Evaluating Node Selection Techniques for Network Visualizations in Virtual Reality

Lucas Joos
lucas.joos@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Uzay Durdu
uzay.durdu@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Jonathan Wieland
jonathan.wieland@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Harald Reiterer
harald.reiterer@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Daniel A. Keim
keim@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Johannes Fuchs
johannes.fuchs@uni-konstanz.de
University of Konstanz
Konstanz, Germany

Maximilian T. Fischer
max.fischer@uni-konstanz.de
University of Konstanz
Konstanz, Germany

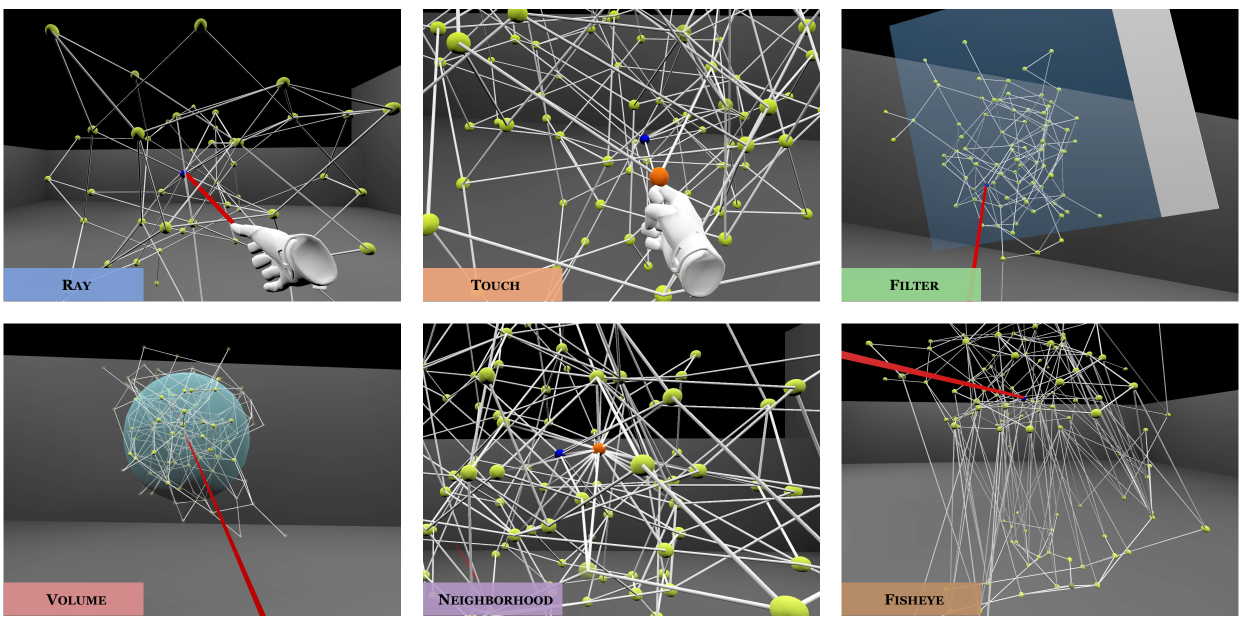

Figure 1: We present and evaluate six interaction techniques for selecting network nodes in a VR environment.

ABSTRACT

The visual analysis of networks is crucial for domain experts to understand their structure, investigate attributes, and formulate new hypotheses. Effective visual exploration relies heavily on interaction, particularly the selection of individual nodes. While node selection in 2D environments is relatively straightforward, immersive 3D environments like Virtual Reality (VR) introduce additional challenges such as clutter, occlusion, and depth perception, complicating node selection. State-of-the-art VR network analysis systems predominantly utilize a ray-based selection method controlled via VR controllers. Although effective for small and sparse graphs, this method struggles with larger and denser network visualizations. To address this limitation and enhance node selection in cluttered immersive environments, we present and compare six distinct node selection techniques through a user study involving 18 participants. Our findings reveal significant differences in the efficiency, physical effort, and user preference of these techniques, particularly in relation to graph complexity. Notably, the filter plane

SUI '24, October 07–08, 2024, Trier, Germany

ACM ISBN 979-8-4007-1088-9/24/10
https://doi.org/10.1145/3677386.3682102

metaphor emerged as the superior method for selecting nodes in dense graphs. These insights advance the field of effective network exploration in immersive environments, and our validations provide a foundation for future research on general object manipulation in virtual 3D spaces. Our work informs the design of more efficient and user-friendly VR tools, ultimately enhancing the usability and effectiveness of immersive network analysis systems.

