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

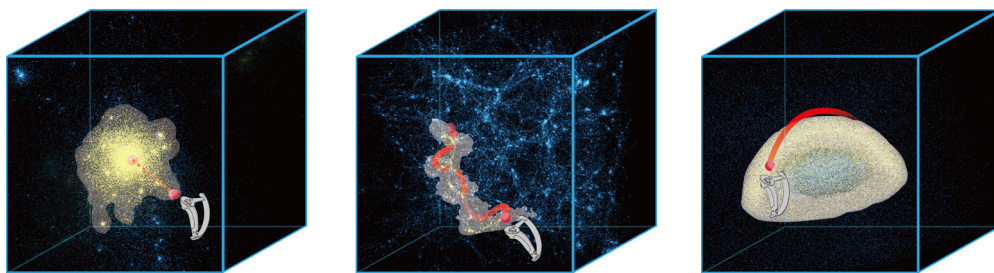Lixiang Zhao 🆔, Tobias Isenberg 🆔, Fuqi Xie 🆔, Hai-Ning Liang 🆔, Lingyun Yu 🆔

Fig. 1: MeTACAST: a family of Target- and Context-aware spatial selection techniques for 3D point cloud data in VR.

Abstract—We propose three novel spatial data selection techniques for particle data in VR visualization environments. They are designed to be target- and context-aware and be suitable for a wide range of data features and complex scenarios. Each technique is designed to be adjusted to particular selection intents: the selection of consecutive dense regions, the selection of filament-like structures, and the selection of clusters—with all of them facilitating post-selection threshold adjustment. These techniques allow users to precisely select those regions of space for further exploration—with simple and approximate 3D pointing, brushing, or drawing input—using flexible point- or path-based input and without being limited by 3D occlusions, non-homogeneous feature density, or complex data shapes. These new techniques are evaluated in a controlled experiment and compared with the Baseline method, a region-based 3D painting selection. Our results indicate that our techniques are effective in handling a wide range of scenarios and allow users to select data based on their comprehension of crucial features. Furthermore, we analyze the attributes, requirements, and strategies of our spatial selection methods and compare them with existing state-of-the-art selection methods to handle diverse data features and situations. Based on this analysis we provide guidelines for choosing the most suitable 3D spatial selection techniques based on the interaction environment, the given data characteristics, or the need for interactive post-selection threshold adjustment.

Index Terms—Spatial selection, immersive analytics, virtual reality (VR), target-aware and context-aware interaction for visualization

---◆---

1 INTRODUCTION

Immersive environments experienced via virtual, augmented, and mixed reality (i. e., extended reality, XR) head-mounted displays (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, 8], 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 [49]). 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 [6, 59]. 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 [63, 64] 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. 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 unknown 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;
- algorithms 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.

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 Region of Interest (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.

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 [1, 34, 39, 45, 60]. 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 [27]. 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. [32] 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 [54] and 3D lasso [30]. 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. [58], 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 [23] or the use of cubes to select brain fibers [11]. 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 [62], 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) [37, 38] and Volume Catcher [44] allow users to flexibly select an ROI in 3D space based on a 2D lasso or a stroke. Besançon et al. [7] 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 [12], 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 [63, 64] 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.

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 [33] 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. [28] 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. [17] developed IntenSelect to aid users in selecting objects in motion within cluttered and occluded VR environments with a scoring function. Baloup et al. [4] introduced a controllable cursor positioned on top of a ray, allowing users to select nearby or occluded objects. Maslych et al. [41] 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. [57] 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 [31] also enhances the accuracy of pointing devices. Finally, Stenholt [52] 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 [21], 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 [26, 55] and cone-based selection volumes [10]. Spatio-Data Coordination [14] and the Embodied Axes [13] 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. [36] 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. [22] 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. [48], e. g., integrated an AR device with a tablet that acts as a tangible device for 3D selection. Montano-Murillo et al. [43] 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. [40] 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. [42] 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. [29], 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 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.

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 their selection strategies to us 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. Both studies (think-aloud, user study) received approval from the University Ethics Committee.

The **size of the visualization** is a crucial factor for the users' selection strategies, especially 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. [33] found that 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 the approach by Danyluk et al. [16] we investigated three different sizes (Fig. 12 in Appx. A): 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** (Fig. 13 in Appx. A). We rendered the target particles in yellow, while we presented the interfering particles in blue. We selected each dataset based on its unique features:

Clusters: 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 [50]).

Filament: a cosmic web simulation with thin filaments (from [51]).

Complex geometries: an empty half-box outside, a half-ball, and a cuboid of particles inside.

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 clearly 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 more intuitive 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 [42].

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.

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.

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 [46,53,65] and developed a handheld 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 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 [63] and CAST [64], instead of relying on particle positions we use a continuous density field $\rho(\mathbf{r})$ that represents
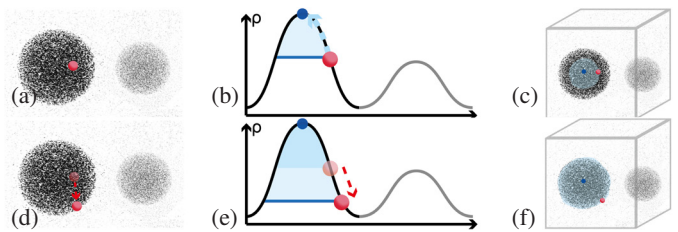

Fig. 2: MeTAPoint: (a) the user points at the target cluster (red); (b) we derive the closest maximum point (blue) and density threshold (schematic representation); (c) we compute the selection volume; (d) the user drags the controller to adjust the density threshold; (e, f) we recompute the density threshold and selection volume.

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 and developed three techniques—MeTAPoint, MeTABrush, and MeTAPaint—that we describe next.

