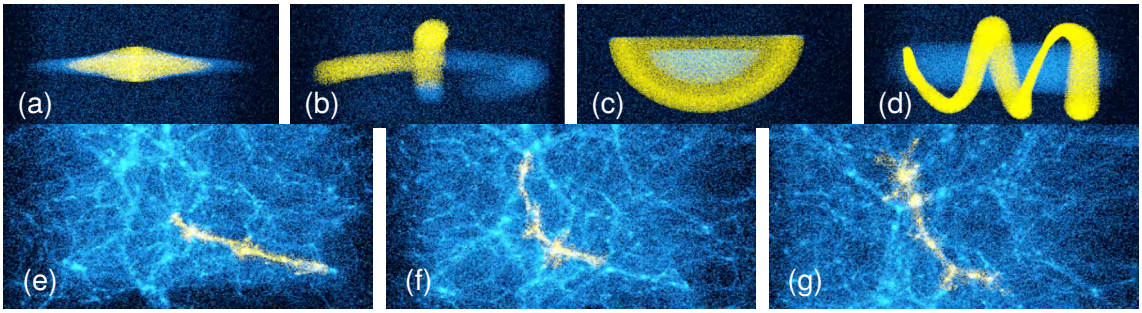


Figure 3.7: Datasets we used in our study: (a) *Disk*, (b) *Rings*, (c) *Shell*, (d) *Strings*, and (e–g) *Filaments*.

Disk: This dataset (Fig. 3.7a) exhibits a gradual decrease in density from its center towards the periphery. The high-density area around the center is the target region.

Rings: This dataset (Fig. 3.7b) consists of two half rings with uniform density, positioned perpendicular to each other with a slight gap between them. The target area comprises a portion of each ring.

Shell: This dataset (Fig. 3.7c) comprises a half-ball of interfering particles that are partially surrounded by a semispherical shell of particles. Both structures have uniform densities and are in close proximity. The semispherical shell is the target structure.

Strings: This dataset (Fig. 3.7d) comprises two string-like structures with non-uniform density. The outer string wraps around the central string and exhibits variations in both density and perimeter along itself. We asked participants to select the outer string.

Filaments: The Millennium-II data subset [108] (Fig. 3.7e) is a complex network of filaments that connect high-density clusters. We chose this real-world data to represent a realistic scenario in astronomy. Unlike the other datasets, we showed the target filaments only for two seconds, to give the participants a general sense of their 3D location and structure. This way we wanted to see what crucial features participants found important and how our methods supported their selections. To avoid a learning effect we thus asked participants to select three distinct filaments in each condition (Fig. 3.7(e–g)).

Task and Procedure. We asked participants to select the yellow and to avoid the blue particles, and to complete the tasks as fast and as precise as possible. We split the study into four tasks (T1–T4) with explicit goals and one task (T5) with an implicit goal. During the explicit goal tasks, which used the datasets *Disk* (Fig. 3.7a), *Rings* (Fig. 3.7b), *Shell*

(Fig. 3.7c), and *Strings* (Fig. 3.7d), the target particles remained highlighted in yellow until they were selected, after which all selected particles turned red. For each combination of trial and dataset, we chose a unique starting orientation, which changed between trials but which we used for all participants. In the implicit goal task (with dataset *Filaments*, Fig. 3.7e) we showed the yellow target particles to participants for only 2 seconds, giving them a limited period to perceive their location and structure. We allowed them, however, to inspect the target particles as many times as needed for an additional 2 seconds each time but did not allow them to make any selections during this time. Once participants had gained a general understanding of the target particles (e. g., location, structure, context, etc.), we asked them to select the ones that possessed "important features" based on their judgment. With this task we aimed to explore the possibility that, by employing context- and target-aware techniques, users could rely on a general understanding of these "important features" when identifying target points, without them continuously being visible.

To prepare participants for the actual study, we first familiarized them with the selection techniques by practicing with additional training data. During the training trials, we instructed the participants to perform the selection using VR controllers and allowed them to take as much time as needed. In the actual study, however, we instructed them to complete their selection tasks with both speed and accuracy but did not inform them if or when they had achieved the selection goal. We provided two possible selection modes: union and subtraction. As Yu et al. [58, 59] had previously discussed, subtraction is best done using region-based techniques, not with structure-aware ones. We thus implemented subtraction with the Baseline technique in all trials. Participants could use the VR joystick to adjust the density threshold and use the VR button to adjust the size of the input marker. In addition, we provided undo and redo functions as well as a reset to the initial unselected state. The whole study lasted $\approx$ 90 minutes. Following each technique, we requested that the participants evaluate their workload and fatigue levels using NASA's Task Load Index (TLX) [116]. After completing all trials, we asked them to indicate which technique they preferred for each dataset and to provide their reasoning.

Design. For the explicit goal tasks with the first four datasets, we counter-balanced the order for the methods and datasets. For a given participant with a specific P_{ID}, where ID is unique and $\in [0, 15]$ (the first batch of 16 participants), we presented the datasets in the same order for each method. We counter-balanced the dataset order using a Latin square as "(P_{ID} / 4) mod 4" and the method order as "P_{ID} mod 4". We repeated the Latin

square order for the second batch of 16 participants. In summary, we had 32 participant × 4 methods × 4 datasets × 3 repetitions $=$ 1536 trials. For the implicit goal tasks with the last dataset (*Filaments*), we used the same order for the three ROIs across all methods, but counter-balanced the method order. In the end, we had 32 participants × 4 methods × 3 cases $=$ 384 trials.

Measures and Analysis. To reduce the impact of the learning effect, we excluded the first repetition of each dataset × method pairs. This left us with 1280 trials. Since NHST has been criticized for analyzing experimental data [117, 118, 119, 120], and APA recommends alternative approaches [121], we report results using estimation techniques with effect sizes and confidence intervals rather than p-value statistics.

Accuracy. Similar to Yu et al. [58, 59], we calculated two distinct accuracy scores, F1 and MCC (Matthews correlation coefficient), to compare the accuracy of our techniques. The accuracy scores for both cases are based on the identification of true positives (TP, # of correctly selected particles, false positives (FP, # of incorrectly selected particles), and false negatives (FN, # of target particles that were not selected). F1 is calculated as $\mathrm{F1} = 2 \cdot (P \cdot R)/(P + R)$, where $P = TP/(TP + FP)$ and $R = TP/(TP + FN)$. While the F1 score provides a measure of the harmonic mean of the precision, it does not consider the true negatives (TN,# of correctly unselected particles) rate. Thus, we also used MCC as our second error metric which is calculated as:

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

The accuracy scores for both cases are normalized. We computed means and 95% bootstrap confidence intervals (CIs; all $n = 32$).

Completion Time. We analyzed the completion time data using exact CIs on log-transformed data (all $n = 32$). We report means thus as geometric means and express comparisons between means as ratios.

3.5.2 Hypotheses

As we designed the three MeTACAST techniques to be adaptable to specific selection intents it is reasonable to expect that their performance varies depending on the given scenario. To anticipate the results of our study, we analyzed the data features and made

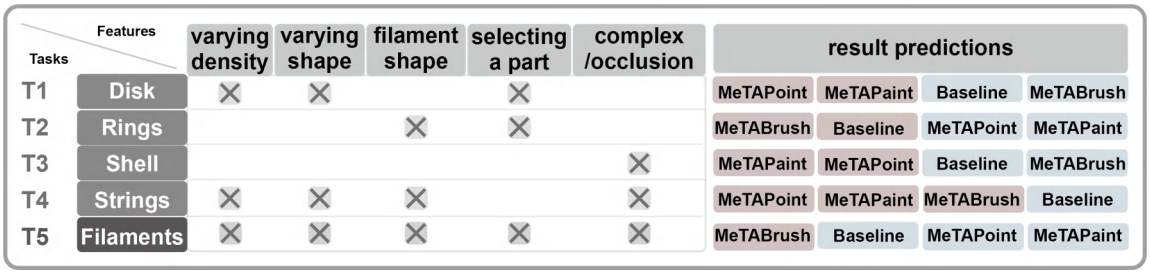


Figure 3.8: Predicted results are based on data features and principles of selection method: red indicates relatively good performance, and blue indicates relatively poor performance.

predictions (Fig. 3.8) based on each technique's principles. The hypotheses are:

H1 MeTACAST adjusts the input toward the local density maximum. All MeTACAST techniques would thus be more accurate than the non-adjusting, region-based Baseline.

H2 Since we explicitly designed MeTABrush for filament-like structures, its completion time and accuracy would be better than that of other techniques for the *Filaments* datasets, especially when the target is part of a filament with varying density and shape.

H3 We designed MeTAPoint and MeTAPaint for selecting complete sub-structures with simple, minimum input, so they would be generally faster than other methods for selecting whole clusters.

H4 Technique preference would depend on the dataset features. In general, for filament-like datasets, participants would prefer MeTABrush and Baseline, while for individual shapes they would prefer MeTAPoint and MeTAPaint.

3.5.3 Results—Explicit Goal Tasks: T1 to T4

As we designed MeTACAST based on different selection strategies and to be able to deal with various datasets, the analysis of the selection effectiveness based on overall completion time and accuracy may not offer a complete picture of their capabilities. We thus discuss the results for the four synthetic datasets and one real-world dataset separately and provide the overall completion time/accuracy results.

Tab. 3.3 shows the overall mean task completion times, accuracy scores and their corresponding 95% confidence intervals. Fig. 3.11 shows the overall geometric means of completion time for each technique. Fig. 3.12 shows the pairwise ratio of completion times for T1–T4. Fig. 3.13 and 3.14 show the overall accuracy for T1–T4. Fig. 3.15–3.24 show

the accuracy per dataset. Fig. 3.28 shows the rank across all datasets (*Disk*, *Rings*, *Shell*, *Strings*, *Filaments*): MeTAPaint (Pa), MeTAPoint (Po), MeTABrush (Br), Baseline (Ba).

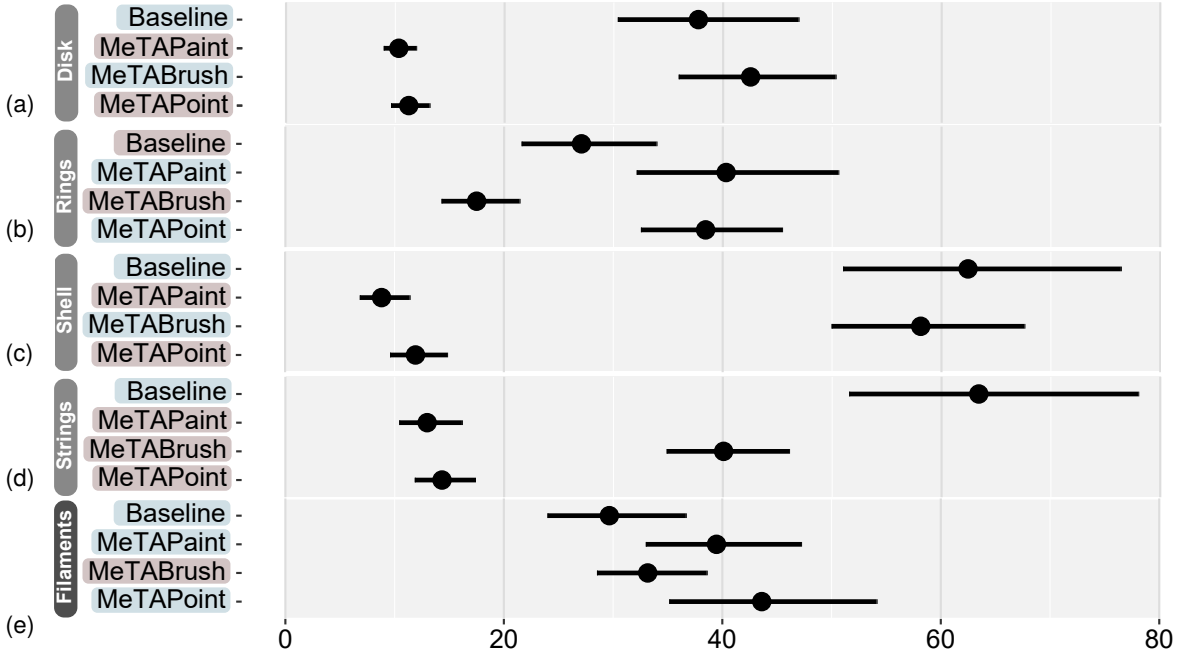


Figure 3.9: The geometric mean completion times in seconds for each selection technique in T1 to T5; (a): *Disk*, (b): *Rings*, (c): *Shell*, (d): *Strings*, (e): *Filaments*. Error bars show 95% confidence intervals (CIs).

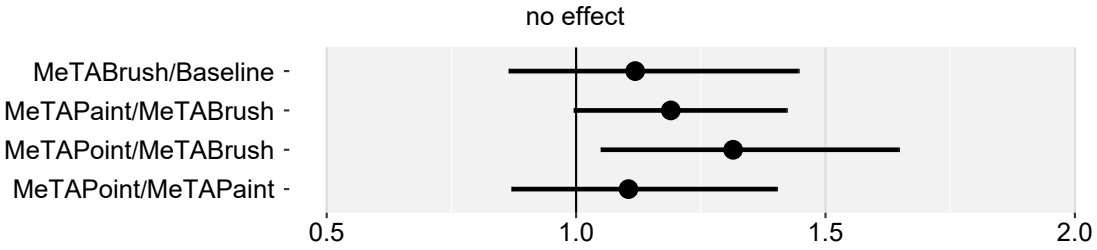


Figure 3.10: Pairwise completion time ratio in T5. Error bars: 95% CIs.

We now focus on the statistical results for each dataset based on our hypotheses. *Disk:* Given that the target area is part of the high-density region, which has varying density and shape, we predicted that the MeTAPoint and MeTAPaint methods would outperform the MeTABrush (designed for filament-like structures) and Baseline (region-based) methods. This is because both MeTAPoint and MeTAPaint facilitate an easy initial selection and

Table 3.2: The mean task completion times, accuracy scores, and their corresponding 95% confidence intervals for T1 to T5.

	Technique	Time	95% CI	F1	95% CI	MCC	95% CI
Disk	MeTAPoint	11s	[9,13]	.95	[.94,.96]	.93	[.91,.95]
	MeTABrush	42s	[35,50]	.92	[.90,.94]	.88	[.85,.91]
	MeTAPaint	10s	[8,11]	.96	[.93,.97]	.95	[.92,.96]
	Baseline	37s	[30,46]	.88	[.84,.89]	.82	[.78,.83]
Rings	MeTAPoint	38s	[32,45]	.97	[.96,.97]	.95	[.94,.96]
	MeTABrush	17s	[14,21]	.97	[.96,.98]	.96	[.94,.97]
	MeTAPaint	40s	[32,50]	.96	[.94,.97]	.94	[.92,.96]
	Baseline	27s	[21,33]	.96	[.93,.97]	.94	[.91,.96]
Shell	MeTAPoint	11s	[9,14]	.97	[.96,.98]	.95	[.92,.97]
	MeTABrush	58s	[49,67]	.97	[.96,.97]	.94	[.92,.95]
	MeTAPaint	8s	[6,11]	.99	[.97,.99]	.98	[.96,.99]
	Baseline	62s	[51,76]	.92	[.82,.96]	.88	[.79,.92]
Strings	MeTAPoint	14s	[11,17]	.99	[.98,.99]	.97	[96,.98]
	MeTABrush	40s	[34,46]	.99	[.98,.99]	.97	[.95,.98]
	MeTAPaint	12s	[10,16]	.98	[.97,.99]	.97	[.95,.98]
	Baseline	63s	[51,77]	.96	[.88,.98]	.93	[.86,.96]
Filaments	MeTAPoint	43s	[35,54]	.69	[.65,.72]	.69	[.65,.72]
	MeTABrush	33s	[28,38]	.87	[.85,.90]	.88	[.85,.90]
	MeTAPaint	39s	[32,47]	.66	[.59,.69]	.66	[.60,.70]
	Baseline	29s	[23,36]	.69	[.64,.72]	.70	[.66,.73]

Table 3.3: The overall (T1 to T4) mean task completion times, accuracy scores, and their corresponding 95% confidence intervals.

Technique	Time	CI	FI	CI	MCC	CI
MeTAPoint	16s	[14,19]	.97	[.96,.98]	.95	[.94,.96]
MeTABrush	36s	[31,41]	.96	[.96,.97]	.94	[.93,.95]
MeTAPaint	14s	[12,17]	.97	[.96,.98]	.96	[.95,.97]
Baseline	44s	[37,54]	.93	[.87,.95]	.89	[.83,.92]

Figure 3.11: The overall (T1 to T4) geometric means of completion time for each selection technique. Error bars: 95% CIs.

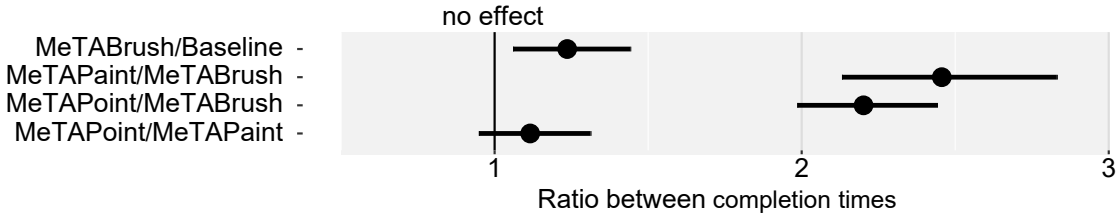


Figure 3.12: The overall (T1 to T4) pairwise ratio of completion time. Error bars show 95% confidence intervals (CIs).

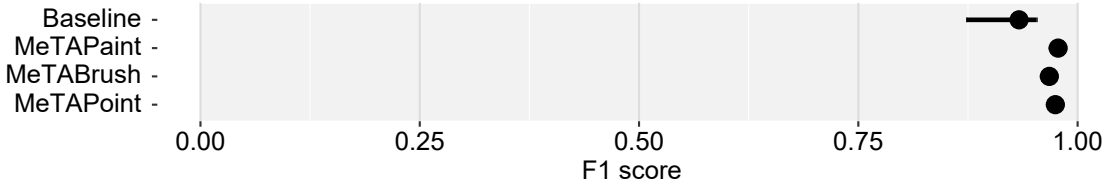


Figure 3.13: The overall (T1 to T4) F1 score. Error bars: 95% CIs.

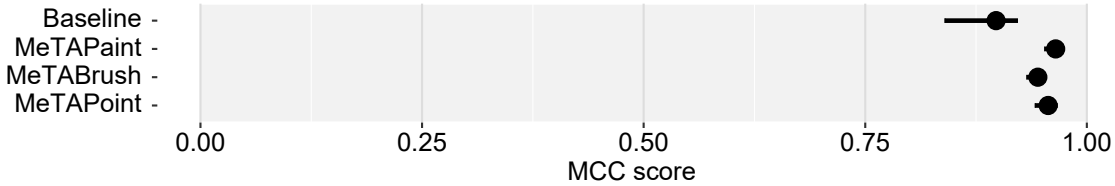


Figure 3.14: The overall (T1 to T4) MCC score. Error bars: 95% CIs.

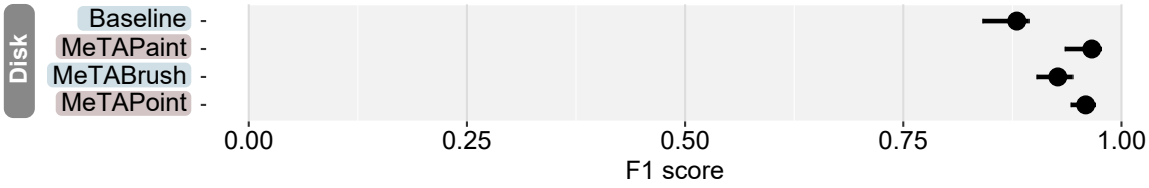


Figure 3.15: The F1 score for T1 (*Disk*). Error bars: 95% CIs.

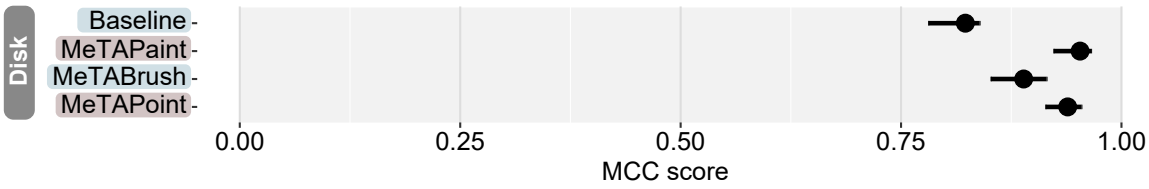


Figure 3.16: The MCC score for T1 (*Disk*). Error bars: 95% CIs.

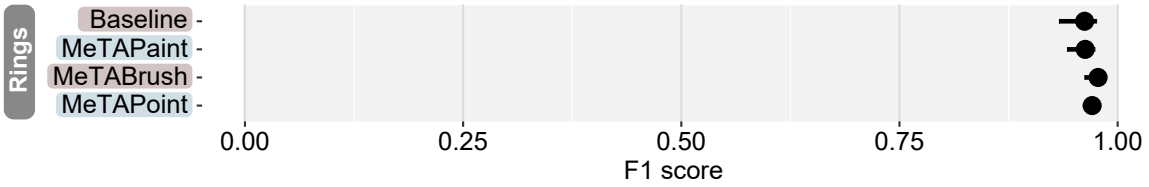


Figure 3.17: The F1 score for T2 (*Rings*). Error bars: 95% CIs.

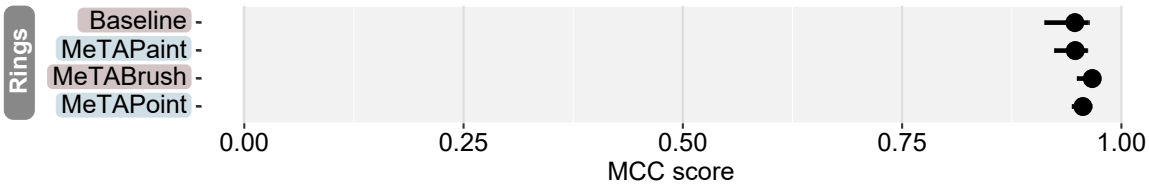


Figure 3.18: The MCC score for T2 (*Rings*). Error bars: 95% CIs.

Figure 3.19: The F1 score for T3 (*Shell*). Error bars: 95% CIs.

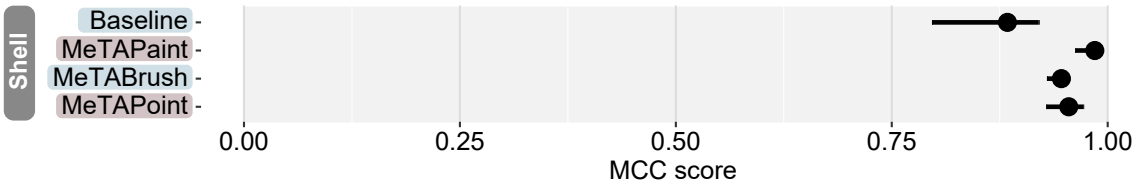


Figure 3.20: The MCC score for T3 (*Shell*). Error bars: 95% CIs.

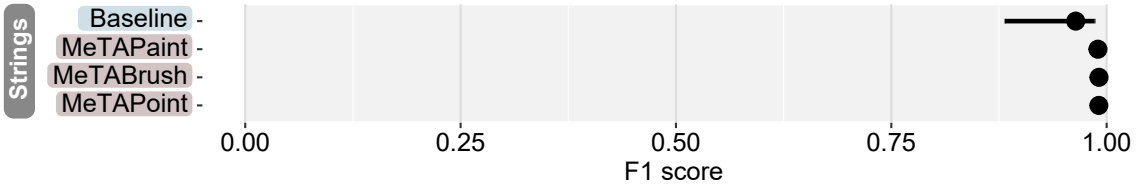


Figure 3.21: The F1 score for T4 (*Strings*). Error bars: 95% CIs.

refinement of results through density threshold adjustments. From the results, we can see MeTAPoint and MeTAPaint were much faster than Baseline and MeTABrush, and Baseline was less accurate than the other methods, which aligns with our prediction.

Rings: As the target area was part of two rings with uniform density and shape, we predicted that both MeTABrush and Baseline would outperform MeTAPoint and MeTAPaint, which were designed to select the entire component and change the overall selection based on the density threshold. We saw that MeTABrush and Baseline were indeed faster. The accuracy scores for all methods are equivalent as we predicted.

Shell: For the *Shell* dataset, the target cluster is a complete component. We thus predicted that MeTAPaint and MeTAPoint would be more effective and that MeTAPoint may be slightly slower than MeTAPaint since its density threshold is determined by the input location. Due to occlusion, however, users may not have precise control over the density adjustment. Our results show that both MeTAPoint and MeTAPaint were indeed faster than the other two methods, with a substantial difference in average completion time. MeTAPoint was slightly slower than MeTAPaint. These results align with our prediction. In addition, our findings indicate that Baseline is the least accurate method.

Strings: Based on the non-uniform density and shape of the target cluster, we predicted that all MeTACAST methods would perform better than the Baseline. We also expected that MeTAPaint and MeTAPoint, requiring minimal input, would be faster than MeTABrush. Although the target was filament-like, MeTABrush may not be as effective as the other two as

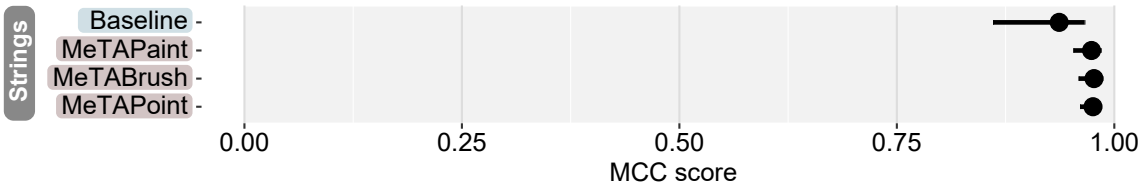


Figure 3.22: The MCC score for T4 (*Strings*). Error bars: 95% CIs.

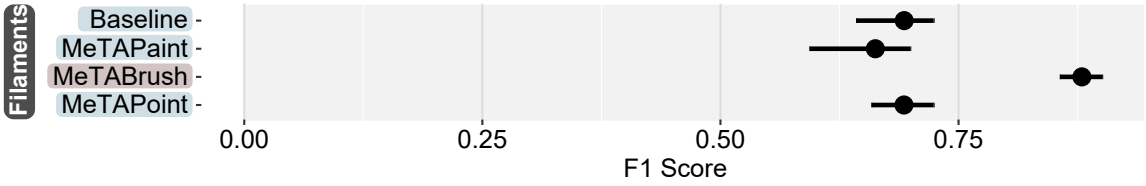


Figure 3.23: The F1 score for T5 (*Filaments*). Error bars: 95% CIs.

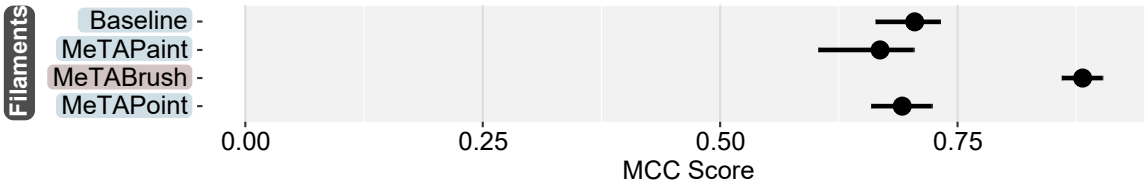


Figure 3.24: The MCC score for T5 (*Filaments*). Error bars: 95% CIs.

it would require participants to brush the whole ROI. Our results show that the MeTACAST methods were both faster and more accurate than Baseline. In addition, MeTABrush was much slower than MeTAPaint and MeTAPoint, which we expected. Moreover, our results show that Baseline was again the least accurate method.

3.5.4 Results—Implicit Goal Task: T5

T5 was a more high-level selection task where participants could only view the features of the target cluster within a short period and make selections based on their understanding of the key data features. In addition to analyzing accuracy and completion time, we were interested in how often participants required to re-check the targets and how our selection strategies supported them in making selections based on data features. We predicted that MeTABrush would outperform other methods since it was specifically designed for filament-like structures in a noisy environment. The results showed that T5 had a lower accuracy level

compared to T1 to T4, while MeTABrush showed considerably higher accuracy (Fig. 3.25) compared to the other methods. Both Baseline and MeTABrush techniques had a shorter

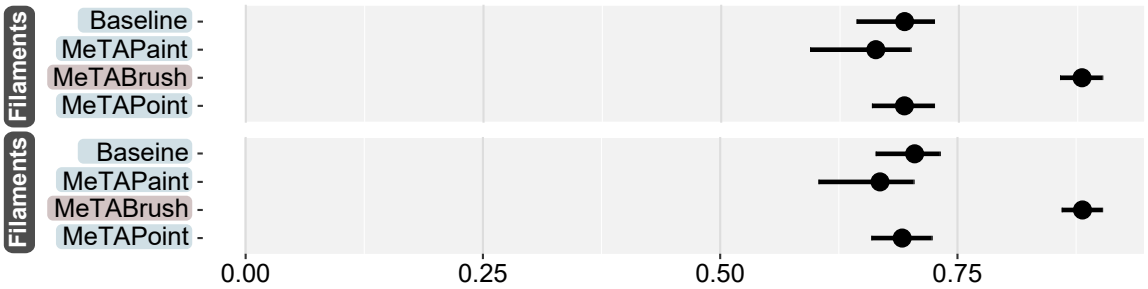


Figure 3.25: The F1 (top) and MCC (bottom) for T5. Error bars: 95% CIs.

mean completion time than MeTAPoint and MeTAPaint (Fig. 3.9(e)), with a small difference. We further provide pairwise ratios of completion times of the MeTACAST techniques in Fig. 3.10. We observed that MeTABrush took only about 0.86–1.44× longer than Baseline. MeTAPoint took about 1.04–1.64× longer, and MeTAPaint took about 1.00–1.42× longer than MeTABrush. Although the filament-based and region-based methods were slightly faster in selecting filament-like structures in the noisy background, the difference was not as substantial as predicted. Overall, we have strong evidence that MeTABrush outperforms other methods in this task.

In conclusion, as noticed in most tasks except T5 (*Filaments*, which also requires user understanding about the data feature), Baseline was the least accurate technique. Therefore, we can **partially support H1**. For the filament datasets (*Rings*, *Strings*, and *Filaments*), the results indicate that MeTABrush was faster than other techniques only in the *Rings* dataset. However, it is important to note that, for the *Strings* dataset where the whole string needs to be selected, MeTAPoint and MeTAPaint were faster. In the case of the *Filaments* task, MeTABrush took about 0.86–1.44× longer than Baseline, but it was substantially faster than other techniques. Yet, MeTABrush showed considerably higher accuracy than other techniques. While we can **partially support H2**, the results thus highlight the importance of MeTABrush in selecting filament-like structures. For the tasks in which the whole dataset needs to be selected (*Shell*, *Strings*), MeTAPaint and MeTAPoint outperform other methods. Thus, **we support H3**.

3.5.5 Results—User Workload and Preference

Workload. After analyzing the data, we found strong evidence that participants felt more efficient, less frustrated, less time-pressured, and needed less mental effort when using MeTAPaint and MeTAPoint in T1–T4, as shown in Fig. 3.26. These findings align with the

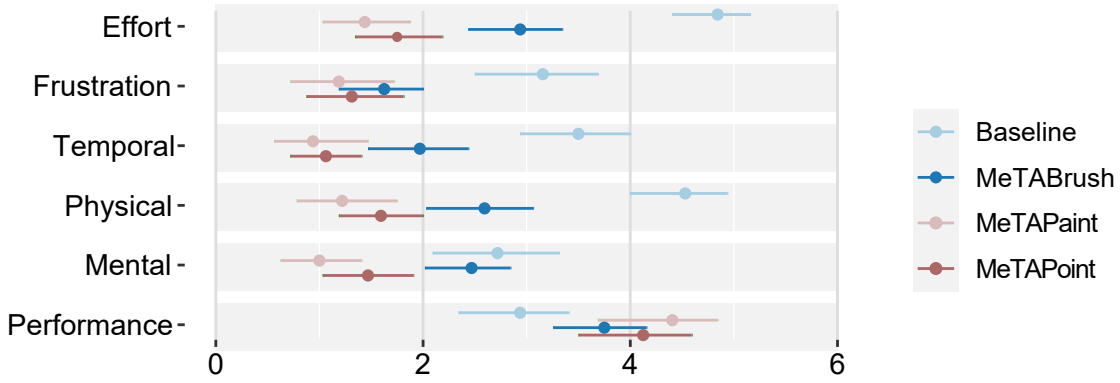


Figure 3.26: User workload and performance, T1–T4. Error bars: 95% CIs.

participants' overall completion times, which indicates that their subjective experiences match their actual performance for these methods. However, for the more complex T5 shown in Fig. 3.27, we saw that MeTABrush was the most effective method, as it led to

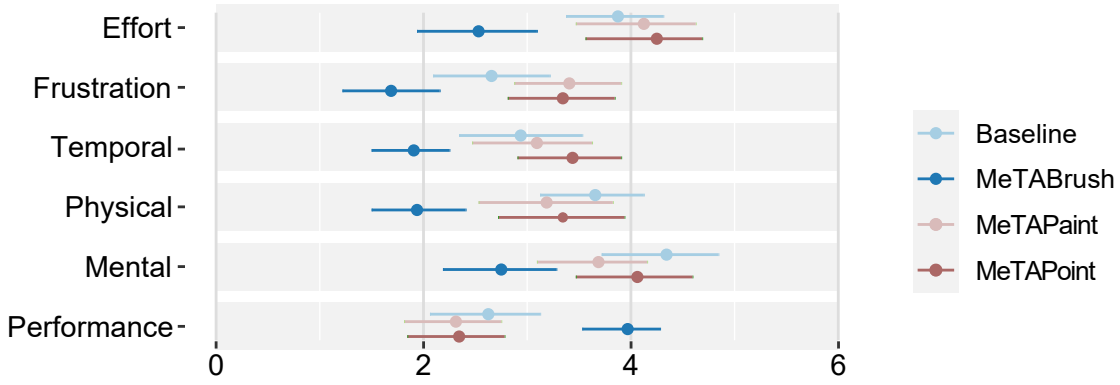


Figure 3.27: User workload and performance, T5. Error bars: 95% CIs.

higher performance, less frustration, less time pressure, and less physical and mental effort. In addition, participants experienced higher mental and physical effort when using Baseline

and felt more frustrated when using MeTAPoint and MeTAPaint in T5.

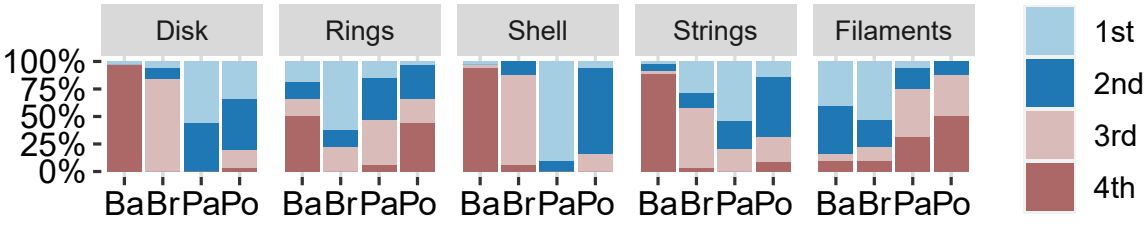


Figure 3.28: User preference for techniques with each dataset.

Preference. After analyzing the techniques' rank across all datasets (T1 to T5) shown in Fig. 3.28, we observe a correlation between the participant preference and their actual performance. Specifically, for the *Disk* and *Shell* datasets, the participants demonstrated a similar pattern in the ranking of the selection techniques (MeTAPaint $>$ MeTAPoint $>$ MeTABrush $>$ Baseline). Interestingly, for the two filament-like datasets, *Rings* and *Filaments*, most participants preferred Baseline and MeTABrush, while MeTAPaint and MeTAPoint were ranked higher for the *Strings* dataset. We believe that this observation may be because participants were required to select the entire string, which can be achieved more easily with MeTAPaint that allows for a large selection with minimal input. Based on these results, **we support H4**.

During the study, we closely observed how participants used each technique to select particles. Our observations revealed that participants demonstrated a clear understanding of the methods' designs and effectively employed strategies to complete the tasks. For example, some participants intentionally brushed a stroke outside of the semispherical shell to ensure that the input stroke remained far away from the half-ball of interfering particles. They made this adjustment with the understanding that the position of the input stroke would be adjusted to align with the shell. In addition, the Baseline technique was also appropriately employed in the study, aligning with its intended design.

To further explore the impact of participants' level of VR fluency, we conducted a separate analysis of task performance on the participants' level of VR fluency. We divided the participants into two groups: 15 VR users (weekly experience) and 17 non-experts (yearly experience and novices). Overall, our comparison revealed that both groups exhibited comparable task performance with MeTACAST methods. However, with Baseline method, both groups were equally fast in the *Disk* and *Rings* tasks. In the *Shell* and *Strings* tasks, which have more complex occlusion features as indicated in Fig. 3.8, VR users were

slower than non-experts. On the other hand, VR users were faster in the most challenging task, *Filaments*. We assume VR users were slower in the complex tasks due to the higher demands of task completion, but they had better direction sense in VR environment and were able achieve better performance in the task which requires a good sense of data structure. This finding indicates performance with MeTACAST is independent of the influence of VR experience—even novice users are able to achieve similar performance to VR users, thanks to the intuitive and effective design of the selection techniques. The following graphs show task performance results for the two groups (black: VR users, gray: non-experts). Fig. 3.29–3.33 show the geometric means of completion times for each technique per dataset, and Fig. 3.34–3.43 show the accuracy results per dataset, comparing two groups.

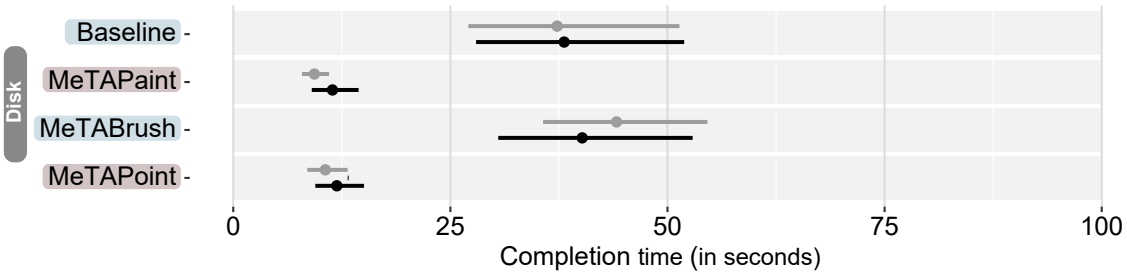


Figure 3.29: The geometric mean completion times in seconds for T1 (*Disk*). VR users (black), non-experts (gray). Error bars: 95% CIs.

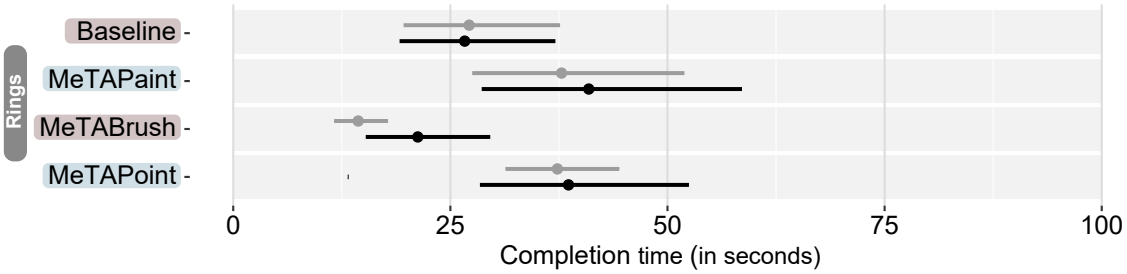


Figure 3.30: The geometric mean completion times in seconds for T2 (*Rings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

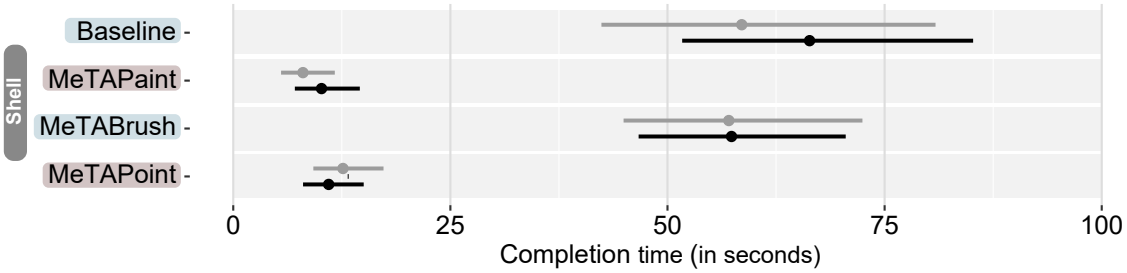


Figure 3.31: The geometric mean completion times in seconds for T3 (*Shell*). VR users (black), non-experts (gray). Error bars: 95% CIs.

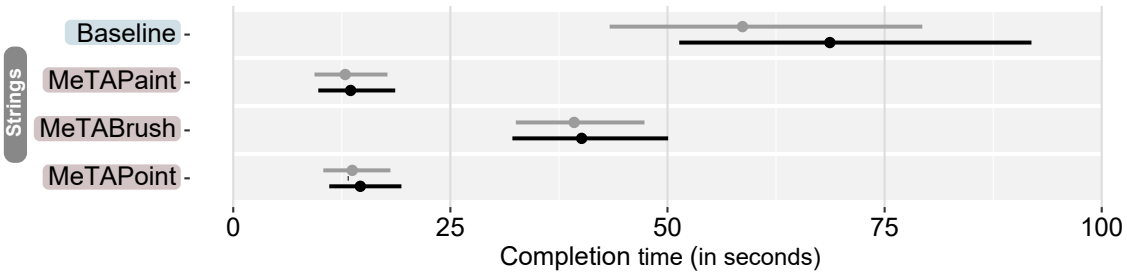


Figure 3.32: The geometric mean completion times in seconds for T4 (*Strings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

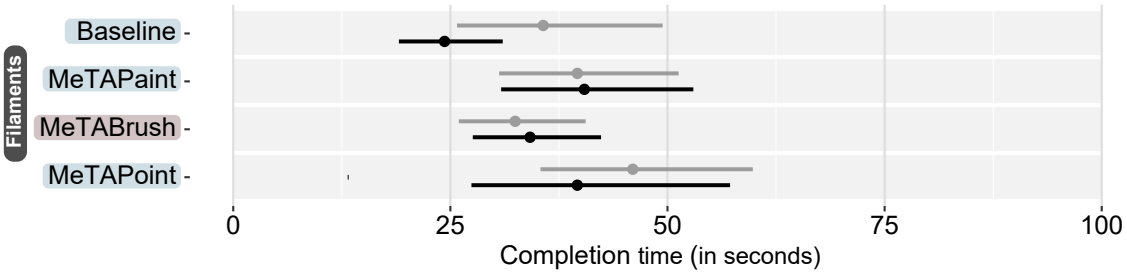


Figure 3.33: The geometric mean completion times in seconds for T5 (*Filaments*). VR users (black), non-experts (gray). Error bars: 95% CIs.

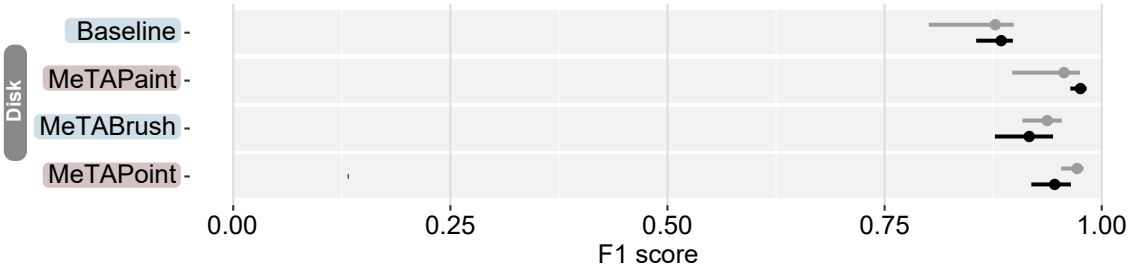


Figure 3.34: The F1 score for T1 (*Disk*). VR users (black), non-experts (gray). Error bars: 95% CIs.

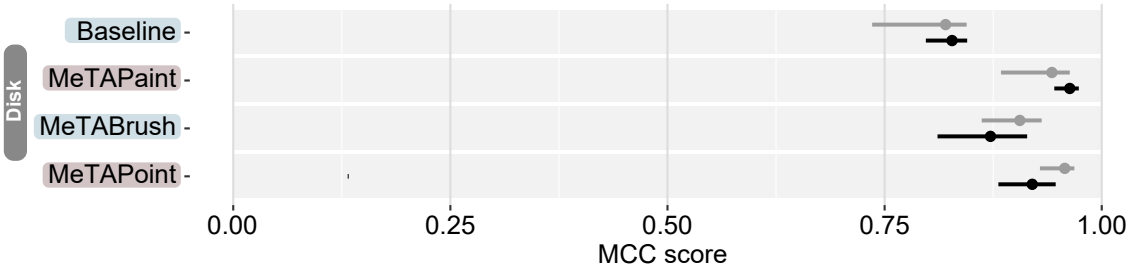


Figure 3.35: The MCC score for T1 (*Disk*). VR users (black), non-experts (gray). Error bars: 95% CIs.