4.2 MeTAPoint

The problems in VR with general free-hand input imprecision and placement with respect to a 3D structure (Sec. 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. 2(a)) to find the local maximum $\mathbf{r}^{(m)}$ of the scalar field (blue point in Fig. 2(b)). We use this input adjustment in all MeTACAST techniques and discuss technical details in Sec. 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 [35,61] 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. 2(b, c)).

The user can continue to explore the target boundary by dragging the controller with the trigger pressed. As they drag the red point, we adjust the density threshold accordingly (Fig. 2(d, e)) and compute a new selection volume (Fig. 2(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, parallel implementation of Marching Cubes. We assign a dedicated thread to each voxel, resulting in smooth threshold modification and efficient triangle generation.

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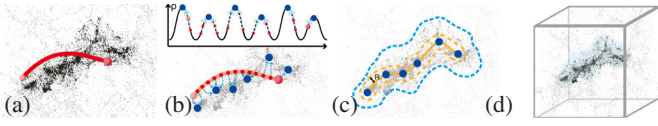
Fig. 3: MeTABrush: (a) the user draws a 3D stroke (red); (b) the stroke points (red) are extracted; by following the direction of gradient (blue arrow), we identify the local maximum points (blue); (c) we construct a tunnel-like volume (yellow dotted region) based on the MaxLine with the radius R; V_{init}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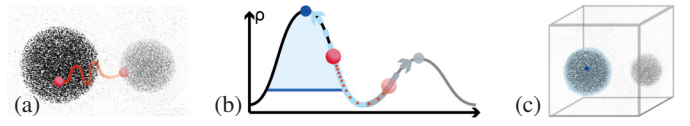

Fig. 4: MeTAPaint: (a) the user draws a 3D stroke (red) near the black cluster, some parts of the input are located near the gray one; (b) the stroke is split into multiple points (red), which flow towards local density maxima (blue and gray points) along the direction of the gradient; (c) the black cluster is selected since it receives most seeds.

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. 13(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(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(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:

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

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(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. 4.5. We thus identify the set of the potential target particles J, which we indicate by the blue dotted line in Fig. 3(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 jth particle (Appx. B 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:

$$V_{init} = \{\mathbf{r} \mid \mathbf{r} \in B, \exists j \in J, \|\mathbf{r}^{(r;j)}\| \leq 1\}, \quad (2)$$

where $\mathbf{r}_k^{(r;j)} = (\mathbf{r}_k - \mathbf{r}_k^{(j)})/\ell_k^{(j)}$ along the kth 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 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.

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. 4(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 ([63, 64]). We then calculate the initial threshold ρ_0 as the arithmetic mean of the grid-nodes' density $\mathbf{r}^{(n)}$ within T, given by

$$\rho_0 = \frac{1}{N_T} \sum_{n=1}^{N_T} \rho(\mathbf{r}^{(n)}), \quad (3)$$

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 destinations $\{\mathbf{r}^{(m_0)}, \mathbf{r}^{(m_1)}, ..., \mathbf{r}^{(m_t)}\}$ (Fig. 4(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. 4(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.

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

$$\nabla\rho(\mathbf{r}) = 0 \text{ and } \lambda_1 < 0, \quad (4)$$

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

$$P : t \in \mathbb{R}^+ \mapsto \mathbb{R}^3, \quad P(0) = \mathbf{r}^{(s)}, \quad P'(t) = \nabla\rho(P(t)). \quad (5)$$

We take $\mathbf{r}^{(m)}$ as the destination of the path P, that is, $\mathbf{r}^{(m)} = dest(P) = \lim_{t\to\infty} P(t)$. According to Morse theory [24], 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 Appx. B. Finally, we obtain the local maximum point $\mathbf{r}^m = dest(P)$ 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. We report on details on the system's performance in Table 1 in Appx. C.
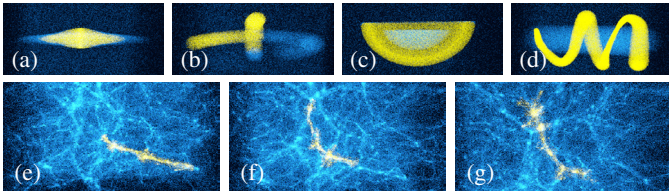
Fig. 5: Datasets we used in our study: (a) *Disk*, (b) *Rings*, (c) *Shell*, (d) *Strings*, and (e–g) *Filaments*.

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 [36], 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 result prediction at osf.io/dvj9n.

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 [9], a PC-based VR head-mounted display (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. 5) have target (yellow) and interfering particles (blue). We designed or selected these datasets to have different features that made the selection of targets challenging:

Disk: This dataset (Fig. 5a) 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. 5b) 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. 5c) 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. 5d) 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 [51] (Fig. 5e) 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. 5(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. 5a), *Rings* (Fig. 5b), *Shell* (Fig. 5c), and *Strings* (Fig. 5d), 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. 5e) 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. [63, 64] 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 ≈ 90 minutes. Following each technique, we requested that the participants evaluate their workload and fatigue levels using NASA's Task Load Index (TLX) [25]. 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 [3, 15, 18, 19], and APA recommends alternative approaches [56], we report results using estimation techniques with effect sizes and confidence intervals rather than *p*-value statistics.

Accuracy. Similar to Yu et al. [63, 64], 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 $F1 = 2 \cdot (P \cdot R)/(P + R)$, 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:

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

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.

Fig. 6: The predicted results are determined by analyzing the data features and the principle of the selection methods: red (relatively good performance), blue (relatively poor performance).

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 predictions (Fig. 6) based on each technique's principles. Based on these pre-registered (osf.io/dvj9n) predictions we formulated the following hypotheses:

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 characteristics. In general, for filament-like datasets, participants would prefer MeTABrush and Baseline, while for individual shapes they would prefer MeTAPoint and MeTAPaint.

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 in Appx. D.