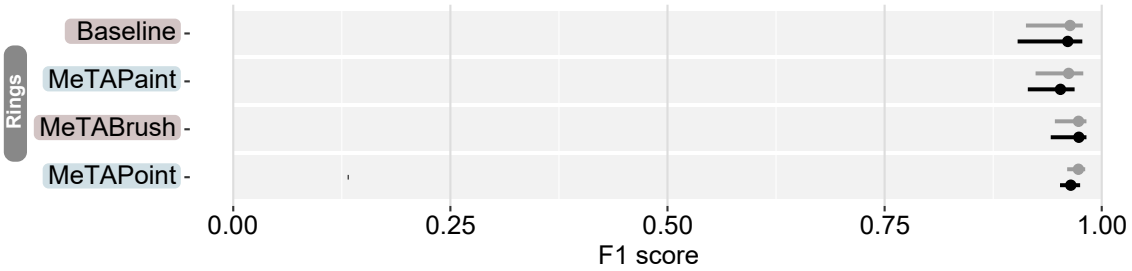


Figure 3.36: The F1 score for T2 (*Rings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

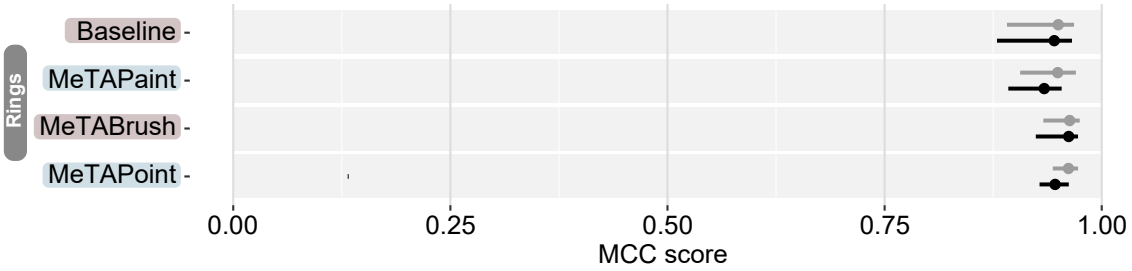


Figure 3.37: The MCC score for T2 (*Rings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

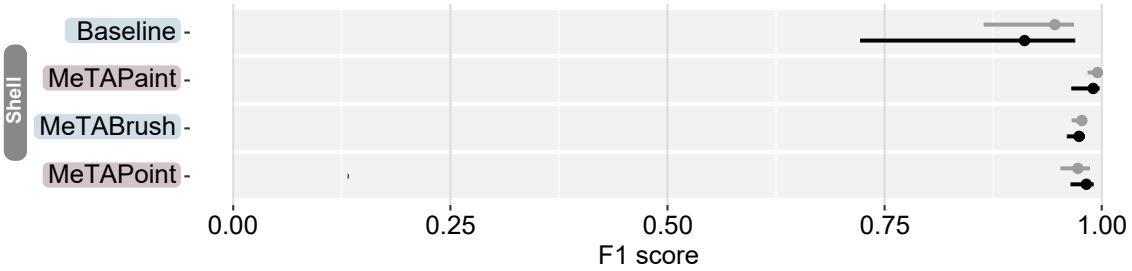


Figure 3.38: The F1 score for T3 (*Shell*). VR users (black), non-experts (gray). Error bars: 95% CIs.

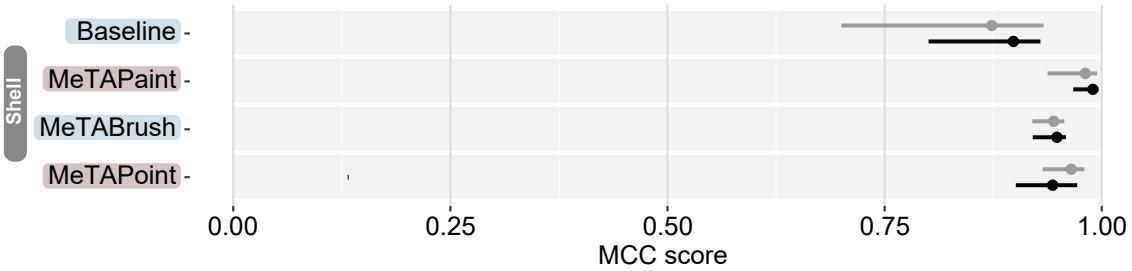


Figure 3.39: The MCC score for T3 (*Shell*). VR users (black), non-experts (gray). Error bars: 95% CIs.

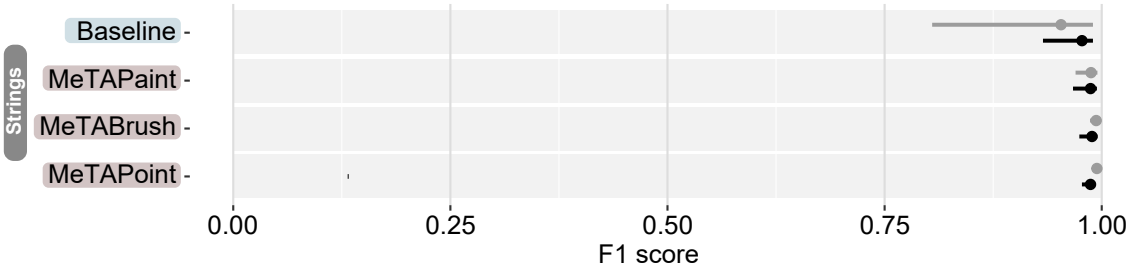


Figure 3.40: The F1 score for T4 (*Strings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

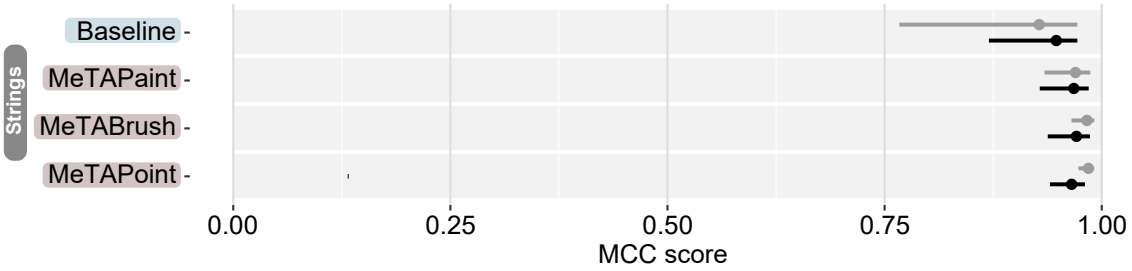


Figure 3.41: The MCC score for T4 (*Strings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

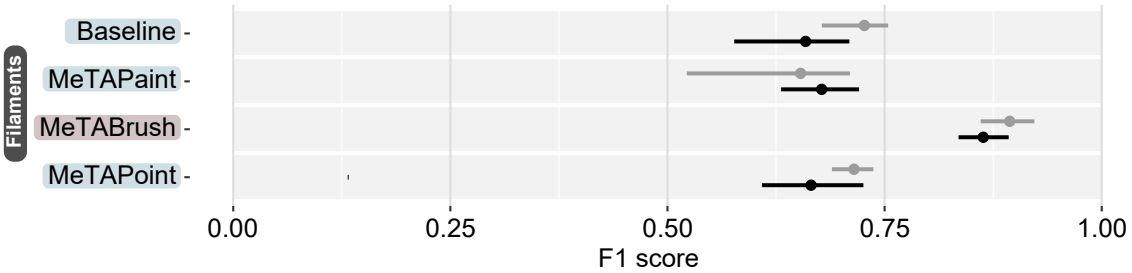


Figure 3.42: The F1 score for T5 (*Filaments*). VR users (black), non-experts (gray). Error bars: 95% CIs.

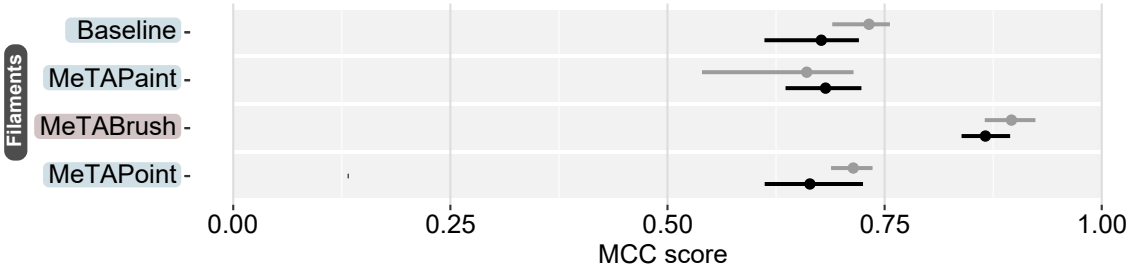


Figure 3.43: The MCC score for T5 (*Filaments*). VR users (black), non-experts (gray). Error bars: 95% CIs.

3.6 Discussion

There are insightful discussion points that can be drawn from our experimental results, as detailed below. We also position MeTACAST within the broader landscape of existing spatial selection techniques to derive general design guidelines for 3D spatial selection.

3.6.1 Design Guidelines for 3D Spatial Selection

Different selection strategies showed advantages in dealing with different data characteristics and situations. When choosing a selection technique, therefore, we need to consider factors, such as data features, selection requirements, and interaction environment.

Target and Context Awareness. The MeTACAST techniques are both target- and context-aware, taking into account various data features during the selection process such as position, structure, and density distribution to realize the user's intention based on their input. MeTAPoint identifies the cluster nearest to the user's pointing location and dynamically adjusts or switches selections based on ongoing input. MeTABrush analyzes the user's input stroke and infers their intention by identifying the primary filament branch close to the path. Finally, MeTAPaint selects targets based on how the user perceives the structure and distinguishes them from the surrounding context. These strategies are well-suited for data selection in immersive environments due to their inherent flexibility, which enables users to focus fully on the data features without separating input location from data position.

Density- vs. Region-based Control. Our results partially support **H1**: MeTACAST techniques are generally more accurate than the region-based Baseline, except in T5 where

the Baseline method was slightly more accurate than MeTAPoint and MeTAPaint—but the difference was minor. In T5, MeTABrush was the second fastest technique (after Baseline), and some participants preferred the region-based interaction when they had limited time to see the target filaments. T5 required participants to rapidly identify features of the target filaments (position, density). Here, if participants were able to quickly recall the target's location, the region-based method offers a straightforward way to complete the task. This observation also explains why some participants ranked Baseline highly in this task, despite being aware that their performance would not be good with this method. Nonetheless, for this dataset, MeTABrush—which considers both the position and density distribution in the selection process—demonstrated clear advantages in terms of accuracy, completion time, and user preference.

Target Shape. Our findings strongly support the notion that different selection strategies are well-adapted to distinct target shapes. Specifically, MeTAPoint and MeTAPaint demonstrated substantially faster performance for *Shell* and *Disk*, whereas MeTABrush and Baseline exhibited advantages in selecting filament-like structures such as *Rings* and *Filaments*. It is important, however, to note that complex data environments typically involve multiple factors, making it challenging to determine the optimal selection method based on a single data feature alone. For instance, although MeTABrush is ideal for filament-like structures, it was not consistently the fastest method in such tasks (e. g., *Strings*). This is primarily due to the fact that, in the *Strings* task, participants had to brush the entire string with MeTABrush and Baseline, while MeTAPaint and MeTAPoint require only minimal painting/pointing on or around the target string.

Partial Selection. In general, our results provide strong evidence to support **H3**: MeTAPoint and MeTAPaint are ***faster*** than the two brushing techniques in ***selecting whole components*** such as for *Disk* and *Shell*, in particular when the target is well-separated from other clusters. This also limits, however, their use when multiple clusters or a subset of cluster are required to be selected (e. g., *Filaments*)—then MeTABrush and region-based techniques are better.

Occlusion. Three datasets had complex occlusion features: *Shell* (partially surrounded half-ball), *Strings* (wrapped inner string), and *Filaments* (comprising many small details). Therefore, participants had to avoid mistakenly selecting interfering particles from other clusters. Our results show that for *Shell* and *Strings*, MeTAPaint and MeTAPoint were faster than MeTABrush and Baseline, and MeTAPaint was slightly faster than MeTAPoint.

The main reason is that in these two tasks, the whole cluster needs to be selected. As long as the user can recognize the partial area of the target cluster, they can accurately and quickly select the entire cluster. A reason why MeTAPoint was slightly slower than MeTAPaint may be that we derive the threshold of MeTAPoint from the density at the input position. Participants thus needed to be careful when dragging the controller. In contrast, the *Filaments* task was particularly challenging for MeTAPaint and MeTAPoint since they treated clusters as individual objects. Future work could explore how to use MeTAPaint and MeTAPoint to identify internal structures such as the half-ball (*Shell*) and inner string (*Strings*) in situations where the user lacks a complete understanding of the target object.

3.6.2 Comparison of Existing Spatial Selection Techniques

Target- and context-aware techniques are especially important in three situations: first, when the user is unfamiliar with the dataset's structure and features; second, when occlusion and perception distortion impede the user's ability to see, judge, or reach the target, or when they need to trace the target's border carefully; and third, when the user has a clear idea about the target and expects to use a simple and fast input to select it. We compared MeTACAST with existing spatial selection techniques' characteristics (summarized in Tab. 3.4) and now discuss their pros and cons under various situations, to offer design guidelines and suggest appropriate methods for different scenarios.

Previous taxonomies have categorized spatial selection techniques according to various factors such as targets (e. g., object, ROI) [122], input strategies and degree of freedom of the input [123], and the users' level of control during the selection process [56]. In our own work we emphasized different aspects of spatial selection, including data characteristics, selection requirements (such as the selection basis and target shape), selection strategies (such as metaphor and selection strategy), and the users' level of control (including precision demand and post-adjustment). Note that the required level of input precision depends on the user's desired level of control and the level of performance needed for the selection. For instance, precise input (2D or 3D) could always be required (high), it could be imprecise or incomplete (low), or imprecise input could be allowed but may cause inaccurate results (medium). To narrow our focus, we only focus on spatial selection where the user specifies a region in space in which the targets are located.

Partial Selection and Interaction Metaphor. An important question is to examine how

Table 3.4: An overview of spatial selection techniques for 3D data.

technique	data type	selection basis	whole / partial	selection strategy	precision needs	metaphor	shape adjustment
Cylinder Selection [88, 88]	ROI	region	whole / partial	semi-autom., 2D	high	lasso	no control
Cloudlasso [58]	point cloud / scalar	value	whole / partial	semi-autom., 2D	medium	lasso	threshold adjust
SpaceCAST [59]	point cloud / scalar	value	whole / partial	semi-autom., 2D	medium	lasso	threshold adjust
Volume Catcher [54]	scalar	value	whole	semi-autom., 2D	low	lasso	no control
TraceCAST [59]	point cloud / scalar	value	whole	semi-autom., 2D	low	lasso	threshold adjust
PointCAST [59]	point cloud / scalar	value	whole	semi-autom., 2D	low	raycasting	threshold adjust
LassoNet [57]	point cloud	deep learning	whole / partial	semi-autom., 2D	low	lasso	no control
WYSIWYP [87]	scalar	value	none	semi-autom., 2D	low	raycasting	no need to control
Slicing Volume [103]	point cloud	region	partial	manual	high	raycasting	region adjust
Hybrid AR selection [35]	point cloud / scalar	region	partial	manual	high	lasso + extrusion	no control
TangibleBrush [56]	point cloud / scalar	region	partial	manual	high	lasso + extrusion	no control
Touching the cloud [101]	point cloud	region	partial	manual	high	brush	region adjust
Embodied Axes [33]	ROI	region	whole / partial	manual	high	tangible (box)	region adjust
Fiducial-Based Tangible [102]	tensor	region	partial	manual	high	brush	no control
Live-Wire [104]	scalar	value	partial	semi-autom., 3D	medium	seed points	no control
Lightweight Tangible [34]	vector	value	whole	semi-autom., 3D	medium	tangible (orien.)	no control
Neuron Tracing [43]	scalar	value	partial	semi-autom., 3D	low	brush	no control
MeTAPoint (this paper)	point cloud / scalar	value	whole	semi-autom., 3D	low	point + drag	threshold adjust
MeTABrush (this paper)	point cloud / scalar	value	whole / partial	semi-autom., 3D	low	brush	threshold adjust
MeTAPaint (this paper)	point cloud / scalar	value	whole	semi-autom., 3D	low	brush	threshold adjust

whole or partial selection intents relate to the chosen interaction metaphor. It would be interesting to investigate, for instance, whether the lasso approach is an effective method for making partial selections. The lasso approach is indeed the most widely used approach for defining selection ranges for multiple targets or partial selection on 2D surfaces (e. g., CylinderSelection, CloudLasso [58], SpaceCast [59], LassoNet[57], Hybrid AR selection [35], and Tangible brush [56]). A potential approach for defining/extending the selection range in 3D is through extrusion using mobile devices, as in Hybrid AR selection [35] and Tangible Brush [56]. 3D brushing provides a direct and intuitive way for users to make partial selections in 3D, as demonstrated in techniques such as Touching the cloud [101], Fiducial-Based Tangible [102], Neuron Tracing [43], and MeTABrush. Unlike the lasso or extrusion, however, the selection range for 3D brushing is not always clear. Designers thus need to consider specific data features and selection contexts when developing such techniques. In contrast to for single-object selection, Raycasting is a less commonly used approach for making partial spatial selections—defining the selection range in 3D through a ray can be challenging. Other possibilities for defining a 3D range include using tangible devices such as Embodied Axes [33], which can achieve partial selection in all dimensions.

Selection Strategy and Precision Needs. Our suggestion for spatial selections is to allow users to express the selection intent in their own way, and then employ selection heuristics that take all relevant input information and data characteristics into account to provide an initial result. Manual selections [103, 56, 33, 102, 35, 101] always require

a high input precision, thus increase the users' physical and mental workload. Several semi-automatic methods are available for context-aware selection on 2D surfaces ([58, 59, 54, 57, 87]) and 3D space (MeTACAST, [104, 34, 43]), specifically designed to consider users' intention and data characteristics. They significantly enhance selection accuracy and efficiency, while also reducing the users' physical workload.

Partial Selection and Selection Strategy and Precision Needs. Following the previous point, a more focused question needs to investigate if the needs for whole or partial selection increase the demand for input precision, leading to greater mental and physical workload. In addition, it would be interesting to explore how selection strategies can mitigate these challenges. We thus focus on selection techniques that have low input precision needs. One scenario is that, for selecting single clusters, users may not need to provide a precise or complete input. The whole target cluster can be selected with a short/incomplete stroke (MeTAPaint, Volume Catcher [54], TraceCast [59]) or a single click (MeTAPoint, PointCast [59], WYSIWYP [87]). Another scenario is that, for selecting a target subset, users may not need to define a precise selection range. The intended selection can, e. g., be made through a complete stroke in LassoNet [57]. Yet two powerful 3D brushing methods, MeTABrush and Neuron Tracing [43], do not require a selection range but still precisely select the intended partial target.

3.6.3 Limitations

As with most work, ours too has some limitations. First, for MeTABrush we currently use a pre-defined range and search for particles that flow into this range by following the gradient direction. While adjusting this range (spatial adjust) and adjusting the density threshold (density adjust) are both possible, it may cause confusion for users about how far the selection volume can expand and whether it can continue to expand as expected. A more reliable approach would be to compute these parameters automatically based on the data distribution, edge, or topological structures. Second, MeTACAST is currently designed for handheld or table-size visualization, and may not be suitable for room-size settings. In such environments users may face difficulties in accurately estimating the density distribution and identifying dense clusters if they are located within them. Third, MeTACAST methods are density-field-based selection techniques which perform well when variations in density distribution are visible. When users encounter difficulties, however, in distinguishing data features, such as when the density remains consistent across the entire point cloud, shape-

based techniques may be more suitable choices. Last, it is important to note that our interaction technique design requires 6DOF input devices.

3.7 Conclusion

We presented a family of spatial 3D data selection methods, MeTACAST (MeTAPoint, MeTABrush, and MeTAPaint), for VR environments. With MeTACAST, users can explore immersive 3D spaces and select data in their preferred way such as pointing, brushing, or drawing. We demonstrated MeTACAST for particle data and density as the measure, yet it can work also with volumetric data and any other scalar field. The basic idea behind our techniques is that users select visually interesting or important locations that contain key information (e. g., density distribution, geometric shape, and position) that causes the feature to be visible. As such, two aspects are essential. First, we need to understand the data and determine which features may guide users' selection intent or attract their attention. This aspect involves analyzing the data features and seeking the expertise of domain specialists to determine the data characteristics and what they are trying to find. Second, we need to understand users and their behavior when encountering data features. To accomplish this, think-aloud studies are effective to collect behavior and intentions.

Our target- and context-aware methods can then infer user intentions from their input, which means that more input can provide more information about the goal. 3D immersive environments thus offer greater flexibility than 2D projection-based approaches, allowing users to flexibly interact with data and seamlessly integrate their actions across modalities. By incorporating information from various modalities (e. g., head and eye movements; path, direction, and speed of input; gestures; user location) more sophisticated target- and context-aware methods can be developed for complex data exploration.

Additional Material Pointers

We share our additional materials (appendix, study results/data and analysis scripts, videos) at `osf.io/dvj9n`. We also make our prototypical implementation of the MeTACAST techniques available at `github.com/LixiangZhao98/MeTACAST`.

CHAPTER 4

ScaleFree: Dynamic KDE for Multiscale Point Cloud Exploration in VR

In the second project, we address **RQ2: How can immersive environments maintain responsiveness when exploring computationally demanding spatial datasets?** The previous project described in **Chapter 3** introduced efficient spatial selection techniques that rely on natural 6DOF input. However, computing the underlying data representation—the scalar field—remains computationally expensive, where we use a precomputation strategy. Yet when users shift their focus to finer scales, the resolution of the precomputed scalar field becomes insufficient to capture the necessary detail. As a result, dynamic re-evaluation of density fields becomes essential to ensure that details are adequately sampled.

In this chapter, we present a GPU-accelerated adaptive KDE algorithm for scalable, interactive multiscale point cloud exploration. With this technique we cater to the massive datasets and complex multiscale structures in advanced scientific computing, such as in cosmological simulations, which contain billions of particles. Exploring such data requires smooth transitions between global context and fine-grained detail. With this approach we can recalculate scalar fields dynamically as users shift their focus across scales. We demonstrate the benefits of our adaptive density estimation through two data exploration tasks: adaptive selection and progressive navigation. Through performance experiments, we demonstrate that our method with GPU-parallel implementation achieves orders-of-magnitude speedups over sequential and multi-core CPU baselines. In a controlled study, we further confirm that our adaptive selection technique improves accuracy and efficiency in multiscale selection tasks.

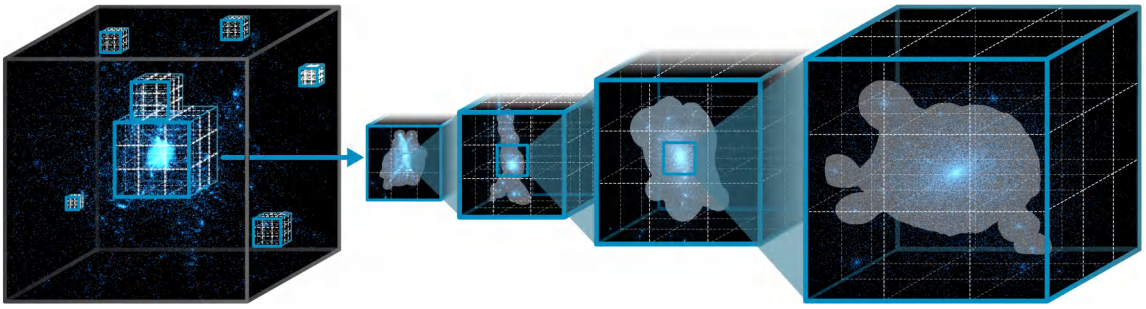


Figure 4.1: ScaleFree dynamically recomputes density fields on the GPU, enabling on-the-fly density estimation of ROI at any scale and position within multiscale point clouds in VR.

4.1 Introduction

Advances in scientific computing have produced datasets that are not only larger in size but also richer in multiscale structural complexity. For example, modern cosmological simulations produce billions of particles, with details that span several orders of magnitude, to capture the formation and evolution of cosmic structures [108, 107]. The fact that meaningful features often emerge at different scales requires analysts to fluidly transition between global and local perspectives—for instance, from cosmic filaments and clusters to fine-grained substructures in astronomical point clouds [108, 124]. To understand the dynamics of structure formation, scientists need to visualize and explore the three-dimensional spatial patterns across multiple scales. Immersive environments delivered through virtual, augmented, and mixed reality HMDs have emerged as superior tools for interpreting complex spatial relationships. Prior research has shown that immersive visualizations can enhance users' understanding of complex spatial structures, including volume data [125], astronomical data [110], and node-link diagrams [126, 127]. These benefits are particularly evident in navigation [128] and 3D manipulation tasks [129], where frequent viewpoint changes are required to inspect spatial structures. The immersive environment not only provide stereoscopic perception of 3D structures but also support seamless transitions between global overviews and localized detail exploration. For instance, a user may zoom in to inspect a galaxy cluster and then zoom out to contextualize its dynamics within the surrounding large-scale structures. This process requires not only an immersive environment, such as VR, but also high-performance computing and visualization systems capable of supporting free navigation across space, time, and scale.

Achieving smooth transitions and effective exploration across multiple scales remains

challenging, however, in particular when data must be transformed into an alternative representation to support further analysis. For example, KDE [71] has been widely used in point cloud visualization to uncover clusters and structural patterns in large multiscale datasets by converting discrete points into a continuous scalar density field. Such density fields are fundamental to spatial interaction, as they provide a robust basis for feature detection [73, 74, 75, 76] and for constructing selection volumes [59, 58, 77], thereby enabling users to identify, isolate, and manipulate regions of interest. Nonetheless, computing scalar density fields on the fly is computationally expensive and often fails to guarantee real-time interaction. Prior density-based selection techniques for point cloud exploration [58, 59, 77] have typically focused on selection interaction itself and relied on precomputed, single-scale density fields, rather than targeting dynamic multiscale recomputation during interaction. Such precomputation, however, implies that the density fields are static during exploration, constraining selection accuracy to the resolution defined *a priori* and leaving finer-scale structures unrepresented. This limitation becomes particularly pronounced in VR. Unlike desktop systems, VR presents users with a single, continuous egocentric view of the data space, where navigation, selection, and scale transitions are tightly coupled with head and body movement. Users frequently shift their focus between global overviews and localized regions, creating a strong demand for density fields that adapt immediately to scale and perspective changes. At the same time, VR imposes strict real-time constraints due to high-frequency stereoscopic rendering and low-latency interaction requirements. These challenges motivate us to explore density estimation techniques that can be dynamically computed at varying scales during real-time immersive exploration.

We thus introduce ScaleFree, a fast, adaptive, and scalable KDE algorithm specifically optimized for interactive point cloud data analysis. Our method leverages GPU acceleration to perform on-the-fly density estimation, enabling scalar fields to be recomputed dynamically as users shift their focus across scales. By maintaining density fields that adapt repeatedly rather than relying on precomputed representations, ScaleFree facilitates more precise and responsive support for core exploration tasks such as *spatial selection*, *feature localization*, and *multiscale navigation* in large point cloud datasets. To evaluate the role of adaptive KDE in interactive VR exploration, we adopt a multi-layered evaluation strategy. We first compare our GPU-based implementation against CPU baselines to establish real-time feasibility, and then evaluate how integrating adaptive KDE affects selection effectiveness through technique demonstrations and a controlled user study, measuring accuracy, efficiency, and

workload against state-of-the-art VR selection techniques. Specifically, we conducted a user study with 24 participants comparing three *selection techniques* that employ different density estimation approaches, including precomputed single-resolution density fields (PS), precomputed multi-resolution mipmap density fields (PM), and our dynamic ScaleFree approach. Results show that selections made with ScaleFree were faster and more accurate than those using precomputed density fields. Participants also reported lower workload and expressed a stronger preference for ScaleFree when conducting multiscale exploration tasks in VR. In summary, our GPU-accelerated adaptive KDE makes the following main contributions to large-scale spatial data exploration in immersive environments:

- **Adaptive and smooth scale transitions:** enabling continuous transitions across scales without disruptive visual artefacts or noticeable delay (demonstrated in both selection Sec. 4.4.1 and navigation Sec. 4.4.1 scenarios).
- **Stable viewpoint-driven exploration:** supporting smooth, orientation-preserving transitions between overview and localized perspectives across scales during exploration of complex datasets (demonstrated in the navigation scenario, Sec. 4.4.1).
- **Fast and responsive computation:** improving selection accuracy and efficiency under varying scales, thereby supporting precise spatial interaction in immersive analytics (shown in the user study results, Sec. 4.5).

4.2 Related Work

Our work builds on prior research in multiscale immersive exploration and density estimation methods that support such exploration.

4.2.1 Multiscale Exploration in Immersive Environment

Multiscale exploration requires users to operate across different levels of scale, ideally without relying on explicit mode switches or commands. Prior studies have demonstrated the advantages of immersive environments in diverse scientific domains such as astronomy [130, 131], geography [132], biology [133], and architecture [134, 135]. A widely adopted technique for multiscale navigation in immersive environments relies on a *World-in-Miniature* (WIM) [136], which presents both the life-size virtual environment and a hand-held, scaled-down replica that functions as an additional viewport. This dual representation en-

ables users to interact with the environment at multiple scales: they can directly manipulate objects via the miniature, specify regions of interest for navigation, and then seamlessly "jump" to the corresponding areas in the full-scale world. Subsequent variations of WIM, such as Scalable WIM [137] and Scaled and Scrolling WIM [138], extended the metaphor by introducing panning and rescaling of the miniature, facilitating navigation across broader spatial ranges and more precise exploration of fine-grained structures, overcoming the scale and resolution constraints of the original WIM. Similarly, *target-based* and *steering-based* approaches, such as the magnifying glass metaphor by Kopper et al. [139], enable navigation through predefined discrete scale levels. Although these methods provide overview-detail support, however, they often fail to preserve users' sense of scale when "jumping" across scale levels.

To address this issue, Zhao et al. [110] linked multiple WIMs into a *hierarchical structure*, allowing users to seamlessly "jump" between different scales. Similarly, Bacim et al. [140] incorporated a hierarchical map to aid travel and wayfinding in anatomy applications. Moving beyond hierarchical structures, *label-based* approaches employ meaningful anchors that reflect data organization, as in Kouřil et al.'s [133] approach, who used active labels to browse hierarchical molecular visualizations. Nevertheless, these methods all depend on well-defined targets and hierarchical algorithms to compute and represent scale levels, with transitions typically occurring in either an end-to-end or step-by-step manner. Similar transition approaches are found in molecular visualization [141, 10], where hierarchical molecular structures provide clearly defined scale levels. By contrast, we focus on unstructured and massive scientific datasets, such as astronomical point clouds, where predefined or meaningful hierarchies are often absent. In such scenarios, "jumping" approaches are less straightforward—we need more flexible techniques that support navigation and interaction at arbitrary scales.

4.2.2 Key Aspects in Multiscale Immersive Exploration

Building on prior work in multiscale immersive exploration, we next synthesize key design aspects and limitations identified across existing systems. In particular, we focus on challenges related to fast and scalable computation, which motivate the need for adaptive density estimation techniques reviewed in the following subsection.

Scalable and flexible exploration. Effective multiscale exploration should enable

users to select or navigate any arbitrary region within an object or space, as features may appear at any scale. When hierarchical information is not explicitly available, such as in astronomical point cloud data [108, 124] or additive manufacturing objects [142, 143] with dense, homogeneous internal structures, localization and navigation become particularly challenging. In such cases, continuous zooming enables more flexible and fluid multiscale exploration. To address this issue, Pavanatto et al. [144] proposed a progressive refinement approach, facilitating, step-by-step, focused inspections via a selection box. Our work shares this motivation but takes a different path: rather than refining predefined regions, we enable users to focus on relatively dense areas across diverse scales. This approach, in turn, requires a dynamic computation of density fields to adaptively support the exploration.

Adaptive and smooth transition. In free exploration, adaptive transitions are essential to reduce disorientation and preserve a natural sense of immersion in multiscale environments. To ensure continuity, techniques should dynamically adjust factors such as computational scale, navigation speed [145, 134], and the visualization's and users' relative scale [134, 146]. Traversing multiscale data, for instance, often requires adaptive "flying" speeds to balance comfort and continuity. Ware and Fleet [145] addressed this point by introducing continuous depth sampling to modulate the navigation speed, while Argelaguet et al. [134] extended the concept with an adaptive navigation technique that facilitates direct camera control, while it also automatically adjusts speed and environmental scale based on object distance and optical flow. More recently, Weissker et al. [146] proposed *teleportation-based* multiscale travel methods that integrate scale adjustments directly into the teleportation process. All this work highlights the need for approaches that enable users to transition smoothly through unstructured multiscale data, with navigation, visualization, and interaction parameters continuously adapted to the new scale, all of which we also strive for in our work.

Fast and responsive computation. Finally, achieving these interaction designs in practice requires scalable computational support: Techniques must remain efficient and responsive when applied to large, complex datasets, ensuring real-time performance. One common approach is to introduce hierarchies of reduced objects or resolutions to balance rendering and interaction costs. Gansner et al. [147], for example, precomputed a hierarchy of coarsened graphs that could be combined on-the-fly for visualization. For massive unstructured datasets such as point clouds, density estimation methods like KDE form the foundation for feature detection, region-of-interest selection, and navigation across scales.

The key challenge, however, is ensuring that KDE can be computed on-the-fly to support such flexible and adaptive interactions across scales—on which we focus in our work and for which we review past work next.

4.2.3 Kernel Density Estimation

KDE [71, 72] is a widely used method for computing data point distributions, which has been applied in diverse domains such as materials science [148], astronomy [149, 150], geoscience [151], ecology [152], and traffic accident studies [153]. Its core idea is that each data point is represented by a probability distribution (kernel; usually Gaussian [154] or Epanechnikov [155, 156, 157] ones) centered at its position, with the bandwidth controlling its spatial influence. Summing all kernels yields a continuous density field.

A key computational challenge in KDE lies in selecting the bandwidth, which controls the level of smoothing. The bandwidth can be specified as a constant (fixed KDE) or adapted to local point densities (adaptive KDE). Fixed KDE [78, 155] applies uniform smoothing across the dataset. While it is computationally less expensive, it often oversmooths dense regions and undersmooths sparse ones, limiting its accuracy in heterogeneous datasets. In contrast, adaptive KDE mitigates this issue by varying the bandwidth according to local density [78, 79]: points in sparse areas are assigned larger bandwidths, while those in dense regions receive smaller ones. This adaptive strategy yields more meaningful estimates in many applications, but comes at a high computational cost. For example, Brunsdon [78] proposed a cross-validation-based method for determining spatially adaptive bandwidths, but its two-dimensional parameter search makes it inefficient for large datasets. For high-dimensional data, researchers proposed hashing-based estimators [158] to achieve sublinear query time and linear space and preprocessing time.

KDE has been widely used in data visualization, with prior work exploring the algorithmic and system-level optimizations. Examples include GPU-accelerated KDE [159] for interactive streaming visualization, efficient kernel density techniques [160] for large-scale 2D hotspot visualization, adaptive GPU-based KDE methods [161] that combine algorithmic optimizations with parallel computation, and more recently, prefix-based spatiotemporal KDE frameworks [162] for large-scale, high-resolution density visualization. While these advanced KDE techniques offer improved theoretical efficiency and scalability and are tailored to different problem settings, our work focuses on immersive interaction integration

and GPU-parallel implementation to support real-time, interactive multiscale exploration.

KDE has also been leveraged to support in the context of immersive analytics. Prouzeau et al. [163] introduced a VR-based technique that uses haptic feedback to convey KDE-derived density information, helping users identify occluded features in dense point cloud data. Zhao et al. [77] employed KDE in immersive spatial selection techniques, enabling users to define selection regions based on density variations rather than individual points. These approaches, however, rely on precomputed single-scale density fields, limiting their ability to capture fine substructures across multiple levels of detail. These efficiency limitations have motivated the exploration of high-performance computing strategies. An effective interaction with large-scale datasets, however, especially in immersive environments, requires not only scalability but also real-time, on-the-fly computation, which was not explicitly addressed in prior research. To this end, we formulate a GPU-parallel algorithm based on a modified Breiman kernel density estimation method with a finite-support, adaptive Epanechnikov kernel, and further propose a parallel optimization method to accelerate adaptive bandwidth calculation, together improving the efficiency of the entire KDE pipeline. To validate our approach, we demonstrate its use in two scientific data exploration scenarios and evaluate its effectiveness through both performance experiments and user studies.

4.3 ScaleFree

We begin our discussion with an overview of the KDE algorithm used in this work (Sec. 4.3.1), then present ScaleFree, our GPU-accelerated KDE for multiscale analysis (Sec. 4.3.3), and finally report on experiments that evaluate its performance (Sec. 4.3.4).

4.3.1 Modified Breiman Kernel Density Estimation

KDE methods produce a smooth, continuous density field by distributing the contribution of each particle over a larger volume using a smoothing kernel function. This kernel function describes the spatial influence of each particle, assigning higher weights to locations closer to the particle center and lower weights to more distant locations. To estimate the scalar density field, similar to past works [77, 58, 59], we apply the modified Breiman kernel density estimation method (MBE) with a finite-support adaptive Epanechnikov kernel [155, 156, 157] as follows: The whole dataset or region of interest is enclosed within a bounding box B

and discretized into a uniform grid of resolution $\mathbf{res}$ (e. g., 128^3), with grid nodes located at positions $\mathbf{r}^{(n)}$, where n denotes the node index. For each spatial axis $k \in \{x, y, z\}$, we calculate the smoothing length ℓ as

$$\ell_k = \frac{2\left(P_k^{(80)} - P_k^{(20)}\right)}{\log N}. \tag{4.1}$$

Here, N is the number of particles in the box B, and $P_k^{(q)}$ represents the q^{th} percentile of the coordinates along axis k. For the grid node at position $\mathbf{r}^{(n)}$, we calculate the pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(n)})$ as:

$$\rho_{\text{pilot}}(\mathbf{r}^{(n)}) = \frac{15}{8\pi N} \frac{1}{\ell_x \ell_y \ell_z} \sum_j E(\|\mathbf{r}^{(n;j)}\|), \tag{4.2}$$

with

$$\mathbf{r}_k^{(n;j)} = (\mathbf{r}_k^{(n)} - \mathbf{r}_k^{(j)})/\ell_k, \tag{4.3}$$

where $\mathbf{r}^{(j)}$ is the the j^{th} particle's position and $E(x)$ is the Epanechnikov kernel:

$$E(x) = \begin{cases} 1 - x^2, & |x| < 1, \\ 0, & |x| \geq 1. \end{cases} \tag{4.4}$$

The function $E(x)$ implies that a particle contributes to pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(n)})$ only if the particle $\mathbf{r}^{(j)}$ lies within the ellipsoid centered at the node position $\mathbf{r}^{(n)}$, with semi-axes ℓ_x, ℓ_y, and ℓ_z. This ellipsoidal restriction effectively limits the kernel bandwidth, ensuring that the resulting density estimate captures the local particle distribution. Then, we compute the pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(j)})$ at the position of the j^{th} particle using multi-linear interpolation with respect to the densities on nearby nodes. Next, we compute the mean pilot density $aveDen$ of all particles:

$$aveDen = \sum_j \rho_{\text{pilot}}(\mathbf{r}^{(j)}) \tag{4.5}$$

We then update the j-th particle's smoothing length by

$$\ell_k^{(j)} = \min\left(\ell_k \cdot \left(\frac{aveDen}{\rho_{\text{pilot}}(\mathbf{r}^j)}\right)^{\frac{1}{3}},\ 5 \cdot \mathsf{s}_k\right), \tag{4.6}$$

where ℓ_k is the initial smoothing length along axis k, s_k denotes the distance between adjacent nodes along the k-axis. To prevent the kernel from becoming excessively large—which could increase computational cost and overly smooth the density—we apply an upper bound of $5 \cdot s_k$. Then, we re-evaluate the density on all the nodes based on the updated smoothing length. For the grid node at position $\mathbf{r}^{(n)}$, we calculate the final density $\rho(\mathbf{r}^{(n)})$

as:

$$\rho(\mathbf{r}^{(n)}) = \frac{15}{8\pi N} \sum_j \frac{1}{\ell_x^{(j)} \ell_y^{(j)} \ell_z^{(j)}} E(\|\mathbf{r}^{(n;j)}\|), \tag{4.7}$$

with

$$\mathbf{r}_k^{(n;j)} = (\mathbf{r}_k^{(n)} - \mathbf{r}_k^{(j)})/\ell_k^{(j)}, \tag{4.8}$$

Finally, we calculate the density $\rho(\mathbf{r})$ at any position $\mathbf{r}$ via trilinear interpolation of the surrounding grid nodes.

4.3.2 Parallel programming framework on the GPU

This part briefly introduces the GPU parallel programming model following HLSL conventions, which is used later in our KDE pipeline. In this model, parallel computation is performed by invoking (programming) kernels, each of which is executed concurrently by multiple GPU threads. A thread is the basic unit of execution, running the same kernel function independently. We can organize threads into groups defined by `numthreads(tx, ty, tz)` before launching a kernel, which specifies the dimension of threads in the group along the x-, y-, and z-dimensions. Threads within a group share fast group memory and can synchronize their data exchange with group memory via the function `GroupMemoryBarrierWithGroupSync()`. Groups remain independent, thus, threads in different groups cannot access each other's group memory. All GPU threads have access to global memory, which serves as the primary space for data exchange during computation. Before launching kernels, we need to transfer the input data from the CPU to global GPU memory. When launching a compute kernel, `Dispatch(kernel, groupCountX, groupCountY, groupCountZ)` specifies the number of thread groups along the x-, y-, and z-dimensions, thereby defining the overall dispatch layout. The total number of threads executed equals the product of threads per group and the number of groups. During kernel execution, each thread identifies its local (within its group), group, and global (among all threads) indices through semantics such as `SV_GroupIndex`, `SV_GroupID`, and `SV_DispatchThreadID`. Once the kernel completes execution, results are written to GPU buffers and can be returned to the CPU for further processing. This programming model provides the execution framework for the GPU-accelerated KDE pipeline described in the following content.

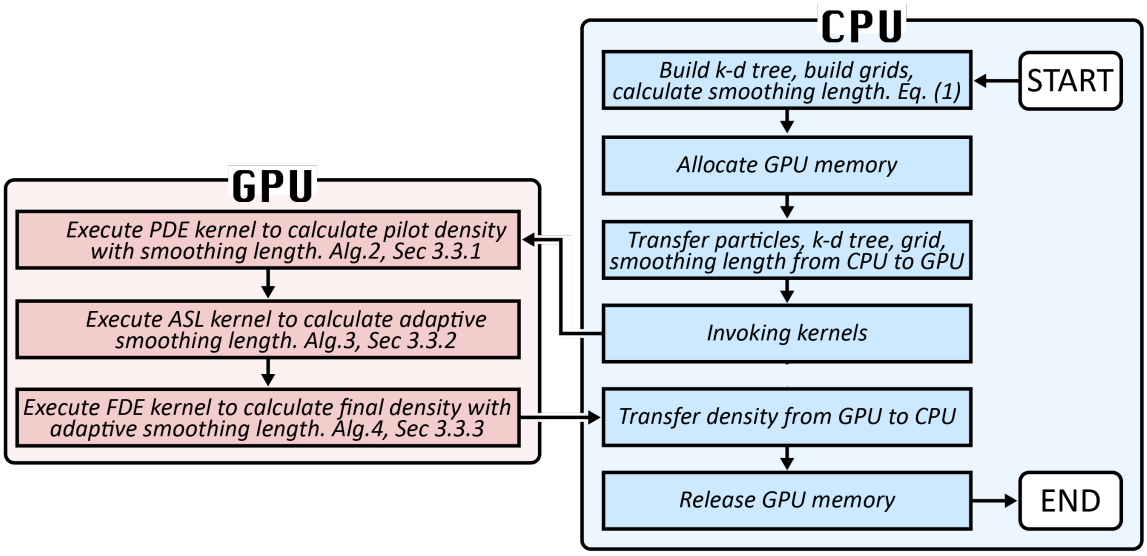


Figure 4.2: Workflow of the GPU-accelerated KDE algorithm.

4.3.3 ScaleFree: Fast and Scalable KDE on GPU

Building on this GPU programming model, we implement the KDE pipeline as three compute kernels with key optimizations for on-the-fly density evaluation (Fig. 4.2). Prior to GPU execution, we perform essential preprocessing on the CPU: we determine the smoothing length (Equa. 4.1) and create a k-d tree spatial index for the point cloud dataset to accelerate future neighborhood queries. We then upload the input data (k-d tree, smoothing length, particle positions, grid node positions, grid resolution, and node spacing) to GPU global memory. In the GPU pipeline, as illustrated in HLSL-style pseudo-code in Alg. 1, we dispatch three compute kernels, each handling a distinct density estimation stage:

- In **Pilot Density Estimation** (Alg. 2) we compute a pilot density field at grid nodes by a uniform smoothing length.
- In **Adaptive Smoothing Length** (Alg. 3) we update per-particle smoothing lengths based on average pilot density.
- In **Final Density Estimation** (Alg. 4) we recompute the density field using the adaptive smoothing lengths.

Algorithm 1: Fast and Scalable GPU-accelerated KDE

1 **function** FSKDE($kdTree, particles, nodes, res, l, s, density$):
  Input: $kdTree$—k-d tree spatial indexing structure
  Input: $particles$—array of positions of all particles
  Input: $nodes$—array of positions of all grid-nodes
  Input: res—resolution of the grid (e.g. $64 \times 64 \times 64$)
  Input: l—array of smoothing length for every particle
  Input: s—node spacing along x, y, z axis
  Output: $density$—array of pilot density at grid-nodes
2 $Dispatch$ (**PDE** $(kdTree, particles, nodes, res, l, density)$, $\frac{res_x}{PDE_{tx}}, \frac{res_y}{PDE_{ty}}, \frac{res_z}{PDE_{tz}}$);
3 $Dispatch$ (**ASL** $(particles, l, s, density)$, $\frac{particles.\texttt{count}}{ASL_{tx}}, 1, 1$);
4 $Dispatch$ (**FDE** $(particles, nodes, res, l, density)$, $\frac{res_x}{FDE_{tx}}, \frac{res_y}{FDE_{ty}}, \frac{res_z}{FSDEtz}$);

Pilot Density Estimation (PDE)

To start, we compute the pilot density at each grid node by accumulating the contributions from neighboring particles. We adopt the $gather$ approach [164], where each thread processes a grid node and collects contributions from neighboring particles, rather than a $scatter$ approach, where each thread processes a particle and distributes its contribution to surrounding nodes. This way we avoid costly atomic operations and improve the performance and scalability for our large-scale particle datasets.

We dispatch the **PDE** kernel (Alg. 2) with $\frac{\textbf{res}_x}{PDE_{tx}} \times \frac{\textbf{res}_y}{PDE_{ty}} \times \frac{\textbf{res}_z}{PDE_{tz}}$ thread groups, each consisting of $PDE_{tx} \times PDE_{ty} \times PDE_{tz}$ threads, to cover the entire grid domain. For each thread with global index idx, we evaluate the pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(idx)})$ at node position $\mathbf{r}^{(idx)}$ using Equa. 4.2. Since only particles within the kernel support contribute to the density (Equa. 4.4), we first identify candidate particles through a spherical range query on the k-d tree, centered at $\mathbf{r}^{(idx)}$ with radius $\max(\ell_x, \ell_y, \ell_z)$, i. e., the largest of the three kernel semi-axes ℓ_x, ℓ_y, and ℓ_z. This query allows us to discard of particles outside the Epanechnikov kernel. In this way, for M nodes and N particles, we reduce the complexity of the original KDE algorithm from $O(MN)$ to $O(M\sqrt{N})$ (where k-d tree searching has complexity $O(\sqrt{N} + K)$, with K the number of neighbors returned [165]). Finally, we compute the pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(idx)})$ at node position $\mathbf{r}^{(idx)}$ using Equa. 4.2 and store it in the idx-th position of density buffer.