Table 2 in Appx. D displays the average task completion times and two accuracy scores for each dataset and technique. For F1, 1 indicates perfect performance, while 0 represents the worst possible result. For MCC, a score of 1 indicates perfect performance, while -1 represents the worst performance. Mean completion times and 95% confidence intervals for each technique are shown in Fig. 7, while we provide the figure for the accuracy scores for each dataset in Appx. D. 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 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 MeTA-Paint 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 MeTA-Paint were indeed faster than the other two methods, with a substantial
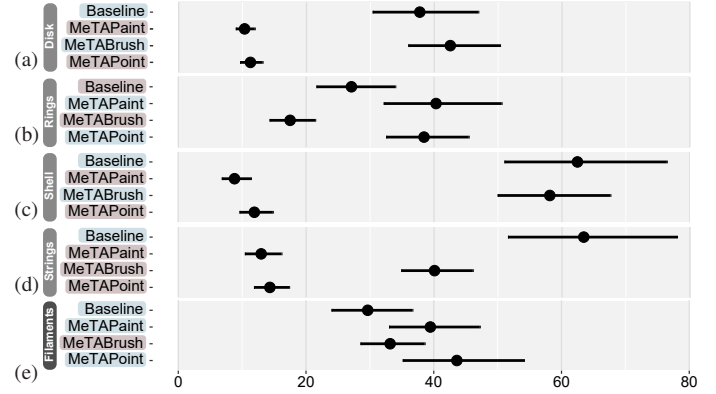

Fig. 7: The geometric mean completion times in seconds for each selection technique in T1 to T5; (a): *Disk*, (b): *Rings*, (c): *Shell*, (d): *Strings*, (e): *Filaments*. Error bars show 95% confidence intervals (CIs).
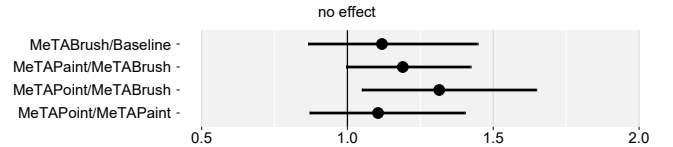

Fig. 8: Pairwise completion time ratio in T5. Error bars: 95% CIs.

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 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.

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. 9) compared to the other methods. Both Baseline and MeTABrush techniques had a shorter mean completion time than MeTAPoint and MeTAPaint (Fig. 7(e)), with a small difference. We further provide pairwise ratios of completion times of the MeTACAST techniques in Fig. 8. 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
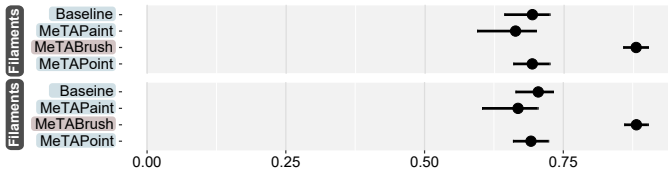
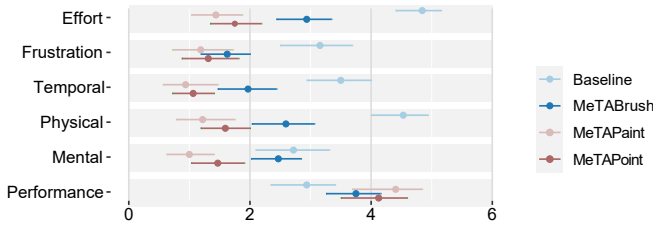Fig. 9: The F1 (top) and MCC (bottom) for T5. Error bars: 95% CIs.

Fig. 10: User workload and performance, T1–T4. Error bars: 95% CIs.
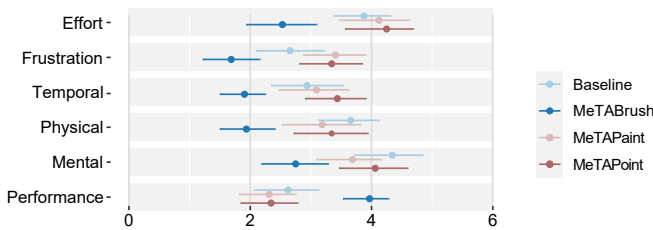

Fig. 11: User workload and performance, T5. Error bars: 95% CIs.

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**.

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. 10. These findings align with the participants' overall completion times, which indicates that their subjective experiences match their actual performance for these methods. However, for the more complex T5 we saw that MeTABrush was the most effective method, as it led to 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.

Preference. After analyzing the techniques' rank across all datasets (T1 to T5) shown in Appx. D, 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 two groups. We provide and discuss the comparison results in Appx. E.

6 DISCUSSION

There are several insightful discussion points that we can derive from our experimental results, as we describe next. We also examine MeTA-CAST within the broader context of existing state-of-the-art selection methods, to derive general design guidelines for spatial selection.

6.1 Selection scenarios

Different selection strategies showed clear advantages in dealing with different data characteristics and situations. When choosing a data selection technique, therefore, we need to consider various 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. MeTA-Point 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 specific 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.

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 Table 4 in Appx. F) 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) [47], input strategies and degree of freedom of the input [2], and the users' level of control during the selection process [7]. 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 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 [63], SpaceCast [64], LassoNet [12], Hybrid AR selection [48], and Tangible brush [7]). A potential approach for defining/extending the selection range in 3D is through extrusion using mobile devices, as in Hybrid AR selection [48] and Tangible Brush [7]. 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 [36], Fiducial-Based Tangible [22], Neuron Tracing [42], 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 [13], 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 [7, 13, 22, 36, 43, 48] 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 ([12, 44, 58, 63, 64]) and 3D space (MeTACAST, [29, 40, 42]), 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 [44], TraceCast [64]) or a single click (MeTAPoint, PointCast [64], WYSIWYP [58]). 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 [12]. Yet there are two powerful 3D brushing methods do not require a selection range but still precisely select the intended partial target: MeTABrush and Neuron Tracing [42].