Algorithm 2: Pilot Density Estimation

$numthreads(PDE_{tx}, PDE_{ty}, PDE_{tz})$
function PDE($kdTree, particles, nodes, res, l, density$):
  Input: $kdTree, particles, nodes, res, l$
  Output: $density$
  $uint3\ DTid$: SV_DispatchThreadID;
  $uint\ idx = DTid_x + DTid_y \cdot res_x + DTid_z \cdot res_x \cdot res_y$;
  if $idx \geq nodes.count$ **then**
    $return$;
  $float3\ query = nodes[idx]$;
  $float\ range = max(l[0]_x, l[0]_y, l[0]_z)$;
  $uint\ retNbrs$; // the number of returned neighbor particles
  $float\ retDists[]$; // the array of neighbor–query distances
  KDTreeSearch($kdTree$, $query$, $range$, $retNbrs$, $retDists$);
  $float\ sum = 0$;
  for $i\ in\ 0 : retNbrs-1$ **do**
    $float3\ r = float3(\frac{retDists[i]_x}{l[0]_x}, \frac{retDists[i]_y}{l[0]_y}, \frac{retDists[i]_z}{l[0]_z})$;
    $sum+ = \max(0, 1 - \|r\|^2)$
    $i++$;
  $density[idx] = \frac{15 \cdot sum}{8\pi \cdot particles.count \cdot l[0]_x \cdot l[0]_y \cdot l[0]_z}$;

Adaptive Smoothing Length (ASL)

Next, we update the smoothing length of each particle based on the result of pilot density estimation. The goal is to assign larger smoothing lengths in sparse regions and smaller ones in dense regions, thereby yielding a smoothly density field in the subsequent stage. Since the update is particle-based, we assign one thread per particle to compute its adaptive smoothing length, which provides a natural and efficient mapping for parallel execution.

We dispatch the **ASL** kernel (Alg. 3) with $\frac{N}{ASL_{tx}} \times 1 \times 1$ thread groups, each containing $ASL_{tx} \times 1 \times 1$ threads, to cover the entire grid domain. For the thread with global index idx, we compute the adaptive smoothing length of the idx^{th} particle as follows.

First, we compute the pilot density $\rho_{\text{pilot}}(\mathbf{r}^{(idx)})$ of the idx^{th} particle using trilinear interpolation and stores the result as `pDen = TrilinearInterp(particles[idx], density)`.

Second, we calculate the arithmetic mean $aveDen$ of the pilot densities across every particle in box B, using Equa. 4.5 for efficient aggregation. To accelerate this step, we design a hierarchical computational method (HCM) that employs parallel reduction with shared memory, thereby reducing costly global memory accesses. Specifically, within each

Algorithm 3: Adaptive Smooth Length

$numthreads(ASL_{tx}, 1, 1)$
function ASL($particles, l, s, density$)**:**
  Input: $particles, l, s, density$
  Output: l
  $uint\ GI$: SV_GroupIndex;
  $uint3\ GID$: SV_GroupID;
  $uint3\ DTid$: SV_DispatchThreadID;
  $uint\ idx = DTid_x$;
  $float\ groupAveDen[\frac{particles.count}{ASL_{tx}}]$;
  $groupshared\ float\ sharedDen[ASL_{tx}]$;
  if $idx \geq particles.count$ **then**
    $return$;
  $float\ pDen = TrilinearInterp(particles[idx], density)$;
  $sharedDen[GI] = pDen$;
  $GroupMemoryBarrierWithGroupSync()$;
  // parallel reduction
  for $i\ in\ ASL_{tx}/2 : 0$ **do**
    if $GI < i$ **then**
      $sharedDen[GI] += sharedDen[GI + i]$;
      $GroupMemoryBarrier()$;
    $i >>= 1$;
  if $GI == 0$ **then**
    $groupAveDen[GID_x] = sharedDen[0]/ASL_{tx}$;
  $WaitForAllThreads()$;
  $float\ aveDen = groupAveDen.Average()$;
  $l[idx]_x = \min\left(l[idx]_x \cdot \left|\frac{aveDen}{pDen}\right|^{\frac{1}{3}},\ 5 \cdot s_x\right)$;
  $l[idx]_y = \min\left(l[idx]_y \cdot \left|\frac{aveDen}{pDen}\right|^{\frac{1}{3}},\ 5 \cdot s_y\right)$;
  $l[idx]_z = \min\left(l[idx]_z \cdot \left|\frac{aveDen}{pDen}\right|^{\frac{1}{3}},\ 5 \cdot s_z\right)$;

thread group, we first allocate a shared memory array `sharedDen`, where each thread stores its locally computed pilot density. We then perform a parallel reduction within the group to compute its average density, which is written to a global memory array `groupAveDen` with N/ASL_{tx} entries. Then, we obtain the overall mean pilot density by averaging all values in `groupAveDen`. Specifically, for thread with global index idx, along with other threads in the same thread group (group index GID), we proceed as follows:

1. Thread idx stores pilot density $pDen$ into the shared memory array `sharedDen` at local

index GI (idx^{th} thread's index within it's thread group), i. e., `sharedDen[GI] = pDen`.

2. All threads in group GID wait until every thread in the same group has updated `sharedDen` with its calculated pilot density, i. e., `GroupMemoryBarrierWithGroupSync()`.
3. All threads in group GID perform a parallel reduction on `sharedDen` to accumulate all density values. After execution, the total value is located at 0^{th} position in `sharedDen` array.
4. The 0^{th} thread in group GID computes the average value of `sharedDen` and then writes it to the global memory array `groupAveDen` at index GID, i. e., `groupAveDen[GID_x] = sharedDen[0]/ASL`$_{\text{tx}}$.
5. All threads wait until every group's 0^{th} thread has written its result to `groupAveDen`, i. e., `WaitForAllThreads()`.
6. Thread idx computes the global average pilot density $aveDen$ as the mean of all elements in `groupAveDen`, i. e., `aveDen = groupAveDen.Average()`.

Third, thread idx computes the smoothing length of the idx^{th} particle using Equa. 4.6 and stores it in the smoothing length buffer.

Algorithm 4: Final Density Estimation

$numthreads(FDE_{tx}, FDE_{ty}, FDE_{tz})$
function FDE($particles, nodes, res, l, density$)**:**
  Input: $particles, nodes, res, l$
  Output: $density$
  $uint3\ DTid$: SV_DispatchThreadID;
  $uint\ idx = DTid_x + DTid_y \cdot res_x + DTid_z \cdot res_x \cdot res_y$;
  if $idx \geq nodes.count$ **then**
    $return$;
  $float\ sum = 0$;
  for $i\ in\ 0 : particles.count-1$ **do**
    $float3\ r$;
    $r_x = \frac{nodes[idx]_x - particles[i]_x}{l[i]_x}$;
    $r_y = \frac{nodes[idx]_y - particles[i]_y}{l[i]_y}$;
    $r_z = \frac{nodes[idx]_z - particles[i]_z}{l[i]_z}$;
    $sum+ = \frac{\max(0, 1-\|r\|^2)}{l[i]_x \cdot l[i]_y \cdot l[i]_z}$;
    $i++$;
  $density[idx] = \frac{15 \cdot sum}{8\pi \cdot particles.count}$;

Final Density Estimation (FDE)

Finally, we recompute the density at each node using the adaptive smoothing length of its contributing particles. We dispatch the **FDE** kernel (Alg. 4), where each thread computes the final density at a single grid node. To cover the grid, we launch $\frac{\mathbf{res}_x}{FDE_{tx}} \times \frac{\mathbf{res}_y}{FDE_{ty}} \times \frac{\mathbf{res}_z}{FDE_{tz}}$ thread groups, each containing $FDE_{tx} \times FDE_{ty} \times FDE_{tz}$ threads. The thread with global index of idx evaluates the final density $\rho(\mathbf{r}^{(idx)})$ at node $\mathbf{r}^{(idx)}$ using Equa. 4.7 and stores the result to the density buffer at position idx. After all threads have completed, we transfer the density buffer from GPU to CPU memory.

4.3.4 Performance Analysis

Before we report our empirical results, we briefly analyze the algorithmic complexity of ScaleFree to highlight its fundamental sources of speedup. These optimizations target the density field computation, which requires recomputation whenever the view changes (e. g., transitioning to a new scale). For pilot density computation, we use a k-d tree to reduce neighborhood search complexity from $O(MN)$ for M grid nodes and N particles to $O(M\sqrt{N})$. To achieve an adaptive smoothing length, we employ a hierarchical method that utilizes thread-group shared memory and parallel reduction, thereby lowering the aggregation complexity from $O(N)$ to $O(\log N)$ within each group of size N. Combined with GPU parallelism, these optimizations enable ScaleFree to scale efficiently to hundreds of thousands of particles and deliver low-latency density field recomputation. To demonstrate performance in practice, next we evaluate our approach on astronomical point cloud data.

Design. To assess the *efficiency* of ScaleFree, following prior work [161], we implemented three density estimation strategies: a sequential version on a single CPU core (SC) as baseline, a parallel version on a multi-core CPU (MC), and the GPU-parallel version (ScaleFree) as described in Sec. 4.3.3. These CPU-based baselines provide a controlled reference that allows us to establish real-time feasibility and to attribute performance differences specifically to GPU parallelization and dynamic recomputation, rather than to algorithmic variation. We compared execution time across the three strategies using the acceleration factor (AF) [166] as a performance metric, defined as $AF = T_{baseline}/T_{target}$. We report the average execution time over 10 runs for all three strategies, with GPU-parallel ScaleFree results excluding CPU–GPU data transfer.

Environment and Implementation. We conducted the experiment on a workstation

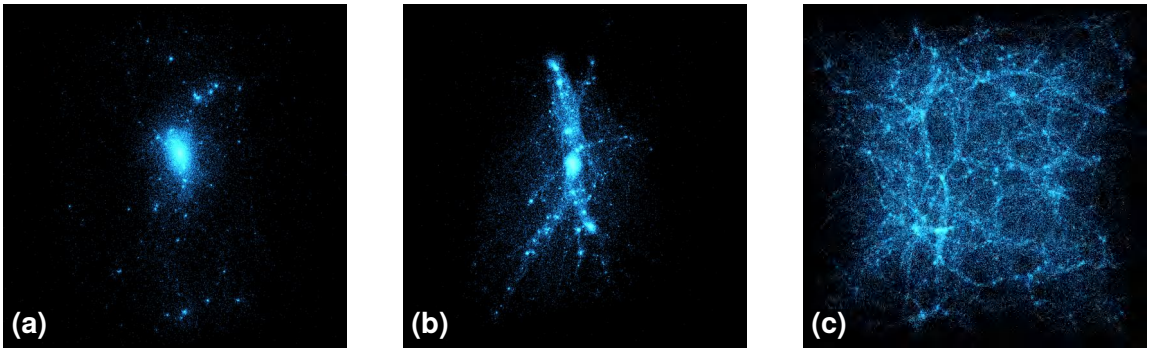


Figure 4.3: Experiment datasets: (a) Nbody 1 (76k points), (b) Nbody 2 (164k points), and (c) Filament (442k points).

running Windows 11 on Intel's 13th generation Core™ i9-13900KF processor (3.0 GHz, 64 GB RAM) and an NVIDIA GeForce RTX 4090 GPU (24 GB of memory) for acceleration. We implemented ScaleFree in Unity3D, with GPU computations realized through compute shaders. We set the resolution res of box B to 64^3 (262k nodes) and the configuration of the thread group dimensions (i. e., $PDE_{tx}, PDE_{ty}, PDE_{tz}, FDE_{tx}, FDE_{ty}, FDE_{tz}$) as 8 and ASL_{tx} as 1024. In addition, we implemented the sequential CPU strategy (SC) using C# and the multi-core CPU strategy (MC) using C#'s Task Parallel Library (TPL).

Datasets. We used three cosmological point cloud datasets: Nbody 1 (76k points, Fig. 4.3(a)), Nbody 2 (164k points, Fig. 4.3(b)), and Filament (442k points, Fig. 4.3(c)). The two N-body datasets feature a dense central cluster surrounded by smaller ones [107], while the Filament dataset depicts a cosmic web with thin filaments[108].

Efficiency. Our results (Tab. 4.1) show that ScaleFree ran substantially faster than SC and MC, with standard deviation consistently within 5% of the mean, indicating stable performance. As dataset size increases, ScaleFree becomes increasingly more efficient relative to CPU implementations, with the acceleration factor rising accordingly. For the 76K-point dataset discretized on a 64^3 grid, ScaleFree achieved an execution time of 0.042 s, corresponding to a sustained rate of about 20 FPS if executed continuously. Although the computation time does not meet the standard 60 FPS, density recomputation only needs to be triggered occasionally, when the view changes substantially. Moreover, the execution times of ScaleFree were generally in the order of 0.3–0.1s or substantially lower, so that the "illusion of animation" remains unaffected [167]. The overall runtime and system fluidity thus remain unaffected.

Table 4.1: Execution time (seconds) and acceleration factor of SC, MC (32 cores), and ScaleFree on a 64^3 grid (262k nodes).

metric	algorithm	Nbody1 (76k)		Nbody2 (164k)		filament (442k)	
		mean	std. dev.	mean	std. dev.	mean	std. dev.
execution time (s)	SC	7.689	0.161	58.373	6.643	395.675	12.458
	MC	1.542	0.041	7.244	0.264	48.514	1.567
	ScaleFree	0.042	0.002	0.119	0.003	0.309	0.011
acceleration factor	SC/ScaleFree	183.1	–	409.5	–	1280.5	–
	MC/ScaleFree	36.7	–	60.8	–	157.0	–

4.4 Multiscale Exploration with ScaleFree

We now discuss how our dynamic KDE approach contributes to improving large-scale scientific data exploration in VR.

4.4.1 Scalable Selection Technique

Selection is a fundamental operation for many subsequent data analyses [47, 27]. Maintaining both efficiency and accuracy, however, is particularly challenging in multiscale point cloud datasets due to their inherent features: highly complex structures, multiple levels of scale, occlusion, and unclear or overlapping boundaries. Designing selection techniques for multiscale point cloud data requires an understanding of user interaction. Users navigate across scales to find a suitable view that allows them to see targets clearly. For ease of use, however, they may also select point clouds even when not at the ideal scale. Selection must thus be supported anywhere and at any scale. As users move across scales, furthermore, they may not want to manually adjust selection parameters—they expect these adjustments to happen automatically. In sum, a selection technique should be dynamic, adaptive, accurate, and efficient, which is exactly what we can realize by fitting our dynamic KDE into the selection workflow. A workflow visualization is provided in Fig. 4.4.

At the start of an interactive session, the global view is initialized. As users navigate to a new scale, we recompute the density field on the GPU using our KDE (Sec. 4.3.3) and then transfer it back to the CPU for subsequent operations. Although we update the density field whenever the scale changes, the GPU pipeline enables rapid recomputation, ensuring smooth navigation.

When users initiate a selection, we invoke a GPU kernel to compute the selection

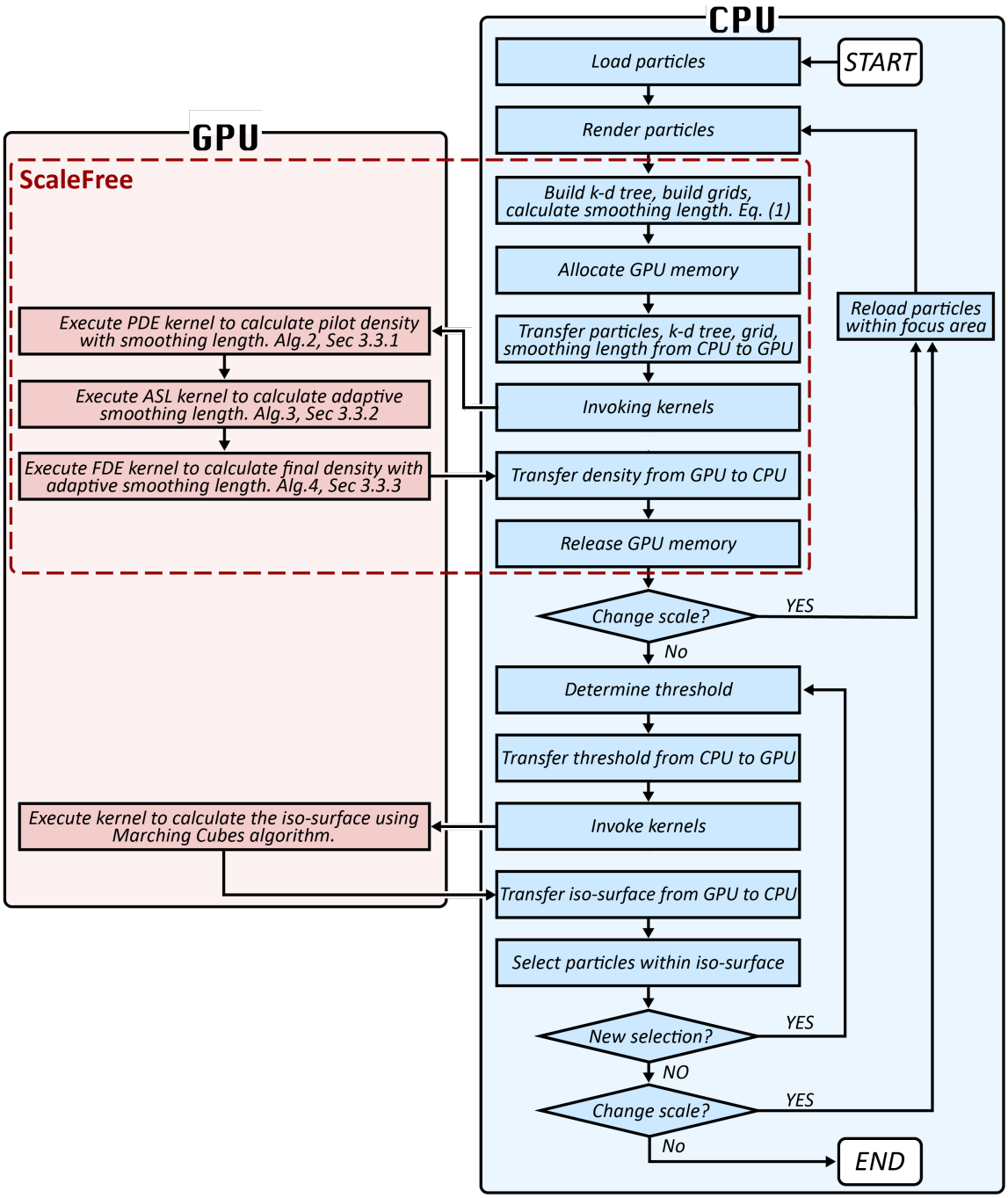


Figure 4.4: Workflow of adaptive selection method.

volume. Specifically, once a selection is initialized, we determine a density threshold for the interacting area and transfer it to the GPU. A kernel then executes the Marching Cubes algorithm [111, 112] to extract the selection volume as an iso-surface based on this threshold. We then select points within the extracted volume. Thanks to our optimized computation, the selection volume is generated rapidly. Even with the MeTAPoint technique [77], which detects density thresholds on-the-fly at the user's interaction point, users can observe updates to the selection volume and assess whether the result aligns with their intent.

Overall, our framework supports adaptive and smooth scale transitions during selection by recomputing density fields on the fly to match users' interactive focus. This enables users to navigate and zoom across scales without noticeable delay. Since the density field is recomputed with our optimized KDE approach once after a view change and remains available for subsequent selections, the technique supports fluid interaction at any scale.

4.4.2 Progressive Navigation Technique

Beyond being a selection technique, in fact, ScaleFree also integrates navigation and zooming into a unified workflow. Navigation is a fundamental exploration task that relocates the viewpoint toward a region of interest while preserving spatial context. It has been widely studied in visualization research across different scenarios, such as gaining insights from volumetric [168] and abstract data [18]. Navigation techniques consist of several components, including wayfinding, travel, and context switching. In large multiscale point clouds, navigation becomes particularly challenging because salient structures are distributed across multiple scales, and dense spatial distribution introduces severe occlusion and clutter. These issues complicate wayfinding and travel, increase the number of travel steps that often occur across scales, and intensify context switching, making it easy for users to lose orientation and spatial context when moving across scales. Navigating such data in immersive environments is even more demanding, as the strong sense of presence can amplify both perceptual challenges and spatial disorientation. To address these challenges, we propose a scale-aware progressive navigation technique that enables users to explore multiscale point cloud data across scales. By defining a region of interest (ROI)—essentially a selection interaction that we have just described—our technique guides users through successive scales (i. e., a combined navigation and zooming interaction) to reach the most suitable viewpoint for observation. Next, we demonstrate how our dynamic KDE can fit into the navigation/zooming workflow to support this process. A workflow visualization is

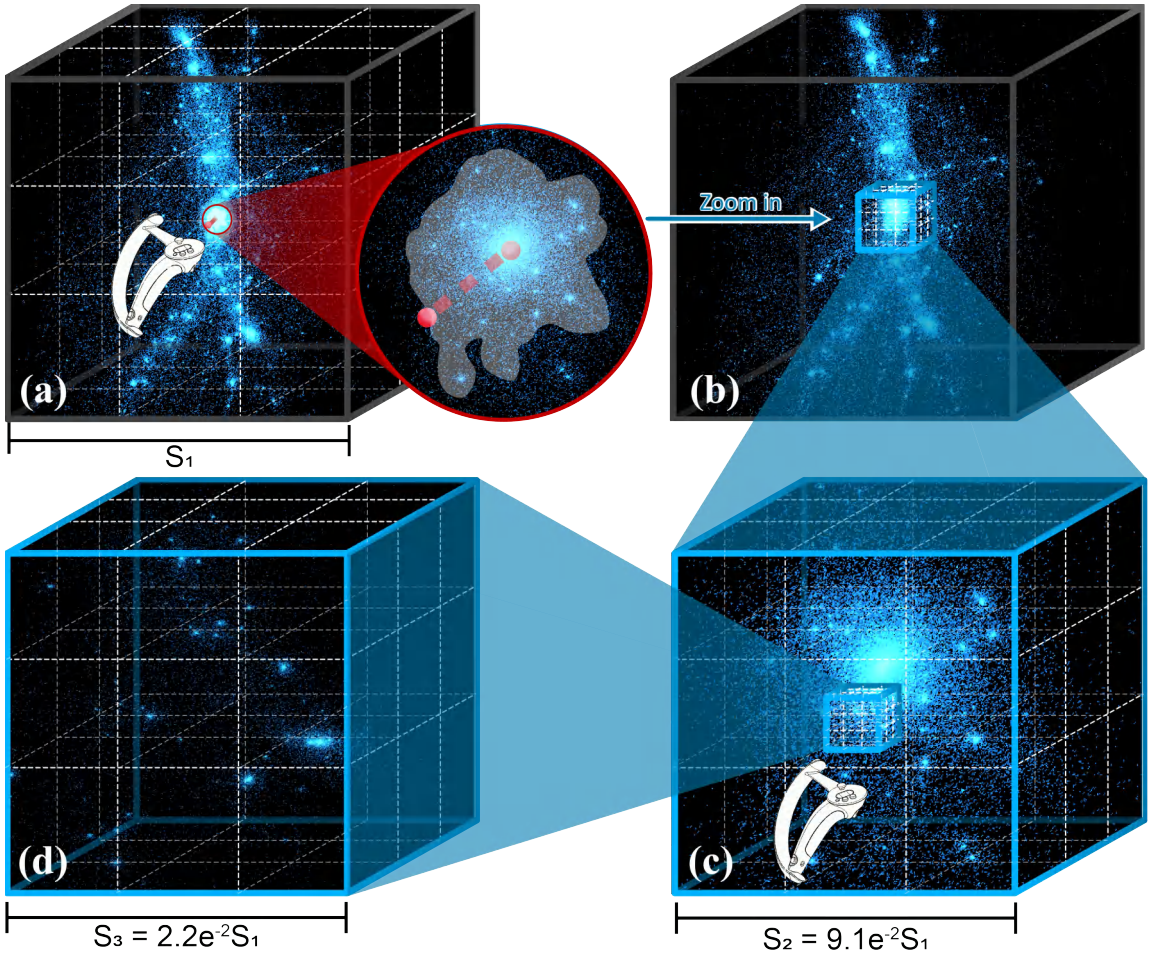


Figure 4.5: Our progressive navigation technique.

provided in Fig. 4.6.

An interactive session starts from a global view of the dataset. As before, we recompute the density field on the GPU using our KDE (Sec. 4.3.3) whenever the view changes substantially. To navigate/zoom to a specific area, users define an ROI with our scalable selection technique (Fig. 4.5(a), Sec. 4.4.1). Because we compute the density field at the current scale, the ROI is immediately identified and we can smoothly transition the viewpoint to the ROI's center (from Fig. 4.5(b) to (c)). We adjust the size of the new viewport to adequately cover the ROI (e. g., the identified selection volume), guiding users to an appropriate scale for observation. Together, this process supports stable viewpoint-driven exploration by preserving spatial orientation and contextual continuity across scale

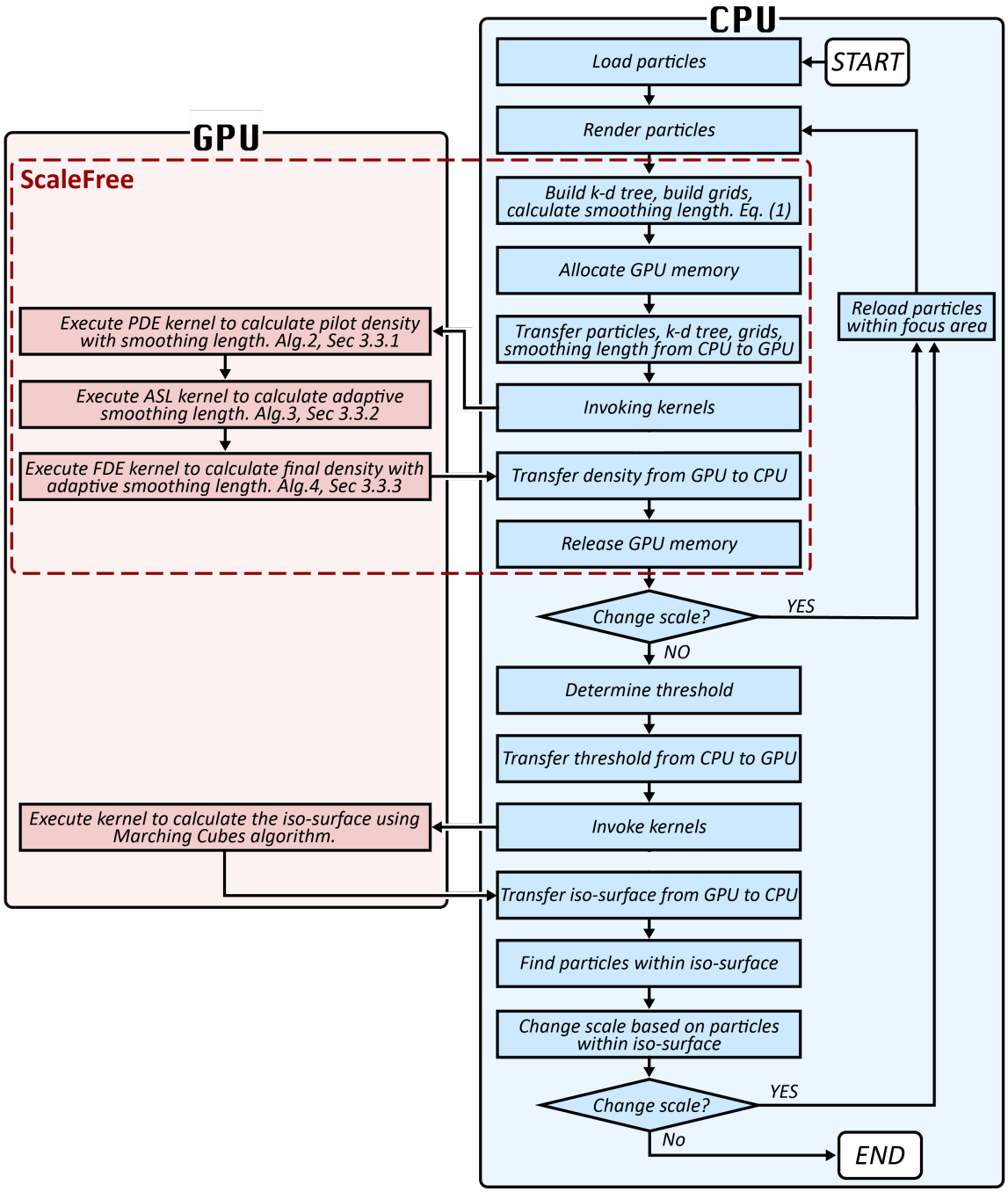


Figure 4.6: Workflow of progressive navigation method.

changes. While the current implementation focuses on the orientation consistency, a promising extension could also be to combine our scale-aware navigation with feature-aware techniques (e. g., [169, 170]) to automatically compute an optimal view orientation, enabling users to travel across scales and arrive at the most informative perspective.

After each viewpoint update, we recompute the density field for the new scale, allowing users to effectively perform further ROI selections that guide subsequent navigation steps (as shown in Fig. 4.5(d)). This progressive "*targeting-travel-context switching*" workflow ensures smooth viewpoint transitions and multiscale navigation without the need for manual wayfinding. In addition, these travel steps can be recorded to enhance spatial awareness during progressive navigation. Overall, our dynamic KDE enables seamless, scale-aware navigation by continuously updating density fields to support progressive ROI selection and smooth view transitions.

4.5 User Study

To evaluate whether our scalable selection method meets the goals of being dynamic, adaptive, accurate, and efficient, we conducted a within-subjects controlled user study comparing it against two density-based strategies: one using a precomputed single resolution and the other using precomputed multiple resolutions. We focus on the selection interaction, as it serves as the foundation for both pure data selection (Sec. 4.4.1) and selection-based progressive data navigation/zooming (Sec. 4.4.1). We pre-registered our study (`osf.io/hfu6e`) and received IRB approval for the protocol (XJTLU University Research Ethics Review Panel, № ER-LRR-00100001205202508130021 21).

4.5.1 Study Design

Participants. We recruited 24 unpaid participants (11 male, 13 female) from the local university, aged 19–30 years (M=23.25, SD=3.22). Among them, 21 were right-handed, two were left-handed, and one was ambidextrous. Twelve participants reported using VR at least once per week, 11 at least once per year, and one had no prior experience with VR devices. Furthermore, 16 participants had obtained a Bachelor's degree or higher. All participants had normal ($n = 16$) or corrected-to-normal ($n = 8$) vision and could clearly distinguish the colors we used in our study.

Apparatus. We used the Vive Pro 2 [171], a PC-based VR head-mounted display (HMD; 2448 × 2448 resolution per eye, 116° field of view, 120 Hz refresh rate). The study was carried out on a PC (Intel 13th Gen Core™ i9-13900KF processor, 3.0 GHz, 64 GB RAM and an NVIDIA GeForce RTX 4090 GPU, 24 GB of memory).

Datasets. We extracted five timesteps from a cosmological N-body simulation [107] and used them as our datasets (shown in Fig. 4.7). These datasets feature stellar clusters distributed across multiple scales, where zoomed-in views reveal progressively detailed substructures. Each dataset contained eight to ten target structures highlighted in yellow, distributed across different scales. Tasks began at varying scales, with all targets becoming visible as participants zoomed in.

Task and Procedure. We instructed participants to select the yellow particles while avoiding the blue ones. Prior to the main experiment and after we had obtained their informed consent, we trained them on the MeTAPoint selection technique [77] using practice datasets. Unlike what Zhao et al. [77] did in their MeTAPoint study, we allowed participants to freely adjust scales (via zooming) and to perform selections at different levels. We also provided undo, redo, and reset functions to restore the initial state if needed. In the main experiment, we asked participants to complete the selections as quickly and accurately as possible. We did not, however, provide suggestions on whether or when the selection was complete. After each condition, we asked participants to report their workload and fatigue using NASA's Task Load Index (TLX) [116]. At the end of all trials, we asked them to indicate their preferred technique and explain their choice, focusing on perceived fluency in multiscale selections. During the interview phase of our user study, we asked our participants the following questions: **Q1. Perceived delay during interaction.** Did you perceive any noticeable delays, lags, or interruptions during scale changes or while making selections when using each technique (PS, PM, and DR)? If yes, please describe when the delay occurred (e.g., during scale transitions, selection, or navigation) and how noticeable it was. **Q2. Impact of delay on interaction experience.** If you noticed any delays or performance slowdowns, did they affect your ability to navigate or select regions effectively? Please explain your reasoning. All participant responses are shared on OSF. A whole session lasted approx. 45 minutes.

Density Field Conditions. We asked our participants to perform all selections using the same density-based MeTAPoint method [77]: they initiated a selection by pointing at or near a target cluster and then dragged along its boundary while holding the VR controller

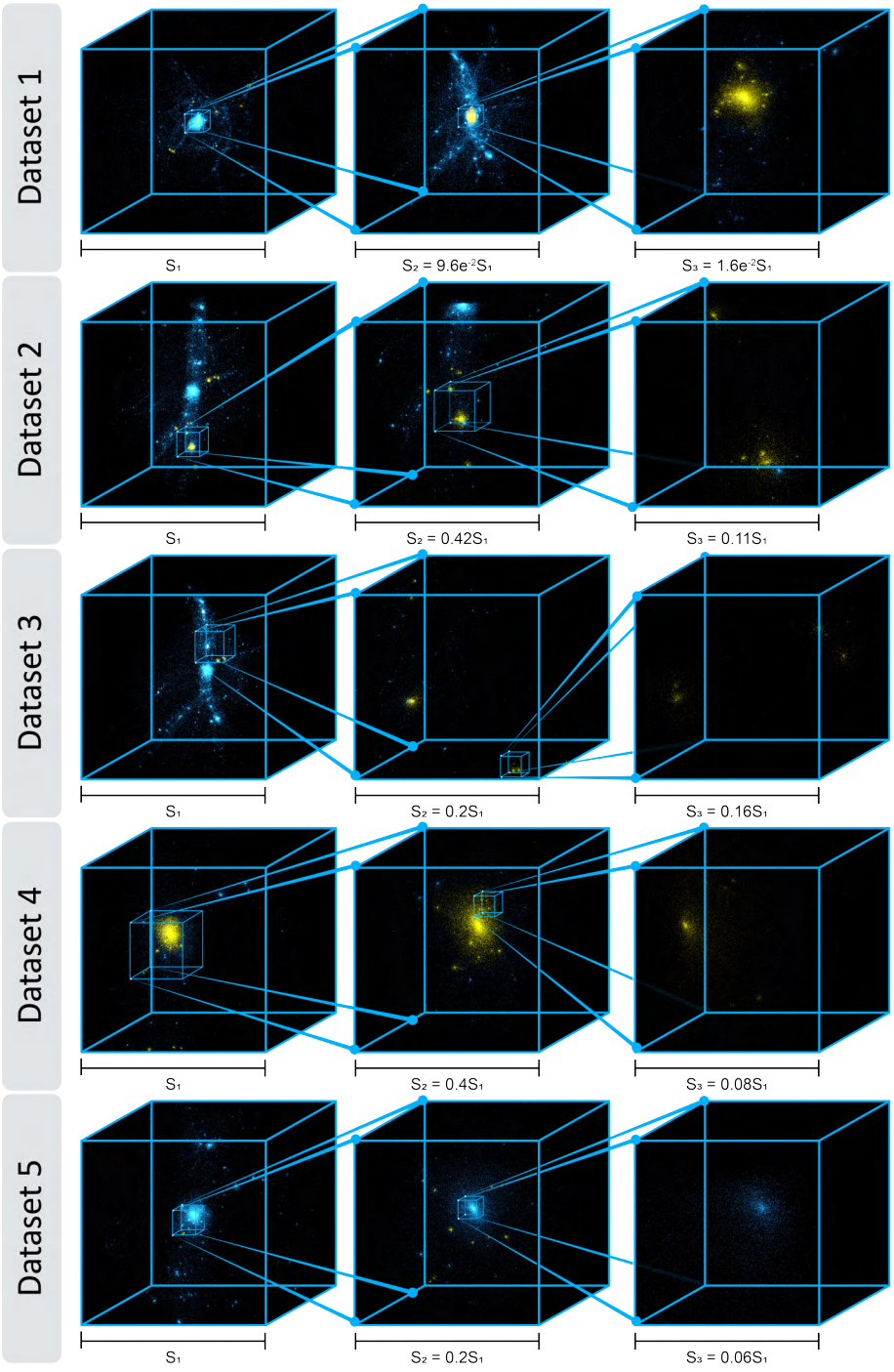


Figure 4.7: Full datasets we used in the study.

trigger. The underlying density field computation, however, differed across approaches, resulting in three experimental conditions:

Precomputed Single-resolution (PS): The density field is precomputed once at a fixed resolution (64^3) before the task. During the task, participants interact with this static field, which remains unchanged across all scale levels.

Precomputed Multi-resolution (PM): The density field is precomputed at two resolutions (64^3 and 128^3). Users interact with the precomputed field interpolated by fields according to their scale level. We use two reference scales, S_{max} and S_{min}, which are predefined—at S_{min} the highest-resolution field is used, whereas at S_{max} the lowest-resolution field is employed. For the current scale S_c during exploration we determine the corresponding mipmap level $ml \in [0, 1]$ by

$$ml = \frac{S_c - S_{min}}{S_{max} - S_{min}}. \tag{4.9}$$

and use ml to obtain the immediate density field via linear interpolation between the two fields with different resolutions.

Dynamic Resolution (DR): We compute the density field in real time via ScaleFree (Sec. 4.4.1). We set the field resolution to 64^3 and configure the thread group dimensions as PDE_{tx}, PDE_{ty}, PDE_{tz}, FDE_{tx}, FDE_{ty}, $FDE_{tz} = 8$, and $ASL_{tx} = 1024$.

While PS represents the conventional approach of using a fixed-resolution density field for density-based selection [58, 59, 77], PM further optimizes this strategy by adopting a mipmap-inspired hierarchy from computer graphics, enabling multiscale access through precomputed density levels without real-time recomputation. By comparing DR against two representative paradigms, we assess dynamic KDE performance in selection tasks.

Design. We counterbalanced the three density field conditions using full permutation (in total six possible orders). We assigned the first six participants one order each based on $P_{ID} \bmod 6$, and repeated the scheme for every subsequent group of six. In total, 24 participants × 3 methods × 5 datasets yielded 360 trials.

Measures and Analysis. We recorded accuracy, completion time, as well as the transition times in the study. Given the criticism of NHST in the analysis of experimental data [117, 118, 119, 120], and APA's advice to seek other methods [121], we present our findings using estimation techniques that report effect sizes and confidence intervals instead of relying on p-value statistics.

Accuracy: Similar to Yu et al. [58, 59, 77], we evaluated the accuracy using two metrics: F1 and MCC (Matthews correlation coefficient). To calculate these, we identified true positives (TP; correctly selected particles), false positives (FP; incorrectly selected particles), false negatives (FN; target particles that were not selected), and true negatives (TN; correctly unselected particles). We defined precision as $P = TP/(TP+FP)$ and recall as $R = TP/(TP+FN)$, and calculated F1 as $\mathrm{F1} = 2 \cdot (P \cdot R)/(P + R)$. While F1 reflects the harmonic mean of precision and recall, it does not account for TN. Thus, we also computed MCC, defined as:

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

We normalized all accuracy scores, and computed means and 95% bootstrap confidence intervals (CIs; $n = 24$).

Completion Time: We analyzed completion times using exact CIs on log-transformed data ($n = 24$), reporting geometric means and mean-comparison ratios.

4.5.2 Hypotheses

We formulated the following hypotheses:

H1: DR yields higher accuracy than PM and PS.
H2: PM yields higher accuracy than PS.
H3: DR requires less completion time than PM and PS.
H4: PM requires less completion time than PS.
H5: DR has a lower cognitive load than PM and PS.
H6: DR has a higher preference than PM and PS.

The rationale behind **H1** is that density fields in DR are dynamically adjusted to the current scale during navigation and zooming. At any scale, the density field is recomputed with a resolution 64^3, ensuring consistent accuracy. By contrast, PS and PM rely on pre-computed density fields: when zoomed in, they use coarser grids that reduce precision. Nevertheless, PM integrates density fields at two scales rather than just one as in PS. Therefore, selections made with PM should be more accurate than those with PS, as stated in **H2**. The rationale behind **H3** is that DR provides the most accurate results, making selections easier for users to accept and thereby reducing completion time. Although the density fields in PS and PM are pre-computed and do not require on-the-fly computation,

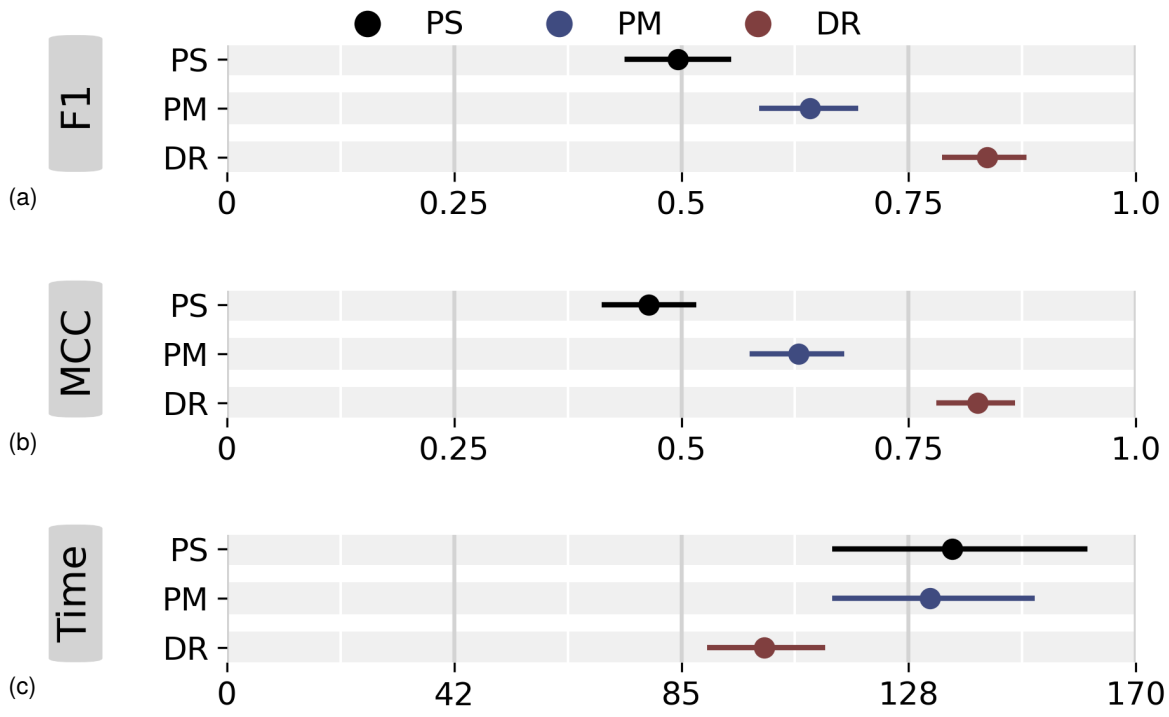


Figure 4.8: The mean of (a) F1, (b) MCC, and (c) time, with 95% CIs.

users may spend additional time refining their selections to achieve greater precision. A similar reasoning applies to **H4**: because PM yields more accurate results than PS, users are more likely to accept the selection outcomes with less refinement, resulting in shorter completion times. In addition, DR enables users to select target points accurately at any scale without the need to find an optimal view or repeatedly refine their selections, which reduces both effort and mental demand—thereby lowering cognitive load (**H5**). Consequently, users are more likely to prefer DR over PS and PM (**H6**).

4.5.3 Results

In Fig. 4.8 we show the mean completion times and two accuracy metrics (F1 and MCC) with 95% confidence intervals across density field conditions (Precomputed Single-resolution (PS), Precomputed Multi-resolution (PM) and Dynamic Resolution (DR)). We further provide the pairwise ratio with 95% confidence intervals in Fig. 4.9. Tab. 4.2 reports the numerical values of average mean task completion times and accuracy scores, whereas Tab. 4.3

presents the pairwise ratios.

We now report statistical results in relation to our hypotheses.

Accuracy. For F1, a score of 1 indicates perfect performance and 0 the worst. For MCC, a score of 1 indicates perfect performance, while -1 represents the worst performance. As shown by the mean F1 scores (Fig. 4.8(a)) and MCC scores (Fig. 4.8(b)), DR outperformed both PM and PS in accuracy. The pairwise comparisons reveals that DR achieved 1.21–1.44× higher F1 score and 1.22–1.49× higher MCC than PM, and 1.58–1.92× higher F1 and 1.67–2.18× higher MCC than PS. **H1 is supported**. In addition, PM outperformed PS in both mean F1 and MCC. The pairwise comparisons show that PM achieved 1.21–1.53× higher F1 and 1.26–1.75× higher MCC than PS. Thus, **H2 is also supported**.

Completion Time. Fig. 4.8(c) shows that selections with DR required less completion time than PM and PS. The pairwise comparisons indicate that DR took 0.70–1.08× as long as PM and 0.66–1.08× as long as PS, suggesting that most of participants completed the tasks more quickly with DR. In a few cases, ratios were only marginally above 1.0, reflecting participants who finished marginally faster with PM or PS. We made related observations during the study: 4 participants ended the tasks early once they realized that they could not achieve satisfactory outcomes with PS or PM. These behaviors, however, do not affect the overall result. We confirm that **H3 is supported**. Our results also showed that PM had a slightly shorter mean completion time than PS. The pairwise comparisons indicate that PS took 0.87–1.40× longer than PM, suggesting that the advantage of PM over PS was modest rather than substantial. Thus, **H4 is not fully supported**.

Table 4.2: The mean task completion times, accuracy scores, and their corresponding 95% confidence intervals.

Technique	Time	CI	FI	CI	MCC	CI
PS	136s	[113,161]	.50	[.44,.55]	.46	[.41,.52]
PM	131s	[113,151]	.64	[.59,.69]	.63	[.58,.68]
DR	100s	[90,112]	.85	[.79,.88]	.83	[.78,.87]

Table 4.3: The pairwise ratio of task completion times, accuracy scores, and their corresponding 95% confidence intervals.

Technique	Time	CI	FI	CI	MCC	CI
DR/PM	0.83	[0.65,1.02]	1.33	[1.21,1.44]	1.35	[1.22,1.49]
DR/PS	0.80	[0.60,1.02]	1.74	[1.58,1.92]	1.90	[1.67,2.18]
PM/PS	0.89	[0.63,1.14]	1.36	[1.21,1.53]	1.48	[1.26,1.74]

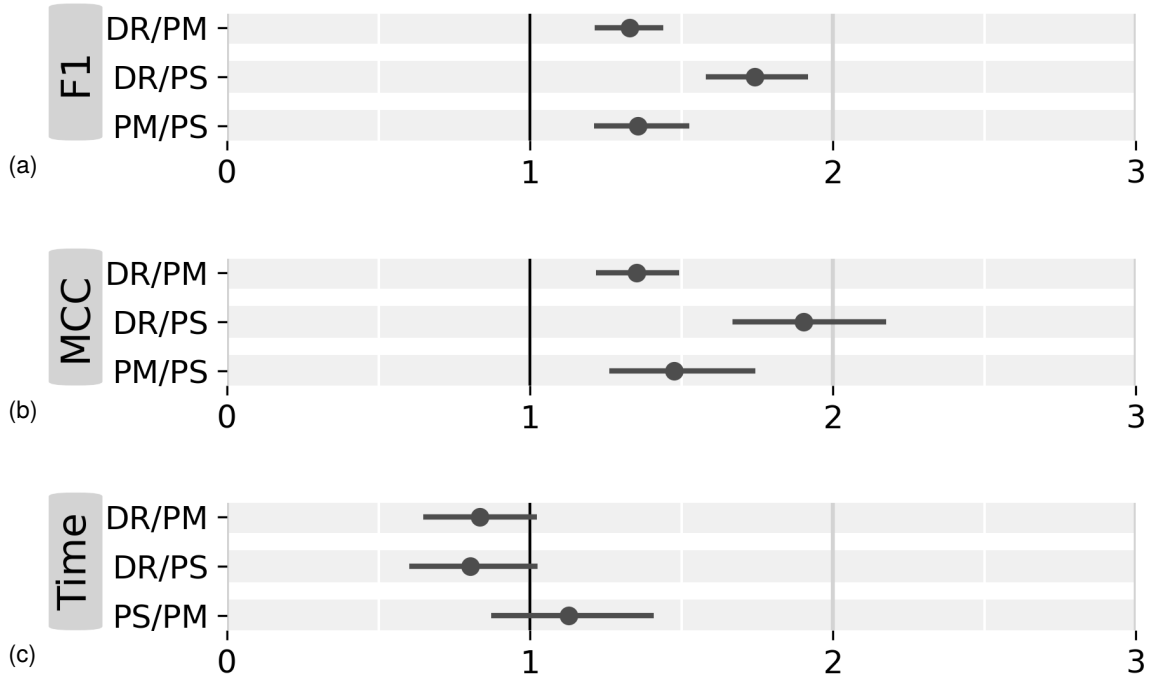


Figure 4.9: Pairwise ratio of (a) F1, (b) MCC, and (c) time, 95% CIs.

Workload and Preference. As shown in Fig. 4.10, participants reported greater efficiency, lower frustration and time pressure, and reduced mental and physical effort with DR compared to PM and PS. They ranked the conditions consistently ($DR > PM > PS$) across speed, accuracy, preference, and fluency (Fig. 4.11). Thus, **H5 and H6 are supported**.

Interview. In follow-up interviews, we asked participants whether they perceived delays during scale changes (where recomputation happens at the end of scale change) and selections across the three techniques. Most participants ($n = 20$) reported no noticeable recomputation delay with DR, while four noted slight delays only when the number of points in view was very large. These delays, however, did not affect their navigation experience. In contrast, a majority of participants ($n = 18$) identified PM as producing the most prominent lag: when using MeTAPoint to select large regions and generate extensive selection volumes, they observed noticeable drops frame rate. This performance degradation stems from the underlying characteristics of PM: higher-resolution grids increase the number of triangles needed to construct the selection volume and impose heavier computational demands when determining selection thresholds. These findings highlight scenarios where

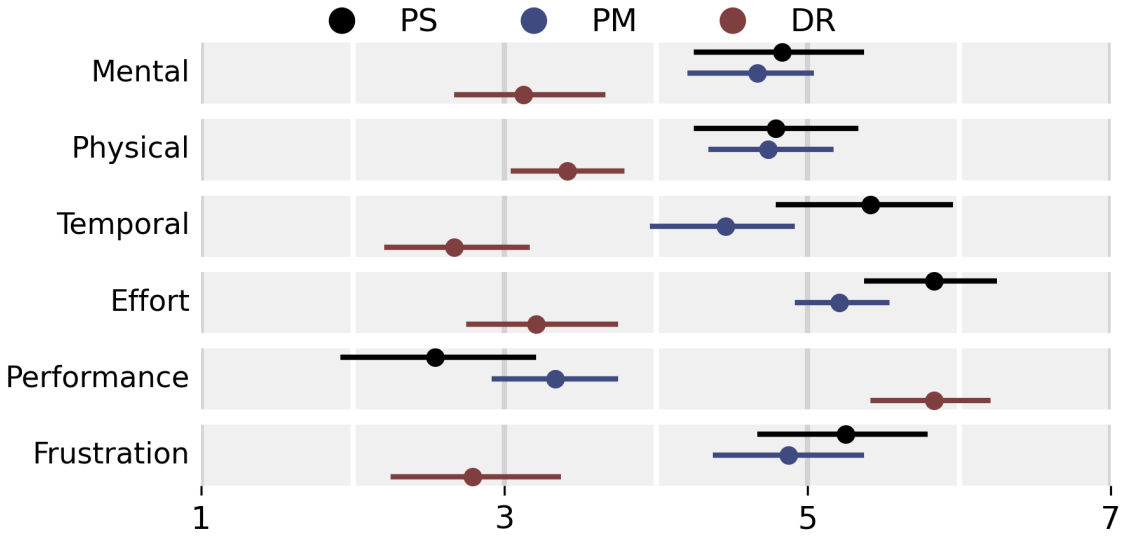


Figure 4.10: User workload and performance. Error bars: 95% CIs.

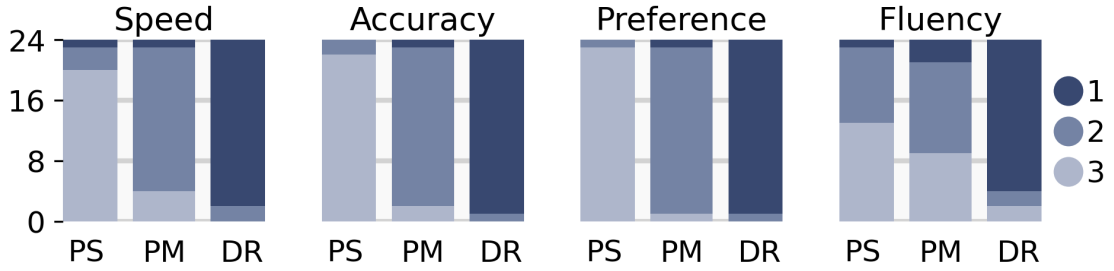


Figure 4.11: Participant rankings on task performance and preference.

dynamic density fields offer substantial advantages over precomputed higher-resolution fields, particularly for sparse or large-scale datasets. In such cases, precomputed fields must rely on densely sampled grids to capture comparable detail, whereas dynamic estimation adapts density computation on demand, enabling the capture of fine-grained structures while maintaining smooth frame rates.

Summary. Improvements in F1 and MCC (**H1**) and reduced completion time (**H3**) show that our dynamic KDE improves both selection accuracy and efficiency over precomputed methods. By maintaining scale-accurate density fields, it enables high-fidelity selection with less effort and lower mental demand (**H5**), and was therefore preferred by participants (**H6**). These results highlight that on-demand recomputation is essential for precise immersive interaction, as precomputed methods miss fine-scale details and require adjustments.

4.6 Discussion

In this section, we discuss how ScaleFree extends to support broader tasks, analyze the trade-offs between dynamic and precomputed fields, limitations and future directions.

4.6.1 Extending ScaleFree

Point cloud data often lacks a natural scale. Such datasets consist of unstructured points scattered unevenly across scales, sometimes with vast empty ranges, as in astronomical simulations. As a result, interaction and visualization approaches must help users orient themselves within scales and navigate across them. Such tasks require effective methods to compute key features, such as density, as the data changes—forming the starting point of our work. Multiscale data often contains fine details embedded within broader structures, creating large gaps between scales. A key feature of our dynamic KDE is that it allows users to explore subtle or complex structures on demand, with control remaining in the users' hands. For example, by dynamically adjusting resolution based on the interaction history and on the local data distribution, the navigation approach can guide users toward scale levels where meaningful features emerge. A future direction of research, however, is to determine how our KDE method can be adapted to better handle time-varying particle distributions in dynamic simulations, where particles move and clusters evolve over time.

4.6.2 Trade-offs

We conducted performance experiments to evaluate ScaleFree, comparing its execution time against density estimation on single-core and multi-core CPUs. We also compared selection techniques using ScaleFree with those based on precomputed density fields in terms of accuracy and efficiency. These experiments highlight the trade-offs between dynamic and precomputed approaches, offering guidance for future researchers in selecting suitable solutions.

When designing an appropriate strategy, factors such as data scale, hardware capacity, and algorithmic complexity must be considered. Our results show that dynamic KDE consistently delivered higher accuracy and faster computation than precomputed density fields. We also found that multi-resolution density fields achieved better accuracy than single-resolution ones, though with only marginal gains in speed. This suggests that for datasets

of moderate scale, precomputing a few density fields may be sufficient—especially when there is only limited memory available—since users can quickly access and switch values without recomputation. For datasets spanning multiple scales, however, dynamic KDE offers clear advantages by computing density fields on demand and supporting seamless multiscale exploration—yet these advantages come with a higher demand for GPU support and GPU/CPU memory and a more complex implementation. Still, an effective strategy is needed to determine when recomputation should occur, as users may notice delays during density field updates. To address this point, we trigger recomputation after users transition to a new scale. While users take time to observe the data at this scale, the system prepares the density field on the GPU, ensuring readiness for subsequent operations.

4.6.3 Limitations and Future Work

Several limitations remain that open avenues for future work.

First, although DR dynamically recomputes the density field at a fixed resolution of 64^3, fine details may still be missed. Although progressive navigation can reveal these features step by step at finer levels, it remains ineffective. Future work could adopt adaptive resolution schemes that allocate denser sampling to regions with high particle density, strong density gradients, or boundary-like structures, so that fine structures within the region of interest can be better preserved.

Second, our dynamic KDE can be further optimized. Currently, recomputation is triggered even for minor scale changes, leading to unnecessary overhead during frequent zooming. To address this issue, we can introduce a threshold for scale changes, ensuring that recomputation occurs only when the change exceeds a meaningful level. This improvement would reduce redundant updates while maintaining responsiveness. Similarly, we could only compute the KDE for subsets of the dataset at any given zoom level (as opposed to the whole data space), such as the size of the viewport plus some margin around it. Then, recomputation for other regions would be triggered when the user approaches the boundary of the currently computed density field during navigation.

Third, the main time cost of the current recomputation delay in our selection technique lies in transferring the computed density field from the GPU back to the CPU. Future implementations could eliminate this bottleneck by migrating the selection algorithm, or any other interaction entirely onto the GPU, thereby avoiding density transfer from GPU to CPU.

Finally, this work is driven by the demands of immersive multiscale point cloud exploration and therefore focuses on GPU-parallel implementation and system-level optimization of adaptive KDE, rather than proposing new density estimation formulations. While recent KDE methods [162, 172] improve theoretical efficiency and memory usage, integrating them into a real-time interactive VR pipeline remains an important direction for future work.

4.7 Conclusion

In this work, we present a GPU-accelerated adaptive KDE technique that dynamically recomputes density fields for interactive multiscale exploration. Through two use cases—navigation and selection—we demonstrate that, as an independent module, our ScaleFree can be directly integrated into diverse pipelines requiring high-performance density estimation in large-scale scientific simulations.

Beyond its technical optimization, our ScaleFree allows us to explore and pursue a novel interaction paradigm for immersive data exploration. Rather than requiring users to learn dedicated 3D navigation techniques—a long-standing challenge due to the complexity and inherent imprecision of 3D interaction—users essentially simply issue commands akin to "*show me that in more detail*." The structure- and context-aware selection paradigm that we paired with the rapid KDE enabled through our ScaleFree approach allows users to simply indicate desired regions of interest to steer their exploration. They can then zoom in to reveal fine-scale detail or adjust their target to focus on another part of the data—ultimately fluidly crossing the scales to find the most informative perspective.

Additional Material Pointers

We share our additional materials (study results/data and analysis scripts, videos) at `osf.io/hfu6e`. We also make our prototypical implementation of the ScaleFree available at the github repository `github.com/LixiangZhao98/ScaleFree`.

APPENDIX 5

Collaborative Multi-scale Spatial Exploration in VR

In this chapter, we address **RQ3: How can immersive environments support effective navigation across multiple spatial scales in spatial datasets?** By default, the user usually explore the spatial data visualization on a single view in the immersive environment. ScaleFree (**Chapter 4**) supports the efficient construction of field representations across scales through a single view. However, a single view offers limited support for understanding the broader multi-scale structure of the data. Without an overview of the global context and the detail views, users often struggle to maintain awareness of their current location, orientation, and scale they are involved in. This chapter focuses on complementary interface designs that provide contextual grounding for navigation across large, unstructured, and multi-scale spatial datasets.

In this chapter, we propose two interface designs that support multi-scale spatial data visualization and navigation in immersive environments. The first design leverages an overview+detail representation to facilitate multi-scale data exploration by managing both global context and local detail view. The second design introduces a linked World-in-Miniature interface, which connects multiple user-specific WiMs, enabling users to easily perceive essential contextual information at a glance, including the relative scales, spatial locations, viewing directions, and surrounding structures.

5.1 Introduction

The immersive environment proposes new prospects for multi-user collaborative data exploration and information exchange in a shared space via HMDs. For instance, the use of immersive environments has applicable potential in the field of astronomy, helping scientists explore the unknown and verify research predictions. Through cosmological simulations, scientists predict that dark matter generates cosmic webs, and the filaments in the web deliver the energy and matter to the high-density cluster area such as galaxies. To discover patterns and confirm their prediction, scientists need to select and explore astronomical data, such as 3D cloud point data from the cosmological simulation [5, 80]. However, one of the inherent features of astronomical data is that these points are widely spread in enormous spatial ranges and contain abundant details at different levels of magnitudes. An immersive virtual reality environment can facilitate user engagement in the data exploration process so that the data can be examined at vastly different spatial resolutions to unveil interesting or unexpected patterns. Moreover, collaborative interaction in such an immersive environment shows great potential in strengthening teamwork and communication, allowing more holistic and efficient spatial exploration.

The *awareness* of collaborators plays a significant role in effectively exploring the immersive environment, including awareness of presence, location, actions, and activities [173]. The awareness can influence the quality of collaborative work by allowing users to find and feel each other at any time. First, visualizing the location, actions and activities of their collaborators improves the user's *spatial and situational awareness*. It endows a sense of knowing where the user is, where other collaborators are, what other collaborators are doing and what is happening around them. Second, the opportunity to visualize collaborators within a shared virtual space prompts *insight sharing*. Insight sharing is one of the critical features of collaborative exploration, especially when the data is unknown and with exhaustive details.

The goal of our study is to design an intuitive and effective VR interface that enhances collaborators' spatial awareness and supports them in sharing views and insights for astrophysical data exploration. For visualizing and representing collaborators, a common approach from the literature is using VR avatars, the virtual representations of users. However, when navigating a large-scale virtual environment, such as the simulation of the astrophysical universe, collaborators may focus on specific parts in different spatial scales

and lose sight of the big picture and other collaborators' locations. More specifically, we identify three challenges as follows:

C1: Multi-scale. The astrophysical data often contains a large number of astronomical structures distributed on numerous scales in the Universe dominated mostly by emptiness. If the collaborators are represented as scaled avatars, visualizing other collaborators becomes challenging since their virtual avatars may look too large or too small in the user's scale of observation. Furthermore, in order to reach another user's location, they need to navigate across huge spatial scales.

C2: Large and empty space. On a specific scale, the astronomical data distributes widely, and empty spaces occupy most regions of the environment [174]. As a result, ROI are spatially segregated from each other. Thus, in the exploration, users may not be able to see their collaborators if they are exploring different parts of the data.

C3: Communication and assistance. In order to identify ROI in the immersive environment, users may need assistance from a third-person point of view. However, sharing views and findings about the large-scale point cloud data in the immersive environment is extremely challenging due to the complex data environment.

In this work, we propose two layouts. One is a set of four Overview+Detail layouts designed to support collaborative exploration of unstructured, multi-scale scientific data in VR. These layouts vary in how global context and local details are distributed, shared, or individualized across collaborators. These four layouts examines what information should be shared—overview, details, or both—and how these representations should be organized to preserve context, support multi-scale navigation and facilitate joint analysis of large-scale structures. The other one is L-WiM (Linked World-in-Miniature), an intuitive and effective user interface for collaborative data exploration in an immersive environment. We used astrophysical data as an example application. With L-WiM, users can observe rendered images of other collaborators and their views, including the contextual environment and the tasks being processed. Users are aware of information about their collaborators' locations and scales. Furthermore, users can share and communicate their insights through interactions and visual cues.

In summary, we contribute

- design considerations for collaborative multi-scale spatial data exploration, focusing on how collaborators maintain shared context, individual exploration space, and

awareness of others' positions and scales.

- a set of four Overview+Detail layouts that distribute overview and detail views across shared and individual workspaces.
- a linked World-in-Miniature interface that organizes collaborators' local exploration contexts into a tree structure, enabling users to be aware of collaborators' positions, scales, and focus regions.
- interaction mechanisms that support communication, information sharing, and navigation among collaborators.

5.2 Related Work

This section reviews the recent work related to collaborative data exploration in VR and, in particular, the world-in-miniature technique and Overview+Detail technique.

5.2.1 Collaborative Data Exploration in VR

Collaborative exploration is an effective approach for analyzing complex datasets [175]. It supports a group of users to perceive, observe, and manipulate the data in a shared space [176]. Users are supported to collaborate in two ways: 1) in the same physical space (co-located collaboration) with distinct virtual avatars in the shared and physical space, communicating via discourse, gestures and gaze or 2) link to the same immersive environment via different hardware in multiple physical locations (remote collaboration).

The Cave Automatic Virtual Environment (CAVE), the responsive workbench [177] and Fishtank [178] were a few of the earliest immersive techniques for co-located collaborative data exploration [179]. Later, the new generation of the CAVE style product CAVE2 was developed [180] and the use cases were proposed by Marai et al. [181]. However, the CAVE-style immersive environment requires multiple screens projected on walls in a large room, which can be costly. In addition, only one tracked user can receive accurate stereoscopic vision, which makes it even less cost-effective.

In recent years, VR HMDs have been gradually employed in immersive collaborative data exploration. Users connect to the shared virtual space via different hardware and possess a unique view. Previous research has studied the relative merits of VR HMDs for collaborative work. The results indicated that VR HMDs achieved the same level of accuracy

as CAVE-based systems while being more portable and affordable in the meantime [182]. Hence, VR HMDs have been utilized in many collaborative system designs. Several notable designs are iVIZ [175], an internet-based collaborative platform for immersive visualization, Telearch [183], a collaborative and interactive archaeological exploration platform and FIESTA [184], a system concentrated on the collaborative pattern in a free-organizing co-located immersive environment with multiple users.

Collaboration is built on the foundation of *workspace awareness*: the understanding of one's collaborators' activities and the status of the environment [185]. People obtain workspace awareness through conversations, gestures, and information produced by bodies and artifacts, which help them coordinate, anticipate each other's actions, and solve problems together [186]. VR environments facilitate workspace awareness by offering a shared workspace for multiple collaborators, enabling them to see each other's embodiment or avatars [173]. Visual cues such as field-of-view frustum and gaze rays, moreover, can be implemented in VR to further indicate each person's focus [187]. Workspace awareness is especially important for the collaborative exploration of large-scale unstructured data, on which we focus in this work, as collaborators may explore features at different scales, rendering themselves invisible to each other in a VR environment. To enhance workspace awareness among collaborators, we employ indicator boxes in the overview to represent each user's location and scale of exploration.

Given VR's support for shared workspaces, prior research has explored collaborative VR environments in various domains, including education [188], training [189], and gaming [190]. Donalek et al. [175] emphasized VR's potential in facilitating collaborative data exploration, highlighting that it allows users to interact with both data and collaborators in an immersive environment. Previous research in VR-supported data exploration, however, has focused more on integrating traditional visualization techniques into immersive environments [191, 192], while studies examining how users employ VR for collaborative data exploration remain relatively limited. We aim to fill this gap by analyzing users' collaboration in conducting visualization tasks through a mixed-methods approach.

Notably, when collaborating in a shared space, collaborators tend to retain individual spaces for personal use, which is identified as *territoriality* [193]. VR supports this behavior by offering flexible mechanisms for users to establish personal workspaces and engage in independent interactions. One method, known as subjective views [194], relies on customized content based on individual tasks and preferences. While effective, this approach

can introduce perceptual discrepancies, potentially impairing workspace awareness and communication. An alternative approach involves partitioning the VR space into multiple subspaces [195, 196], creating separate manipulation areas within a cohesive shared workspace. These subspaces can be further configured with access controls to balance collaborative and individual work. Questions remain, however, on which content should be shared and which should remain private.

5.2.2 Overview + Detail

The core idea of the Overview+Detail paradigm is to provide two coordinated display spaces that present the same underlying data at different levels of scale. By separating global structure from fine-grained details across distinct views, Overview+Detail maintains visual continuity and allows users to access both contextual and local information simultaneously. This technique enables simultaneous access to global and local data representations, supporting user orientation and analysis.

Prior work has applied Overview+Detail techniques in the single-use immersive environment. Yang et al. [197] conducted a systematic study of immersive navigation techniques for 3D scatterplots, by comparing room-sized visualization and zooming interfaces, each evaluated with and without an overview visualization. Coffey et al. [2] introduced Interactive Slice WIM, a table+wall VR framework that extends the World-in-Miniature metaphor to provide an Overview+Detail visualization of volumetric data by combining a floating 3D miniature with touch-controlled 2D slice projections for efficient navigation and interactive exploration. Sorger et al. [198] introduced an Overview+Detail navigation metaphor for large dynamic network exploration that combines flying for global overview with controller-based teleportation to view the detailed structures, showing through expert feedback that maintaining an overview is essential for preserving users' spatial orientation. Usher et al. [66] proposed a VR-based neuron tracing system that incorporates a minimap-style Overview+Detail design, where the global dataset visualization, current focus region, and previously traced structures are presented in a compact overview to support spatial orientation while users perform fine-grained tracing operations in the detailed 3D focus view.

Building on this line of work, our research investigates how Overview+Detail visualization can be extended to collaborative settings. Specifically, we evaluate multiple layout configurations and how they support shared awareness, coordinated navigation, and joint

reasoning during multi-scale scientific data exploration in VR.

5.2.3 World-in-Miniature

World-in-Miniature (WiM) [16] is a technique that provides a scaled-down version of the whole or a part of the virtual environment. The technique has been effectively applied to architecture [199, 200, 201] and medical visualization [2]. More recently, Danyluk et al. [202] defined WiM as "a scale replica within and linked to the original region through virtual feedback or interaction". When a user manipulates objects through a WiM, the changes to the WiM will be synchronized into the original world. Similarly, modifications to items from the original world will be reflected in WIM objects. Users can utilize WiMs to edit items that are out of reach. Furthermore, WiM can also be utilized as a navigation tool [2, 203]. A limitation of WiMs is that the data visualized with a broad scope (large-scale) contains many invisible details. Li et al. [174] proposed SWiM to support fast scale transition in an astronomical environment based on the Power-law spatial scaling method. They leveraged logarithmical mapping approach to visualize the miniature spaces, providing an overall view of the whole astronomical spreading on a wide range. Wingrave et al. [138] introduced SSWIM to overcome the large-scale problem in city simulated data. SSWIM allows the WiM to show parts of the whole data with a user-defined visualization scale and region, supporting the navigation and manipulation of the buildings in a vast city in an immersive environment. Although the exploration of large-scale data via WiM has been investigated, a relatively small body of literature is concerned with collaborative application with WiMs [202]. Stafford et al. [204] proposed "god-like collaborative interaction" in which the user indoor manipulates a WiM projected by a tabletop display system, while the user outdoor observes the reconstructed model reflected from the indoor WiM via a mobile augmented system. Chheang et al. [205] developed the group WiM (GWiM) system, which provided a team navigation method via WiM guided by a leader. The WiM technique in collaborative data exploration has not been studied. A single WiM is able to display the contextual information and support interactions that are out of reach. We believe WiMs, combined with small multiples [206], is an effective technique for collaborative exploration in spatial data.

5.3 Design One: Overview + Detail

In the first design, we present Overview+Detail techniques that allow collaborators to perform fine-grained exploration in their own spaces while simultaneously accessing a shared global view, aligning with the multi-view, multi-scale, and semi-public workspace considerations for immersive large-scale 3D data exploration.

5.3.1 Design Considerations

Drawing on the characteristics of large-scale data and prior work in data exploration, collaboration and Overview+Detail, we propose three design considerations.

DC1: Multi-view exploration. As users are prone to losing orientation after navigating across scales in VR [197], the environment design should enable users to maintain an awareness of the overall data distribution. To achieve this goal, multiple views are needed: an overview to provide contextual awareness and support orientation, and detail views that allow users to focus on specific areas of interest.

DC2: Multi-scale transition. Exploring large-scale scientific data with complex structures and rich detail requires users to frequently zoom and shift positions. Therefore, the camera movement speed over vast distances, multiple instances, and multiple orders of magnitude must be carefully controlled to achieve seamless transitions between spatial locations of significantly varying scales. An alternative approach is teleportation [17], where users define spatial points and jump directly between them without physically traversing the distance. This approach, however, may confuse users when determining locations and understanding transitions across scales.

DC3: Semi-public workspace. Considering that collaborators often engage in group and individual tasks in different areas [193], it is essential to provide a semi-public workspace. This setup allows users to communicate and coordinate with partners in a shared space while independently exploring their personal areas, promoting collaboration and minimizing mutual interference, particularly when collaborators focus on different scales.

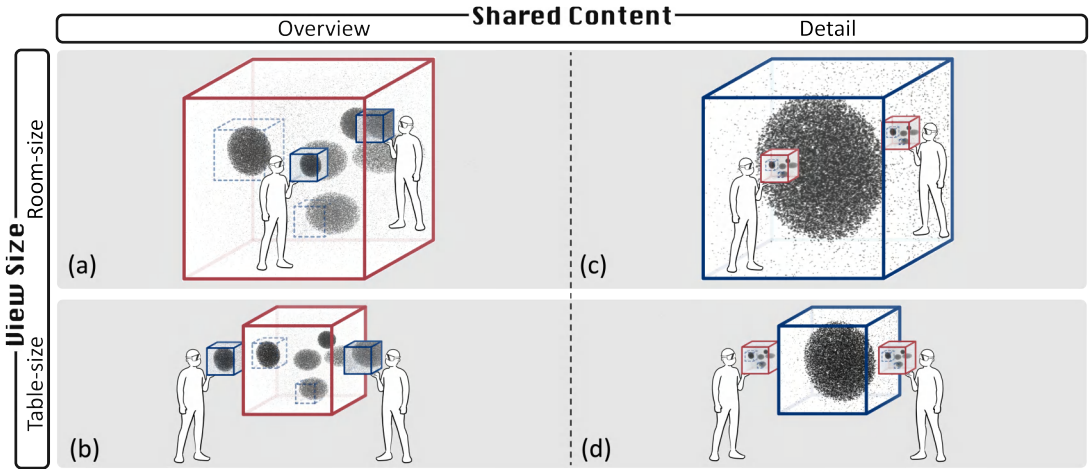


Figure 5.1: Four Overview+Detail designs in VR, developed based on two key perspectives: shared content and view size, including (a) RO (Room-size Overview and hand-size detail), (b) TO (Table-size Overview and hand-size detail), (c) RD (Room-size Detail and hand-size overview), and (d) TD (Table-size Detail and hand-size overview).

5.3.2 Four Overview + Detail Designs

Based on **DC1**, we adopt Overview+Detail designs, which fulfill the need to view both overall and detailed information. The overview provides a global context, helping users orient themselves within the data space, while the detail view facilitates a close examination of specific regions of interest (ROI). We limit each type of view to one to investigate their roles in collaborative exploration within a controlled setting. To facilitate seamless multi-scale or long-distance navigation as emphasized in **DC2**, we incorporate indicator boxes into the overview to represent each user's focus area. These boxes are visible to all collaborators, thereby enhancing workspace awareness. If a user zooms in or out, the corresponding indicator box dynamically resizes to reflect the current scale. As the focus location shifts, the box transitions smoothly across the overview, visually tracing the transition path.

We explored three sizes for both the overview and the detail view:

Hand-size: A cubic space of 30 cm side length, held by the user's non-dominant hand. We attached the center to the controller with a 35 cm bias to ensure it does not interfere with dominant hand interactions, similar to MeTACAST [77].

Table-size: A 1 m cubic space placed 1 m above the floor.

Room-size: A 2 m cubic space placed on the floor where users are immersed, similar to Yang et al.'s room-sized interface [197].

The table- and room-size views are visible to all users, whereas the hand-size view is only visible to the individual holding it, creating a semi-public workspace (**DC3**). While users may desire to move content between private and shared spaces, we fixed the visibility of each view to assess the impact of view shareability on collaboration. Thus, we have two types of views—overview and detail—and two size options for shared views: table-size and room-size. This setup results in the following four combinations (shown in Fig. 5.1):

RO: (**R**oom-size **O**verview + Hand-size Detail). RO features a large, shared overview at room-size and individual detail views at hand-size. As shown in Fig. 5.1(a), the room-size overview immerses users in the data. A hand-size detail view is attached to each user's non-dominant hand, allowing users to explore different details independently.

TO: (**T**able-size **O**verview + Hand-size Detail). As illustrated in Fig. 5.1(b), similar to RO, TO provides hand-size individual detail views to users. However, it employs a shared overview smaller than RO and positioned in front of users.

RD: (**R**oom-size **D**etail + Hand-size Overview). RD offers an individual hand-size overview and a collaborative room-size detail view. As shown in Fig. 5.1(c), the overview is anchored to the user's non-dominant hand. Users project selected features from the overview into the detail view, and if another user initiates this action, the content in the detail view is replaced.

TD: (**T**able-size **D**etail + Hand-size Overview). As presented in Fig. 5.1(d), users share a table-size detail view, with each person controlling an individual hand-size overview. Compared to RD, detailed features in TD are more centrally concentrated.

The abbreviations signify the content in the shared view and whether a table-size or room-size view is being employed. **RO**, for instance, indicates the use of a room-size view to present the overview of the dataset, which is accessible to multiple users, while the individual hand-size view is used to show detailed data specific to the data subset onto which the user currently focuses.

5.4 Design Two: Linked World-in-Miniatures

In the second design, we propose a WiM-based method for multi-scale data collaborative exploration. In previous design, users reference indicator boxes in the overview to understand their and collaborators' positions, scales, and workspace context. As the number of collaborators grows beyond two, however, these indicator boxes begin to overlap and nest

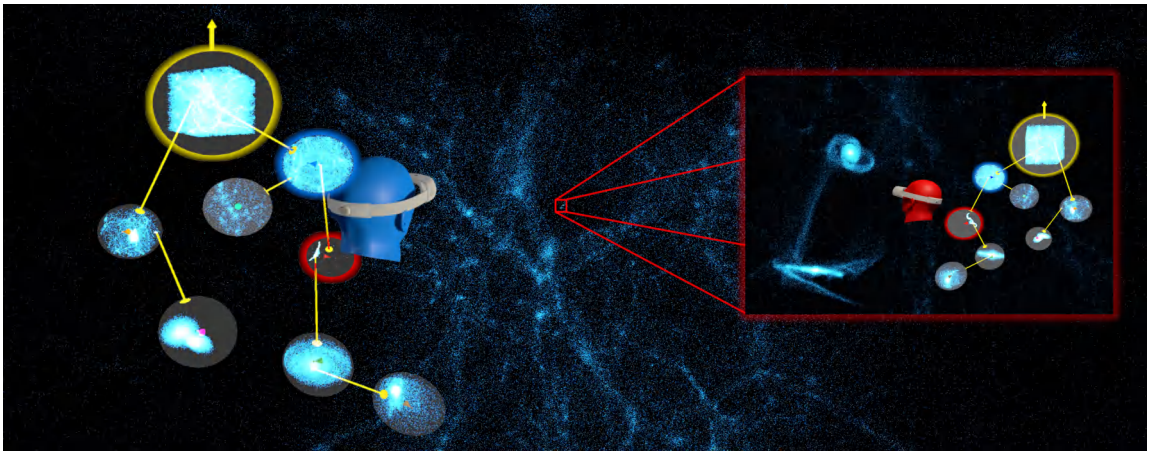


Figure 5.2: Seven users are exploring a large-scale simulation of astronomical data in the immersive virtual reality environment. The blue user possesses a larger scale, viewing the cosmic web, while the red user is observing the galaxy in a relatively small scale. Each user is assigned a WiM which shows a miniature of the virtual environment at the level of the scale. All the WiMs are connected in the L-WiM tree.

within one another, significantly reducing the clarity of each collaborator's information. In this section, each collaborator is given a WiM as their personal exploration space, and all WiMs are linked together through a connecting thread, forming an L-WiM (Linked World-in-Miniatures) tree. This L-WiM tree provides an all-knowing third-person perspective on the collaborators' collective observations.

5.4.1 Design Considerations

The system aims to enhance the collaborator's context, position, and scale awareness in immersive exploration work. Therefore, we summarize four design considerations (DCs).

DC1 Context Awareness: Illustrate the virtual environment at the user's level of scale. The main idea of the WiM metaphor is to offer a dynamic view through a miniature of the virtual environment so that the users are aware of the surrounding data. Therefore, it is critical to provide contextual information in the WiM, including the surrounding virtual environment, the user's position and orientation in that environment.

DC2 Position Awareness: Demonstrate the relative positions among the collaborators. The relative positions of collaborators are vital for collaborative exploration, especially when exploring multi-scale data in the immersive VR environment. Thus,

it is important to display all users' WiMs in one view so that people will know where their partners are or how to find them.

DC3 Scale Awareness: Indicate the level of scales among collaborators. The unique feature of exploring multi-scale astrophysical data in the immersive environment is that people can navigate through different scales and focus on the data in different regions. Thus, the system should indicate users' scales in the view and display what people are looking at from their positions.

DC4 Collaboration: Support communication and information sharing. There are many ways to achieve communication and information sharing, for instance, co-located and synchronous sharing through voice messages, visual highlights, viewpoints sharing. To ensure effective collaboration, the system should provide an optimized way to support people 1) to focus on their own data analysis, and 2) to exchange their insights and share their findings when needed.

5.4.2 WiM

To address **DC1**, every user is assigned a WiM that miniaturizes the virtual environment at the level of scale where the user is. Immersed in astronomical data that spans multiple orders of magnitude, users can traverse multiple scales and explore different regions. Through the WiM, the user can see a scaled-down copy of the contextual environment, where the user perceives their location, view direction, surroundings, as well as other collaborators' avatars if they are on the same scale.

We introduce the "size" and "scope" concepts used in the L-WiM system. Danyluket al. [202] explains these two terms, where "size" refers to the spatial size of a WiM, and "scope" refers to the range of miniature data visualized by the replica. For instance, a cube-shaped WiM (size: 1 meter on each edge) in the immersive environment holds a replica of the Milky Way galaxy with a radius of 520 thousand light-years (scope).

We constructed the user's surrounding environments using a spherical WiM (see Fig. 5.3). Within the spherical WiM, the user's location is represented as a cone, and the user's view direction is illustrated by the opening (base) of the cone. For ease of visualization, the user is always placed in the center of the WiM sphere. Note that the WiM only highlights data that is within the visible scope of the user. The scope of the visualization is computed by $c \cdot s_{user}$, in which s_{user} refers to the size of the user and c is a user-defined

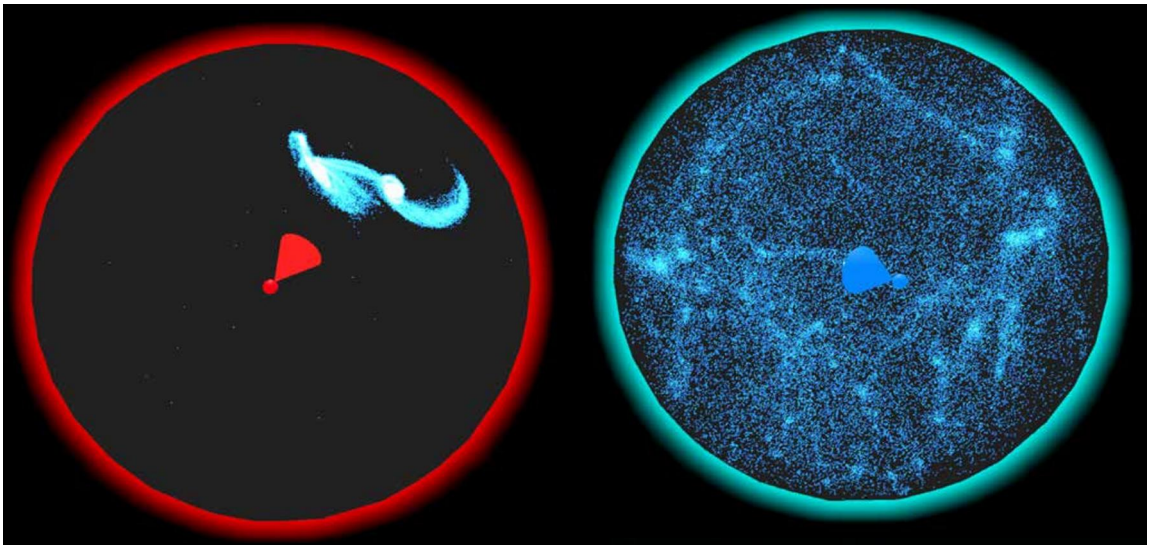


Figure 5.3: The WiMs of the blue user and the red user are shown. The user-centered WiM demonstrates the virtual environment at the corresponding level of scale.

constant factor (for our use case, the value is set to 5). We calculate the distance between the datapoints and the user (located in the center of the WiM)

$$||p_{user} - p_{particle}|| < c \cdot s_{user} \tag{5.1}$$

where $\boldsymbol{p_{user}}$ is the position of the user and $\boldsymbol{p_{particle}}$ is the position of each particle in the dataset. The particles located within the scope are visualized in the WiM.

5.4.3 L-WiM

To address **DC2** and **DC3** and guide the design of the L-WiM tree, we propose the following design principles:

DP1 If a user (U_i) travels to a different scale and appears in another user's (U_j) virtual environment, i.e., if U_i is located within the scope of the U_j's world, the two WiMs are connected directly.

DP2 If two WiMs are connected through a thread, the user can learn the relative position of the other user by looking at the endpoints of the thread.

DP3 The position of WiM in the tree along a pre-defined axis indicates the user's level of

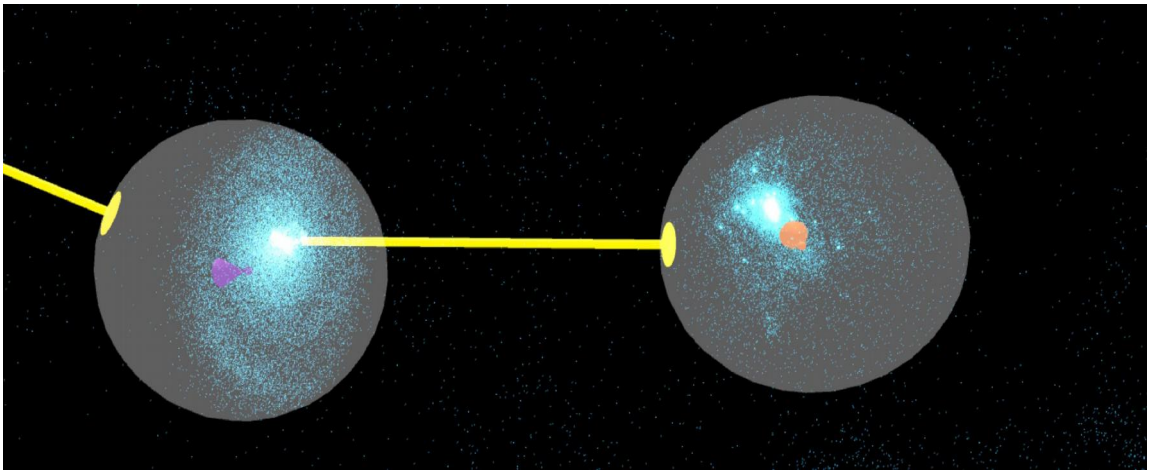


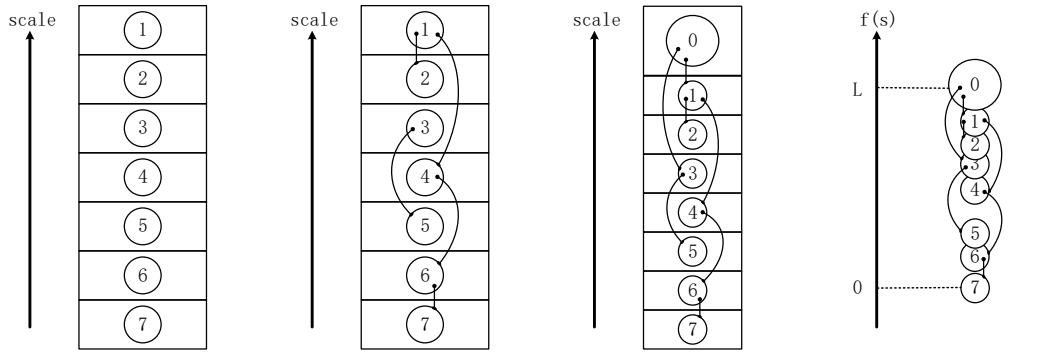
Figure 5.4: The father node (WiM with the purple user) and the child node (WiM with the orange user) are connected through the yellow thread. The endpoint of the thread located within the father node illustrates the relative position of the orange user to the purple user in the purple user's world.

scale compared to other users.

To address **DP1** and **DP2**, we leverage a father-child relation to demonstrate the relative position among collaborators. We define WiM W_2 as the father node of WiM W_1 if W_1 is located in the scope of W_2 and has a smaller scale than W_2, i.e.,

$$\begin{cases} ||p_1 - p_2|| < c \cdot s_2 \\ s_1 < s_2 \end{cases} \tag{5.2}$$

where p_1 and p_2 are the positions of W_1 and W_2. s_1 and s_2 are the scale of W_1 and W_2. If such a father-child relation holds, the two WiMs can then be connected by yellow threads (shown in Fig. 5.4). The thread's endpoint in the father node indicates the relative position of the child node to the father node. However, in some cases, one WiM may have multiple father nodes or none. In order to obtain an intuitive and clear hierarchical L-WiM tree, we determine the father node for each WiM by the following steps. We first sort the n WiMs based on the corresponding user avatar scale from the largest value to the smallest, where n represents the number of the WiMs, and create a WiM list (shown in Fig. 5.5a). Then, we pick one WiM W_i from the list. Following a bottom-up order from the list, we search for its father node W_j which meets two requirements: 1) user i's avatar appears within the scope of W_j and 2) W_j's avatar has a greater value of scale than W_i until we find the first satisfied WiM. We perform this bottom-up search on every WiM in the list and eventually obtain the



(a) Sorting (b) Linking (c) Adding node #0 (d) Mapping

Figure 5.5: The building process of the L-WiM tree

hierarchical relationships among the WiMs (shown in Fig. 5.5b). Note that, not all WiMs in the list can be connected in the same branch, and some WiMs may have no father node because they do not belong to any scope of the other WiMs. For instance, node #3 and node #5 are isolated from the main tree without a connection. Finally, in order to link all the WiMs to the tree, we add a node #0 at the top of the L-WiM tree (shown in Fig. 5.5c). The node #0 illustrates the replica of the whole environment to ensure all WiMs are eventually included in the tree. It is highlighted in yellow (shown in Fig. 5.2) and has a larger size than the other WiMs.

To address **DP3**, initially we attempted to map the scale information to the size of the WiMs, i.e., users observing at a larger scale owns a bigger WiM. However, visual depth perception in immersive environment influences the judgement of the WiM size. A large size WiM located far away might look smaller than a nearby WiM with a small size. Therefore, we decided to illustrate the spatial scales through the vertical organization of the individual WiMs in the L-WiM tree. We leverage an additional axis to map the users' scale information. In other words, the WiM projected on a higher-up position in the tree along the axis indicates that the user is at a larger level of scale of observation. To implement this, we first set the position of the node with the minimum scale (node #7) as $(P_{x_{min}}, P_{y_{min}}, P_{z_{min}})$ in the virtual view, the direction of the scale axis denoted by the unit vector $\overrightarrow{n}$ and a total length L for the L-WiM. Next, we map the user scale interval (s_{min}, s_{max}) to the range $(0, L)$ based on the function

$$f(s) \in (0, L), s \in (s_{min}, s_{max}), f(s_{min}) = 0, f(s_{max}) = L \tag{5.3}$$

and it also satisfies

$$s1 \in (s_{min}, s_{max}), s2 \in (s_{min}, s_{max}), s1 < s2, f(s1) < f(s2) \tag{5.4}$$

In terms of the design of $f(s)$, we considered the situation that many users may explore data at a similar scale. Thus, to avoid clustering of WiMs within a small-scale range, we use the non-linear $f(s)$. In our design, the first derivative of $f(s)$, denoted by $f(s)'$, is related to the WiM distribution density spread in (s_{min}, s_{max}). In a dense distributed region, $f(s)'$ has a large value. This way, if many WiMs have similar scales, bigger range along the scale axis will be preserved and these WiMs with similar user scales will distribute sparser. The position of WiM along the scale axis $(P_{xaxis_i}, P_{yaxis_i}, P_{zaxis_i})$ is calculated as:

$$(P_{xaxis_i}, P_{yaxis_i}, P_{zaxis_i}) = (P_{x_{min}}, P_{y_{min}}, P_{z_{min}}) + f(s_i) * \overrightarrow{n} \tag{5.5}$$

Subsequently, the WiMs are ranked vertically along the scale axis so that the higher-up WiM possesses a bigger scale (shown in Fig. 5.5d). Furthermore, to avoid the overlapping issue and maintain the correct scale information, we push the WiMs one by one to the sparse area in the direction perpendicular to the scale axis, denoted by the unit vector $\overrightarrow{v_i}$ until they do not overlap and are separated by a certain distance away from each other. The final position of the WiM $(P_{x_i}, P_{y_i}, P_{z_i})$ is:

$$(P_{x_i}, P_{y_i}, P_{z_i}) = (P_{xaxis_i}, P_{yaxis_i}, P_{zaxis_i}) + r_i * \overrightarrow{v_i} \tag{5.6}$$

where r_i is an adaptive constant representing the distance being pushed and $\overrightarrow{v_i}$ is an adaptive unit vector indicating the most sparse direction at $(P_{xaxis_i}, P_{yaxis_i}, P_{zaxis_i})$ perpendicular to $\overrightarrow{n}$:

$$\overrightarrow{v_i} \cdot \overrightarrow{n} = 0 \tag{5.7}$$

The interface is shown in Fig. 5.6. To demonstrate the direction of spatial scale, we provide an additional yellow arrow on top of the #0 node showing the direction of increasing scale.

5.4.4 Communication and Information Sharing

To address **DC4**, we design a direct approach to support users to "visit" others' WiMs (sharing findings, observing data, offering guidance) through visual cues and voice messages. We consider the key principles of communication and information sharing in collaborative

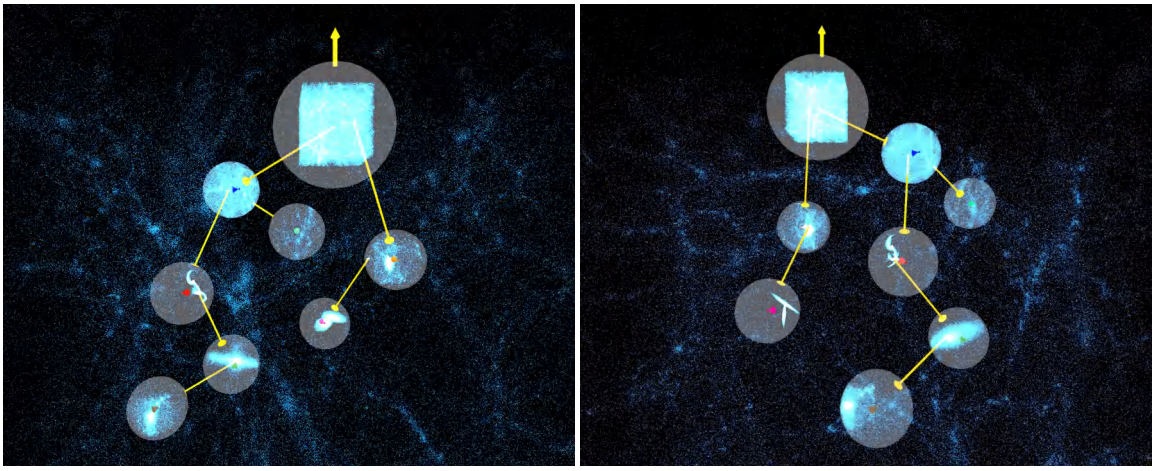


Figure 5.6: The structure of L-WiM tree in a front view (left) and side view (right). The yellow arrow on top of node#0 indicates the positive direction of the scale axis.

exploration from two aspects.