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. 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-based methods 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.

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 what is considered key information or what should attract a user's 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 the users and how they behave when encountering key information. To accomplish this part, think-aloud studies are effective to collect user behavior and their 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.

ACKNOWLEDGMENTS

SUPPLEMENTAL MATERIAL POINTERS

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

IMAGES/GRAPHS/PLOTS/TABLES/DATA LICENSE/COPYRIGHT

REFERENCES

[1] F. Argelaguet and C. Andujar. Efficient 3D pointing selection in cluttered virtual environments. *IEEE Comput Graph Appl*, 29(6):34–43, 2009. doi: 10.1109/MCG.2009.117

[2] F. Argelaguet and C. Andujar. A survey of 3D object selection techniques for virtual environments. *Comput Graph*, 37(3):121–136, 2013. doi: 10.1016/j.cag.2012.12.003

[3] T. Baguley. Standardized or simple effect size: What should be reported? *Br J Psychol*, 100(3):603–617, Aug. 2009. doi: 10.1348/000712608X377117

[4] M. Baloup, T. Pietrzak, and G. Casiez. Raycursor: A 3D pointing facil-itation technique based on raycasting. In *Proc. CHI*, pp. 101:1–101:12. ACM, New York, 2019. doi: 10.1145/3290605.3300331

[5] J. Bechtold. The Lyman-Alpha forest near 34 quasi-stellar objects with $z > 2.6$. *Astrophys J Suppl*, 91:1–78, 1994. doi: 10.1086/191937

[6] L. Besançon, A. Ynnerman, D. F. Keefe, L. Yu, and T. Isenberg. The state of the art of spatial interfaces for 3D visualization. *Computer Graphics Forum*, 40(1):293–326, 2021. doi: 10.1111/cgf.14189

[7] 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: 10.1111/cgf.13710

[8] J. Bond, L. Kofman, and D. Pogosyan. How filaments of galaxies are woven into the cosmic web. *Nature*, 380(6575):603–606, 1996. doi: 10.1038/380603a0

[9] R. Brown. Valve index. Web site: `vr-compare.com/headset/valveindex`. Last accessed: March 2023.

[10] 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: 10.1109/IMMERSIVE.2016.7932377

[11] 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: 10.1109/TVCG.2009.112

[12] 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: 10.1109/TVCG.2019.2934332

[13] M. Cordeil, B. Bach, A. Cunningham, R. 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. 486:1–486:12. ACM, New York, 2020. doi: 10.1145/3313831.3376613

[14] 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: 10.1109/PACIFICVIS.2017.8031578

[15] G. Cumming. The new statistics: Why and how. *Psychol Sci*, 25(1):7–29, Jan. 2014. doi: 10.1177/0956797613504966

[16] 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: 10.1109/TVCG.2020.3023336

[17] G. de Haan, M. Koutek, and F. H. Post. IntenSelect: Using dynamic object rating for assisting 3D object selection. In *Proc. EGVE*, pp. 201–209. EG Assoc., Goslar, 2005. doi: 10.2312/EGVE/IPT_EGVE2005/201-209

[18] 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: 10.1007/978-3-319-26633-6_13

[19] P. Dragicevic, F. Chevalier, and S. Huot. Running an HCI experiment in multiple parallel universes. In *CHI Extended Abstracts*, pp. 607–618. ACM, New York, 2014. doi: 10.1145/2559206.2578881

[20] B. Ferdosi, H. Buddelmeijer, S. Trager, M. Wilkinson, and J. Roerdink. Comparison of density estimation methods for astronomical. *Astron Astro-phys*, 531:A114:1–A114:16, 2011. doi: 10.1051/0004-6361/201116878

[21] A. Franzluebbers, C. Li, A. Paterson, and K. Johnsen. Virtual reality point cloud annotation. In *Proc. SUI*, pp. 14:1–14:11. ACM, New York, 2022. doi: 10.1145/3565970.3567696

[22] S. R. Gomez, R. Jianu, and D. H. Laidlaw. A fiducial-based tangible user interface for white matter tractography. In *Proc. ISVC*, vol. 2, pp. 373–381. Springer, Berlin, 2010. doi: 10.1007/978-3-642-17274-8_37

[23] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3D volumetric displays. In *Proc. UIST*, pp. 3–12. ACM, New York, 2006. doi: 10.1145/1166253.1166257

[24] M. Guest. Morse theory in the 1990's. arXiv preprint math/0104155, 2001. doi: 10.48550/arXiv.math/0104155

[25] S. Hart. Nasa-task load index (NASA-TLX); 20 years later. *Proc Hum Factors Ergon Soc Annu Meet*, 50(9):904–908, 2006. doi: 10.1177/154193120605000909

[26] 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: 10.1109/VR.2009.4811041

[27] 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: 10.20380/GI2021.33

[28] 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: 10.1109/TVCG.2018.2865191

[29] 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: 10.1109/TVCG.2013.121

[30] D. F. Keefe, R. C. Zeleznik, and D. H. Laidlaw. Tech-note: Dynamic dragging for input of 3D trajectories. *Proc. 3DUI*, pp. 51–54, 2008. doi: 10.1109/3DUI.2008.4476591

[31] 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: 10.1007/978-3-642-03655-2_73

[32] 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: 10.1109/3DUI.2011.5759219

[33] 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: 10.1109/TVCG.2019.2934395

[34] S. Lee, J. Seo, G. J. Kim, and C.-M. Park. Evaluation of pointing tech-niques for ray casting selection in virtual environments. In *Proc. SPIE*, vol. 4756, pp. 38–44. SPIE, Bellingham, 2003. doi: 10.1117/12.497665

[35] 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: 10.1145/37402.37422

[36] 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: 10.1109/3DUI.2014.6798885

[37] J. F. Lucas. Design and evaluation of 3D multiple object selection tech-niques. Master's thesis, Virginia Polytechnic Institute and State University, USA, 2005. hdl: 10919/31769.

[38] J. F. Lucas, D. A. Bowman, J. Chen, and C. A. Wingrave. Design and eval-uation of 3D multiple object selection techniques. Report, Virginia Poly-technic Institute and State University, USA, 2005. url: `researchgate.net/publication/228764561`.

[39] 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: 10.1109/WPC.2001.921711

[40] 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: 10.1007/11907350_56

[41] 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: 10.1109/VR55154.2023.00061

[42] 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: 10.1109/TVCG.2020.3030363

[43] R. A. Montano-Murillo, C. Nguyen, R. H. Kazi, S. Subramanian, S. Di-Verdi, 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: 10.1109/VR46266.2020.00023

[44] S. Owada, F. Nielsen, and T. Igarashi. Volume catcher. In *Proc. I3D*, pp. 111–116. ACM, New York, 2005. doi: 10.1145/1053427.1053445

[45] 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: 10.1145/253284.253303

[46] 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: 10.1109/VRW55335.2022.00307

[47] 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: 10.1006/jvlc.1998.0112

[48] 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: 10.1111/cgf.14550

[49] 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: 10.1109/VL.1996.545307

[50] V. Springel, J. Wang, M. Vogelsberger, A. Ludlow, A. Jenkins, A. Helmi, J. F. Navarro, C. S. Frenk, and S. D. White. Aquarius project: The subhaloes of galactic haloes. *Mon Not R Astron Soc*, 391(4):1685–1711, 2008. doi: 10.1111/j.1365-2966.2008.14066.x

[51] 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: 10.1038/nature03597

[52] R. Stenholt. Efficient selection of multiple objects on a large scale. In *Proc. VRST*, pp. 105–112. ACM, New York, 2012. doi: 10.1145/2407336.2407357

[53] 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: 10.1145/223904.223938

[54] 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: 10.1007/978-3-540-78640-5_82

[55] R. van Teylingen, W. Ribarsky, and C. van der Mast. Virtual data visualizer. *IEEE Trans Vis Comput Graph*, 3(1):65–74, 1997. doi: 10.1109/2945.582350

[56] G. R. VandenBos, ed. *Publication Manual of the American Psychological Association*. APA, Washington, DC, 6th ed., 2009. url: apastyle.org/manual.

[57] 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*, pp. 283:1–283:14. ACM, New York, 2023. doi: 10.1145/3544548.3581042

[58] 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: 10.1109/TVCG.2012.292

[59] 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: 10.1109/INFVIS.1996.559216

[60] 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: 10.1109/VR.2005.1492770

[61] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for *soft* objects. *Vis Comput*, 2(4):227–234, 1986. doi: 10.1007/BF01900346

[62] 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: 10.1145/2366145.2366161

[63] 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: 10.1109/TVCG.2012.217

[64] 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: 10.1109/TVCG.2015.2467202

[65] 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: 10.1109/ISMAR-Adjunct57072.2022.00031

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

Appendix

In this appendix we provide additional tables, plots, and charts that show data beyond the material that we could include in the main paper due to space limitations or because it was not essential for explaining our approach. For access to the source code, application, datasets, and analysis scripts used in this work, please refer to osf.io/dvj9n and to github.com/LixiangZhao98/MeTACAST.

A ADDITIONAL ILLUSTRATIONS FROM THE ELICITATION STUDY

Fig. 12 shows a visual representation of the visual presentation size concepts that we used in the study in Sec. 3, while Fig. 13 shows the four datasets we used in this experiment.
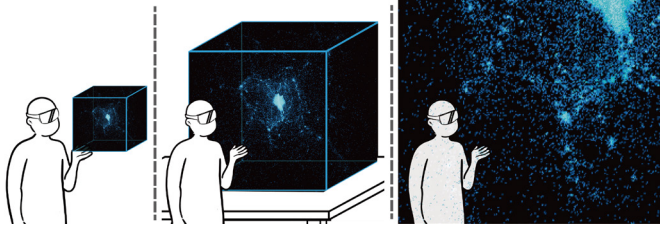
Fig. 12: Three visualization sizes: (a) hand, (b) table, and (c) room size.
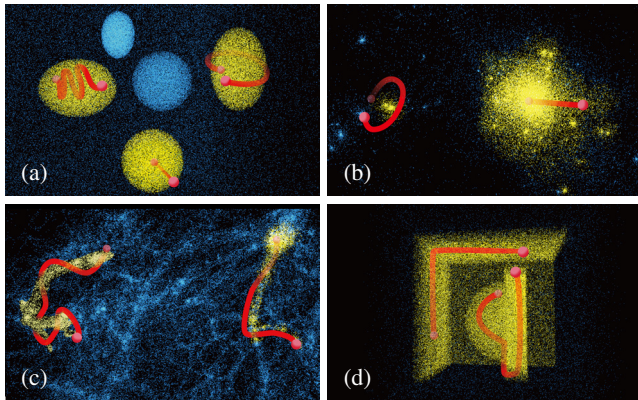
Fig. 13: User input trajectories in selecting four point cloud datasets employed in the think-aloud study: (a) *Clusters*; (b) *N-body Simulation*; (c) *Filament*; (d) *Complex geometries*.