DP4 The user should be able to decide if they want to be observed or visited by others.

DP5 The user should be able to communicate with voice.

DP6 The user can select the data in their collaborators' WiMs.

DP7 The user can visit their collaborators' WiMs whenever they want if it is allowed.

DP4 is proposed for personal focus and productivity. The users should be able to focus on their own work without being disturbed. Thus, they should have the option to decide if they would like to receive messages and allow others to see what they are doing. To achieve DP4, our L-WiM provides a switch-like functionality. By default, all WiMs can be visited. However, if a user chooses to disable visitor view permission by turning off the switch, the user's WiM is rendered invisible in other's view (the grey WiM in Fig. 5.7).

DP5 ensures that users can establish direct and immediate communication whenever needed. In L-WiM, users can initiate communication by selecting any collaborator's WiM in the L-WiM tree with the controller by the ray interaction (shown in Fig. 5.7(a)). The chosen WiM is highlighted, and a copy of it is attached to the user's VR controller (shown in Fig. 5.7(b)). With this hand-held WiM, users are able to activate **voice communication** with the corresponding collaborator.

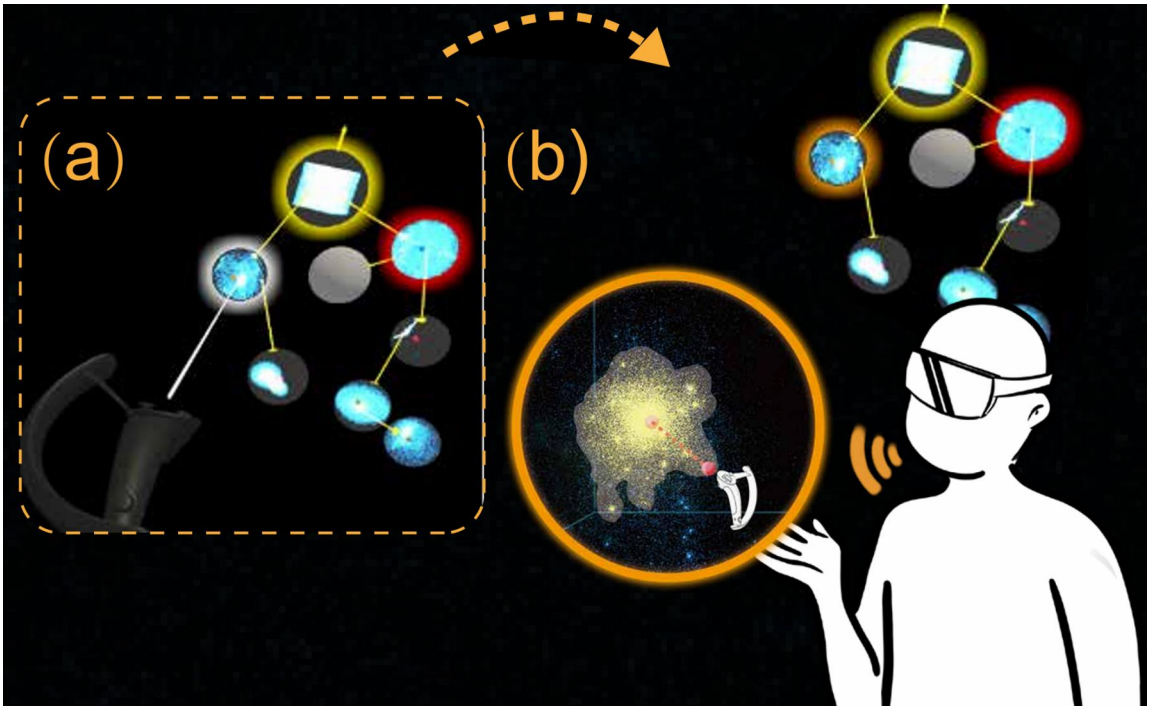


Figure 5.7: (a): The WiM is highlighted in white when the user is pointing at it with the write ray. (b): A copy of the selected WiM is attached to the hand by clicking the "trigger". After that, both the WiM in the L-WiM tree and the copy will be highlighted with the same color as the corresponding collaborator, indicating the selected state. The grey opaque WiM is inaccessible.

DP6 enables users to select regions of interest, allowing them to initiate discussions about specific data structures whenever necessary. After selecting another collaborator's WiM, a duplicated copy is held in the user's hand. Users can then directly select data on the WiM using the MeTACAST technique introduced in **Chapter 3** (shown in Fig. 5.7(b)). The selected data on the WiM copy is also synchronized in the collaborator's environment. Thanks to the fast GPU-accelerated KDE method introduced in **Chapter 4**, the density field within the data space enclosed by the WiM can be recomputed in real time whenever the underlying data changes—for example, when the collaborator moves to a different location.

DP7 supports users in navigating across scales. After selecting a collaborator's WiM, users can navigate to the collaborator's environment through two mechanisms: (1) by specifying a destination via clicking on the handheld WiM using the controller, similar to the point-and-click teleportation metaphor [18]; and (2) by using the progressive navigation

technique introduced in **Chapter 4**, which enables scale transitions to finer levels by directly selecting target structures with the MeTACAST technique. The virtual camera then moves smoothly to the target region, adjusting the user's scale accordingly.

5.4.5 Interactions

Our prototype supports the manipulation of the L-WiM and the individual WiMs. We leverage the "World grab" metaphor [3] for manipulating the L-WiM. By holding the trigger button on the right VR controller, users can drag, push and rotate the L-WiM in the view. Users can also use the left-hand VR controller to select and manipulate an individual WiM through the Ray-casting pointing method by the white ray. The boundary of the WiM being pointed will be highlighted in white when the ray moves over (shown in Fig. 5.7(a)). Users can press the trigger button to select the WiM. The selected WiM will be highlighted in the same color as the corresponding user, and at the same time, it will be copied and attached to the virtual left-hand controller (shown in Fig. 5.7(b)). Users can manipulate the controller to get a closer look at the selected WiM from different directions. The WiM can be deselected and detached from the left-hand controller by pressing the trigger button again.

5.5 Discussion

Our design—Overview+Detail and L-WiM techniques—both support multi-user collaborative exploration of spatial data in immersive environments, allowing collaborators to move across spatial scales, inspect fine-grained structures, and remain aware of each other's positions, scale and contextual information.

The two designs are different in many aspects. First, they differ substantially in how they manage views. In L-WiM, the full dataset is rendered as a global overview in the immersive space, while each user's WiM acts as a detail view; the linked collection of these WiMs allows collaborators to see all users' detailed focus regions simultaneously. By contrast, Overview+Detail gives each user a single handheld view—either overview or detail—encouraging focused local exploration while referencing the overview only when needed. The second difference lies in the awareness of scale. In L-WiM, a user's current scale is explicitly encoded by the vertical position of their WiM, allowing collaborators to precisely compare who is operating at a smaller or larger scale. In contrast, Overview+Detail

provides only a coarse indication of scale: users infer another's viewing scale from the size of the color-coded cubic frame shown in the overview. This cue does not support precise comparison. When a collaborator inspects very small-scale details, the frame can become too small to perceive, making it difficult—or sometimes impossible—to estimate their current scale. Third, they differ in collaborator awareness. In L-WiM, users transition across scales by resizing their avatars. At very small scales, collaborators can no longer perceive a user's physical avatar and must rely on the user's WiM to infer their location and focus. In Overview+Detail design, collaboration occurs at a constant spatial scale, so avatars remain visible, enabling partners to observe gestures, identify whether others are viewing overview or detail, and communicate through pointing and embodied cues.

5.6 Conclusion

Despite extensive research into collaboration in VR, the challenges of exploring complex data visualization and large-scale features remain significant, which are very commonly used in scientific domains, such as astronomical and physical simulations. However, the inherent advantages of a 3D perspective view make VR ideal for exploring such data, which motivated us to initiate this study. Collaborative exploration goes beyond sharing understanding and completing well-defined tasks. More importantly, it enhances collaborators' experiences and leverages their diverse knowledge and creativity, especially when navigating unknown and complex environments. In VR, this exploration requires enhanced awareness, robust support, and the flexibility to experiment and share new findings. This is exactly the key elements that come into play: shared spatial/scale awareness, storing findings, flexible content sharing, adjustable view size, and importantly, individual exploration space. These elements create a dynamic and effective collaborative environment, allowing participants to seamlessly integrate their insights while fostering a shared understanding. We acknowledge several limitations and directions for future exploration. The effectiveness of L-WiM requires further evaluation through a formal user study. Future work will examine how the linked WiM structure supports collaborator awareness, cross-scale navigation, and shared understanding during collaborative multi-scale exploration. In addition, a broader understanding of the needs of collaborative exploration across research domains is still required. Future work may extend our design to other collaborative settings involving large-scale datasets from fields such as geography and environmental science.

APPENDIX 6

SpatialTouch: Exploring Spatial Data Visualizations in Cross-Reality

In the fourth project, we address **RQ4: How can immersive environments support the effective presentation and integration of spatial datasets that rely on multiple 2D and 3D visual representations?** In many scientific workflows, researchers rely heavily on 2D representation—such as slice views, projections, quantitative plots, or abstract diagrams—to support domain-specific analysis tasks. Although VR/AR techniques can also host 2D visualization, 2D displays remain the native and effective medium for presenting and interacting with 2D visualizations. This need for 2D information raises an important question regarding how immersive settings can incorporate both 2D and 3D views in a coherent and meaningful way.

In this chapter, we propose and study a novel cross-reality environment that seamlessly integrates a monoscopic 2D surface (an interactive screen with touch and pen input) with a stereoscopic 3D space (an AR HMD) to jointly host spatial data visualizations. This innovative approach combines the best of two conventional methods of displaying and manipulating spatial 3D data, enabling users to fluidly explore diverse visual forms using tailored interaction techniques. Providing such effective 3D data exploration techniques is pivotal for conveying its intricate spatial structures—often at multiple spatial or semantic scales—across various application domains and requiring diverse visual representations for effective visualization.

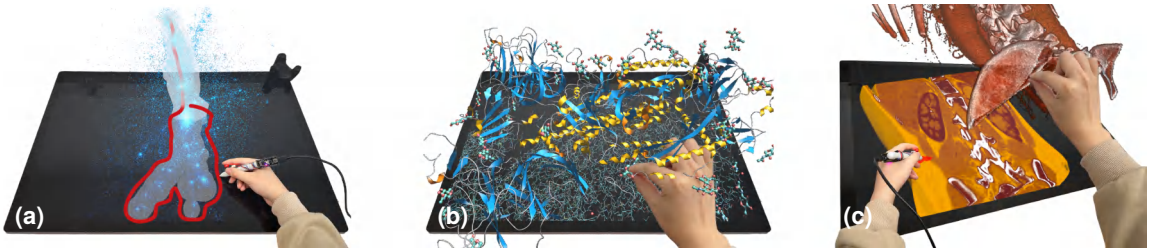


Figure 6.1: Data exploration in SpatialTouch: (a) select astronomical points, (b) navigate molecular visualization, (c) annotate medical data.

6.1 Introduction

As computing and simulation technologies have advanced, the complexity of 3D structures within scientific datasets, alongside their contextual information, has grown substantially. Approaches to interactively explore 3D data have emerged to help users understand complex spatial structures across various disciplines. These datasets often encompass elaborate details, comprising millions of intricate components such as unstructured points in astronomical simulations [107, 108], flood simulation [207] or biological structures spanning multiple scales [208]. To gain a deeper understanding of such complex data, researchers often need to selectively focus on different regions. They may adjust visualization parameters to filter the data, emphasize key features, or alter data representations to highlight specific data attributes. This exploration can involve switching between 2D and 3D representations or displaying a combination of both for an in-depth analysis of data structures. Abstractocyte [20], e. g., is an interactive system designed to assist biologists in exploring the morphology of astrocytes using various levels of visual abstraction, while simultaneously analyzing neighboring neurons and their connectivity. This tool allows users to manipulate a visualization widget of 2D abstraction space to smoothly transition between the different 3D and 2D abstraction levels, to identify morphological features for cells of interest. Furthermore, various visual representations, such as ball-and-stick models, are employed in molecular visualization to display 3D structures with different data features [209, 210, 211].

A common approach for supporting researchers to observe diverse combinations of data features, whether in 2D [212, 213, 214] or in 3D [215, 216], is through multiple linked views, facilitating connections among different representations. An alternative approach is to integrate 2D and 3D visual representations within a unified view [217]. Achieving a seamless interpretation of visual transitions and interaction with the data in such combined

representations, however, is challenging—especially when dealing with complex data that contains a lot of detail. These challenges prompted us to explore a different display environment for presenting complex 3D data, one that integrates various visual representations and enables users to work with and seamlessly switch between different representations in a single setting.

To this end, we introduce SpatialTouch—a CR environment [218] that merges a monoscopic interactive 2D surface (an interactive screen with touch and pen input) with a stereoscopic 3D space (an AR HMD). We can visualize data simultaneously on both the 2D surface and in the 3D space such that the depiction on both displays remains in sync. Rather than imposing restrictions on which visual representations should be displayed in each space, we approach our setup as an integrated system in which the user decides where to display the data and how to interact with it. The integration of both views facilitates smooth transitions for data visualization between the 2D surface and the 3D space, ensuring a continuous situational awareness. With our CR SpatialTouch environment, we establish a platform that is capable of accommodating a wide range of data types that can be displayed and interactively explored as needed. While creating many possibilities for visualization and interaction design, our setup also poses questions on how to best organize the data exploration.

A central question when hosting data visualizations within SpatialTouch is how interaction techniques should be designed, ensuring that users can seamlessly continue their tasks, without having to consider how interactions should be performed in the respective other display space. Touch-enabled screens facilitate direct input on the 2D screen that typically shows projected views of the spatial 3D data; and many interaction techniques have been developed for such settings (e. g., for navigation [219, 220, 221, 29], selection [222, 58, 59], positioning [223, 224]). While the exploration of 3D spatial visualizations via a 2D view can pose challenges (e. g., it often requires users to rotate the data to explore its 3D aspects), due to the ubiquity of monoscopic screens, people have already developed a solid mental concept of how interactions are performed on the 2D screen. At the same time, however, they have also established a solid understanding of how gesture interactions are performed in the 3D space. When a visualization spans both of these spaces as with SpatialTouch, the question arises whether people can maintain their spatial awareness and are able to seamlessly transition between the different interaction techniques in the hybrid environment. To clarify, “spatial awareness” in our context refers to users’ understanding of both the

spatial position of interactions within the CR environment and the spatial status of the data, including its structure, orientation, and position, which guides to perform specific actions.

Another crucial question is, when users explore spatial data, how they react as they engage with diverse data representations within a CR environment. We can assume that people decide how interactions should be executed based on their perception and understanding of the specific data structures. When selecting subsets of unstructured point cloud data, e. g., individuals may circle around them to define a range, whereas when selecting string-like structures they may brush along the structure. Any interaction pattern, however, may undergo additional alterations when the data transitions from 3D space to the 2D surface or vice versa. When selecting point cloud data which is rendered partially on the 2D screen and partially in 3D space, for instance, people may realize at some point during the selection action they can no longer circle around the target points in the depth direction. Consequently, the question arises if people maintain or change their interaction strategies when a visual representation is used to depict data across both spaces.

Other important considerations when merging 2D and 3D display spaces and constructing a CR environment include the challenge of effectively presenting 2D/3D visualizations and determining the appropriate level of detail. We can often choose from a diverse set of visual representations that each focus on a particular set of features or contain various scales or visual abstractions, which in our case may be partially rendered stereoscopically in 3D space and partially projected on the 2D screen. We thus ask: how should we represent the data such that users gain a comprehensive understanding of it? Moreover, understanding the best specific setup of our CR environment (in particular, the optimal angle or angle range for the 2D surface) is crucial to facilitate the user's interaction experience. In summary, we contribute

- a CR environment that combines a monoscopic 2D surface with a stereoscopic 3D space to facilitate joint spatial data exploration;
- a study on user actions and interaction strategies while interacting with spatial data;
- a design space of how CR supports 3D spatial data analysis across various visual representations, scales, and data abstractions; and
- three uses cases for demonstrating data manipulation, selection, annotation and measurements in three distinct domains.

6.2 Related Work

Our work focuses on three key aspects of CR environments: the environment itself, data visualization, and interaction techniques employed within these CR settings. We review related work on these topics next.

6.2.1 Cross-reality Environments

The reality-virtuality continuum, as conceptualized by Milgram and Kishino [12], provides a framework for data analysis and visualization techniques. It spans from visual analytics on 2D displays, to stereoscopic visualizations with immersive technologies, including augmented reality (AR), augmented virtuality (AV), and virtual reality (VR). Cross-reality [218] emphasizes seamless integrations and transitions among these visualization environments, offering users effective visual and algorithmic assistance tailored for maximal cognitive and perceptual suitability based on data, tasks, and user requirements.

Recent surveys have extensively explored cross-reality/virtuality environments from various perspectives, such as design and human factors [225], visualization and interaction techniques [27] and collaborative analysis [226]. Fröhler et al. [227] and Auda et al. [228] conducted comprehensive reviews of existing work, classifying the literature regarding different stages in the reality-virtuality continuum, visualization and view transition techniques, collaboration, visualization and visual analytics techniques, evaluation methods, and application domains. These opportunities and challenges have been recognized by many researchers and extensively discussed in conference workshops [229, 230, 218].

The main purposes of CR environments for domain research span from observation to collaborative analysis. An initial CR prototype was created by Kijima and Ojika [231] based on a projective see-through HMD (PHMD) and 2D monitor, demonstrating how **object manipulation** can be performed using keyboard and mouse. The most obvious benefit of combined 2D and 3D environments is the inherent display environments for **observing 2D and 3D data representations**. Seraji and Stuerzlinger [232] introduced an immersive visual analytics tool that allows users to conduct analysis at either end of the reality-virtuality continuum. It demonstrates that users can experience a lower cognitive load while viewing both 2D and 3D representations from two environments and task-switching between these virtuality modes. CR environments are also employed for **displaying and connecting**

different views. Reipschläger et al. [233] proposed the combination of large interactive displays with personal head-mounted AR for displaying information in various views to facilitate data analysis. Their approach demonstrates how CR can be designed to address challenges encountered in solely large displays, such as perception, multi-user support, and managing data density and complexity. Such environments also show great benefits in making the best use of 2D and 3D interfaces and providing users with familiar ways **perceiving, creating and manipulating 3D contents**. Dedual et al. [234] explored hybrid user interfaces that combined a 2D multi-touch tabletop display with a 3D head-tracked video see-through head-worn display, including an urban visualization scenario in which the tabletop provided a 2D map while the head-worn display showed complementary 3D building information above the tabletop. Reipschläger and Dachselt presented an augmented design workstation [46], which seamlessly integrated an interactive surface displaying 2D views with a stereoscopic AR HMD, demonstrating how this combined space facilitates 3D model creation. For facilitating complex data exploration, decision making, and **collaborative analysis**, Butscher et al. [235] investigated the combination of AR and touch-sensitive tabletops for collaborative multi-dimensional data analysis through 3D parallel coordinates, using established touch interactions with visualizations anchored to the tabletop.

All these purposes for using CR heavily inspired the design of our own CR environment, including the selection of devices, layout, and setup. Typical such devices encompass 2D desktops, tablet/mobile devices, tabletop, wall displays, CAVEs, as well as AR and VR HMDs. We discuss these devices and the implication of their use in detail related to our intended visualization tasks and domains in **Chapter 7**.

6.2.2 Visualization and its Tasks in Immersive Environments

Immersive environments provide a stereoscopic view to convey complex 3D structural arrangements, with great potential in exploring many types of scientific data. Molecular data, e. g., often consists of densely packed 3D structures with intricate internal detail that spans multiple scales—making it challenging to comprehend the contextual information of the data. Alharbi et al. [15], e. g., introduced a guided-tour generator for immersive environments to navigate and communicate multi-scale, crowded, scientifically accurate 3D models. Similarly, point clouds such as astronomical simulations exhibit complexities: 3D occlusions, non-uniform feature density, or intricate shapes. Zhao et al. [77] proposed target- and context-aware selection techniques for users to select sub-regions based on

understanding of the 3D structures yet without the need for high input precision.

A key aspect of these visualizations lies in their emphasis on spatial attributes and their positions within the data context. When exploring or creating data of this nature, researchers often consider these spatial features displayed in different views or using different representations. That is where CR comes into play. In medical imaging, e. g., detailed views that focus on key regions are crucial while maintaining the overall data context. Coffey et al. [2] addressed this issue by presenting 3D medical data in the air alongside a detailed stereoscopic 3D view. This 3D data was complemented, moreover, by a 2D overview presented on a table, enabling users to interact with the data using familiar 2D touch gestures. Furthermore, Sereno et al. [35] proposed a spatial selection technique for 3D point data, wherein selections are performed on a tablet, while a stereoscopic view is provided by an AR HMD. With AR showing an overview of the point distribution, users can perform selections on the tablet and precisely control the selection in depth.

When CR is used to expand the limited display space, various visual representations can overlay the presented information to indicate connections or provide additional context. Langner et al. [236], e. g., introduced a conceptual framework that extends a 2D scatterplot displayed on a mobile device with superimposed 3D trajectories shown in AR. Similarly, Reipschläger et al. [233] proposed a system that augments data on large displays via AR. Satriadi et al. [40, 237] envisioned how to present multivariate data around physical scale models such as tangible globes, with relevant data attributes being displayed on and around the display or tangible interfaces. There are usually no strict constraints on how data should be displayed within and across multiple views, as different views may show different aspects. The crucial issue is that the provided information should be spatially linked to facilitate easy interpretation. This point is particularly important in our context. When using multiple visual representations to illustrate complex spatial data, it is crucial to ensure that users consistently perceive and understand the spatial relationships and features of the data.

When transitioning a data visualization across various spaces, a central design principle is to ensure that users understand this transformation and can seamlessly continue their tasks, without losing focus. For this purpose, Schwajda et al. [238] developed and evaluated different variants of transformation to seamlessly transition graph visualizations from 2D to 3D representations and from a 2D surface to 3D AR space, facilitating the development of a mental model in both environments. Fröhler et al.'s survey [227] introduced various

visual/view transition techniques, including portal, fade, and off-screen transition. These techniques guide users when a visualization shifts along the reality-virtuality continuum, such as moving from reality to AR. Lee et al. [22] presented a design space for data visualization transformations between a 2D screen and 3D AR, along with the interactions that facilitate this transformation. Their focus primarily lies on abstract data, such as transforming between 2D and 3D scatterplots, histograms, and parallel coordinate plot extrusions. To the best of our knowledge, there has been limited research addressing visual transitions for spatial data visualization. In our work we thus target smooth visual transitions of spatial data visualizations between 2D and 3D spaces, with an emphasis on preserving the intrinsic 3D structures and spatial distribution to maintain continuous situational awareness in CR environments.

6.2.3 Spatial/touch Interaction in Cross-reality

Drawing from the CR environments we discussed, typical input techniques include touch input, mid-air gestures, pen input, tangible and haptic interactions, as well as input through mobile devices or HMD controllers. This input predominantly occurs "inside" the stereoscopic view of 3D data presented in the air, on the 2D visualization displayed on a 2D interface, or directly with the mobile devices. Jackson et al. [239], e. g., proposed a prop-based **tangible interface** to control visualizations of thin 3D fiber structures. Fröhlich et al. [240] created a cubic tangible input device to precisely manipulate the slice with data visualization on a stereoscopic display. In the realm of model design or 3D content creation, multiple interactions facilitate the checking a 3D design, while creating and revising on 2D surfaces. DesignAR [46], e. g., allowed users to create and refine 3D models on a 2D surface with **touch and pen input**, while manipulating the virtual model with **mid-air gestures** in 3D. Similarly, Mockup [241] used sketching tools to construct models on a tabletop, extruding the sketches from the 2D surface to 3D space using mid-air gestures. Hybridaxes [232] demonstrated how to transition 2D or 3D data from a display to AR, allowing users to switch interfaces using **free-hand interaction or controllers**. Wu et al. [242] proposed interactions that harnesses physical affordances to assist digital interaction in AR with hand gestures.

Interestingly, regardless of where these interactions occur, people typically have a solid mental concept of how interactions should be performed on a selected interface. On mobile/touch interfaces, e. g., users usually use two touches for scaling, and one touch for

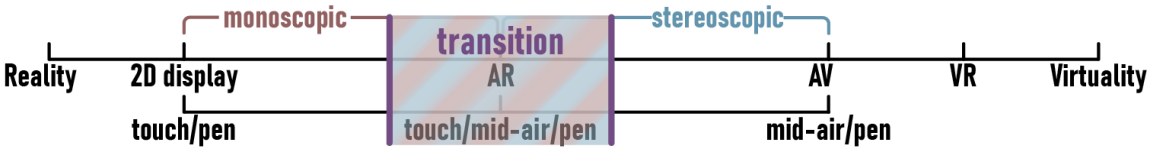


Figure 6.2: SpatialTouch's placement within the reality-virtuality continuum.

either x- and y-translation or trackball rotation [29]. Inspired by touch-based interaction, users tend to use two pinch gestures in the air to zoom content in or out [243]. As users perform a pinch gesture, their fingers naturally come into physical contact, creating explicit haptic feedback to reinforce the virtual action [242]. Thus, Lubos et al. [101] allow users to touch a 3D point cloud in mid-air and transform it with pinch gestures.

To maximize the benefits of CR environments, interactions can be performed on familiar interfaces or interfaces best suited for the task. The Interactive slice WIM [2], for instance, projected a data overview on the table and allows users to interact with the 3D data through familiar touch interaction. Similarly, López et al. [244] allowed viewers to use touch-based interactions to navigate and control the visualization on a monoscopic tablet, while observing data on a large stereoscopic display. In these cases, however, it is crucial to define how 2D interactions can be effectively mapped to the 3D visualization tasks.

In conclusion, we saw that most studies typically regard the CR environment as comprising distinct environments, each with its inherent and tailored interaction paradigm. It is thus interesting to investigate seamless interactions across different levels of virtuality—which is what we do. If users perceive the entire CR as an integrated environment, there is likely a substantial potential for interactions across the diverse interfaces to facilitate continuous actions, in a way that eases also the mental connection between the different output spaces.

6.3 SpatialTouch: A Cross-reality Rnvironment

Before we detail our experiments, we first introduce our CR environment SpatialTouch (for its placement in the reality-virtuality continuum see Fig. 6.2) with its camera settings, input techniques, interaction devices, and general implementation. As we do so, we discuss the key considerations drove the design. Our focus lies solely on creating an integrated visualization environment for displaying spatial data, without yet considering specific interaction designs.

SpatialTouch comprises two dedicated display areas: a monoscopic 2D interactive

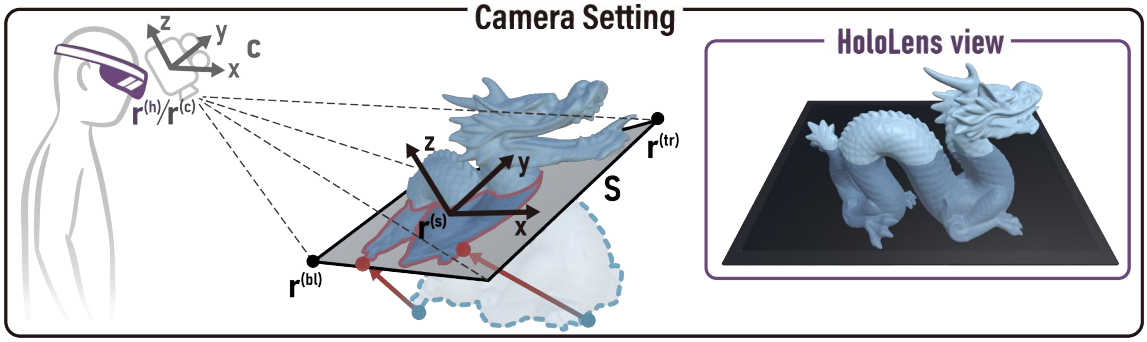


Figure 6.3: The configurations of the AR camera and the Surface camera $\mathbf{c}$ are depicted, illustrating how the virtual content below the Surface (marked by a blue dotted line) is projected onto the Surface (a red solid line). The HoloLens view shows what users perceive in SpatialTouch.

surface (Microsoft Surface Studio) and a stereoscopic 3D space (Microsoft HoloLens). We designed it to accommodate 3D spatial data and its associated information. As an integrated (hybrid) visualization space, the data can be positioned and manipulated anywhere within the CR environment—above, on, or below the surface, or spanning across it. To emphasize 3D structures with different data features, we want the visual representations that we use to show it to be able to take various forms, including 2D or 3D representations, various scales and forms of visual abstraction [245, 246], or combined forms such as the ball-and-stick representation in molecular visualization [210]. These requirements of having multiple forms of representation—later to be adjusted to the specific depiction location—set our approach apart from others as most existing CR applications fully transition data from one visual format to its 3D counterpart when moving it across different spaces (e. g., converting a 2D node-link diagram into a 3D visualization [238]). In addition, we not only focus on how data or views should be presented, arranged, and transformed in CR but we also provide a perspective view of the spatial data within the whole environment to ensure that viewers correctly perceive the 3D structures and their spatial distribution, to be able to maintain a continuous situational awareness while transitioning the data between both display spaces.

Camera Settings. In our design, regardless of the viewing angle, the data rendered on the surface and through the Hololens seamlessly combines into a cohesive 3D representation. To achieve this effect, we developed a rendering algorithm that aligns the visualization content displayed on the 2D surface with the AR visualization. We set two virtual cameras in our system, (1) an AR HMD camera rendering the 3D view and (2) a surface camera $\mathbf{c}$ for rendering 2D view (Fig. 6.3). For the AR HMD camera, we employ a perspective projection,

a standard method used in AR/VR applications. For the Surface camera $\mathbf{c}$, inspired by the Fishtank VR concept [247], we first align $\mathbf{c}$ to the HMD's position and then adopt an oblique perspective projection. This way we dynamically calculate $\mathbf{c}$'s parameters—position $\mathbf{r}^{(c)}$, orientation, and projection matrix $\mathbf{m}^{(c)}$—in each frame based on the AR HMD camera's position $\mathbf{r}^{(h)}$ and screen surface $\mathbf{sc}$'s center position $\mathbf{r}^{(sc)}$. The whole configuration process of $\mathbf{c}$'s parameters involves the following five steps in each frame:

1. We translate $\mathbf{c}$ to the HMD's position, in which $\mathbf{r}^{(c)}$=$\mathbf{r}^{(h)}$.
2. We adjust the forward direction of the $\mathbf{c}$ (negative z-axis) to $\mathbf{s}$ perpendicularly, in which the z-axis of $\mathbf{c}$ aligns with the z-axis of $\mathbf{sc}$.
3. We modify the x-axis of $\mathbf{c}$ to align with the x-axis of $\mathbf{sc}$, maintaining consistency in horizontal orientation between $\mathbf{c}$ and $\mathbf{sc}$.
4. We compute the surface's center position $\mathbf{r}^{(sc)}$, the bottom-left corner $\mathbf{r}^{(bl)}$, and the top-right corner $\mathbf{r}^{(tr)}$ in local coordinates w.r.t. $\mathbf{r}^{(c)}$.
5. The screen surface $\mathbf{sc}$ serves as the near projection plane of $\mathbf{c}$. The distance of $\mathbf{c}$'s far projection plane is denoted by f. Then, we subsequently derive the projection matrix $\mathbf{m}^{(c)}$ as:

$$\mathbf{m}^{(c)} = \begin{Bmatrix} \frac{2\mathbf{r}_z^{(sc)}}{\mathbf{r}_x^{(tr)} - \mathbf{r}_x^{(bl)}} & 0 & \frac{\mathbf{r}_x^{(tr)} + \mathbf{r}_x^{(bl)}}{\mathbf{r}_x^{(tr)} - \mathbf{r}_x^{(bl)}} & 0 \\ 0 & \frac{2\mathbf{r}_z^{(sc)}}{\mathbf{r}_y^{(tr)} - \mathbf{r}_y^{(bl)}} & \frac{\mathbf{r}_y^{(tr)} + \mathbf{r}_y^{(bl)}}{\mathbf{r}_y^{(tr)} - \mathbf{r}_y^{(bl)}} & 0 \\ 0 & 0 & \frac{-(f + \mathbf{r}_z^{(sc)})}{f - \mathbf{r}_z^{(sc)}} & \frac{-2f\mathbf{r}_z^{(sc)}}{f - \mathbf{r}_z^{(sc)}} \\ 0 & 0 & -1 & 0 \end{Bmatrix} \tag{6.1}$$

We share an open-source simulator of the camera setup in our supplemental material.

Touch/Pen/Mid-air Interactions. Microsoft's API Surface Studio captures multi-touch input, while we can obtain mid-air input from the AR HMD. For precise interaction needs such as selection and annotation, however, we want to augment touch and mid-air input from the Surface Pen—both on the surface *and* in mid-air. While the Surface Pen input can easily be captured on the Surface, it is normally not detected when used in the air. To overcome this limitation, we attached an Arduino board to the pen, allowing us to detect events when users press the physical button on the pen without it resting on the surface.

Devices and Implementations. A 28-inch Microsoft Surface Studio (637 mm × 438 mm; 4,500 × 3,000 px) serves as the 2D surface of our SpatialTouch CR setup, which

captures both pen and multi-touch input. It can be adjusted smoothly from a vertical position to a nearly horizontal orientation, the latter resulting in a slight inclination of $\approx 20°$ from the horizontal. As AR HMD we used Microsoft's HoloLens 2 (2,048 × 1,080 px per eye, 52° FoV), equipped with spatial tracking and gesture recognition. We connected both to a PC (Intel Core™ i9, 3.5 GHz, 64 GB RAM, GeForce RTX3090, 24 GB video memory), the AR HMD via holographic remoting. We attached a physical button to the Surface pen, along with the Arduino nano board on the back so that users could initiate actions both in 3D space and on the 2D surface. When the Arduino detects a button press, we compute the pen position based on the user's index finger position—tracked by the HoloLens using computer vision—, to which we add a constant offset to arrive at the pen tip. The resulting pen precision was sufficient for our prototype implementation. For more precise input, however, we could explore advanced technologies such as Logitech's VR Ink Pilot Edition.

We developed our prototype implementations with C# in the Unity 3D engine and ran on both the Surface and the HoloLens. We realized the communication between both display spaces through Universal Windows Platform sockets. After starting the program on the Hololens, we calibrate the setup by aligning the two coordinate systems via QR code tracking and manually manipulating the anchor point, as in past work [46]. After calibration, we employ HTC Vive trackers to follow the position and orientation of the Surface Studio.

6.4 Elicitation Study

To explore potential interaction designs for SpatialTouch, we conducted an elicitation study to gauge users' reactions to this new environment. Similar to Wang et al. [248], our study encouraged participants to propose any interactions they could imagine. We captured their responses and interactions, subsequently incorporating this feedback into our final interaction techniques design. We pre-registered our study (`osf.io/avxr9`) and received IRB approval for the protocol (XJTLU University Research Ethics Review Panel, № 20240201174957).

6.4.1 Study Setup

Based on SpatialTouch that we just introduced in Sec. 6.3, we had the following goals: (G1) to determine whether an optimal position (or range) exists for the 2D surface to facilitate

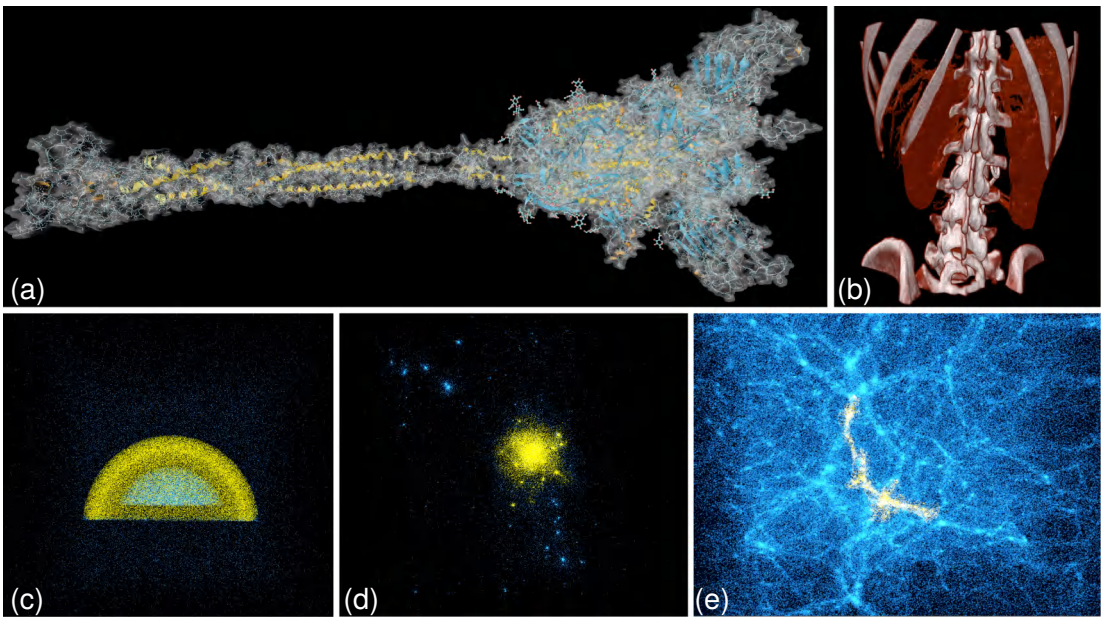


Figure 6.4: Elicitation study datasets: (a) Protein data (data from [249]) , (b) MRI multi-slice anatomical data, (c) filament.

the observation of 3D data and to enhance user interaction; (G2) to understand how users perceive information within a CR environment; and (G3) to observe the participants' interactions and strategies while they complete exploration tasks with 3D spatial data, as well as to identify the physical locations of where the interactions occur.

Participants. We recruited eight voluntary participants from the local university, with ages ranging from 24 to 41 years old (mean 26.5, SD 5.5). The participants' past VR/AR experience included weekly use (3×), annual use (3×), and no past experience (2×). On average, they had 5.3 years of experience in the visualization or human-computer interaction fields. One participant was left-handed. All participants had normal or corrected-to-normal vision with no color deficiency, ensuring a clear ability to perceive the data visualization and colors in our study.

Datasets. To represent a variety of application domains, we used five datasets with a diverse set of data features, including:

Structured surface data: The molecular structure data (Fig. 6.4(a)) is a spike protein from the SARS-CoV-2 virus [249], reconstructed from the electron microscopy images. It is rendered as a ribbon diagram, which includes lines, sheets, and helix structures.

Volume visualization data: The MRI volume data (Fig. 6.4(b)) comprises multiple slices

and facilitates interactive cutting plane exploration to focus on specific regions of the anatomical structures.

Unstructured point cloud data: We used three point cloud datasets: (1) a synthetic semi-spherical shell of particles (Fig. 6.4(c)) that partially encompasses a half-ball of interfering particles; (2) a cosmological N-body simulation [107] (Fig. 6.4(d)) with densely populated central cluster encircled by numerous smaller clusters; and (3) Millennium-II data subset [108] (Fig. 6.4(e)), a complex network of filaments connecting high-density clusters.

Task and Procedure. After providing their written informed consent, we asked participants to comfortably sit on a chair ≈ 0.5 m above the floor. We placed the surface screen on the table, ≈ 0.75 m from the floor, with an initial angle of $\approx 20°$ to the horizontal. Participants could adjust the chair height if desired. We then asked them to complete the three tasks described below. Importantly, for both manipulation and selection tasks, we did not show any feedback to the participants on their proposed interactions. We only provided them with a data context with highlighted (exploration or selection) targets and asked them to imagine how actions should be performed according to the given tasks.

Task 1. We instructed the participants to adjust the screen angle to achieve a suitable position for an optimal observation of the 3D structures. In addition, we reminded them of the importance of considering interaction on both the 2D Surface and in 3D space for a comprehensive understanding. We also informed the participants that they could re-adjust the screen angle whenever needed, throughout the study.

Task 2. We asked the participants to manipulate the five spatial datasets. In all manipulation tasks, we provided them with a target object and a target position. Specifically, we asked them to: (1) translate data displayed below and above the surface within each display space; (2) translate data displayed below and above the surface to the other display space; (3) translate data displayed partially in both spaces to a target position above or below the surface; and (4) rotate and scale data located in 3D space, within a 2D surface, or partially in both spaces.

Task 3. We asked the participants to select a given highlighted target object or highlighted regions of interest.

We recorded the screen angle chosen by participants, their actions (incl. head/hand position, orientation), and device input events for our analysis (which we make available at `osf.io/avxr9`). After completing the three tasks, we conducted semi-structured interviews with them to discuss their thoughts and suggestions on the interaction.

6.4.2 Findings

Our participants demonstrated that the integrated 3D data visualization in our environment allowed them to gain a comprehensive understanding of the 3D content. They mainly concentrated on the 3D visualization in the AR space and attempted to view it from various angles. Their decision on where to interact with the data was influenced by the location of key information such as target data or target location. Their preference for direct manipulation methods remained consistent, however, such as touching the data on the surface or brushing it in the air. Below we present our findings on surface angle, visualization techniques, and interaction techniques tailored to our CR environment.

Surface Angle

From the initial $\approx$20° incline w.r.t. the horizontal table, some participants made slight adjustments at the beginning of the study, with angles ranging from 18.2° to 22.6° (mean 21.0°). After the initial setup, however, they did not make any further adjustments, despite being informed that they could do so. During interviews, they said the initial setting provided adequate space for observing stereoscopic 3D data through the AR HMD, while offering a convenient position and orientation for interacting with data presented on the surface.

We also found that, when comparing our setup with a past stereoscopic multi-touch system [250, 223], a notable difference in display orientation: Butkiewicz et al.'s screen was nearly perpendicular to the table. This difference initially seems strange because Butkiewicz et al. [250, 223] had also relied on innovative interaction techniques using two-finger pinch gestures on the touch screen for tasks such as positioning the cursor under the display for 3D exploration and selection. The only difference appears to be that they had employed an auto-stereoscopic touch display, while we used an AR HMD. A main reason for the difference then appears to be the additional 3D space visible through the AR HMD, *decoupled from the touchscreen*, which substantially reduced people's reliance on the 2D surface. In the recorded data, in tasks where the visualization spanned both two spaces we saw that $\geq$67.5% of the time was spent on observing 3D data in AR space. Users apparently preferred the stereoscopic AR space and only referred to the 2D surface when needed. Our environment also encouraged users to actively move around to find a good view, rather than adjusting the surface angle. All these factors diminished the importance and constraints of the surface angle. Moreover, the $\approx$20° angle is also appropriate for

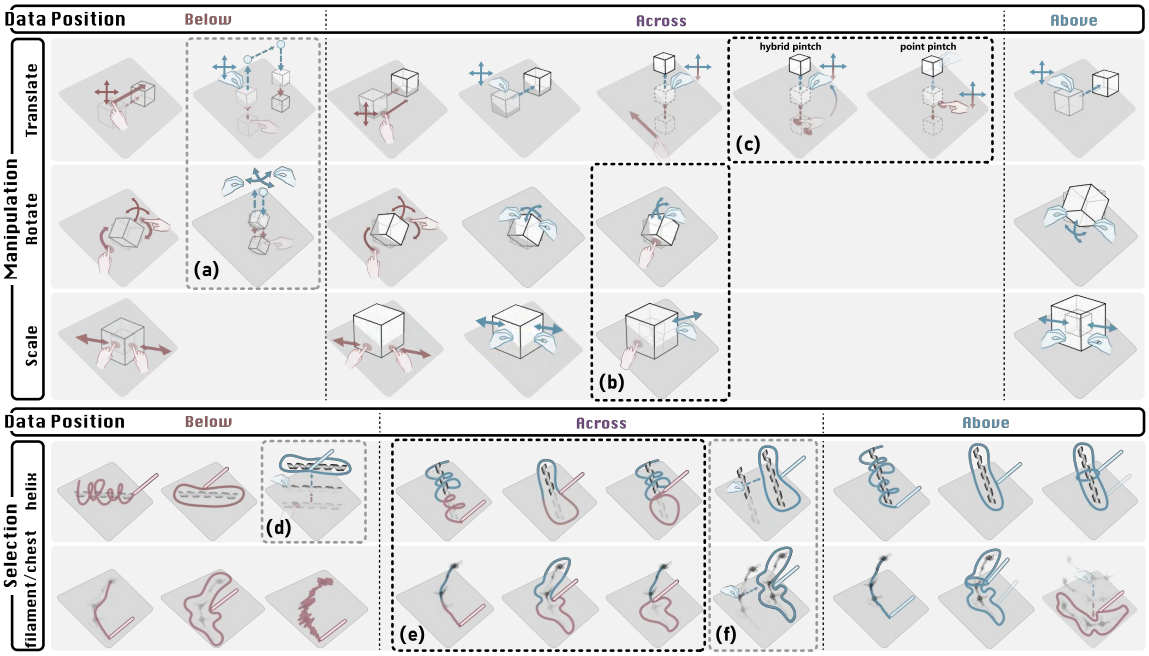


Figure 6.5: Design space for interaction techniques for two visualization tasks: data manipulation and selection. Red: interactions on 2D surface; Blue: interactions in 3D space. **Below**, **Across**, and **Above**: positions of target data/location. (a), (d), (f): move data above the surface for interaction. (b) (c) and (e): interaction transitions across spaces.

design work [46] as it provides stability and physical support from the base, enabling users to interact confidently and securely.

Visualization in CR

In addition to the Fish Tank VR view [247] on the 2D surface, we also asked participants to compare it with a static orthogonal view and a 2D slice view as alternatives. All participants expressed a strong preference for the perspective Fish Tank VR view, highlighting its effectiveness in creating a seamless 3D experience. Several participants recognized the significance and necessity of 2D slice view on the surface, particularly for tasks such as adjusting 2D image slices or marking MRI volume data. Furthermore, they also noted that the see-through AR HMD allowed them to look through the stereoscopic rendering directly to the 2D surface visualization. The resulting experience thus combined a comprehensive 3D data understanding with the ability to observe projected information, and the environment is flexible in how and where the information should be presented.