B DENSITY ESTIMATION

We used the same density estimation method as Yu et al. [63, 64]. 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 $100 \times 100 \times 100$ grid. For each direction $k = x, y, z$, we define the smoothing length

$$\ell_k = 2(P_k^{(80)} - P_k^{(20)})/\log N \quad (6)$$

where N is the particle count in the box B and $P_j^{(q)}$ is coordinate k's q-th percentile value. For the ith 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 [20], given by a

$$\rho(\mathbf{r}^{(i)}) = \frac{15}{8\pi N} \sum_j \frac{1}{\ell_x^{(j)} \ell_y^{(j)} \ell_z^{(j)}} E(\|\mathbf{r}^{(j;i)}\|), \quad (7)$$

with

$$\mathbf{r}_k^{(j;i)} = (\mathbf{r}_k^{(j)} - \mathbf{r}_k^{(i)})/\ell_k^{(j)} \quad (8)$$

Fig. 14: The overall (T1 to T4) geometric means of completion time for each selection technique. Error bars: 95% CIs.
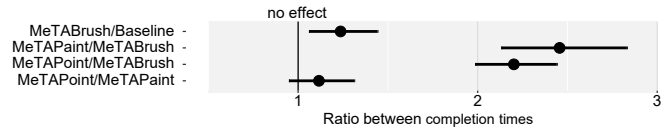
Fig. 15: The overall (T1 to T4) pairwise ratio of completion time. Error bars show 95% confidence intervals (CIs).

where $\mathbf{r}^{(j)}$ is the position of the jth particle and $\ell_k^{(j)}$ are the smoothing lengths of the jth particle along the kth direction ($k = x, y, z$) generated from the pilot density calculation [63, 64]. The Epanechnikov kernel $E(x)$ is given by

$$E(x) = \begin{cases} 1 - x^2, & |x| < 1, \\ 0, & |x| \geq 1. \end{cases} \quad (9)$$

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

C SYSTEM PERFORMANCE

We have tested the performance of the MeTACAST with Unity on Intel Core™ i9, GeForce RTX3090. Table 1 shows the performance of the MeTACAST, including point cloud dataset sizes, the performance of density estimation, selection algorithm and geometry construction.

Table 1: MeTACAST performance. Times are in seconds.

Dataset	Particle Size	Density Estimation (offline)	Selection Algorithm	Marching Cubes
Fig. 1(left)	76k	0.38	0.06	0.005
Fig. 1(mid)	442k	3.47	1.69	0.008
Fig. 1(right)	146k	0.80	0.17	0.009

D ADDITIONAL RESULT DATA FROM THE STUDY

The following tables and graphs show additional results from the experiment we described in Sec. 5. Table 2 shows all mean task completion times, accuracy scores, and their corresponding 95% confidence intervals for T1 to T5. Table 3 shows the overall mean task completion times, accuracy scores, and their corresponding 95% confidence intervals. Fig. 14 shows the overall geometric means of completion time for each technique and Fig. 15 shows the pairwise ratio of completion times for T1–T4. Fig. 16 and 17 show the overall accuracy scores for T1–T4. Fig. 18–27 show the accuracy results per dataset. Fig. 28 shows the rank across all datasets (*Disk*, *Rings*, *Shell*, *Strings*, *Filaments*): MeTAPaint (Pa), MeTAPoint (Po), MeTABrush (Br), Baseline (Ba).

Table 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: 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]
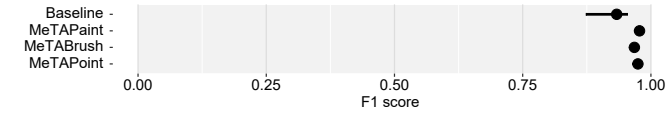
Fig. 16: The overall (T1 to T4) F1 score. Error bars: 95% CIs.
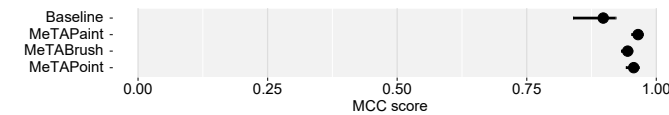
Fig. 17: The overall (T1 to T4) MCC score. Error bars: 95% CIs.

Fig. 18: The F1 score for T1 (*Disk*). Error bars: 95% CIs.
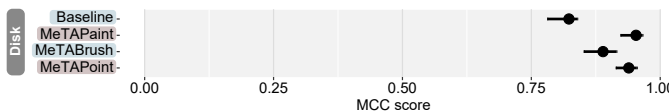
Fig. 19: The MCC score for T1 (*Disk*). Error bars: 95% CIs.

Fig. 20: The F1 score for T2 (*Rings*). Error bars: 95% CIs.
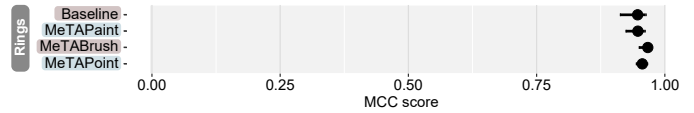
Fig. 21: The MCC score for T2 (*Rings*). Error bars: 95% CIs.
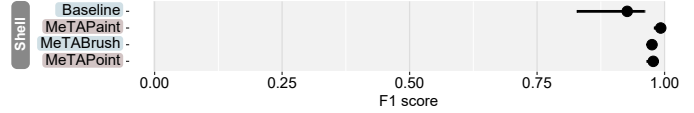
Fig. 22: The F1 score for T3 (*Shell*). Error bars: 95% CIs.
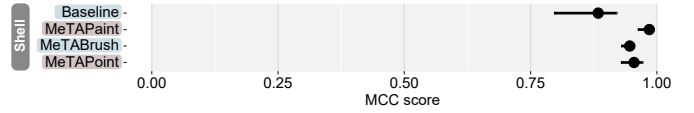

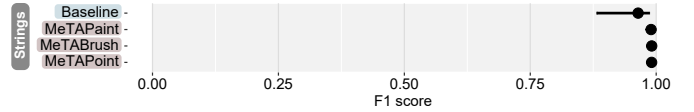Fig. 23: The MCC score for T3 (*Shell*). Error bars: 95% CIs.

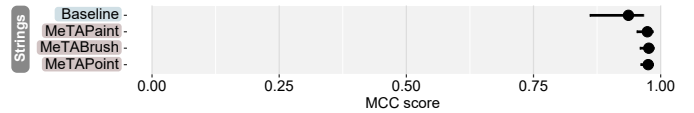Fig. 24: The F1 score for T4 (*Strings*). Error bars: 95% CIs.

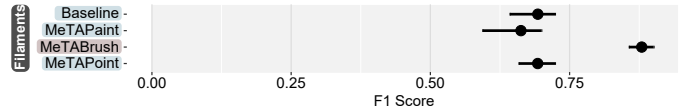Fig. 25: The MCC score for T4 (*Strings*). Error bars: 95% CIs.

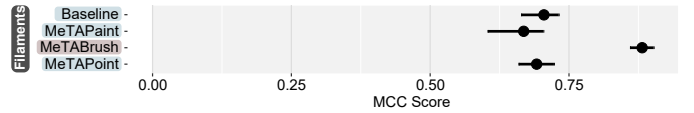Fig. 26: The F1 score for T5 (*Filaments*). Error bars: 95% CIs.

Fig. 27: The MCC score for T5 (*Filaments*). Error bars: 95% CIs.
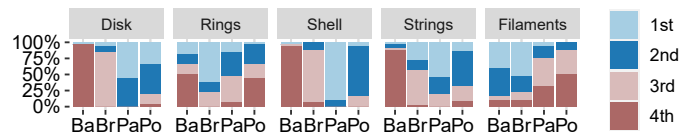

Fig. 28: User preference for techniques with each dataset.
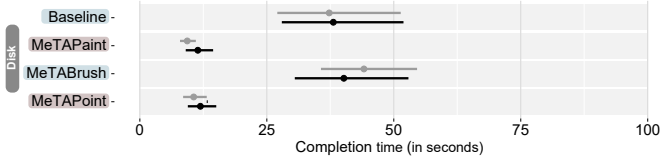
Fig. 29: The geometric mean completion times in seconds for T1 (*Disk*). VR users (black), non-experts (gray). Error bars: 95% CIs.
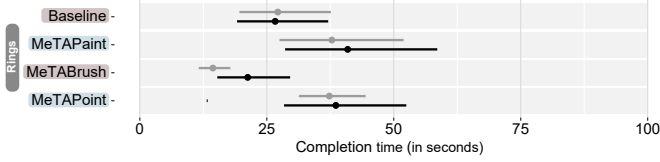

Fig. 30: The geometric mean completion times in seconds for T2 (*Rings*). VR users (black), non-experts (gray). Error bars: 95% CIs.
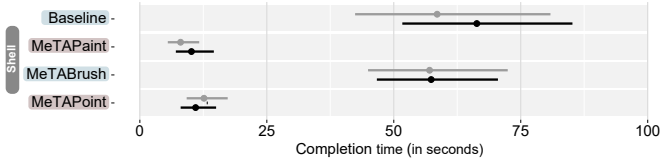

Fig. 31: The geometric mean completion times in seconds for T3 (*Shell*). VR users (black), non-experts (gray). Error bars: 95% CIs.
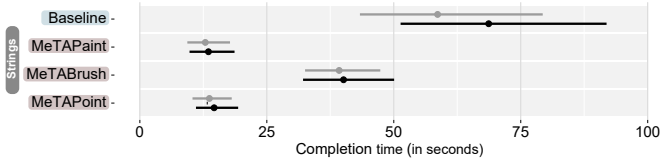

Fig. 32: The geometric mean completion times in seconds for T4 (*Strings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

E COMPARISON OF TASK PERFORMANCE BETWEEN TWO USER GROUPS

We further conducted a separate analysis of task performance, taking into consideration 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. 6, VR users were slower than non-experts. On the other hand, VR users were faster in the most challenging task, *Filaments*. We assume that 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 location and structure. This finding also indicates that task 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. 29–33 show the geometric means of completion times for each technique per dataset, and Fig. 34–43 show the accuracy results per dataset, comparing the two user groups.

F COMPARISON OF EXISTING SPATIAL SELECTION TECHNIQUES

Table 4 shows an overview of spatial selection techniques for 3D data focusing on data characteristics, selection requirements (such as the selection basis and target shape), selection strategies (such as metaphor and selection strategy), and users' level of control (including precision demand and post-adjustment), as described in Sec. 6.2.
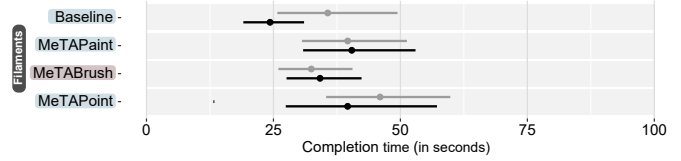

Fig. 33: The geometric mean completion times in seconds for T5 (*Filaments*). VR users (black), non-experts (gray). Error bars: 95% CIs.
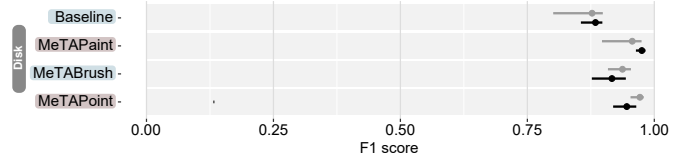

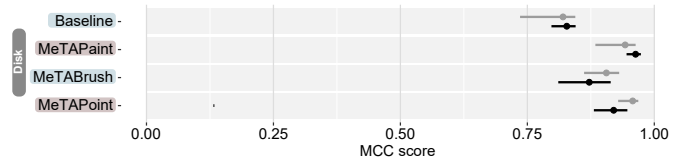Fig. 34: The F1 score for T1 (*Disk*). VR users (black), non-experts (gray). Error bars: 95% CIs.