User Interaction in CR

Interactions in 2D or 3D display spaces. When observing our participants engaging in either of the two individual methods of displaying and manipulating spatial 3D data—projected on the 2D surface or stereoscopically in 3D space—we noticed that they have a well-established mental concept of how an interaction should be performed. As we show in Fig. 6.5 (Manipulation, **Below**), when participants manipulated data presented below the surface, they used one finger to pan the visualization and two fingers to rotate and scale it on the surface. Conversely, when they saw data rendered in the AR HMD above the surface (Fig. 6.5, Manipulation, **Above**), they used one or two hands to grab and manipulate the data directly. Similarly, for the selection tasks, when the data was presented below the surface (Fig. 6.5, Selection, **Below**), participants preferred to brush or circle data on the surface. When the data was entirely above the screen (Fig. 6.5, Selection, **Above**), however, brushing and circling interactions were performed in 3D space.

Interestingly, some participants also used the benefits of the environment to aid their understanding of the data structure or to make more precise inputs. When both the target data and target location were below the surface, for instance, some participants would grab it out and put it back underneath to position the target. They explained in the interview that this extra action would provide them with more precise control over the depth in the positioning. Another example is that, to make precise selections, some participants would pull the data entirely out of the surface so that they can see the whole dataset stereoscopically or, alternatively, push it entirely below the surface for precise selection input. We highlighted these interactions in Fig. 6.5 (a) and Fig. 6.5 (d,f).

Interactions across Two Display Spaces. Two types of tasks required interaction across both spaces: if the data was visualized across both spaces or needed to be translated from one to the other. We show actions in **Across** in Fig. 6.5 (Manipulation, Selection).

In the rotation and scaling tasks with a visualization spanning both spaces, participants performed actions on either the surface, in 3D space, or using both spaces for different tasks. They mainly made this decision based on where they perceived the key information. In addition, 4 participants favored using both spaces, most of them with a notable strategy: they (3×) used one finger to designate a rotation/scaling center on the surface, while using their other hand to execute the rotation or scaling actions around the selected center (Fig. 6.5(b)). In the interviews, these participants mentioned for them that this approach

was a more precise and secure method of manipulation.

In the selection tasks in which we placed the data across both spaces, we also observed intriguing patterns. Most participants (6×) approached our environment as an integrated system and used consistent selection strategies across both spaces. For example, participants followed the 3D structure and brushed along the data in 3D space, then continued brushing the rest along the structure projected on the 2D surface (Fig. 6.5 (e, left)). Or they drew a large lasso across both spaces to enclose the data (Fig. 6.5 (e, middle)). These selection strategies remained consistent even when they physically reached the surface and noticed that they could no longer follow the spatial structure in the depth direction.

The most interesting finding is that participants maintained their mental model when interacting with data across both display spaces. Some employed a mixed approach, brushing data in 3D space and circling data on the 2D surface, or vice versa (Fig. 6.5 (e, right)), which indicates that they maintained spatial awareness and seamlessly transitioned between different interaction techniques in the hybrid environment.

In tasks requiring transition across both spaces, we observed two distinct transition methods (Fig. 6.5 (c)). First, when the data was situated below the surface and needed to be moved out, some participants initially employed 2D pinch gestures to pull it out. Once their two fingers naturally came into physical contact, they seamlessly transitioned to 3D pinch gestures to continue the action. They used the same method to put the data back in its original position. Second, some participants pointed at the data on the 2D surface, assuming that it would automatically approach their finger position. They then continued the manipulation using 3D pinch gestures to pull it out. In the AR space, all participants uniformly used 3D pinch gestures, which facilitated positioning of the data—whether in 2D or 3D space—in a rapid manner.

Interactions on Visual Representations. Participants employed a consistent approach in manipulating all three datasets. They exhibited, however, varied approaches for selecting data with different representations in our environment. For the structured surface data, participants selected it by adhering to its 3D structure, such as wrapping a protein helix with a helical stroke. Conversely, for the unstructured point cloud data, they attempted to enclose it with a freeform lasso, such as drawing a lasso around the astronomical point cloud data. We observed that participants relied on their established mental models to determine how the data should be selected, based on their perception and understanding of the specific data structures. Therefore, most participants did not alter their selection strategies when

interacting across both spaces—even though their interactions were physically limited to reach the visualization below the surface: they believed that through continuous input they could successfully select data within the CR environment.

6.5 SpatialTouch for Domain-specific Use

Based on these findings, we then focused on developing SpatialTouch sample applications tailored to three distinct domains: astronomical point cloud analysis, molecular visualization, and medical anatomy imaging—each with its unique challenges for interactive visualization. Based on the optimal angle for the 2D surface identified in the elicitation study (G1), we set the angle of the surface at a 21° incline w.r.t. the horizontal table. We use the Fishtank VR view on the Surface camera and a perspective projection in AR HMD camera to ensure a seamless visualization (G2). In the following description we highlight, in particular, specific realizations within SpatialTouch for the given type of data or domain.

6.5.1 Astronomical Point Cloud Visualization

Astronomical simulation datasets typically comprise billions of spatial points [107, 108]. Researchers often need to navigate in 3D to obtain a clear view of the structures to select and explore the regions of interest. The task of data selection becomes paramount in this context, serving as a critical step in data visualization and exploration [47, 27]. While much research has been dedicated to developing selection techniques for 2D surfaces [58, 59, 251, 57, 88] and in 3D space [77, 103, 96, 252, 253], these methods are predominantly designed for a single display. In 3D space, although users can clearly see the spatial data distribution, limitations persist in their ability to precisely delineate selection regions. Conversely, the 2D surface facilitates accurate input for target inclusion but struggles with depth prediction, posing a challenge for selecting specific data regions. SpatialTouch bridges this gap by supporting data observation in AR, affording users a comprehensive understanding of data density distribution and context, while also facilitating precise data selection on a 2D surface. As an integrated environment—users are empowered with the flexibility to decide both *where* the selection occurs (whether on the 2D surface, within 3D space, or spanning both) and *how* selections are made (via a freeform lasso or a brush).

Based on the interaction techniques detected from the elicitation study (G3, Fig. 6.5), we

developed two new spatial selection techniques to facilitate a seamless transition from 3D selection to 2D selection, and vice versa. These techniques are based on the context- and target-aware selection metaphors CloudLasso [58], WYSIWYP [87], and MeTABrush [77]. These methods analyze the density distribution and data features within the local area of user interaction so that users can identify and select critical features of interest, without the need for precise input. Our first new technique, *BrushWYP*, draws inspiration from the user interactions illustrated in Fig. 6.5 (e, left), MetaBrush [77], and WYSIWYP [87]. It enables users to trace the string-like shape of 3D point cloud data by brushing over these structures in the 3D space and continuing to brush the rest along the structure on the 2D surface. We designed our second new technique, *BrushLasso*, based on Fig. 6.5 (e, right), MetaBrush [77], and CloudLasso [58]. This method allows users to begin their selection by brushing over the data in the 3D space and draw a lasso on the 2D surface to enclose the targeted points. Beyond the ability to transition between two spaces, both techniques are flexible in that they also allow users to brush the target points on the 2D surface and in 3D space, or encircle the target points on the 2D surface. Moreover, similar to our previous works [58, 59, 77], subtraction can be achieved using region-based techniques. A potential action to activate subtraction could be, for instance, to turn the Surface pen over and to rub the end of the pen over the selected data.

Technically similar to MeTACAST [77], we leveraged a continuous density field $\rho(\mathbf{r})$ to represent the particle density at location $\mathbf{r}$. We pre-compute the density of the field at all nodes i of the regular $128\times128\times128$ 3D grid box B that covers the dataset, denoted as $\rho(\mathbf{r}^{(i)})$, offline on GPU. We use the modified Breiman kernel density estimation with a finite-support adaptive Epanechnikov kernel [114]. This approach allows us to apply our selection not only to point clouds but also to volumetric data, which samples a scalar field that represents important data aspects in a visually salient way (not limited to density).

BrushWYP. We employ MeTABrush to allow users to select structures in 3D space by direct tracing along 3D structures. When the input transitions from the 3D space to 2D surface, however, the direct brushing on the target location is constrained. We address this issue with a modified WYSIWYP method, enabling users to select points below the surface by identifying the correct depth of the point of interest (POI).

Our algorithm initiates by sampling points along the input stroke on the surface, denoted as $\mathbf{r}^{(s)} = \{\mathbf{r}^{(s_0)}, \mathbf{r}^{(s_1)}, ...\mathbf{r}^{(s_n)}\}$, as well as above the surface, represented by $\mathbf{r}^{(a)} = \{\mathbf{r}^{(a_0)}, \mathbf{r}^{(a_1)}, ...\mathbf{r}^{(a_m)}\}$. For each sampled point $\mathbf{r}^{(s_i)}$ in $\mathbf{r}^{(s)}$, we project a ray from $\mathbf{r}^{(s_i)}$ toward

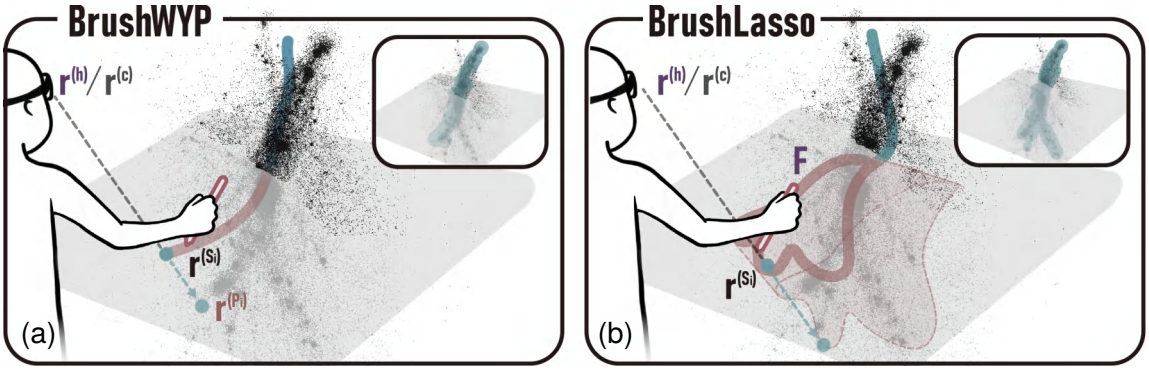


Figure 6.6: Point cloud data selections: (a) BrushWYP, (b) BrushLasso.

the direction of $\mathbf{r}^{(s_i)} - \mathbf{r}^{(h)}$, where $\mathbf{r}^{(h)}$ denotes the position of the AR HMD (Fig. 6.6(a)). Then we traverse along each emitted ray in fixed steps and search for the POI $\mathbf{r}^{(p_i)}$ that exhibits the maximum density along the ray. This approach is different from the original WYSIWYP [87], which identifies the highest jump of accumulated scalar value along the ray. After traversing $\mathbf{r}^{(s)}$, we get a list of POIs, $\mathbf{r}^{(p)} = \{\mathbf{r}^{(p_0)}, \mathbf{r}^{(p_1)}, ...\mathbf{r}^{(p_n)}\}$. Subsequently, we obtain the input array of MeTABrush as $\mathbf{r}^{(i)} = \{\mathbf{r}^{(p)}, \mathbf{r}^{(a)}\}$ and select the target points with the MeTABrush method.

BrushLasso. We integrate MeTABrush and CloudLasso to provide users with the ability to brush target points in mid-air and encircle points on the surface through a single, seamless input. The original CloudLasso selects point cloud clusters of high density that fall within the input lasso. Directly merging both methods, however, could lead to disconnected selection volumes—one above the surface and several isolated ones below the surface—which might not align with the user's intention of making a continuous selection from the 3D space to the 2D surface. To solve this issue, we implemented a modified CloudLasso method that ensures that the selection volume is smooth and continuous.

This algorithm initiates by sampling points along the input stroke on the surface. For each sampled point $\mathbf{r}^{(a)}$ above the surface, we calculate an initial volume of interest (VOI) V_{init} with MeTABrush algorithm [77]. We then remove the parts below the screen and keep only the VOI above the surface $V_{\text{a}} = V_{\text{init}} \cap V_{\text{3D}}$, where V_{3D} is the space above the surface. We then connect the sampled points on the surface, $\mathbf{r}^{(s)}$, to form a lasso, **L** (Fig. 6.6(b)). We then map the particle coordinates to the view coordinates of the surface camera, **c**, using the model-view transformation. Similar to CloudLasso [58], we then compute the lasso frustum **F** based on the first-level binding stage. We can then obtain the VOI below

the surface $V_{\mathrm{b}} = \mathrm{F} \cap V_{\mathrm{2D}}$, where V_{2D} is the space below the surface. Finally, we combine both VOIs to get the interconnected VOI, $V_{\mathrm{CR}} = V_{\mathrm{b}} \cup V_{\mathrm{a}}$. We compute the initial density threshold ρ_0 as:

$$\rho_0 = \frac{1}{N_{V_{CR}}} \sum_{n=1}^{N_{V_{CR}}} \rho(\mathbf{r}^{(n)}), \tag{6.2}$$

where $N_{V_{\mathrm{CR}}}$ is the number of grid-nodes, $\mathbf{r}^{(n)}$, inside of the combined VOI, V_{CR}. We select the volume V with density ρ above ρ_0 within V_{CR}:

$$V = \{\mathbf{r} \,|\, \mathbf{r} \in B,\, \mathbf{r} \in V_{CR},\, \rho(\mathbf{r}) > \rho_0\}, \tag{6.3}$$

We generate the iso-surface with Marching Cubes based on the density threshold ρ_0. Both resulting selection techniques facilitate a seamless and natural transition between 2D and 3D selections.

6.5.2 Molecular Visualization

Molecular visualization is a field rich with diverse data representations, each designed to highlight various aspects of molecular structures and interactions. These include space-filling diagrams, ball-and-stick models, ribbon models, licorice visualization, backbone, and surface visualizations, etc. Given the complexity of molecular interactions, it is common to use these representations as a combination to provide a comprehensive view of the molecules. For biologists and chemists, gaining a deep understanding of how drug molecules interact within larger molecular structures is essential. They require insights into the precise location and distribution of these molecules to assess their interaction and affinity. Traditionally, they need to select specific features or regions of interest from biological sequences and link these on a 3D visualization rendered on a 2D display. The task of selecting spatial features and observing the dynamic behavior of molecules through 2D display, however, presents significant challenges. Even within a pure 3D environment, maintaining spatial awareness can be a struggle for users, given the inherent complexity of the involved datasets.

With SpatialTouch we address this need by merging the distinct display spaces to facilitate a concurrent visualization of different representations and abstractions. As we illustrate in Fig. 6.7, e. g., we can show the hyperball representation [255] stereoscopically in

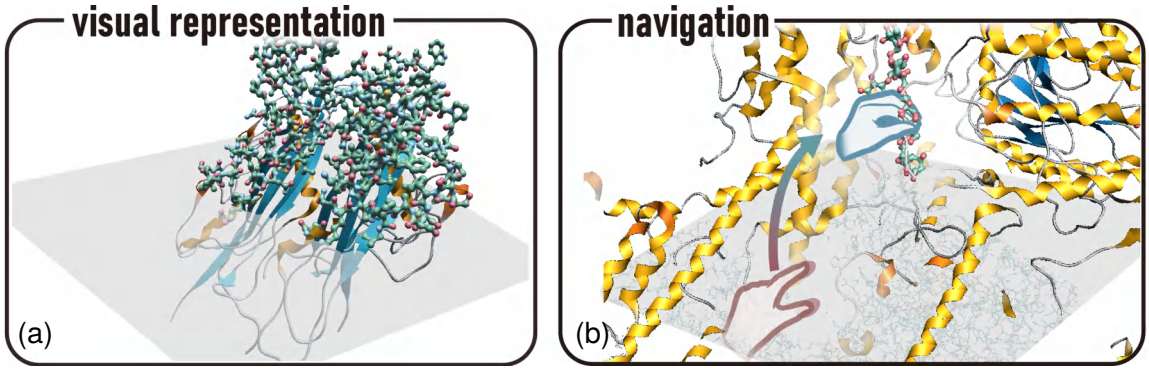


Figure 6.7: Proteins visualized with UnityMol [254]. PDB ID: (a) 4fpq, (b) 8rfe. Users can "grab" the 3D visualization directly to check the local regions.

AR space for a detailed view of the molecule, while a ribbon model is rendered on the surface for context. Alternatively, we can put a ribbon diagram in AR space, complemented by a licorice diagram on the surface. SpatialTouch allows users to combine these representations based on their preferences and exploration needs. In addition, we enable them to adjust visual representations in the specific regions as needed, both in AR space and on the surface (Fig. 6.7(a)). Users can also directly grab the visualization to check details of local regions, as we show in Fig. 6.7(b).

We enable users to interact with data rendered below the surface using familiar touch interactions or above the surface through gesture interactions, as illustrated in Fig. 6.5 (manipulation, **Below** and **Above**). In addition, inspired by the findings in the elicitation study as illustrated in Fig. 6.5(c), we developed two seamless interaction transition techniques to enable users to move data across two spaces. The first method supports users to employ a familiar 2D pinch gesture on the touch surface to "pull" the data to the surface level (akin to Hancock et al. [256]), then continue with a 3D pinch gesture to "grab" it further into the air above the surface, as illustrated in Fig. 6.5(c, left) and Fig. 6.7(b). Conversely, performing this gesture in reverse enables users to seamlessly return the data to below the surface. This approach supports users to directly interact with and manipulate regions of interest, whether transferring them from the 3D space directly onto the 2D surface for extended analysis or relocating them to any other area within the CR environment for closer inspection. It is worth noting, however, that 2D pinch gestures are typically associated with zooming interactions. The second method thus allows users to press the screen to "push" data further into the depth of the surface or to "pull" it to the outside. The 3D pinch interaction in AR HMD remains, as we illustrate in Fig. 6.5(c, right). While both methods

provide seamless translations across two spaces, we also implemented scaling and rotation: users can point at a particular feature on the 2D surface and use 3D pinch gestures to rotate or scale the data around that chosen point, as illustrated in Fig. 6.5(b).

6.5.3 Medical Anatomical Visualization

Medical imaging plays a central role in many healthcare practices for diagnosis, treatment planning, and patient care. In particular, 3D anatomical visualization has been used for treatment planning for various medical procedures in radiology [257] as well as for teaching [13]. It often relies on volume rendering to show internal structures in detail. While 2D slices are commonly used and are particularly effective for many diagnostic purposes, 3D visualizations are often necessary for addressing complex cases such as those involving intricate fractures. Physicians also frequently interact with both 2D slices and 3D visualizations for a range of tasks, including assessing injuries and devising treatment plans [258]. When planning surgical interventions for complicated orthopaedic injuries, for example, the integration of 2D and 3D visualizations allows surgeons to accurately visualize and navigate the affected areas [259]. This combined approach significantly enhances the precision of both planning and executing surgical procedures.

Prior work has explored the use of 2D displays [260] and 3D environments [261] for exploring 3D medical visualizations within clinical research tasks. Particularly relevant is Slice WIM [2], which uses a stereoscopic display and a multi-touch table to present both an overview and detailed views of 3D medical data, and projects 2D slices onto a wall or table display. For SpatialTouch, in contrast, we use an integrated environment that allows users to view 2D slices directly on the surface, while also observing stereoscopic renderings superimposed on these slices. Furthermore, users can use an interactive Surface pen for precise annotation and distance measurements. Moreover, with our method selected 2D slices can be saved and set aside on the surface, facilitating quick observation and navigating to specific local structures (Fig. 6.8(b)).

With this example we want to highlight that domain experts often need to engage in precise interactions, based on their observations of 3D structures—our environment being able to fulfill these requirements. Physicians can make annotations and measure distances directly on the 2D touch surface, while simultaneously viewing the 3D representation in AR space. To provide a clear view of the 2D slices and facilitate accurate interactions

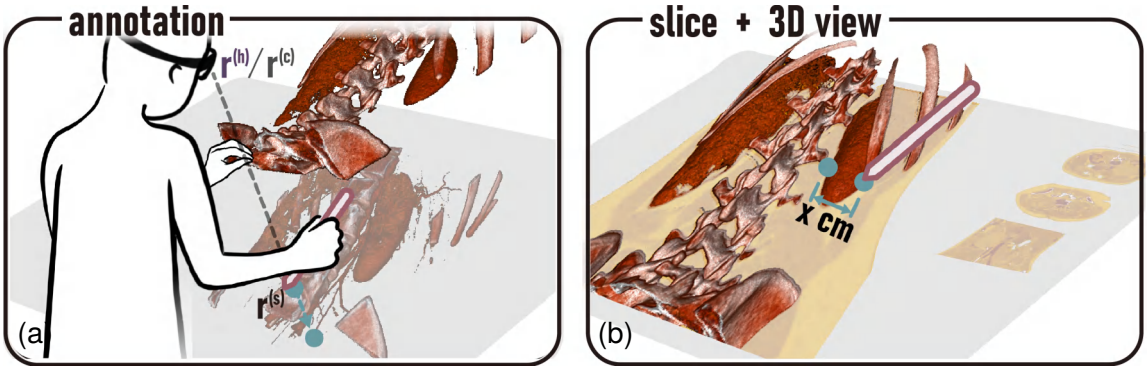


Figure 6.8: (a) Annotation on 2D surface while grabbing 3D data; (b) 3D visualization superimposed on 2D slices to facilitate distance measurement.

on the surface, we implement a "lifting" feature. It is activated with a 3D pinch gesture, which temporarily elevates the 3D volume visualization away from the 2D slice on the touch surface. This separation facilitates unobstructed annotation or marking. Once these tasks are complete, releasing the pinch gesture returns the 3D volume visualization to its original position on the screen, seamlessly integrating the new annotations with the remaining data.

In addition, with SpatialTouch interaction can happen anywhere within CR, including making annotations directly on the 2D slice (on the surface), marking features on the 3D visualization (above the surface), and selecting features that appear in depth but are rendered below the surface. We realized the latter interaction based on the WYSIWYP principle [87]. To identify the depth of a ROI, our method detects a significant change in the accumulated scalar value along the ray. This ray originates from the contact point on the surface, denoted as $r^{(s)}$, and projects it in the direction of $r^{(s)} - r^{(h)}$, as shown in Fig. 6.8(a). Our integrated environment thus offers users a seamless and precise interaction with medical 3D data for a detailed immersive examination.

6.5.4 Illustrations of the Application Cases

This section presents visual illustrations of the three application cases, demonstrating how SpatialTouch adapts to different domains.

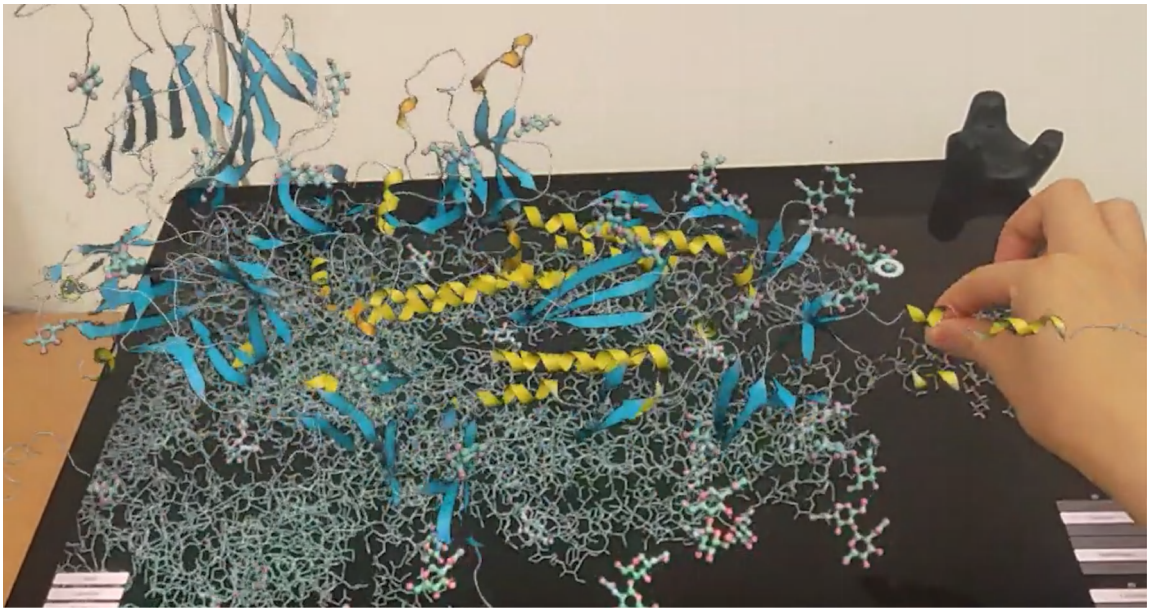


Figure 6.9: Protein data visualization [249] with the ribbon (shown in AR) and the licorice (shown on the surface) visual representations.

Molecular Visualization

As shown in Fig. 6.9, we merge two distinct display spaces (AR HMD and Surface) to facilitate a concurrent visualization of different representations and abstractions. We developed a seamless interaction transition technique to enable users to move data across two spaces. It supports users to employ a familiar 2D pinch gesture on the touch surface to "pull" the visualization below the surface to the surface level and continue to use pinch gesture in 3D space to manipulate the visualization (shown in Fig. 6.10).

Medical Anatomical Visualization

SpatialTouch allows users to view 2D slices directly on the surface, while also observing stereoscopic renderings superimposed to the slice (Fig. 6.11(a)). Users can pinch up the 3D volume visualization away from the 2D slice on the touch surface (Fig. 6.11(b)). In this way, users can obtain a clear view of the 2D slices and perform accurate interaction on the surface, such as precise annotation (Fig. 6.12(a)) and distance measurements (Fig. 6.12(b)). SpatialTouch supports marking features on 2D slice directly (on the surface). This method is useful when features appear in depth but are rendered on the surface.

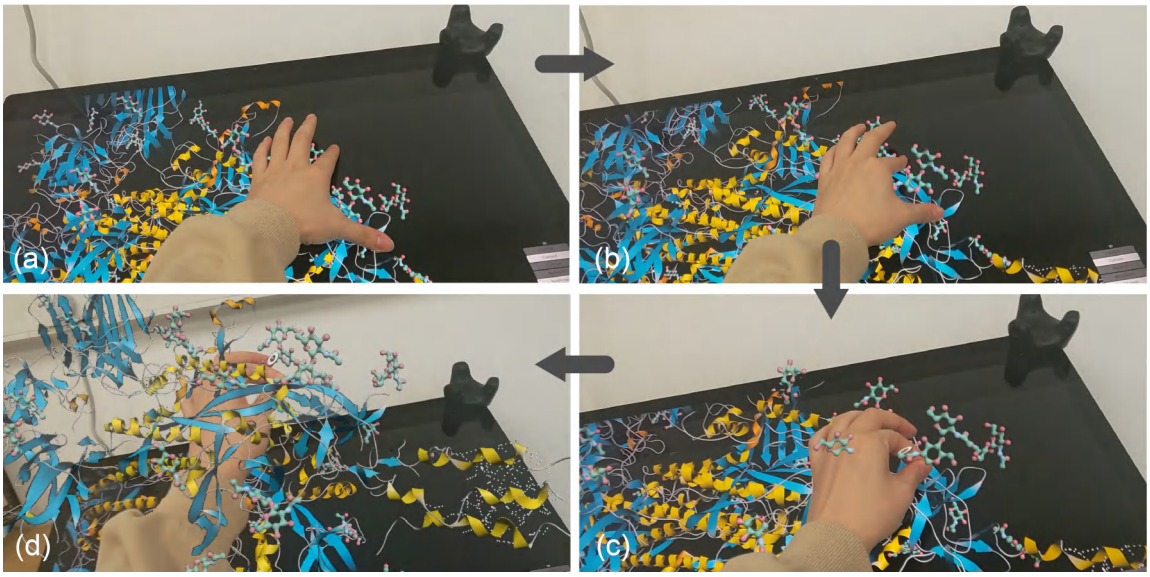


Figure 6.10: The interaction transition technique to move the data visualization from 2D to 3D spaces. (a) start pinching on the surface to "pull" the visualization from 2D to 3D. (b) during the pinching, the visualization "moves towards 3D space" with the distance between the thumb and index finger decreasing. (c) The two fingers merge and the visualization is pulled up to near the surface. (d) Continue to move the visualization by pinching in the air.

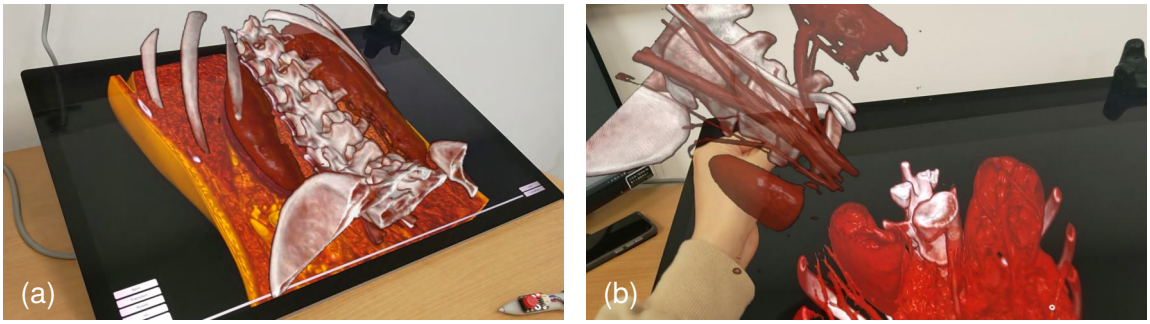


Figure 6.11: SpatialTouch allows (a) visualizing 3D anatomic data on the surface and AR device with different transfer function settings and (b) picking up the stereo rendering with the pinch gesture.

Astronomical Point Cloud Visualization

As shown in Fig. 6.13, users can view 3D point cloud visualizations in AR space to gain a comprehensive understanding of data density distribution and context, while performing precise data analysis, such as selection or annotation, on a 2D surface. We developed two seamless spatial selection techniques for point cloud visualization. With BrushWYP

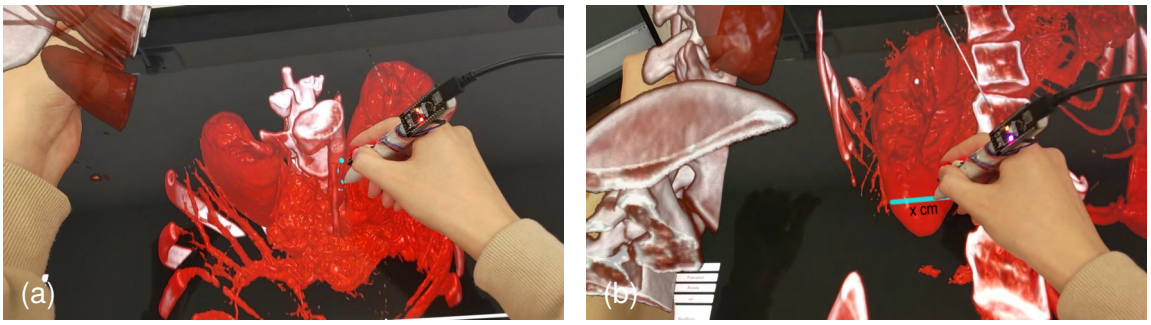


Figure 6.12: Lifting the data and (a) annotating the features in depth, (b) measuring the width of the lung with a pen on a 2D surface.

Figure 6.13: Cosmological N-body simulation visualization [107] across 2D (dark blue) and 3D environments (light blue).

(Fig. 6.14), users can brush over the string-like shape of 3D point cloud data in the 3D space and continue to brush the rest along the structure on the 2D surface. With BrushLasso (Fig. 6.15), users can brush target points in mid-air and encircle points on the surface through a single, seamless input. In addition, users are able to brush target points only in mid-air with MeTABrush [77] (Fig. 6.16), or draw a lasso around them on the surface with CloudLasso [58] (Fig. 6.17).

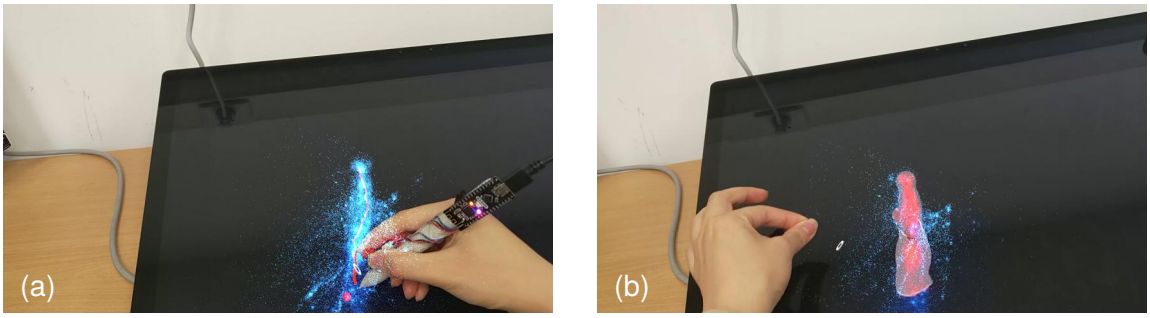


Figure 6.14: BrushWYP: (a) direct brushing on the target points across two spaces, and (b) the selection result.

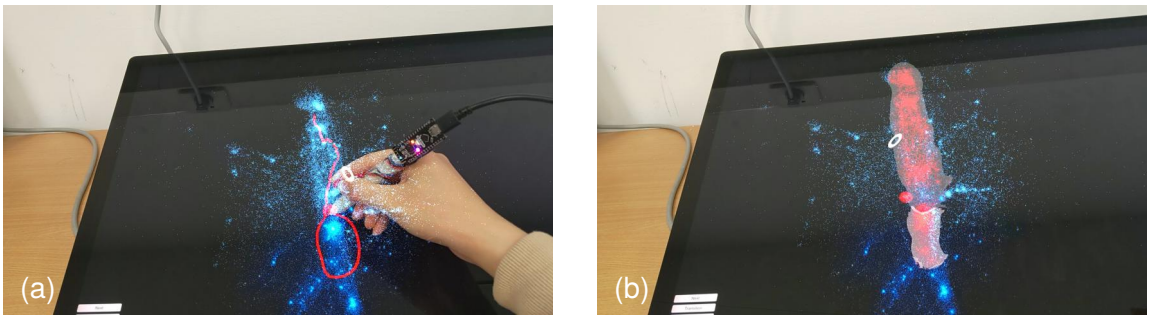


Figure 6.15: BrushLasso: (a) brushing the target points in mid-air and drawing a lasso on the surface, (b) the selection results.

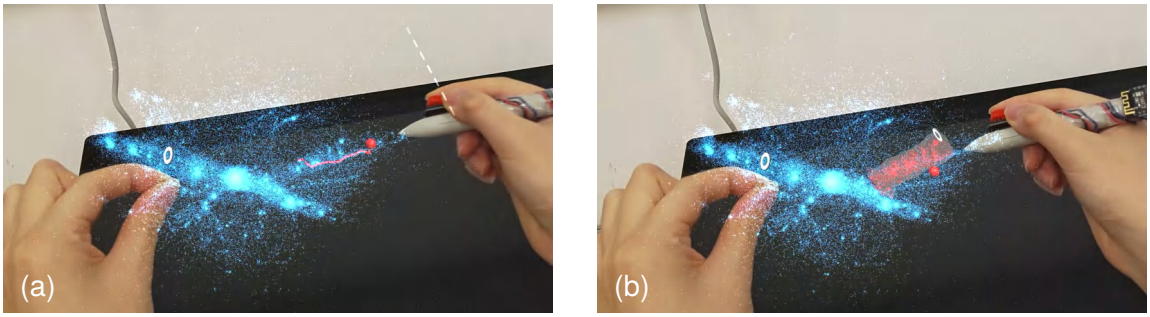


Figure 6.16: MeTACAST: (a) brushing the target points directly in mid-air, (b) the selection results.

6.6 Evaluation

We ran two evaluations with domain experts, both pre-registered (`osf.io/avxr9`) and IRB-approved (XJTLU University Research Ethics Review Panel, № 20240201174957). First, we evaluated SpatialTouch's usability with domain experts, focusing on its use in the domain. Second, we assessed the interaction with VR/AR/MR experts.

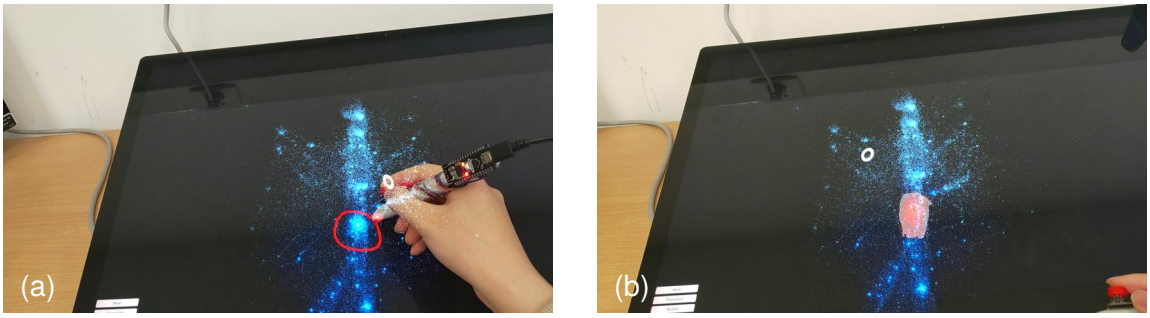


Figure 6.17: CloudLasso: (a) drawing a lasso around the target points on the surface, (b) the selection results.

6.6.1 Evaluation with Domain Experts

We presented SpatialTouch to four domain experts: two biologists and one astronomer from the local university, and one doctor from a local hospital. All experts had $\geq$10 years of professional experience.

With the biologists, our demonstration focused on the SARS-CoV-2 virus spike protein [249] (Fig. 6.1(b)). The domain experts appreciated that the 2D and 3D perspective views offer a thorough understanding of the molecular data. They were highly interested in transitioning the visualization between 2D and 3D spaces, finding it intuitive to grab the visualization out from the 2D screen to check its 3D structure. They mentioned several scenarios where SpatialTouch could significantly benefit their research. First, the stereoscopic rendering in 3D space enhances molecular structure comprehension. It enables them to estimate distances between structures and gain a better understanding of molecular interactions and alignments. Second, 3D space provides a flexible interaction platform with more degrees of freedom. One expert noted that "this environment would be particularly useful for data analysis. I can analyze two molecular datasets on the surface and grab them to the 3D space to test various angles for fitting them together and observing their interaction. Afterward, I can return them back to the surface for further analysis." Third, the integration of 2D surface and 3D space combines their unique strengths, enhancing usability. Linking 2D representations on the surface with 3D visualizations in AR space, for instance, facilitates simultaneous editing of legends and observation of molecular interactions. Users can create and compare multiple copies within the environment. Moreover, the experts discussed how SpatialTouch could change their approach to data selection. Currently, they select data based on amino acid sequences. This process is cumbersome when the

target region is spread over multiple sequence segments. To optimize the ligand affinity, for instance, they need to check the surrounding amino acids and select the pieces one by one from the sequence. Thus, they look forward to intelligent spatial selection techniques that facilitate efficient selections across 2D and 3D representations to streamline this process.

The astronomer specializes in computational astrophysics, planetary system dynamics, few-body systems, and star cluster dynamics. After seeing the N-body simulation visualizations (Fig. 6.1(a)) in SpatialTouch, the expert mentioned that 3D stereoscopic rendering enhances the understanding of data structures, providing an intuitive grasp of spatial relationships and dynamic processes within the data. Its capability is particularly beneficial for exploring simulations of physical phenomena and identifying specific data patterns. Conversely, the 2D surface can be used for coding to perform complex data analysis tasks. The combination of both visual interfaces can allow users to leverage their existing data analysis practices, while motivating them to test their hypotheses and observe 3D simulations more easily. The expert did not show a strong preference for whether the selection interaction should occur on the 2D surface or in 3D space. Instead, they highly valued the flexibility to select based on where the data feature is perceived.

The doctor's research focuses on orthopaedics. After observing and manipulating the anatomical data visualization (Fig. 6.1(c)), he expressed strong interest in SpatialTouch and believed that it would play a significant role in surgical planning, where multi-views of 3D structures, 2D measurements, and precise interaction are required. The smooth and flexible transition between 2D and 3D allows doctors to view medical structures and the surgical plan from different perspectives. In addition, he mentioned that it is often difficult to control the depth at which a needle or a screw is inserted, which can harm neighboring tissues or organs. With SpatialTouch, they can view specific treatment locations on the 2D interface and test, compute, and plan where and how needles or screws should be inserted in 3D space.

6.6.2 Evaluation with MR Experts

We then invited three MR experts with each over three years of experience and daily technology use. After obtaining their informed consent, we introduced them to SpatialTouch, in two sessions. First, we focused on the manipulation tasks, including translation, rotation, and scaling. After a brief tutorial, we asked them to use our techniques to manipulate the

molecular visualization from Sec. 6.5.2, comprising a combination of ribbon and licorice diagrams (Fig. 6.1(b)). We particularly evaluated four interaction transitions—two for translation (Fig. 6.5(c)) and rotation and scaling (Fig. 6.5(b))—and gathered their feedback in interviews. Our second session focused on our two selection techniques, BrushWYP and BrushLasso (Sec. 6.5.1). Following these two sessions, we asked them to fill in a questionnaire to assess and compare accuracy, efficacy, and user experience of both selection and manipulation techniques using 7-point Likert scales. In addition, throughout the study we encouraged them to share any feedback they may have.

The experts showed high engagement with our environment and its interactivity. They actively compared the visual effects and differences between the two visual representations across both spaces. All three experts expressed that they enjoyed the feeling of interacting with data seamlessly throughout the environment, appreciating the intuitive nature of the interactions without experiencing any confusion about how interactions should be performed. The experts particularly focused on the approaches for transferring data between the two displays. In their comparison of the two translation techniques, they mentioned that both methods provided a natural and fluid movement of data across spaces. They felt, however, the 2D pinch + 3D pinch gesture to be more precise than the 2D pointing + 3D pinch one. One reason was that the 2D pinch gave them precise control of data depth—allowing them to directly correlate the distance between their fingers with depth adjustment of the data below the surface. Depth control with a single finger-pointing gesture, in contrast, was challenging as it depended on visual feedback from below the 2D surface. An interesting observation was that all three experts did not realize that the familiar 2D pinch gesture had been repurposed from scaling to facilitate translation from **Below** to **Above** the surface. They seamlessly adapted to using an integrated 2D pinch + 3D pinch gesture to "grab" data across spaces. Moreover, while finding all interaction designs intuitive and easy to remember, the experts suggested that additional icons would be helpful visual cues for interaction, especially for transitions between the two distinct spaces. All experts also appreciated the design of both selection methods. Two of them favored BrushWYP for its uniform brushing approach from start to finish, aligning with their expectation of how data selection should work—even when transitioning across two distinct spaces. The other expert felt that either method could be effective. In comparing the accuracy of both, all experts felt that both methods can support them in selecting the intended target data above and below the surface. Interestingly, we did not observe that using a lasso on the 2D screen

would enhance their perception of accuracy. This may be because their main challenge was not to achieve precise 2D input but rather the inability to track the data visualization below the surface, unlike in AR space. We solved this issue, however, with our target- and context-aware selection technique.

6.7 SpatialTouch and CR Visualization and Interaction

Based on the design considerations of SpatialTouch and other CR solutions, and their potential use in domain research, we can now revisit the initial questions we posed: (1) how to represent data to ensure users understand what they see and (2) how to design interaction techniques that allow users to seamlessly continue their tasks without the need to deliberate on how interactions translate between different display spaces. We experimented with a variety of data representation strategies to elucidate data attributes (e. g., ball-stick, surface, stripes). We also investigated different visualization placements, **Above** or **Below** the surface, employing various levels of detail and abstraction. We also explored the use of perspective views, orthographic views, or exclusively 2D cutting planes for presenting data content on 2D surfaces. One crucial lesson emerged from this work: it is not viable to impose strict limits on which visual representations should be displayed in each space. Instead, we need to treat CR environments as integrated systems, in which users can decide where data is displayed and how to interact with it. The focus of our design considerations thus shifted toward managing **transitions** effectively, breaking down into two essential facets: visualization transformation and interaction transition.

Visualization transformation involves the seamless migration of data representations between two distinct spaces, such as from a 2D display to 3D space. This process demands thoughtful design to ensure that the essence and clarity of the data are preserved during the transition to not break a user's mental model. Lee et al. [22] have thoroughly explored visualization transformation approaches that can be optimally designed to support visualization tasks. The transition of 3D spatial data from 2D to 3D is inherently facilitated by the data's existing 3D structure if perspective views are employed. The merging of 2D visualizations on flat surfaces with 3D visualizations in spatial environments to represent data features coherently, however, presents a considerable challenge: visual complexity. This complexity arises as users are presented with a blend of visual representations—2D or 3D, with varied visual elements—tailored to highlight distinct data features. For instance,

when data moves from a 2D display to 3D space and the visual representations shift from surfaces and lines to stripes and curves, there is a concern about whether users can maintain their situational awareness. This challenge of visual complexity is not unique but is inherent across many environment designs within CR settings, especially when incorporating both situated and embedded visualizations to enhance user understanding of data features. After discussions with domain experts, we identified that a promising approach is to enable users to adjust visual representations locally—irrespective of being in 2D, 3D, or intermediate space—based on their specific needs. The shallow-depth area across two spaces, in particular, may be an appropriate place for this kind of adjustment. With SpatialTouch, users can effortlessly switch between different representations to dynamically examine data features. This flexibility in adapting visual representations aligns with Schwajada et al.'s [238] findings, who highlight the benefits of user-controlled transformation in enhancing task performance efficiency by catering to the users' requirements.

Interaction transition. To the best of our knowledge, there has not been a thorough exploration of interaction transition within CR environments. We thus introduce the notion of interaction transition as *the process in which interactions extend across various levels of virtuality within a CR environment, ensuring all actions maintain coherence, thus creating a continuous and seamless interactive experience.* Several factors influence this process, including the tactile feedback upon touching a physical surface, constraints in interacting with both depth and 2D surfaces, established mental models guiding 2D and 3D interactions, and how visual perception of data features impacts interaction.

Central to our understanding is the goal to preserve a user's **spatial awareness** to facilitate natural interaction within CR. This goal is twofold: first, we need to ensure that users maintain spatial awareness of their surrounding space (whether 2D or 3D), allowing them to seamlessly continue tasks without needing to reconsider interaction modalities for different display spaces. When using *BrushLasso* (Fig. 6.6(b)) in CR, e. g., users intuitively brush data in 3D and then draw a circle to enclose target points on the 2D surface. Second, it is crucial that users maintain spatial awareness of their data (structure, orientation, position). This awareness guides them in understanding where and how to perform specific actions. Users can trace, for instance, the 3D structure of data with *BrushWYP* (Fig. 6.6(a)) and seamlessly continue their interaction by following the structure onto the 2D surface.

To support such seamless transitions, our techniques need to be context- and target-aware. While a user's intention may be clear in one space, it may become ambiguous when

transitioning to another. Considering the previous example, if a user is unable to trace a 3D point cloud on the 2D surface any technique needs to predict their selection intentions. This leads to another critical awareness we need to maintain: **situational awareness**—users should be able to predict the results of their actions in CR. Fig. 6.7(b) demonstrates this case—users use two pinch gestures to "grab" data from and to the surface. Their fingers naturally come into physical contact, promoting a continuous 3D pinch gesture to "grab" data into the 3D space. During this process, users have expected that the subsequent action would extract the data further. Another interaction technique, "lifting" (Fig. 6.8(a)), allows users to temporarily elevate the 3D volume for annotating a 2D slice, expecting that releasing the pinch gesture returns the data to its original position on the surface once the annotation is complete. These interaction designs preserve spatial and situational awareness, thereby ensuring a deep engagement in task completion in the CR environment. Thus, they were highly favored by AR/VR experts in our evaluation.