Fig. 35: The MCC score for T1 (*Disk*). VR users (black), non-experts (gray). Error bars: 95% CIs.
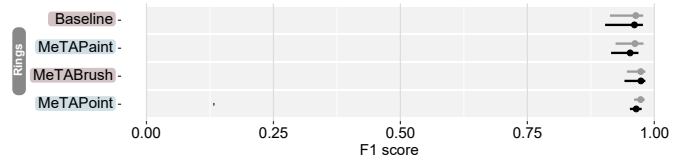

Fig. 36: The F1 score for T2 (*Rings*). VR users (black), non-experts (gray). Error bars: 95% CIs.
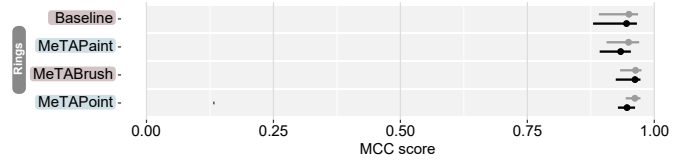

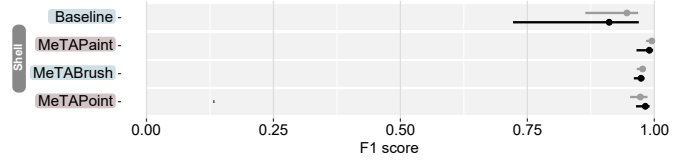Fig. 37: The MCC score for T2 (*Rings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

Fig. 38: The F1 score for T3 (*Shell*). VR users (black), non-experts (gray). Error bars: 95% CIs.
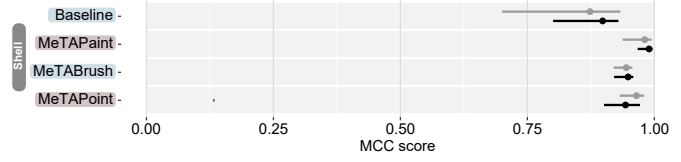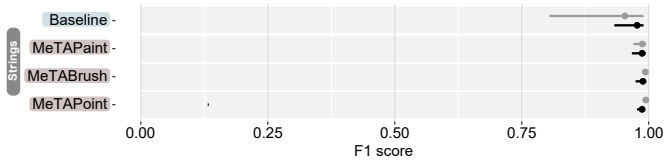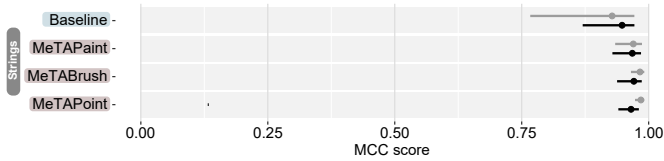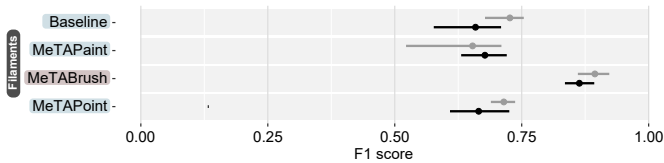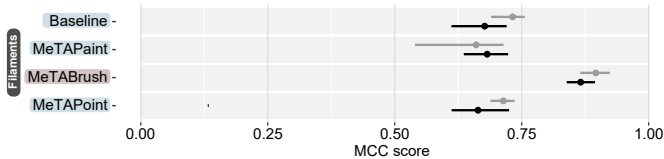

Fig. 39: The MCC score for T3 (*Shell*). VR users (black), non-experts (gray). Error bars: 95% CIs.

IMAGES/GRAPHS/PLOTS/TABLES/DATA LICENSE/COPYRIGHT

Table 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 [37, 38]	ROI	region	whole / partial	semi-autom., 2D	high	lasso	no control
Cloudlasso [63]	point cloud / scalar	value	whole / partial	semi-autom., 2D	medium	lasso	threshold adjust
SpaceCAST [64]	point cloud / scalar	value	whole / partial	semi-autom., 2D	medium	lasso	threshold adjust
Volume Catcher [44]	scalar	value	whole	semi-autom., 2D	low	lasso	no control
TraceCAST [64]	point cloud / scalar	value	whole	semi-autom., 2D	low	lasso	threshold adjust
PointCAST [64]	point cloud / scalar	value	whole	semi-autom., 2D	low	raycasting	threshold adjust
LassoNet [12]	point cloud	deep learning	whole / partial	semi-autom., 2D	low	lasso	no control
WYSIWYP [58]	scalar	value	none	semi-autom., 2D	low	raycasting	no need to control
Slicing Volume [43]	point cloud	region	partial	manual	high	raycasting	region adjust
Hybrid AR selection [48]	point cloud / scalar	region	partial	manual	high	lasso + extrusion	no control
TangibleBrush [7]	point cloud / scalar	region	partial	manual	high	lasso + extrusion	no control
Touching the cloud [36]	point cloud	region	partial	manual	high	brush	region adjust
Embodied Axes [13]	ROI	region	whole / partial	manual	high	tangible (box)	region adjust
Fiducial-Based Tangible [22]	tensor	region	partial	manual	high	brush	no control
Live-Wire [40]	scalar	value	partial	semi-autom., 3D	medium	seed points	no control
Lightweight Tangible [29]	vector	value	whole	semi-autom., 3D	medium	tangible (orien.)	no control
Neuron Tracing [42]	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

Fig. 40: The F1 score for T4 (*Strings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

Fig. 41: The MCC score for T4 (*Strings*). VR users (black), non-experts (gray). Error bars: 95% CIs.

Fig. 42: The F1 score for T5 (*Filaments*). VR users (black), non-experts (gray). Error bars: 95% CIs.

Fig. 43: The MCC score for T5 (*Filaments*). VR users (black), non-experts (gray). Error bars: 95% CIs.

the permission to be used here. We also make them available under the Creative Commons Attribution 4.0 International (ⓒⓘ CC BY 4.0) license and share them at osf.io/dvj9n.