Limitations. Similar to all visualization environments, SpatialTouch has its intrinsic limitations. SpatialTouch has limitations in hosting spatial visualizations. First, the stereoscopic rendering and the 2D monoscopic rendering may interfere with each other, potentially affecting the users' understanding of the spatial data. In our case, we use the HoloLens optical see-through HMD (OST)—when the 2D screen is too bright, it can interfere with the comprehension of the 3D rendering. Similarly, if a video see-through HMD (VST) is used, the stereoscopic rendering may occlude the 2D monoscopic rendering. In addition, the different visualization luminance between 2D displays and AR HMDs may also lead to a sense of disconnection. Therefore, blending the two renderings to obtain a coherent visualization is a meaningful research question for future development. Various factors should be considered to ensure an optimal color blending method, such as the specific AR device property (OST or VST), brightness, and other rendering-related settings. Second, 2D rendering is limited by the size of the screen, whereas AR theoretically provides unlimited space for 3D rendering. As a result, parts of the data visualization in the environment may not appear complete. While domain experts tend to focus on the local region of the data during exploration, we did not encounter any issues in the interviews. However, future researchers may need to take this into consideration when large-size visualizations are used in cross-reality interfaces that include fixed-size screens. Finally, collaborative data exploration is currently limited due to the design of FishTank VR monoscopic rendering on the 2D surface. Although users from different viewing angles can have their own view of

the data in AR space, they share the same view rendered on the 2D surface. So only one user is able to see a cohesive 3D representation. A potential solution for the future would be to render several views on the screen at high frequency and use a technique similar to "shutter glasses," allowing each user to see their 2D rendering on the shared 2D surface.

SpatialTouch also has limitations caused by specific hardware settings (Surface Studio and Hololens HMD). First, SpatialTouch requires high precision in the registration between the 2D surface and the 3D space to ensure accurate alignment between the two views. In our prototype, the HoloLens and Surface Studio run the program separately as two clients, with visualization and interaction states synchronized via the server. Similar to past work [46], after starting the application on Hololens, we need to manually align the two coordinate systems. Thus, although the view synchronization algorithm presented in Sec. 6.3 works accurately, as shown in the supplemental demo, it was time-consuming and difficult to achieve a high level of precision in the coordinate system registration. For future development, we recommend the researchers leverage automatic registration techniques with high precision or develop targeted CR display ecology with a unique coordinate system. Second, the limited FOV and the restricted gesture recognition area of the HoloLens significantly hinder the user's immersive experience. The narrow FOV confines the visualization to a small window in the user's vision, which breaks the sense of immersion as users are constantly reminded of the boundaries of the display. The constrained gesture recognition area means that users need to perform interactions within a small space in front of the body. It increases the cognitive load of users as they need to be conscious of keeping their hands within the detectable zone. We recommend that researchers leverage HMDs or glasses with large FOVs and extensive gesture recognition areas. For the FOV, current mainstream video see-through HMDs, such as the Oculus Quest 3, can sample the real world and map the results onto display devices with a large FOV. However, the video quality of VST HMDs is not yet sufficient for users to clearly observe the data visualizations on 2D surfaces in the real world. We believe that in the future, the image quality of VST HMDs will continue to improve, and issues such as video stream latency and image distortion will be resolved. For gesture recognition, external sensors such as Leap Motion can be introduced to facilitate gesture recognition and tracking.

6.8 Conclusion

Cross-reality is thus more than simply merging various levels of virtuality, where data is either positioned at a single space or moved to another place. Instead, it stands as an innovative and integrated environment for data presentation and exploration. Holding this vision, data can appear in any form and at any corner of the environment, tailored to the specific needs of each domain. Through the interviews with domain experts, we learned that transitioning their data between 2D and 3D spaces significantly motivates them to view and analyze their data from previously unexplored angles. In this light, we shared design insights we gained through experimenting with various configurations for the CR environment, resulting in our recommendations to respect the users' mental models of the spaces in which they interact as well as of their data. Moreover, whether for understanding data or completing tasks, it is crucial to allow users to control data transformations, such as transforming data across spaces or changing representations. Yet, the most critical aspect to consider is why CR is the appropriate choice for the task and data. This rationale shapes all other design aspects: the choices of environment, visualization, and interaction designs.

Supplemental material pointers

We share our additional material at `osf.io/avxr9`. We also make our SpatialTouch simulator available at `github.com/LixiangZhao98/Cross-Reality-Environment-SpatialTouch` and the point cloud visualization and density estimation available at `github.com/LixiangZhao98/PointCloud-Visualization-Tool`.

APPENDIX 7

Discussion and Conclusion

Through the explorations presented in this thesis, we investigated visualization, interaction, and computational strategies across multiple immersive environments. Specifically, we examined how to design precise interaction techniques despite imprecise input (**Chapter 3**), how to achieve responsive performance when handling large-scale spatial data (**Chapter 4**), how to create layouts and transitions that support multiscale navigation (**Chapter 5**), and how cross-reality approaches can present 2D and 3D representations simultaneously while enabling seamless transitions between them (**Chapter 6**). Taken together, these explorations position us to address the research questions outlined at the beginning of this thesis. Building on these answers, we broaden the discussion to consider what the results reveal more generally about immersive visualization—highlighting key takeaways, design recommendations, and potential avenues for future research.

7.1 Research Questions

RQ1. How can immersive environments support precise, fine-grained spatial data exploration despite the inherent imprecision of 6DOF input modalities?

RQ1 addresses a fundamental tension in immersive visualization: although 6DOF input enables natural, flexible, and expressive mid-air interaction, it is intrinsically imprecise due to depth-judgment limitations, hand instability, and the absence of physical constraints. This raises a core question: How can immersive environments meaningfully support tasks that require fine-grained spatial accuracy?

We examined this question in two projects across distinct data domains. **Chapter 3** explored point-cloud selection in cosmological simulations—an unstructured, large-scale dataset with no clear boundaries and complex multiscale features. **Chapter 6** investigated precise annotation of medical volumetric data, where placing anatomical landmarks requires accurate localization within dense 3D structures. In both scenarios, users possess clear analytic intent (which region to select, which point to annotate), yet they cannot physically express this intent precisely using standard 6DOF controllers or touch inputs. Interestingly, users often do not perceive this imprecision themselves—immersive interaction feels natural, creating the illusion that approximate pointing is sufficient.

Across both studies, we find that immersive environments *can* support precise scientific operations, but only when interaction techniques explicitly account for both *user input behaviour*—lasso paths, gesture shapes, pointing trajectories; and *data characteristics*—density distributions, feature coherence, structural continuity, or boundary cues. User input provides approximate evidence of intent, while data context constrains the space of plausible interpretations by indicating which structures exist and how they are spatially organized. By combining these two sources of information, the system can infer a more complete model of user intention and generate precise results even when the raw physical input is imprecise.

Collectively, these studies demonstrate that *precise, fine-grained scientific exploration is achievable in immersive environments*. Precision does not come from the input device itself, but from the design of the interaction techniques that interpret, constrain, and refine that input. Effective methods introduce mechanisms such as:

1. inferring user intent from coarse input,
2. refining, constraining, or snapping interactions to meaningful data structures, and
3. supporting iterative refinement or threshold adjustment to confirm results.

RQ2. How can immersive environments maintain responsiveness when exploring computationally demanding spatial datasets?

Spatial datasets in scientific domains, such as cosmological simulations, medical volumes, and molecular structures, require continuous computation during exploration-filtering, density estimation, feature extraction, and view-dependent updates. In immersive environments, this demand becomes even more critical: users naturally walk around, change viewpoints, and interact fluidly with the data, which forces the system to update computations in real time. If these operations cannot be performed at interactive frame rates, latency

disrupts visual comfort and breaks users' sense of presence. RQ2 therefore asks whether—and how—immersive systems can remain responsive when handling large, high-resolution, and computationally intensive spatial datasets.

We explored this question in **Chapter 4** through the development of a fast GPU-based kernel density estimation framework designed for multiscale, billion-element point-cloud datasets. Traditional density-based methods struggle in immersive settings because they rely on precomputed density fields for fixed scales, making them imprecise when users zoom. In contrast, we use an adaptive, on-the-fly KDE estimator that dynamically recomputes the density field covering only users' current focus region with high resolution, leveraging GPU parallelism strategy. This adjustment of computation enables the system to maintain real-time responsiveness even as users navigate freely through massive datasets.

Taken together, these findings show that *immersive environments can remain responsive even when operating on computationally demanding spatial datasets*—but only when computation is restructured around incremental updates, multiscale efficiency, and GPU-accelerated pipelines rather than naïve full-field recomputation. Responsiveness depends not merely on raw processing speed, but on designing data structures and algorithms that minimize redundant work, decouple computation from viewpoint changes, and support progressive or partial updates. Effective strategies include:

1. precomputing scale-relevant information that can be reused across multiple levels of detail;
2. encoding multiscale structure so that fine-grained detail is accessed or computed only when required; and
3. leveraging massively parallel GPU processing to amortize expensive operations and sustain real-time interaction.

RQ3. How can immersive environments support effective navigation across multiple spatial scales in spatial datasets?

Along with the previous question, spatial datasets often exhibit rich multiscale structure: meaningful features appear at widely different spatial resolutions, and analysts frequently need to move fluidly between global context and fine-grained detail to form a coherent understanding. While immersive environments allow users to walk, lean, and reposition themselves naturally around the data, such embodied interaction does not solve the perceptual and cognitive challenges of navigating across vastly different spatial scales. As users transition across scales, they may lose awareness of how their current view relates to the

larger structure, making it difficult to maintain orientation or understand where local features lie within the global context. This challenge becomes even more pronounced in large-scale or collaborative exploration scenarios, where users must repeatedly re-locate themselves—and sometimes their collaborators—across multiple levels of scale. RQ3, therefore, asks how immersive systems can support seamless, interpretable, and orientation-preserving navigation across spatial scales.

We explored this question through a series of projects on multiscale cosmological point-cloud exploration, where features such as halos, subhalos, and filaments emerge only at different density scales. First, **Chapter 4** introduces a scale-free KDE framework that supports real-time density computation, adaptive refinement, and progressive updates. This enables smooth, continuous transitions across scales while preserving structural continuity during navigation. **Chapter 5** tackles multiscale navigation through spatial framing and hierarchical representation. Linked World-in-Miniature views let users inspect fine-scale features locally while maintaining a persistent global frame of reference. **Chapter 5** addresses cross-scale navigation using an *Overview and Detail* metaphor, enabling users to manipulate detailed views while grounding their actions within a shared global overview.

Together, these studies demonstrate that *immersive environments can support effective multiscale navigation*, provided that visualization and interaction techniques explicitly preserve context and orientation across scales. Effective approaches incorporate:

1. Continuous scale transitions that reveal fine detail without breaking the user's understanding of the larger structure;
2. Context-preserving layouts that maintain global reference frames and prevent spatial disorientation; and
3. Interaction techniques that allow users to fluidly move between scales without losing their mental model or sense of place.

RQ4. How can immersive environments support the effective presentation and integration of spatial datasets that rely on multiple 2D and 3D visual representations?

If the previous three research questions examine whether immersive environments can overcome key challenges—precision of interaction (RQ1), computational responsiveness (RQ2), and multiscale navigation (RQ3)—then RQ4 moves one step further. It asks not only whether immersive environments can *handle* these challenges, but whether they can provide *unique advantages* for scientific analysis. A central motivation for this question comes from a defining property shared across many scientific domains: meaningful insight often

requires switching between, or jointly interpreting, heterogeneous visual representations. Traditional desktop systems approximate this need through multiple coordinated views, but these are spatially constrained, visually fragmented, and cognitively demanding. Immersive environments, by contrast, offer multiple levels of immersion and spatial presentation that may fundamentally reshape how heterogeneous representations are integrated.

We explored this question in **Chapter 6** through a cross-reality environment that unifies a monoscopic 2D touchscreen with a stereoscopic 3D AR workspace. This system is used to explore three spatial datasets—molecular structures, cosmological point clouds, and anatomical CT data—each of which naturally requires both 2D and 3D visualization. Molecular data benefit from 3D spatial understanding of folds and interactions while relying on 2D diagrams for functional or sequence-based abstractions. Cosmological point clouds require 3D perception of spatial distribution yet depend on 2D projections or slices for precise quantitative inspection. Anatomical CT volumes integrate 3D context with 2D slice-based measurement for clinical interpretation. Across all three domains, analysts must fluidly switch between these modalities or use them in conjunction to construct a coherent understanding. Through our domain expert interviews, we found that the cross-reality environment can support this multi-representation reasoning effectively when 2D and 3D views are spatially co-located, embodied, and cognitively aligned. Users consistently benefited from being able to arrange, align, and cross-reference heterogeneous representations within a shared immersive workspace. Experts reported that this integrated setup allowed them to maintain global spatial understanding in 3D while performing precise, detail-oriented operations in 2D. Importantly, users preferred workflows where 2D and 3D representations complemented, rather than replaced, one another.

These findings indicate that *immersive environments have strong potential as a powerful medium for scientific analysis*—especially for tasks that require integrated 2D/3D perspectives. However, this potential can be realized only if visualization and interaction designs preserve mental continuity and cognitive coherence across representations. Several principles emerge: immersive systems can effectively support multi-representation scientific reasoning when they

1. preserve stable and meaningful spatial relationships between 2D and 3D views;
2. enable embodied comparison by arranging representations around the user;
3. support seamless transitions that avoid disrupting users' mental models; and
4. provide fluid interactions that operate consistently across the entire CR environment.

Taken all four research questions together, our findings indicate that immersive environments can not only *accommodate* the core challenges—precision, responsiveness, multiscale navigation, and multi-representation integration—but can also *introduce new opportunities* for richer, more integrated scientific reasoning. These insights allow us to answer our central question:

How should immersive environments be designed to support spatial data exploration effectively and efficiently?

Our work demonstrates that immersive environments can effectively support scientific spatial data exploration when visualization, interaction, and computational strategies are carefully designed to leverage their unique affordances while mitigating inherent limitations.

7.2 Key Takeaways and the Path Forward

Drawing this thread forward, we now summarize the key takeaways that emerge from our work and outline directions for future research in immersive and CR visualization systems. To contextualize these discussions, we refer to Fig. 7.1, which illustrates a range of existing CR designs representing the broader solution space.

7.2.1 Motivation: Why Combine Multiple Spaces?

When designing a CR environment for spatial data, the foundational question is why multiple spaces should be combined at all. The motivation shapes every subsequent decision—representation, interaction, spatial alignment, and collaboration models.

Augmenting Representations for Understanding. Immersive and CR environments allow designers to combine heterogeneous representations to reveal complementary facets of the data. The CR designs illustrated in Fig. 7.1 all demonstrate this capability. Different types of information—additional data dimensions, contextual cues, or regions of interest—often require distinct visual forms and levels of abstraction, and CR environments should be designed to accommodate these needs. Examples include AR paired with a traditional PC-based analysis interface for 3D data [262] (Fig. 7.1(a)), overview and detail techniques for navigating volumetric data [2] (Fig. 7.1(d)), embedded AR visualizations for multivariate analysis [233] (Fig. 7.1(f)), and our SpatialTouch system (**Chapter 6**), which integrates 3D spatial visualization with multiple 2D slice-based representations (Fig. 6.8(b)).

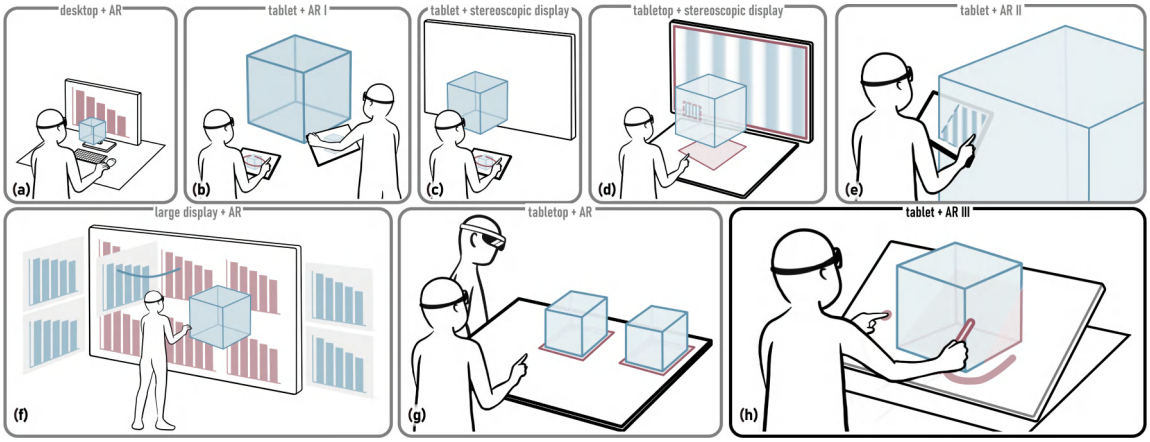


Figure 7.1: Existing CR environments for visualization: (a) monitor + AR, (b) tablet + AR, (c) tablet + Stereoscopic display, (d) tablet + stereoscopic display, (e) tablet + AR, (f) large display + AR, (g) tablet + AR, (h) tablet + AR, SpatialTouch.

Design Takeaway: Identify focal regions, place each representation in the space where it supports reasoning best, and structure surrounding views to preserve continuity across representations.

Providing Interaction Spaces that Support Precision. CR environments introduce physically grounded interaction spaces that can substantially improve precision. As users better perceive spatial relationships and identify regions of interest, their actions become more deliberate and targeted. For example, users can accurately place a slice view inside a medical volume on a touch table while simultaneously maintaining awareness of the global anatomy (Fig. 7.1(d)). In **Chapter 6** (SpatialTouch), users first reason about the 3D structure of a point cloud in AR, then perform precise lasso operations on the stable 2D surface beneath it (Fig. 7.1(h)). By separating where coarse spatial reasoning occurs (in 3D space) from where fine-grained, stability-dependent operations occur (on 2D surfaces), CR environments help mitigate the inherent imprecision of mid-air input. A natural design strategy is therefore to assign rough exploration and spatial understanding to 3D space, while offloading precise, detail-oriented actions to grounded 2D interaction surfaces. However, in practice, users often interact wherever they find something interesting—whether in 2D or 3D—so designers must assume that both spaces may be used for both types of actions. This highlights the need to create interaction techniques that behave coherently across spaces and preserve users' mental models regardless of where the interaction is initiated.

Design Takeaway: Assign coarse, spatial reasoning to 3D space and reserve fine-

grained, stability-dependent actions for grounded 2D surfaces; CR environments should guide—but not restrict—users toward the interaction space best suited for their task.

Supporting Collaboration. A significant advantage of CR environments is their ability to balance shared and personal spaces for collaborative scientific analysis. For instance, users in a shared AR environment can manipulate the same 3D visualization while maintaining personal tablet views for private exploration [35] (Fig. 7.1(b)). Other systems offer shared large-display content augmented with private AR overlays [31, 233] (Fig. 7.1(f, g)). Some integrate personal tablets directly into AR space with precise spatial registration (Fig. 7.1(e)).

Design Takeaway: Balance shared global views with personal analysis spaces, enabling collaborators to fluidly transition between collective sense-making and individual deep exploration.

Leveraging Multiple Levels of Abstraction. CR environments naturally support the coordination of multiple levels of abstraction. **Chapter 6** (SpatialTouch) demonstrates the value of integrating multiple abstraction layers—raw spatial data, intermediate abstractions such as slices or projections, and higher-level derived representations—across both the 2D surface and the surrounding AR space. CR designs such as those in Fig. 7.1(e, h) show how tightly coupling these abstraction layers enables users to relate structural, functional, and contextual aspects of the data more effectively. By placing abstractions in close spatial relation to the 3D content, users can fluidly shift between global understanding, focused inspection, and interpretive analysis.

Design Takeaway: Design abstraction layers as an integrated ecosystem—spatially linked, visually coherent, and cognitively continuous—so users can seamlessly move between raw data, intermediate views, and high-level interpretations.

7.2.2 Environment & Dataset Considerations: Where and What to Visualize?

Effective immersive visualization requires aligning the environmental configuration with the characteristics of the dataset being explored. A typical CR system combines monoscopic displays (e. g., tablets, touch surfaces, desktops) with stereoscopic AR/VR views (Fig. 7.1). These spaces serve complementary roles: 2D displays provide stability and precision, while 3D environments reveal spatial structure and support embodied reasoning. For certain

analytical workflows, CR environments may also be constructed entirely from VR and AR spaces, offering flexible combinations of immersion levels tailored to specific tasks.

Designing Coherent Connections Between Spaces. A central challenge lies in creating meaningful, coherent connections between these heterogeneous spaces. Existing systems (Fig. 7.1(a–d)) often rely on coordinated visual cues (e. g., color, highlighting), linked views, or animated transitions to help users relate information across displays. More recent designs (Fig. 7.1(e–h)) emphasize precise **spatial alignment** between AR and display content, bringing representations into a shared physical frame of reference. For instance, Reipschläger and Dachselt [46] previously defined three levels of spatial proximity: AR content positioned in front of or behind the display, AR content arranged close to or at the edge of the display, and AR content rendered with no spatial relation to the display. Analyzing these setups, Fig. 7.1(f, g) use the display as a frame of reference for AR content placement. Designs in Fig. 7.1(e, h)—including SpatialTouch—leverage the spatial nature of 3D data that is rendered in AR space and position the 2D display inside the AR visualization. This approach is particularly effective for demonstrating the relationship between contents in two different spaces, as demonstrated in spatial selection tasks [35] and the visualization of medical volumes using cutting planes in SpatialTouch.

Design Takeaway: Use spatial alignment to unify 2D and 3D representations whenever possible; alignment reduces cognitive load and strengthens users' mental models across spaces.

Transition Areas: Designing the "In-Between" Space. The most challenging—yet most promising—region in any CR environment is the transition area between two distinct spaces, such as the boundary between the AR volume and the touch screen (as seen in Fig. 7.1) or the boundary between AR and VR spaces. These regions are where shifts naturally occur: **shifts in abstraction** (from raw 3D structure to sliced or projected views), **shifts in representation** (from 3D stereoscopic depth to 2D precision), and **shifts in interaction modality** (from mid-air gestures to grounded touch input). Because so many changes converge here, transition areas can easily confuse or disorient users if not carefully designed. Yet, when used effectively, they can guide users smoothly between tasks, help them shift analytical focus, and maintain continuity across heterogeneous representations. The design goal is not to force users into one mode or the other, but to provide gentle orientation cues and consistent spatial relationships that help them understand how one interaction space leads into the next.

Design Takeaway: Treat the transition area as an intentional design space: use alignment, visual continuity, and interaction cues to guide users smoothly between 2D and 3D representations, preserving orientation and analytical flow.

7.3 Summary

Visualization helps the domain experts to more effectively understand the data. By supporting iterative cycles of data representation, interactive exploration, and decision-making, visualization enables users to gradually develop insights into their data. In recent years, immersive visualization—delivered through VR, AR, and MR environments and technologies—has become a major focus within the visualization research community due to its potential for 3D spatial data analysis. Immersive environments provide stereoscopic rendering, enabling users to examine 3D structures from arbitrary viewpoints, walk around and inspect them up close to understand spatial data features. Immersive environments also support embodied interaction, allowing users to directly interact with the data visualization through natural actions using 6DOF hand-held input devices or mid-air gestures. Despite these advantages, their actual benefits and limitations in spatial data exploration remain insufficiently examined.

In this thesis, we focused on answering a central question: How should immersive environments be designed to support spatial data exploration effectively and efficiently? We focus on the visualization of three types of spatial datasets—cosmological simulations of galaxy dynamics, volumetric medical imaging data (CT/MRI), and biological molecular structures—chosen for their complexity, large scale, and multi-scale characteristics, which we described in Chapter 2. These properties pose fundamental challenges for immersive environments, affecting users' ability to understand and explore data visualizations across scales, increasing the demand for precise interaction techniques, and highlighting the importance of high-performance computing techniques.

For the first part, we focus on the precise interaction technique. Despite the intuitiveness of 6DOF input, the inherent imprecision of the input due to the lack of physical support prompted us to investigate whether immersive environments can truly support precise, fine-grained interactions for spatial data exploration. We focused on selection tasks on the cosmological simulation dataset, as the task demands a precise specification of regions of interest of the complex target structures. In Chapter 3, we introduce the data selection

technique for scalar-field-based data with 6DOF natural stroke and point input, called MeTACAST (target- and context-aware selection technique). Selection is essential for spatial data exploration, as it enables users to isolate regions of interest from complex 3D datasets before further analysis and follow-up interactions. The challenge lies in interpreting inherently imprecise mid-air input and translating users' approximate gestures into selection volumes that accurately reflect their intent. Our techniques support the selection of ROIs through simple actions—such as pointing, dragging, or drawing strokes—while determining the final selection volume by integrating these inputs with structural properties of the data, including density information. Our selection technique exhibits strong generality. Although its effectiveness is demonstrated on selecting high-density regions in cosmological datasets in this work, it can be readily extended to any scalar-field–based data selection scenario, provided that an appropriate target selection rule is defined.

For the second part, we focus on high-performance computing techniques. A key fact is that our selection techniques (Chapter 3) rely on a precomputed underlying data representation—the scalar field—because computing this field is computationally expensive for a large resolution. However, as users navigate to increasingly fine-grained scales, the precomputed field becomes insufficient to represent small-scale structures. To faithfully capture these finer details while maintaining interactivity, the scalar field must be (re)computed at higher resolutions in real time. In Chapter 4, we introduce a GPU-accelerated method for fast field reconstruction that recomputes the scalar field at higher sampling resolutions—at a speed imperceptible to the user—whenever they shift their focus to finer scales, ensuring that small-scale structural details remain accurately represented. This capability allows users—true to the name "ScaleFree"—to freely query field values at any scale in real time, thereby enabling further fine-grained interactions. The core value of ScaleFree lies in its fast point-to-field computation capability, which enables real-time reconstruction of detailed field representations for any local region during large-scale data exploration, thereby supporting not only selection but a broad range of field-based interaction techniques in real-time interactive exploration.

For the third part, we focus on the multi-scale visualization and navigation techniques. When exploring multi-scale datasets, a single view (Chapter 3 and Chapter 4) is often insufficient to convey both local details and global structure. As a result, users struggle to understand the hierarchical relationships across scales, become disoriented about their position within the data, and are unsure how to navigate toward relevant regions. In Chapter 5,

we introduce two layout strategies—a WiM-based approach and an overview-and-detail approach—that support multi-scale, collaborative exploration in immersive environments. These layouts maintain users' awareness of position, orientation, and scale throughout transitions across multiple orders of magnitude. Combined with interactive tools that allow users to perform any exploration or analysis operation, these techniques are well-suited for collaborative data analysis, enabling not only contextual understanding but also the execution of complex, detail-oriented tasks. L-WiM interface is highly extensible and compatible with a wide range of collaborative tasks, enabling various interaction and computation techniques to be embedded seamlessly. For example, the MeTACAST technique introduced in Chapter 3 can be applied directly on the handheld L-WiM, allowing users to select data within a collaborator's world through the copied miniature model. Similarly, the progressive navigation technique described in Chapter 4 can also be applied directly to the L-WiM to support scale transitions and target-driven navigation. It is important to note, however, that both selection and navigation rely fundamentally on the fast field-computation capability introduced in Chapter 4. Whenever a collaborator changes their position, this capability ensures that the density field enclosed by the WiM is updated rapidly and accurately, enabling responsive and reliable interaction across environments. Together, techniques such as fast field estimation, target- and context-aware selection, and progressive navigation—developed in earlier chapters—can be readily extended and applied within the L-WiM interface, forming a cohesive and powerful tool for collaborative spatial data exploration.

For the fourth part, we investigate how 2D and 3D representations can be presented and combined within immersive environments. In many scientific workflows, 2D representations remain essential for data analysis, even when the data is explored in VR or AR. Although immersive systems can host 2D visualizations, traditional 2D displays continue to be the most familiar, native, and effective medium for presenting 2D content in everyday practice. In Chapter 6, we design a hybrid display environment that integrates a touch-enabled 2D display with an AR workspace. Before developing specific visualization and interaction techniques, we conducted a preliminary study to understand what users expect from the two environments and how they prefer to interact with spatial data in such a environment. We found that preserving a accurate understanding of 3D structure is crucial for spatial data visualization. That is why we developed a technique and environment that seamlessly aligns the rendering on the 2D display with the stereoscopic rendering in 3D space. Interestingly, in such an environment users naturally perceived the integrated system as a single unified

environment and showed a strong willingness to interact across the two spaces seamlessly. These observations provide valuable guidance for designing visualization and interaction techniques for spatial data exploration in cross-reality environments.

Finally, in Chapter 7, we return to the four research questions and address each individually, outlining the specific contributions our work makes toward resolving them. These questions capture fundamental challenges in visualization, interaction, and computing that arise when exploring large-scale spatial data in immersive environments. Our investigations thus establish the foundation needed to answer the central question of this thesis. Moreover, the cross-reality techniques explored here demonstrate a concrete path toward bridging immersion levels in spatial data exploration and reveal the significant potential of cross-reality environments for immersive data analysis. While this work marks only an initial step, much remains ahead. Future techniques and systems must be tailored to the requirements of diverse tasks and data characteristics, with thoughtful integration of visualization, interaction, and computing.

Bibliography

[1] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014. doi: 10/gd3xgq

[2] D. Coffey, N. Malbraaten, T. B. Le, I. Borazjani, F. Sotiropoulos, A. G. Erdman, and D. F. Keefe. Interactive Slice WIM: Navigating and interrogating volume data sets using a multisurface, multitouch VR interface. *IEEE Trans Vis Comput Graph*, 18(10):1614–1626, 2012. doi: 10/fcds9g

[3] A. Fonnet and Y. Prié. Survey of immersive analytics. *IEEE Trans Vis Comput Graph*, 27(3):2101–2122, 2021. doi: 10/gf73gm

[4] R. Fraedrich, J. Schneider, and R. Westermann. Exploring the millennium run-scalable rendering of large-scale cosmological datasets. *IEEE Trans Vis Comput Graph*, 15(6):1251–1258, 2009. doi: 10/cbhr2f

[5] J. Bond, L. Kofman, and D. Pogosyan. How filaments of galaxies are woven into the cosmic web. *Nature*, 380(6575):603–606, 1996. doi: 10/fq7bzc

[6] M. Ruiz, A. Bardera, I. Boada, I. Viola, M. Feixas, and M. Sbert. Automatic transfer functions based on informational divergence. *IEEE Trans Vis Comput Graph*, 17(12):1932–1941, 2011. doi: 10/fw7dnc

[7] M. Levoy. Display of surfaces from volume data. *IEEE Comput Graph Appl*, 8(3):29–37, 1988. doi: 10/bsn9df

[8] N. Max. Optical models for direct volume rendering. *IEEE Trans Vis Comput Graph*, 1(2):99–108, June 1995. doi: 10/fm2mw5

[9] G. Kindlmann and J. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proc. Volume Vis*, pp. 79–86. IEEE Comp. Soc., Los Alamitos, 1998. doi: 10/cw5jv8

[10] S. Halladjian, H. Miao, D. Kouřil, M. E. Gröller, I. Viola, and T. Isenberg. Scale Trotter: Illustrative visual travels across negative scales. *IEEE Trans Vis Comput Graph*, 26(1):654–664, 2020. doi: 10/kt3k

[11] D. S. Goodsell. Visual methods from atoms to cells. *Structure*, 13(3):347–354, 2005. doi: 10/ds2x9n

[12] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. In *Proc. Telemanipulator and Telepresence Technologies*, pp. 282–292. SPIE, Bellingham, 1995. doi: 10/dh8jnv

[13] L. Yu, J. Ouwerling, P. Svetachov, F. H. J. van Hoesel, P. M. A. van Ooijen, and J. Kosinka. VeLight: A 3D virtual reality tool for CT-based anatomy teaching and training. *J Vis*, 25(2):293–306, 2022. doi: 10/mpq7

[14] F. Lu, V. Nanjappan, P. Parsons, L. Yu, and H.-N. Liang. Effect of display platforms on spatial knowledge acquisition and engagement: An evaluation with 3D geometry visualizations. *J Vis*, 26(3):667–686, 2023. doi: 10/grd3tz

[15] R. Alharbi, O. Strnad, L. R. Luidolt, M. Waldner, D. Kouřil, C. Bohak, T. Klein, E. Gröller, and I. Viola. Nanotilus: Generator of immersive guided-tours in crowded 3D environments. *IEEE Trans Vis Comput Graph*, 29(3):1860–1875, 2023. doi: 10/gtnn26

[16] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proc. CHI*, 8 pages, pp. 265–272. ACM, New York, 1995. doi: 10.1145/223904.223938

[17] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey. Point & teleport locomotion technique for virtual reality. In *Proc. CHI PLAY*, pp. 205–216. ACM, New York, 2016. doi: 10/ggcj62

[18] Y. Yang, M. Cordeil, J. Beyer, T. Dwyer, K. Marriott, and H. Pfister. Embodied navigation in immersive abstract data visualization: Is overview+detail or zooming better for 3D scatterplots? *IEEE Trans Vis Comput Graph*, 27(2):1214–1224, 2021. doi: 10/ghgt52

[19] D. Kuťák, M. N. Selzer, J. Byška, M. L. Ganuza, I. Barišić, B. Kozlíková, and H. Miao. Vivern a virtual environment for multiscale visualization and modeling of dna nanostructures. *IEEE Trans Vis Comput Graph*, 28(12):4825–4838, 2022. doi: 10/mpwt

[20] H. Mohammed, A. K. Al-Awami, J. Beyer, C. Cali, P. Magistretti, H. Pfister, and M. Hadwiger. Abstractocyte: A visual tool for exploring nanoscale astroglial cells. *IEEE Trans Vis Comput Graph*, 24(1):853–861, 2018. doi: 10/gcqk6d

[21] R. C. Mota, A. Rocha, J. D. Silva, U. Alim, and E. Sharlin. 3de interactive lenses for visualization in virtual environments. In *Proc. SciVis*, pp. 21–25. IEEE Comp. Soc., Los Alamitos, 2018. doi: 10/hb6rjr

[22] B. Lee, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer. A design space for data visualisation transformations between 2D and 3D in mixed-reality environments. In *Proc. CHI*, article no. 25, 14 pages. ACM, New York, 2022. doi: 10/kvj4

[23] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisco, 1999. urn:oclc :record:1036805840.

[24] X. Chen, J. Z. Self, L. House, J. Wenskovitch, M. Sun, N. Wycoff, J. R. Evia, S. Leman, and C. North. Be the data: Embodied visual analytics. *IEEE Trans Learn Technol*, 11(1):81–95, 2018. doi: 10/gdc3q2

[25] J. Jankowski and M. Hachet. Advances in interaction with 3d environments. 34(1):152–190, 2015. doi: 10/f649pp

[26] K. Joseph, O. Kwon, K. Ma, and P. Eades. A study of mental maps in immersive network visualization. pp. 1–10. IEEE Comp. Soc., Tianjin, 2020. doi: 10/gjbr4c

[27] L. Besançon, A. Ynnerman, D. F. Keefe, L. Yu, and T. Isenberg. The state of the art of spatial interfaces for 3D visualization. *Comput Graph Forum*, 40(1):293–326, 2021. doi: 10/gjbpxp

[28] S. Bruckner, T. Isenberg, T. Ropinski, and A. Wiebel. A model of spatial directness in interactive visualization. *IEEE Trans Vis Comput Graph*, 25(8):2514–2528, 2018. doi: 10/gjbdnn

[29] L. Yu, P. Svetachov, P. Isenberg, M. H. Everts, and T. Isenberg. FI3D: Direct-touch interaction for the exploration of 3D scientific visualization spaces. *IEEE Trans Vis Comput Graph*, 16(6):1613–1622, 2010. doi: 10/fc2df8

[30] R. Li, T. Huang, H. Liang, B. Han, X. Zhang, and H. Liao. 3d volume visualization and screen-based interaction with dynamic ray casting on autostereoscopic display. In *Proc. ISMAR-Adjunct*, pp. 240–245. IEEE Comp. Soc., Bari, 2021. doi: 10/hb6rjt

[31] B. Ens, S. Goodwin, A. Prouzeau, F. Anderson, F. Y. Wang, S. Gratzl, Z. Lucarelli, B. Moyle, J. Smiley, and T. Dwyer. Uplift: A tangible and immersive tabletop system for casual collaborative visual analytics. *IEEE Trans Vis Comput Graph*, 27(2):1193–1203, 2021. doi: 10/ghgt5x

[32] H. Ishii and B. Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proc. CHI*, pp. 234–241. ACM, New York, 1997. doi: 10/chjrkd

[33] M. Cordeil, B. Bach, A. Cunningham, B. Montoya, R. T. Smith, B. H. Thomas, and T. Dwyer. Embodied axes: Tangible, actuated interaction for 3D augmented reality data spaces. In *Proc. CHI*, pp. 1–12. ACM, Honolulu, 2020. doi: 10/grv5tk

[34] B. Jackson, T. Y. Lau, D. Schroeder, K. C. Toussaint, and D. F. Keefe. A lightweight tangible 3D interface for interactive visualization of thin fiber structures. *IEEE Trans Vis Comput Graph*, 19(12):2802–2809, 2013. doi: 10/gh38r2

[35] M. Sereno, S. Gosset, L. Besançon, and T. Isenberg. Hybrid touch/tangible spatial selection in augmented reality. *Comput Graph Forum*, 41(3):403–415, 2022. doi: 10/gqq53j

[36] L. Besançon, P. Issartel, M. Ammi, and T. Isenberg. Mouse, tactile, and tangible input for 3d manipulation. In *Proc. CHI*, pp. 4727–4740. ACM, New York, 2017. doi: 10/gpmzm8

[37] T. Rau, T. Isenberg, A. Koehn, M. Sedlmair, and B. Lee. Traversing dual realities: Investigating techniques for transitioning 3d objects between desktop and augmented reality environments. In *Proc. CHI*, pp. 1–16. ACM, New York, 2025. doi: 10/pj4f

[38] S. Frees, G. D. Kessler, and E. Kay. Prism interaction for enhancing control in immersive virtual environments. *ACM Trans Comput Human Interac*, 14(1):2–es, 2007. doi: 10/bp8wmq

[39] N. Osawa. Two-handed and one-handed techniques for precise and efficient manipulation in immersive virtual environments. In *Proc. Advances in Visual Computing*, pp. 987–997. Springer, 2008. doi: 10/d78382

[40] K. Satriadi, A. Cunningham, B. Thomas, A. Drogemuller, A. Odi, N. Patel, C. Aston, and R. Smith. Augmented scale models: Presenting multivariate data around physical scale models in augmented reality. In *Proc. ISMAR*, pp. 54–63. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/gtnn2p

[41] L. Besançon, M. Sereno, L. Yu, M. Ammi, and T. Isenberg. Hybrid touch/tangible spatial 3d data selection. In *Comput Graph Forum*, vol. 38, pp. 553–567. Wiley Online Library, 2019. doi: 10/gjbgt8

[42] O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz. Interactions in the air: adding further depth to interactive tabletops. In *Proc. UIST*, pp. 139–148. ACM, New York, 2009. doi: 10/bmpntq

[43] T. McDonald, W. Usher, N. Morrical, A. Gyulassy, S. Petruzza, F. Federer, A. Angelucci, and V. Pascucci. Improving the usability of virtual reality neuron tracing with topological elements. *IEEE Trans Vis Comput Graph*, 27(2):744–754, 2021. doi: 10/pfkh

[44] R. Arora and K. Singh. Mid-air drawing of curves on 3d surfaces in virtual reality. *ACM Trans Graph*, 40(3):1–17, 2021. doi: 10/gskhs4

[45] H. Yao, L. Zhao, H.-N. Liang, Y. Liu, Y. Li, and L. Yu. Exploring embodied asymmetric two-handed interactions for immersive data exploration. In *Proc. CHI EA*, article no. 140, 10 pages. ACM, New York, 2024. doi: 10/g899s6

[46] P. Reipschläger and R. Dachselt. DesignAR: Immersive 3D-modeling combining augmented reality with interactive displays. In *Proc. ISS*, pp. 29–41. ACM, New York, 2019. doi: 10/gtdktf

[47] G. J. Wills. Selection: 524,288 ways to say “this is interesting”. In *Proc. InfoVis*, pp. 54–60. IEEE Comp. Soc., Los Alamitos, 1996. doi: 10/bjq6dh

[48] F. Argelaguet and C. Andujar. Efficient 3D pointing selection in cluttered virtual environments. *IEEE Comput Graph Appl*, 29(6):34–43, 2009. doi: 10/fws5k7

[49] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3D immersive environments. In *Proc. I3D*, pp. 39–43. ACM, New York, 1997. doi: 10/fmwqt6

[50] S. Lee, J. Seo, G. J. Kim, and C.-M. Park. Evaluation of pointing techniques for ray casting selection in virtual environments. In *Proc. SPIE*, vol. 4756, pp. 38–44. SPIE, Bellingham, 2003. doi: 10/bj28fq

[51] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. In *Proc. UIST*, 10 pages, pp. 3–12. ACM, New York, 2006. doi: 10/b9g33c

[52] W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A novel interface for interactive exploration of DTI fibers. *IEEE Trans Vis Comput Graph*, 15(6):1433–1440, 2009. doi: 10/c43p7m

[53] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Tech-note: Dynamic dragging for input of 3D trajectories. In *Proc. 3DUI*, pp. 51–54. IEEE Comp. Soc., Los Alamitos, 2008. doi: 10/dh9sfm

[54] S. Owada, F. Nielsen, and T. Igarashi. Volume catcher. In *Proc. I3D*, pp. 111–116. ACM, New York, 2005. doi: 10/d36dg8

[55] P. Xu, H. Fu, O. K.-C. Au, and C.-L. Tai. Lazy selection: A scribble-based tool for smart shape elements selection. *ACM Trans Graph*, 31(6):142:1–142:9, 2012. doi: 10/hb6rj3

[56] L. Besançon, M. Sereno, L. Yu, M. Ammi, and T. Isenberg. Hybrid touch/tangible spatial 3D data selection. *Comput Graph Forum*, 38(3):553–567, 2019. doi: 10/gjbgt8

[57] Z. Chen, W. Zeng, Z. Yang, L. Yu, C.-W. Fu, and H. Qu. LassoNet: Deep lasso-selection of 3D point clouds. *IEEE Trans Vis Comput Graph*, 26(1):195–204, 2020. doi: 10/gmghvt

[58] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. Efficient structure-aware selection techniques for 3D point cloud visualizations with 2DOF input. *IEEE Trans Vis Comput Graph*, 18(12):2245–2254, 2012. doi: 10/f4fv9z

[59] L. Yu, K. Efstathiou, P. Isenberg, and T. Isenberg. CAST: Effective and efficient user interaction for context-aware selection in 3D particle clouds. *IEEE Trans Vis Comput Graph*, 22(1):886–895, 2016. doi: 10/kt5n

[60] D. E. Shaw, J. Grossman, J. A. Bank, B. Batson, J. A. Butts, J. C. Chao, M. M. Deneroff, R. O. Dror, A. Even, C. H. Fenton, et al. Anton 2: raising the bar for performance and programmability in a special-purpose molecular dynamics supercomputer. In *Proc. High Performance Computing, Networking, Storage and Analysis*, pp. 41–53. IEEE, 2014. doi: 10/gfz7d3

[61] J. W. Wadsley, B. W. Keller, and T. R. Quinn. Gasoline2: a modern smoothed particle hydrodynamics code. *Monthly Notices of the Royal Astronomical Society*, 471(2):2357–2369, 2017. doi: 10/gbwzvm

[62] O. Fuhrer, T. Chadha, T. Hoefler, G. Kwasniewski, X. Lapillonne, D. Leutwyler, D. Lüthi, C. Osuna, C. Schär, T. C. Schulthess, et al. Near-global climate simulation at 1 km resolution: establishing a performance baseline on 4888 gpus with cosmo 5.0. *Geoscientific Model Development*, 11(4):1665–1681, 2018. doi: 10/hb6rjn

[63] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Proc. NeurIPS*, vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020. doi: 10/gpmv43

[64] D. D. Bock, W.-C. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid. Network anatomy and in vivo physiology of visual cortical neurons. *Nature*, 471(7337):177–182, 2011. doi: 10/b9bpfk

[65] A. Chrysostomou, C. Taljaard, R. Bolton, L. Ball, S. Breen, and A. van Zyl. Operating the square kilometre array: the world's most data intensive telescope. In *Proc. Observatory Operations: Strategies, Processes, and Systems*, vol. 11449, pp. 156–170. SPIE, 2020. doi: 10/hb6rjs

[66] W. Usher, P. Klacansky, F. Federer, P.-T. Bremer, A. Knoll, J. Yarch, A. Angelucci, and V. Pascucci. A virtual reality visualization tool for neuron tracing. *IEEE Trans Vis Comput Graph*, 24(1):994–1003, 2018. doi: 10/gcqcfr

[67] C. Hurter, N. H. Riche, S. M. Drucker, M. Cordeil, R. Alligier, and R. Vuillemot. FiberClay: Sculpting three dimensional trajectories to reveal structural insights. *IEEE Trans Vis Comput Graph*, 25(1):704–714, 2019. doi: 10/kvhw

[68] Y. Jiang, C. Yu, T. Xie, X. Li, Y. Feng, H. Wang, M. Li, H. Lau, F. Gao, Y. Yang, et al. Vr-gs: A physical dynamics-aware interactive gaussian splatting system in virtual reality. In *Proc. Siggraph*, pp. 1–1. ACM, New York, 2024. doi: 10/hbwcmk

[69] Z. Luo, Z. Cui, S. Luo, M. Chu, and M. Li. Vr-doh: Hands-on 3d modeling in virtual reality. *ACM Trans Graph*, 44(4):1–12, 2025. doi: 10/hb6rjq

[70] N. Deng, Z. He, J. Ye, B. Duinkharjav, P. Chakravarthula, X. Yang, and Q. Sun. Fov-nerf: Foveated neural radiance fields for virtual reality. *IEEE Trans Vis Comput Graph*, 28(11):3854–3864, 2022. doi: 10/gr8wgc

[71] P. Diggle. A kernel method for smoothing point process data. *J R Stat Soc C (Appl Stat)*, 34(2):138–147, 1985. doi: 10/bsh8ht

[72] J. E. Burt, G. M. Barber, and D. L. Rigby. *Elementary Statistics for Geographers*. Guilford Press, New York, 3rd ed., 2009. doi: 10/br6ngd

[73] Y.-C. Chen, S. Ho, P. E. Freeman, C. R. Genovese, and L. Wasserman. Cosmic web reconstruction through density ridges: Method and algorithm. *Mon Not R Astron Soc*, 454(1):1140–1156, 2015. doi: 10/f7wrzq

[74] S. Pfeifer, N. I. Libeskind, Y. Hoffman, W. A. Hellwing, M. Bilicki, and K. Naidoo. COWS: A filament finder for Hessian cosmic web identifiers. *Mon Not R Astron Soc*, 514(1):470–479, 2022. doi: 10/p5fx

[75] N. Shivashankar, P. Pranav, V. Natarajan, R. van de Weygaert, E. P. Bos, and S. Rieder. Felix: A topology based framework for visual exploration of cosmic filaments. *IEEE Trans Vis Comput Graph*, 22(6):1745–1759, 2016. doi: 10/f8q5rz

[76] M. Cautun, R. van de Weygaert, and B. J. Jones. NEXUS: Tracing the cosmic web connection. *Mon Not R Astron Soc*, 429(2):1286–1308, 2013. doi: 10/f4wxgz

[77] L. Zhao, T. Isenberg, F. Xie, H.-N. Liang, and L. Yu. MeTACAST: Target-and context-aware spatial selection in VR. *IEEE Trans Vis Comput Graph*, 30(1):480–494, 2024. doi: 10/gtnn25

[78] C. Brunsdon. Estimating probability surfaces for geographical point data: An adaptive kernel algorithm. *Comput GeoSci*, 21(7):877–894, 1995. doi: 10/dnt6g8

[79] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, New York, 1986. doi: 10/gqbn6x

[80] J. Bechtold. The Lyman-Alpha forest near 34 quasi-stellar objects with $z > 2.6$. *Astrophys J Suppl*, 91:1–78, 1994. doi: 10/dpdh8j

[81] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. VL*, pp. 336–343. IEEE Comp. Soc., Los Alamitos, 1996. doi: 10/fwdq26

[82] C. Wingrave, R. Tintner, B. Walker, D. Bowman, and L. Hodges. Exploring individual differences in raybased selection: Strategies and traits. In *Proc. VR*, pp. 163–170. IEEE Comp. Soc., Los Alamitos, 2005. doi: 10/fwv6nk

[83] J. I. Maletic, J. Leigh, A. Marcus, and G. Dunlap. Visualizing object-oriented software in virtual reality. In *Proc. IWPC*, pp. 26–35. IEEE Comp. Soc., Los Alamitos, 2001. doi: 10/dr8299

[84] J. Hong, F. Argelaguet, A. Trubuil, and T. Isenberg. Design and evaluation of three selection techniques for tightly packed 3D objects in cell lineage specification in botany. In *Proc. GI*, pp. 213–223. CHCCS, Mississauga, 2021. doi: 10/kt3q

[85] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3D selection by progressive refinement. In *Proc. 3DUI*, pp. 67–74. IEEE Comp. Soc., Los Alamitos, 2011. doi: 10/ckcczx

[86] C. Tietjen, K. Mühler, F. Ritter, O. Konrad, M. Hindennach, and B. Preim. METK – The medical exploration toolkit. In *Bildverarbeitung für die Medizin*, pp. 407–411. Springer, Berlin, 2008. doi: 10/dvrjj3

[87] A. Wiebel, F. M. Vos, D. Foerster, and H.-C. Hege. WYSIWYP: What you see is what you pick. *IEEE Trans Vis Comput Graph*, 18(12):2236–2244, 2012. doi: 10/f4ft8h

[88] J. F. Lucas, D. A. Bowman, J. Chen, and C. A. Wingrave. Design and evaluation of 3D multiple object selection techniques. Technical report, Virginia Polytechnic Institute and State University, USA, 2005. url: `researchgate.net/publication/228764561`.

[89] M. Kraus, N. Weiler, D. Oelke, J. Kehrer, D. A. Keim, and J. Fuchs. The impact of immersion on cluster identification tasks. *IEEE Trans Vis Comput Graph*, 26(1):525–535, 2020. doi: 10/gtnn23

[90] G. de Haan, M. Koutek, and F. H. Post. IntenSelect: Using dynamic object rating for assisting 3D object selection. In *Proc. EGVE*, 9 pages, pp. 201–209. EG Assoc., Goslar, 2005. doi: 10/hb6rj2

[91] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3D pointing facilitation technique based on raycasting. In *Proc. CHI*, pp. 101:1–101:12. ACM, New York, 2019. doi: 10/gskhtf

[92] M. Maslych, Y. Hmaiti, R. Ghamandi, P. Leber, R. K. Kattoju, J. Belga, and J. J. LaViola. Toward intuitive acquisition of occluded VR objects through an interactive disocclusion mini-map. In *Proc. VR*, pp. 460–470. IEEE Comp. Soc., Los Alamitos, 2023. doi: 10/gsnbhs

[93] Y. Wei, R. Shi, D. Yu, Y. Wang, Y. Li, L. Yu, and H.-N. Liang. Predicting gaze-based target selection in augmented reality headsets based on eye and head endpoint distributions. In *Proc. CHI*, article no. 283, 14 pages. ACM, New York, 2023. doi: 10/hb6rjz

[94] W. A. König, J. Gerken, S. Dierdorf, and H. Reiterer. Adaptive pointing—Design and evaluation of a precision enhancing technique for absolute pointing devices. In *Proc. INTERACT*, pp. 658–671. Springer, Berlin, 2009. doi: 10/dbbtjk

[95] R. Stenholt. Efficient selection of multiple objects on a large scale. In *Proc. VRST*, 8 pages, pp. 105–112. ACM, New York, 2012. doi: 10/hb6rjx

[96] A. Franzluebbers, C. Li, A. Paterson, and K. Johnsen. Virtual reality point cloud annotation. In *Proc. SUI*, article no. 14, 11 pages. ACM, New York, 2022. doi: 10/gtn4jk

[97] R. van Teylingen, W. Ribarsky, and C. van der Mast. Virtual data visualizer. *IEEE Trans Vis Comput Graph*, 3(1):65–74, 1997. doi: 10/c4w4md

[98] B. Hentschel, M. Wolter, and T. Kuhlen. Virtual reality-based multi-view visualization of time-dependent simulation data. In *Proc. VR*, pp. 253–254. IEEE Comp. Soc., Los Alamitos, 2009. doi: 10/fn32r6

[99] N. Brunhart-Lupo, B. W. Bush, K. Gruchalla, and S. Smith. Simulation exploration through immersive parallel planes. In *Workshop on Immersive Analytics*, pp. 19–24. IEEE Comp. Soc., Los Alamitos, 2016. doi: 10/hb6rj6

[100] M. Cordeil, B. Bach, Y. Li, E. Wilson, and T. Dwyer. Design space for spatio-data coordination: Tangible interaction devices for immersive information visualisation. In *Proc. PacificVis*, pp. 46–50. IEEE Comp. Soc., Los Alamitos, 2017. doi: 10/gh38z6

[101] P. Lubos, R. Beimler, M. Lammers, and F. Steinicke. Touching the cloud: Bimanual annotation of immersive point clouds. In *Proc. 3DUI*, pp. 191–192. IEEE Comp. Soc., Los Alamitos, 2014. doi: 10/gjb2n8

[102] S. R. Gomez, R. Jianu, and D. H. Laidlaw. A fiducial-based tangible user interface for white matter tractography. In *Proc. ISVC*, pp. 373–381. Springer, Berlin, 2010. doi: 10/bkf5ps

[103] R. A. Montano-Murillo, C. Nguyen, R. H. Kazi, S. Subramanian, S. DiVerdi, and D. Martinez-Plasencia. Slicing-Volume: Hybrid 3D/2D multi-target selection technique for dense virtual environments. In *Proc. VR*, pp. 53–62. IEEE Comp. Soc., Los Alamitos, 2020. doi: 10/gjb2j2

[104] F. Malmberg, E. Vidholm, and I. Nyström. A 3D live-wire segmentation method for volume images using haptic interaction. In *Proc. DGCI*, pp. 663–673. Springer, Berlin, 2006. doi: 10/dpnfhw

[105] B. Jackson, D. Coffey, and D. F. Keefe. Force Brushes: Progressive Data-Driven Haptic Selection and Filtering for Multi-Variate Flow Visualizations. In M. Meyer and T. Weinkaufs, eds., *Proc. EuroVis-short*. EG Assoc., Goslar, 2012. doi: 10/hb6rj7

[106] K. Danyluk, T. T. Ulusoy, W. Wei, and W. Willett. Touch and beyond: Comparing physical and virtual reality visualizations. *IEEE Trans Vis Comput Graph*, 28(4):1930–1940, 2022. doi: 10/ghk9ww

[107] V. Springel, J. Wang, M. Vogelsberger, A. Ludlow, A. Jenkins, A. Helmi, J. F. Navarro, C. S. Frenk, and S. D. White. The Aquarius project: The subhaloes of galactic haloes. *Mon Not R Astron Soc*, 391(4):1685–1711, 2008. doi: 10/fsjgzw

[108] V. Springel, S. D. M. White, A. Jenkins, C. S. Frenk, N. Yoshida, L. Gao, J. F. Navarro, R. J. Thacker, D. J. Croton, J. C. Helly, J. A. Peacock, S. Cole, P. A. Thomas, H. M. P. Couchman, A. E. Evrard, J. M. Colberg, and F. R. Pearce. Simulations of the formation, evolution and clustering of galaxies and quasars. *Nature*, 435:629–636, 2005. doi: 10/c3cmxr

[109] J. Pivovar, J. DeGuzman, and E. S. Rosenberg. Virtual reality on a SWIM: Scalable world in miniature. In *Proc. VR: Abstracts and Workshops*, pp. 912–913. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/gt2485

[110] L. Zhao, N. Cao, S. He, H.-N. Liang, and L. Yu. L-WiM: Collaborative exploration in immersive environments. In *Proc. ISMAR-Adjunct*, pp. 118–123. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/gtmh3d

[111] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for *soft* objects. *Vis Comput*, 2(4):227–234, 1986. doi: 10/dndmwc

[112] W. E. Lorensen and H. E. Cline. Marching Cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Comput Graph*, 21(4):163–169, 1987. doi: 10/ft9gsh

[113] M. Guest. Morse theory in the 1990's. arXiv preprint math/0104155, 2001. doi: 10.48550/arXiv.math/0104155

[114] B. Ferdosi, H. Buddelmeijer, S. Trager, M. Wilkinson, and J. Roerdink. Comparison of density estimation methods for astronomical. *Astron Astrophys*, 531, article no. A114, 16 pages, 2011. doi: 10/cj3s5m

[115] R. Brown. Valve index. Web site: `vr-compare.com/headset/valveindex`. Last accessed: March 2023.

[116] S. Hart. Nasa-task load index (NASA-TLX); 20 years later. *Proc Hum Factors Ergon Soc Annu Meet*, 50(9):904–908, 2006. doi: 10/fzvtd4

[117] T. Baguley. Standardized or simple effect size: What should be reported? *Br J Psychol*, 100(3):603–617, 2009. doi: 10/bnw2nb

[118] G. Cumming. The new statistics: Why and how. *Psychol Sci*, 25(1):7–29, 2014. doi: 10/5k3

[119] P. Dragicevic. Fair statistical communication in HCI. In J. Robertson and M. Kaptein, eds., *Modern Statistical Methods for HCI*, chap. 13, pp. 291–330. Springer, Cham, 2016. doi: 10/ggd8gc

[120] P. Dragicevic, F. Chevalier, and S. Huot. Running an HCI experiment in multiple parallel universes. In *Proc. CHI EA*, pp. 607–618. ACM, New York, 2014. doi: 10/gpcs8n

[121] G. R. VandenBos, ed. *Publication Manual of the American Psychological Association*. APA, Washington, DC, 6th ed., 2009. url: apastyle.org/manual.

[122] I. Poupyrev and T. Ichikawa. Manipulating objects in virtual worlds: Categorization and empirical evaluation of interaction techniques. *J Vis Lang Comput*, 10(1):19–35, 1999. doi: 10/dnsmpz

[123] F. Argelaguet and C. Andujar. A survey of 3D object selection techniques for virtual environments. *Comput Graph*, 37(3):121–136, 2013. doi: 10/f4ws5z

[124] O. Hahn, C. Porciani, C. M. Carollo, and A. Dekel. Properties of dark matter haloes in clusters, filaments, sheets and voids. *Mon Not R Astron Soc*, 375(2):489–499, 2007. doi: 10/dxbdjj

[125] B. Laha, K. Sensharma, J. D. Schiffbauer, and D. A. Bowman. Effects of immersion on visual analysis of volume data. *IEEE Trans Vis Comput Graph*, 18(4):597–606, 2012. doi: 10/fx9b5g

[126] R. J. García-Hernández, C. Anthes, M. Wiedemann, and D. Kranzlmüller. Perspectives for using virtual reality to extend visual data mining in information visualization. In *Proc. Aerospace Conf.*, pp. 1151–1161. IEEE Comp. Soc., Los Alamitos, 2016. doi: 10/gtmkhw

[127] M. Whitlock, S. Smart, and D. A. Szafir. Graphical perception for immersive analytics. In *Proc. VR*, pp. 616–625. IEEE Comp. Soc., Los Alamitos, 2020. doi: 10/gktm78

[128] B. Sousa Santos, P. Dias, A. Pimentel, J.-W. Baggerman, C. Ferreira, S. Silva, and J. Madeira. Head-mounted display versus desktop for 3D navigation in virtual reality: A user study. *Multimedia Tools Appl*, 41(1):161–181, 2009. doi: 10/ccq4dq

[129] A. Bueckle, K. Buehling, P. C. Shih, and K. Börner. 3D virtual reality vs. 2D desktop registration user interface comparison. *PloS one*, 16(10):e0258103, 2021. doi: 10/qmkr

[130] D. Song and M. Norman. Looking in, looking out: Exploring multiscale data with virtual reality. *IEEE Comput Sci Eng*, 1(3):53–64, 1994. doi: 10/csw826

[131] C.-W. Fu, W.-B. Goh, and J. A. Ng. Multi-touch techniques for exploring large-scale 3D astrophysical simulations. In *Proc. CHI*, pp. 2213–2222. ACM, New York, 2010. doi: 10/drxzf7

[132] I. Cho, J. Li, and Z. Wartell. Multi-scale 7DOF view adjustment. *IEEE Trans Vis Comput Graph*, 24(3):1331–1344, 2018. doi: 10/gcx789

[133] D. Kouřil, T. Isenberg, B. Kozlíková, M. Meyer, M. E. Gröller, and I. Viola. Hyperlabels: Browsing of dense and hierarchical molecular 3D models. *IEEE Trans Vis Comput Graph*, 27(8):3493–3504, 2021. doi: 10/kt4m

[134] F. Argelaguet and M. Maignant. GiAnt: Stereoscopic-compliant multi-scale navigation in VEs. In *Proc. VRST*, pp. 269–277. ACM, New York, 2016. doi: 10/p46w

[135] J. J. LaViola, D. A. Feliz, D. F. Keefe, and R. C. Zeleznik. Hands-free multi-scale navigation in virtual environments. In *Proc. I3D*, pp. 9–15. ACM, New York, 2001. doi: 10/bdmkfc

[136] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a WIM: Interactive worlds in miniature. In *Proc. CHI*, pp. 265–272. ACM, New York, 1995. doi: 10/bx762s

[137] J. Pivovar, J. DeGuzman, and E. S. Rosenberg. Virtual reality on a SWIM: Scalable world in miniature. In *Proc. VRW*, pp. 912–913. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/gt2485

[138] C. A. Wingrave, Y. Haciahmetoglu, and D. A. Bowman. Overcoming world in miniature limitations by a scaled and scrolling WIM. In *Proc. 3DUI*, pp. 11–16. IEEE Comp. Soc., Los Alamitos, 2006. doi: 10/bbp9k4

[139] R. Kopper, T. Ni, D. A. Bowman, and M. Pinho. Design and evaluation of navigation techniques for multiscale virtual environments. In *Proc. VR*, pp. 175–182. IEEE Comp. Soc., Los Alamitos, 2006. doi: 10/cjs6g6

[140] F. Bacim, D. Bowman, and M. Pinho. Wayfinding techniques for multiscale virtual environments. In *Proc. 3DUI*, pp. 67–74. IEEE Comp. Soc., Los Alamitos, 2009. doi: 10/fg3dbh

[141] S. Halladjian, D. Kouřil, H. Miao, M. E. Gröller, I. Viola, and T. Isenberg. Multiscale unfolding: Illustratively visualizing the whole genome at a glance. *IEEE Trans Vis Comput Graph*, 28(10):3456–3470, 2022. doi: 10/kt3m

[142] V. Chheang, B. T. Weston, R. W. Cerda, B. Au, B. Giera, P.-T. Bremer, and H. Miao. A virtual environment for collaborative inspection in additive manufacturing. In *Proc. CHI*, article no. 26, 7 pages. ACM, New York, 2024. doi: 10/g92v9w

[143] P. Klacansky, H. Miao, A. Gyulassy, A. Townsend, K. Champley, J. Tringe, V. Pascucci, and P.-T. Bremer. Virtual inspection of additively manufactured parts. In *Proc. PacificVis*, pp. 81–90. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/g92v9x

[144] L. Pavanatto, A. Giovannelli, B. Giera, T. Bremer, H. Miao, and D. A. Bowman. Exploring multiscale navigation of homogeneous and dense objects with progressive refinement in virtual reality. In *Proc. VR*, pp. 228–237. IEEE Comp. Soc., Los Alamitos, 2025. doi: 10/p5fw

[145] C. Ware and D. Fleet. Context sensitive flying interface. In *Proc. I3D*, pp. 127–130. ACM, New York, 1997. doi: 10/drbfvq

[146] T. Weissker, M. Franzgrote, and T. Kuhlen. Try this for size: Multi-scale teleportation in immersive virtual reality. *IEEE Trans Vis Comput Graph*, 30(5):2298–2308, 2024. doi: 10/p5ft

[147] E. Gansner, Y. Koren, and S. North. Topological fisheye views for visualizing large graphs. *IEEE Trans Vis Comput Graph*, 11(4):457–468, 2005. doi: 10/cgmnfg

[148] K. Saito, M. Yano, H. Hino, T. Shoji, A. Asahara, H. Morita, C. Mitsumata, J. Kohlbrecher, and K. Ono. Accelerating small-angle scattering experiments on anisotropic samples using kernel density estimation. *Sci Rep*, 9(1), article no. 1526, 10 pages, 2019. doi: 10/p5f4

[149] Z. Yuan, M. J. Jarvis, and J. Wang. A flexible method for estimating luminosity functions via kernel density estimation. *Astrophys J Suppl Ser*, 248(1), article no. 1, 18 pages, 2020. doi: 10/p5qc

[150] Z. Yuan, X. Zhang, J. Wang, X. Cheng, and W. Wang. A flexible method for estimating luminosity functions via kernel density estimation. II. Generalization and Python

implementation. *Astrophys J Suppl Ser*, 260(1), article no. 10, 15 pages, 2022. doi: 10/p5f2

[151] M.-T. Pelz, M. Schartau, C. J. Somes, V. Lampe, and T. Slawig. A diffusion-based kernel density estimator (diffKDE, version 1) with optimal bandwidth approximation for the analysis of data in geoscience and ecological research. *GeoSci Model Dev*, 16(22):6609–6634, 2023. doi: 10/p5f3

[152] C. H. Fleming and J. M. Calabrese. A new kernel density estimator for accurate home-range and species-range area estimation. *Methods Ecol Evol*, 8(5):571–579, 2017. doi: 10/f98k6z

[153] Z. Xie and J. Yan. Kernel density estimation of traffic accidents in a network space. *Comput Env Urban Syst*, 32(5):396–406, 2008. doi: 10/b7gxpq

[154] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, Hoboken, 2nd ed., 2015. doi: 10/p5f5

[155] V. A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory Probab Appl*, 14(1):153–158, 1969. doi: 10/bjdbsc

[156] B. Ferdosi, H. Buddelmeijer, S. Trager, M. Wilkinson, and J. Roerdink. Comparison of density estimation methods for astronomical datasets. *Astron Astrophys*, 531, article no. A114, 16 pages, 2011. doi: 10/cj3s5m

[157] M. H. F. Wilkinson and B. C. Meijer. DATAPLOT: A graphical display package for bacterial morphometry and fluorimetry data. *Comput Methods Programs Biomed*, 47(1):35–49, 1995. doi: 10/dnr53p

[158] A. Backurs, P. Indyk, and T. Wagner. Space and time efficient kernel density estimation in high dimensions. *Adv Neural Inf Process Syst*, 32, article no. 9251, 10 pages, 2019. url: papers.nips.cc/paper_files/paper/2019/hash/a2ce8f1706e52936dfad516c23904e3e-Abstract.html.

[159] O. D. Lampe and H. Hauser. Interactive visualization of streaming data with kernel density estimation. In *Proc. Pacific Vis*, pp. 171–178. IEEE Comp. Soc., Los Alamitos, 2011. doi: 10/c866v9

[160] T. N. Chan, R. Cheng, and M. L. Yiu. QUAD: Quadratic-bound-based kernel density visualization. In *Proc. SIGMOD*, pp. 35–50. ACM, New York, 2020. doi: 10/gkpt7c

[161] G. Zhang, A.-X. Zhu, and Q. Huang. A GPU-accelerated adaptive kernel density estimation approach for efficient point pattern analysis on spatial big data. *Int J Geogr Inf Sci*, 31(10):2068–2097, 2017. doi: 10/ggfmp5

[162] T. N. Chan, P. L. Ip, B. Zhu, D. Wu, J. Xu, C. S. Jensen, et al. Large-scale spatiotemporal kernel density visualization. In *Proc. ICDE*, pp. 99–113. IEEE Comp. Soc., Los Alamitos, 2025. doi: 10/qmw4

[163] A. Prouzeau, M. Cordeil, C. Robin, B. Ens, B. H. Thomas, and T. Dwyer. Scaptics and highlight-planes: Immersive interaction techniques for finding occluded features in 3D scatterplots. In *Proc. CHI*, article no. 325, 12 pages. ACM, New York, 2019. doi: 10/gnkk76

[164] B. He, N. K. Govindaraju, Q. Luo, and B. Smith. Efficient gather and scatter operations on graphics processors. In *Proc. SC*, article no. 46, 12 pages. ACM, New York, 2007. doi: 10/fgtjzk

[165] H. M. Kakde. Range searching using KD tree. Technical report, Dept. of Computer Science, Florida State Univ., 2005. Available: `https://users.cs.utah.edu/%7Elifeifei/cis5930/kdtree.pdf`.

[166] G. Zhang, Q. Huang, A.-X. Zhu, and J. H. Keel. Enabling point pattern analysis on spatial big data using cloud computing: Optimizing and accelerating Ripley's K function. *Int J Geogr Inf Sci*, 30(11):2230–2252, 2016. doi: 10/gjkw5j

[167] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualizer, an information workspace. In *Proc. CHI*, pp. 181–186. ACM, New York, 1991. doi: 10/cvtdps

[168] W.-H. Hsu, Y. Zhang, and K.-L. Ma. A multi-criteria approach to camera motion design for volume data animation. *IEEE Trans Vis Comput Graph*, 19(12):2792–2801, 2013. doi: 10/f5h3m3

[169] Z. Cao, J. Han, S. Yang, and X. Jin. Fast best viewpoint selection with geometry-enhanced multiple views and cross-modal distillation. *Vis Comput*, 41(7):5075–5086, 2025. doi: 10/p5p3

[170] C. Yang, Y. Li, C. Liu, and X. Yuan. Deep learning-based viewpoint recommendation in volume visualization. *J Vis*, 22(5):991–1003, 2019. doi: 10/p5p4

[171] R. Brown. HTC Vive Pro2. Web site: vr-compare.com/headset/htcvivepro2. Last accessed: Sep. 2025.

[172] K. Olsen, R. M. H. Lindrup, and M. Mørup. Think global, adapt local: Learning locally adaptive k-nearest neighbor kernel density estimators. *PMLR*, 238:4114–4122, 2024. Available: proceedings.mlr.press/v238/olsen24a.html.

[173] T. T. Huyen Nguyen and T. Duval. A survey of communication and awareness in collaborative virtual environments. In *Proc. 3DCVE*, article no. 1, 8 pages. IEEE Comp. Soc., Los Alamitos, 2014. doi: 10/gf5f2h

[174] Y. Li, C.-W. Fu, and A. Hanson. Scalable wim: Effective exploration in large-scale astrophysical environments. *IEEE Trans Vis Comput Graph*, 12(5):1005–1012, 2006. doi: 10/b8wr2w

[175] C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, et al. Immersive and collaborative data visualization using virtual reality platforms. In *Proc. Big Data*, pp. 609–614. IEEE Comp. Soc., Los Alamitos, 2014. doi: 10/gf82hr

[176] P. Isenberg, N. Elmqvist, J. Scholtz, D. Cernea, K.-L. Ma, and H. Hagen. Collaborative visualization: Definition, challenges, and research agenda. *Information Visualization*, 10(4):310–326, 2011. doi: 10/b93p9h

[177] W. Krüger, C.-A. Bohn, B. Fröhlich, H. Schüth, W. Strauss, and G. Wesche. The responsive workbench: A virtual work environment. *Computer*, 28(07):42–48, 1995. doi: 10/cvbntz

[178] J.-C. Lombardo. A virtual reality environment for macromolecules exploration. In *Proc. Laval virtual, First International Virtual Reality meeting*, pp. 1–10. Citeseer, 1999.

[179] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 135–142, 1993. doi: 10/bwf45x

[180] A. Febretti, A. Nishimoto, T. Thigpen, J. Talandis, L. Long, J. Pirtle, T. Peterka, A. Verlo, M. Brown, D. Plepys, et al. Cave2: a hybrid reality environment for immersive simulation and information analysis. In *Proc. The Engineering Reality of Virtual Reality*, vol. 8649, pp. 9–20. SPIE, 2013. doi: 10/ggw56c

[181] G. E. Marai, A. G. Forbes, and A. Johnson. Interdisciplinary immersive analytics at the electronic visualization laboratory: Lessons learned and upcoming challenges. In *Proc. Workshop on Immersive Analytics*, pp. 54–59. IEEE, 2016. doi: 10/hb6rjp

[182] M. Cordeil, T. Dwyer, K. Klein, B. Laha, K. Marriott, and B. H. Thomas. Immersive collaborative analysis of network connectivity: Cave-style or head-mounted display? *IEEE Trans Vis Comput Graph*, 23(1):441–450, 2016. doi: 10/f92fxv

[183] G. . i. e. j. Kurillo and M. Forte. Telearch–integrated visual simulation environment for collaborative virtual archaeology. *Mediterranean Archaeology and Archaeometry*, 12(1):11–20, 2012. Available: `www.maajournal.com/index.php/maa/article/view/488`.

[184] B. Lee, X. Hu, M. Cordeil, A. Prouzeau, B. Jenny, and T. Dwyer. Shared surfaces and spaces: Collaborative data visualisation in a co-located immersive environment. *IEEE Trans Vis Comput Graph*, 27(2):1171–1181, 2020. doi: 10/ghgt5v

[185] C. Gutwin and S. Greenberg. A descriptive framework of workspace awareness for real-time groupware. *Comput Supported Cooperative Work*, 11(3):411–446, 2002. doi: 10/c43wsg

[186] C. Gutwin and S. Greenberg. *The importance of awareness for team cognition in distributed collaboration.* American Psychological Association, 2004. doi: 10/d3kd4w

[187] T. Piumsomboon, A. Dey, B. Ens, G. Lee, and M. Billinghurst. The effects of sharing awareness cues in collaborative mixed reality. *Front Rob AI*, 6, article no. 5, 18 pages, 2019. doi: 10/ggtr7k

[188] T. Drey, P. Albus, S. der Kinderen, M. Milo, T. Segschneider, L. Chanzab, M. Rietzler, T. Seufert, and E. Rukzio. Towards collaborative learning in virtual reality: A comparison of co-located symmetric and asymmetric pair-learning. In *Proc. CHI*, article no. 610, 19 pages. ACM, New York, 2022. doi: 10/pjm8

[189] X. Xu, A. Puggioni, D. Kilroy, and A. G. Campbell. User experience of collaborative co-located mixed reality: A user study in teaching veterinary radiation safety rules. In *Proc. ISMAR*, pp. 583–590. IEEE Comp. Soc., Los Alamitos, 2023. doi: 10/pjm9

[190] F. Born, P. Sykownik, and M. Masuch. Co-located vs. Remote gameplay: The role of physical co-presence in multiplayer room-scale VR. In *Proc. CIG*, pp. 81–88. IEEE Comp. Soc., Los Alamitos, 2019. doi: 10/ggqpd9

[191] H. Nguyen, B. Ward, U. Engelke, B. Thomas, and T. Bednarz. Collaborative data analytics using virtual reality. In *Proc. VR*, pp. 1098–1099. IEEE Comp. Soc., Los Alamitos, 2019. doi: 10/pjmw

[192] A. Yassien, Y. Emad, and S. Abdennadher. CDVVAR: VR/AR collaborative data visualization tool. In *Proc. VRW*, pp. 599–600. IEEE Comp. Soc., Los Alamitos, 2021. doi: 10/g5h6xr

[193] S. D. Scott, M. S. T. Carpendale, and K. Inkpen. Territoriality in collaborative tabletop workspaces. In *Proc. CSCW*, pp. 294–303. ACM, New York, 2004. doi: 10/dh2wc5

[194] G. Smith and J. Mariani. Using subjective views to enhance 3D applications. In *Proc. VRSI*, pp. 139–146. ACM, New York, 1997. doi: 10/b5njck

[195] A. Kunert, T. Weissker, B. Froehlich, and A. Kulik. Multi-window 3D interaction for collaborative virtual reality. *IEEE Trans Vis Comput Graph*, 26(11):3271–3284, 2020. doi: 10/gtmh3j

[196] H. Xia, S. Herscher, K. Perlin, and D. Wigdor. Spacetime: Enabling fluid individual and collaborative editing in virtual reality. In *Proc. UIST*, pp. 853–866. ACM, New York, 2018. doi: 10/gjbvfr

[197] Y. Yang, M. Cordeil, J. Beyer, T. Dwyer, K. Marriott, and H. Pfister. Embodied navigation in immersive abstract data visualization: Is overview+detail or zooming better for 3D scatterplots? *IEEE Trans Vis Comput Graph*, 27(2):1214–1224, 2021. doi: 10/ghgt52

[198] J. Sorger, M. Waldner, W. Knecht, and A. Arleo. Immersive analytics of large dynamic networks via overview and detail navigation. In *Proc. AIVR*, pp. 144–1447. IEEE Comp. Soc., Los Alamitos, 2019. doi: 10/ghp4j5

[199] A. Bluff and A. Johnston. Don't panic: Recursive interactions in a miniature metaworld. In *Proc. SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, article no. 33, 9 pages. ACM, New York, 2019. doi: 10/hb6rjm

[200] L. Chittaro, V. K. Gatla, and S. Venkataraman. The interactive 3d breakaway map: A navigation and examination aid for multi-floor 3d worlds. In *Proc. Cyberworlds*, pp. 8 pp.–66. IEEE Comp. Soc., Los Alamitos, 2005. doi: 10/b9pj7t

[201] C. Hull and W. Willett. Building with data: Architectural models as inspiration for data physicalization. In *Proc. CHI*, pp. 1217–1264. ACM, New York, 2017. doi: 10/hb6rkb

[202] K. Danyluk, B. Ens, B. Jenny, and W. Willett. A design space exploration of worlds in miniature. In *Proc. CHI*, pp. 1–15. ACM, New York, 2021. doi: 10/gj5snp

[203] E. Vote, D. A. Feliz, D. H. Laidlaw, and M. S. Joukowsky. Discovering petra: Archaeological analysis in vr. *IEEE Comput Graph Appl*, 22(5):38–50, 2002. doi: 10/bzxhhm

[204] A. Stafford, W. Piekarski, and B. H. Thomas. Implementation of god-like interaction techniques for supporting collaboration between outdoor ar and indoor tabletop users. In *Proc. ISMAR*, pp. 165–172. IEEE Comp. Soc., Los Alamitos, 2006. doi: 10/cgcxzr

[205] V. Chheang, F. Heinrich, F. Joeres, P. Saalfeld, B. Preim, and C. Hansen. Group wim: A group navigation technique for collaborative virtual reality environments. In *Proc. VRW*, pp. 556–557. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/hb6rj9

[206] S. van den Elzen and J. J. van Wijk. Small multiples, large singles: A new approach for visual data exploration. *Comput Graph Forum*, 32(2):191–200, 2013. doi: 10/gbdbgf

[207] S. Boorboor, Y. Kim, P. Hu, J. M. Moses, B. A. Colle, and A. E. Kaufman. Submerse: Visualizing storm surge flooding simulations in immersive display ecologies. *IEEE Trans Vis Comput Graph*, 13 pages, 2024. doi: 10/gt2smx

[208] L. A. Garrison, I. Kolesar, I. Viola, H. Hauser, and S. Bruckner. Trends & opportunities in visualization for physiology: A multiscale overview. *Comput Graph Forum*, 41(3):609–643, 2022. doi: 10/gtmk6z

[209] S. I. O'Donoghue, D. S. Goodsell, A. S. Frangakis, F. Jossinet, R. A. Laskowski, M. Nilges, H. R. Saibil, A. Schafferhans, R. C. Wade, E. Westhof, and A. J. Olson.

Visualization of macromolecular structures. *Nat Methods*, 7(3):S42–S55, 2010. doi: 10/df36n3

[210] D. Kuťák, P.-p. Vázquez, T. Isenberg, M. Krone, M. Baaden, J. Byška, B. Kozlíková, and H. Miao. State of the art of molecular visualization in immersive virtual environments. *Comput Graph Forum*, 42(6), article no. e14738, 29 pages, 2023. doi: 10/kt4n

[211] M. van Der Zwan, W. Lueks, H. Bekker, and T. Isenberg. Illustrative molecular visualization with continuous abstraction. *Comput Graph Forum*, 30(3):683–690, 2011. doi: 10/c893rz

[212] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987. doi: 10/gf7bgr

[213] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *Proc. VIS*, pp. 156–163. IEEE Comp. Soc., Los Alamitos, 1991. doi: 10/c7wvqq

[214] G. Wills. Linked data views. In C.-h. Chen, W. Härdle, and A. Unwin, eds., *Handbook of Data Visualization*, chap. II.9, pp. 217–241. Springer, Berlin, 2008. doi: 10/dkvn57

[215] M. Meuschke, S. Voss, O. Beuing, B. Preim, and K. Lawonn. Combined visualization of vessel deformation and hemodynamics in cerebral aneurysms. *IEEE Trans Vis Comput Graph*, 23(1):761–770, 2017. doi: 10/f92d8h

[216] M. Schäfer, N. Brich, J. Byška, S. M. Marques, D. Bednář, P. Thiel, B. Kozlíková, and M. Krone. Invado: Interactive visual analysis of molecular docking data. *IEEE Trans Vis Comput Graph*, 30(4):1984–1997, 2024. doi: 10/gtnn2n

[217] J. Hong, R. Hnatyshyn, E. A. Santos, R. Maciejewski, and T. Isenberg. A survey of designs for combined 2D+3D visual representations. *IEEE Trans Vis Comput Graph*, 30(6):2888–2902, 2024. doi: 10/gtnn2v

[218] A. Riegler, C. Anthes, H.-C. Jetter, C. Heinzl, C. Holzmann, H. Jodlbauer, M. Brunner, S. Auer, J. Friedl, B. Fröhler, C. Leitner, F. Pointecker, D. Schwajda, and S. Tripathi. Cross-virtuality visualization, interaction and collaboration. In *Proc. ISS*. ACM, New York, 2020. Available: `ceur-ws.org/Vol-2779/paper1.pdf`.

[219] A. Martinet, G. Casiez, and L. Grisoni. Integrality and separability of multitouch interaction techniques in 3D manipulation tasks. *IEEE Trans Vis Comput Graph*, 18(3):369–380, 2012. doi: 10/c6g7jt

[220] P. Song, X. Yan, W. B. Goh, A. Q. Chen, and C.-W. Fu. Hand-posture-augmented multitouch interactions for exploratory visualization. In *Proc. SIGGRAPH ASIA Technical Briefs*, article no. 27, 4 pages. ACM, New York, 2016. doi: 10/gtnn2k

[221] X. Wang, L. Besançon, M. Ammi, and T. Isenberg. Augmenting tactile 3D data navigation with pressure sensing. *Comput Graph Forum*, 38(3):635–647, 2019. doi: 10/gjbgt9

[222] A. Olwal, S. Feiner, and S. Heyman. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proc. CHI*, pp. 295–304. ACM, New York, 2008. doi: 10/dgz29t

[223] T. Butkiewicz, A. H. Stevens, and C. Ware. Multi-touch 3D positioning with the Pantograph technique. In *Proc. I3D*, article no. 13, 9 pages. ACM, New York, 2019. doi: 10/gtnn2m

[224] S. Strothoff, D. Valkov, and K. Hinrichs. Triangle cursor: Interactions with objects above the tabletop. In *Proc. ITS*, pp. 111–119. ACM, New York, 2011. doi: 10/b6jrqq

[225] A. Çöltekin, I. Lochhead, M. Madden, S. Christophe, A. Devaux, C. Pettit, O. Lock, S. Shukla, L. Herman, Z. Stachoň, P. Kubíček, D. Snopková, S. Bernardes, and N. Hedley. Extended reality in spatial sciences: A review of research challenges and future directions. *ISPRS Int J Geo-Inf*, 9(7), article no. 439, 29 pages, 2020. doi: 10/ghk9xk

[226] M. Sereno, X. Wang, L. Besançon, M. J. Mcguffin, and T. Isenberg. Collaborative work in augmented reality: A survey. *IEEE Trans Vis Comput Graph*, 28(6):2530–2549, 2022. doi: 10/gjkq7w

[227] B. Fröhler, C. Anthes, F. Pointecker, J. Friedl, D. Schwajda, A. Riegler, S. Tripathi, C. Holzmann, M. Brunner, H. Jodlbauer, H.-C. Jetter, and C. Heinzl. A survey on cross-virtuality analytics. *Comput Graph Forum*, 41(1):465–494, 2022. doi: 10/gtnn2w

[228] J. Auda, U. Gruenefeld, S. Faltaous, S. Mayer, and S. Schneegass. A scoping survey on cross-reality systems. *ACM Comput Surv*, 56(4), article no. 83, 38 pages, 2023. doi: 10/gtnn22

[229] H.-N. Liang, L. Yu, and F. Liarokapis. Workshop: Mixing realities: Cross-reality visualization, interaction, and collaboration. In *Proc. VRW*, pp. 298–300. IEEE Comp. Soc., Los Alamitos, 2023. doi: 10/gtnn2q

[230] H.-N. Liang, H.-C. Jetter, F. Maurer, U. Gruenefeld, M. Billinghurst, and C. Anthes. *1st Joint Workshop on Cross Reality*. 2023. doi: 10/gtnn2r

[231] R. Kijima and T. Ojika. Transition between virtual environment and workstation environment with projective head mounted display. In *Proc. VR*, pp. 130–137. IEEE Comp. Soc., Los Alamitos, 1997. doi: 10/fwtdxw

[232] M. R. Seraji and W. Stuerzlinger. Hybridaxes: An immersive analytics tool with interoperability between 2D and immersive reality modes. In *Proc. ISMAR-Adjunct*, pp. 155–160. IEEE Comp. Soc., Los Alamitos, 2022. doi: 10/gtnn2x

[233] P. Reipschläger, T. Flemisch, and R. Dachselt. Personal augmented reality for information visualization on large interactive displays. *IEEE Trans Vis Comput Graph*, 27(2):1182–1192, 2021. doi: 10/ghgt5w

[234] N. J. Dedual, O. Oda, and S. K. Feiner. Creating hybrid user interfaces with a 2d multi-touch tabletop and a 3d see-through head-worn display. In *Proc. ISMAR*, pp. 231–232. IEEE Comp. Soc., Los Alamitos, 2011. doi: 10/bw9r4x

[235] S. Butscher, S. Hubenschmid, J. Müller, J. Fuchs, and H. Reiterer. Clusters, trends, and outliers: How immersive technologies can facilitate the collaborative analysis of multidimensional data. In *Proc. CHI*, pp. 1–12. ACM, New York, 2018. doi: 10/ggfwqj

[236] R. Langner, M. Satkowski, W. Büschel, and R. Dachselt. MARVIS: Combining mobile devices and augmented reality for visual data analysis. In *Proc. CHI*, article no. 468, 17 pages. ACM, New York, 2021. doi: 10/gksmhm

[237] K. A. Satriadi, J. Smiley, B. Ens, M. Cordeil, T. Czauderna, B. Lee, Y. Yang, T. Dwyer, and B. Jenny. Tangible globes for data visualisation in augmented reality. In *Proc. CHI*, article no. 505, 16 pages. ACM, New York, 2022. doi: 10/grv5tm

[238] D. Schwajda, J. Friedl, F. Pointecker, H.-C. Jetter, and C. Anthes. Transforming graph data visualisations from 2D displays into augmented reality 3D space: A quantitative study. *Front Virtual Reality*, 4, article no. 1155628, 19 pages, 2023. doi: 10/kvj9

[239] B. Jackson, T. Y. Lau, D. Schroeder, K. C. Toussaint, and D. F. Keefe. A lightweight tangible 3D interface for interactive visualization of thin fiber structures. *IEEE Trans Vis Comput Graph*, 19(12):2802–2809, 2013. doi: 10/gh38r2

[240] B. Fröhlich and J. Plate. The Cubic Mouse: A new device for three-dimensional input. In *Proc. CHI*, pp. 526–531. ACM, New York, 2000. doi: 10/dgg8k8

[241] B. R. De Araújo, G. Casiez, J. A. Jorge, and M. Hachet. Mockup Builder: 3D modeling on and above the surface. *Comput Graph*, 37(3):165–178, 2013. doi: 10/f4wv2s

[242] S. Wu, D. Byrne, and M. W. Steenson. "megereality": Leveraging physical affordances for multi-device gestural interaction in augmented reality. In *Proc. CHI EA*, pp. 1–4. ACM, New York, 2020. doi: 10/gmkzwj

[243] H. Benko and A. D. Wilson. Multi-point interactions with immersive omnidirectional visualizations in a dome. In *Proc. ITS*, pp. 19–28. ACM, New York, 2010. doi: 10/cxw789

[244] D. López, L. Oehlberg, C. Doger, and T. Isenberg. Towards an understanding of mobile touch navigation in a stereoscopic viewing environment for 3D data exploration. *IEEE Trans Vis Comput Graph*, 22(5):1616–1629, 2016. doi: 10/f8ghxc

[245] I. Viola and T. Isenberg. Pondering the concept of abstraction in (illustrative) visualization. *IEEE Trans Vis Comput Graph*, 24(9):2573–2588, 2018. doi: 10/gd3k7m

[246] I. Viola, M. Chen, and T. Isenberg. Visual abstraction. In *Foundations of Data Visualization*, chap. 2, pp. 15–37. Springer, Berlin, 2020. doi: 10/gk874c

[247] C. Ware, K. Arthur, and K. S. Booth. Fish tank virtual reality. In *Proc. CHI*, pp. 37–42. ACM, New York, 1993. doi: 10/cqjwdf

[248] N. Wang, D. Zielasko, and F. Maurer. User preferences for interactive 3D object transitions in cross reality – An elicitation study. In *Proc. AVI*, article no. 22, 9 pages. ACM, New York, 2024. doi: 10/gt2smz

[249] N. Nguyen, O. Strnad, T. Klein, D. Luo, R. Alharbi, P. Wonka, M. Maritan, P. Mindek, L. Autin, D. S. Goodsell, and I. Viola. Modeling in the time of COVID-19: Statistical and rule-based mesoscale models. *IEEE Trans Vis Comput Graph*, 27(2):722–732, 2021. doi: 10/k8sh

[250] T. Butkiewicz and C. Ware. Multi-touch 3D exploratory analysis of ocean flow models. In *Proc. OCEANS*, pp. 746–755–10. IEEE Comp. Soc., Los Alamitos, 2011. doi: 10/gjb2jq

[251] G. Perelman, E. Dubois, A. Probst, and M. Serrano. Visual transitions around tabletops in mixed reality: Study on a visual acquisition task between vertical virtual displays and horizontal tabletops. *Proc ACM Hum-Comput Interact*, 6(585):660–679, 2022. doi: 10/grd27g

[252] M. Krüger, T. Gerrits, T. Römer, T. Kuhlen, and T. Weissker. IntenSelect+: Enhancing score-based selection in virtual reality. *IEEE Trans Vis Comput Graph*, 30(5):2829–2838, 2024. doi: 10/gtn4jm

[253] R. Shi, Y. Wei, X. Qin, P. Hui, and H.-N. Liang. Exploring gaze-assisted and hand-based region selection in augmented reality. *Proc ACM Hum-Comput Interact*, article no. 160, 19 pages, 2023. doi: 10/gt2sm2

[254] S. Doutreligne, T. Cragnolini, S. Pasquali, P. Derreumaux, and M. Baaden. Unitymol: interactive scientific visualization for integrative biology. In *Proc. LDAV*, pp. 109–110. IEEE, 2014. doi: 10/m53v

[255] M. Chavent, A. Vanel, A. Tek, B. Levy, S. Robert, B. Raffin, and M. Baaden. Gpu-accelerated atom and dynamic bond visualization using hyperballs: A unified algorithm for balls, sticks, and hyperboloids. *Journal of computational chemistry*, 32(13):2924–2935, 2011. doi: 10/c3tdp5

[256] M. Hancock, T. ten Cate, S. Carpendale, and T. Isenberg. Supporting sandtray therapy on an interactive tabletop. In *Proc. CHI*, pp. 2133–2142. ACM, New York, 2010. doi: 10/dv4k73

[257] S. Jadhav and A. E. Kaufman. MD-Cave: An immersive visualization workbench for radiologists. *IEEE Trans Vis Comput Graph*, 29(12):4832–4844, 2022. doi: 10/gt2smv

[258] M. Pfeiffer, H. Kenngott, A. Preukschas, M. Huber, L. Bettscheider, B. Müller-Stich, and S. Speidel. Imhotep: virtual reality framework for surgical applications. *International journal of computer assisted radiology and surgery*, 13:741–748, 2018. doi: 10/gdj9p9

[259] M. Benmahdjoub, A. Thabit, M.-L. C. van Veelen, W. J. Niessen, E. B. Wolvius, and T. v. Walsum. Evaluation of AR visualization approaches for catheter insertion into the ventricle cavity. *IEEE Trans Vis Comput Graph*, 29(5):2434–2445, 2023. doi: 10/mpq6

[260] C. Lundström, T. Rydell, C. Forsell, A. Persson, and A. Ynnerman. Multi-touch table system for medical visualization: Application to orthopedic surgery planning. *IEEE Trans Vis Comput Graph*, 17(12):1775–1784, 2011. doi: 10/bzwxj8

[261] M. Sousa, D. Mendes, S. Paulo, N. Matela, J. Jorge, and D. S. o. Lopes. VRRRRoom: Virtual reality for radiologists in the reading room. In *Proc. CHI*, pp. 4057–4062. ACM, New York, 2017. doi: 10/mn9z

[262] X. Wang, L. Besançon, D. Rousseau, M. Sereno, M. Ammi, and T. Isenberg. Towards an understanding of augmented reality extensions for existing 3D data analysis tools. In *Proc. CHI*, article no. 528, 13 pages. ACM, New York, 2020. doi: 10/gm4jj8