CCS CONCEPTS

• **Human-centered computing → Virtual reality**; **User studies**; **Information visualization**.

KEYWORDS

Virtual reality, network analysis, selection, interaction, evaluation

ACM Reference Format:
Lucas Joos, Uzay Durdu, Jonathan Wieland, Harald Reiterer, Daniel A. Keim, Johannes Fuchs, and Maximilian T. Fischer. 2024. Evaluating Node Selection Techniques for Network Visualizations in Virtual Reality. In *ACM Symposium on Spatial User Interaction (SUI '24), October 07–08, 2024, Trier, Germany*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3677386.3682102

1 INTRODUCTION

Visualizing networks to gain insights into structure, node attributes, connection strengths, and other factors has a long tradition and has proven beneficial for various research and application domains [3, 11, 14, 39]. Moreover, the visual analysis of networks, in combination with the domain knowledge of experts, often leads to new hypotheses that can be evaluated using mathematical concepts and statistical tests. For this purpose, adjacency matrices and node-link diagrams are the predominant visual representations expressing the network structures and attributes [34]. These visualization types are often created and explored using classical 2D setups, i.e., a 2D monitor with mouse and keyboard interaction, for which multiple popular tools and libraries, such as Cytoscape [40], Gephi [4], or NetworkX [18] exist. However, with the rising field of Immersive Analytics (IA), 3D network representations interactively explorable in immersive environments, such as Virtual Reality (VR), Augmented Reality (AR), or CAVEs become increasingly popular, showing benefits compared to 2D setups in multiple user evaluations [16, 24, 29, 53]. Especially node-link diagrams tend to benefit from the third dimension, stereoscopic vision, and the advanced interaction capabilities of immersive setups [20].

The effective analysis of data visualizations requires multiple interaction capabilities, for instance, *navigation*, *filtering*, *annotating*, and *arranging* visual elements [7]. These high-level interaction concepts typically require the *selection* of elements as an initial, low-level interaction [27]. This is also the case for the exploration of node-link visualizations, where the selection of nodes plays a central role, e.g., for retrieving attribute information, annotating it, removing it, highlighting its neighbors, initiating calculations based on the node, adding edges to other nodes, or revealing the underlying structure if the node represents an aggregated subnetwork [5, 39, 49]. Layout algorithms for 2D node-link diagrams optimize for multiple criteria and usually ensure that nodes are drawn without overlap. Therefore, selecting individual nodes with the mouse tends to be rather easy. In contrast, 3D node-link representations typically lead to node occlusion and clutter, depending on the user perspective [19]. Therefore, the selection of nodes in a VR setting–often based on pointing a visual ray with a VR controller at a desired node while pressing a button–can be cumbersome and frustrating, especially for larger graphs and when overlap and clutter prevent precise selections.

We highlight the three **major contributions** of our research:

- The implementation of six controller-based VR selection concepts adapted to the use case of network structures.
- A detailed user study with 18 participants providing valuable insights and validation regarding the *efficiency*, *physical effort*, *user preference*, and individual *user feedback*.
- A comprehensive analysis and discussion of the study results, opening up new research questions and informing the design of interaction concepts for similar use cases and applications.

Our research does not only contribute to improved visual network analysis in VR but also validates the potential to enhance user interaction and efficiency in a variety of fields that rely on immersive data visualization and IA, including, but not limited to, bioinformatics, social network analysis, and educational tools, providing insights for data and object interaction in immersive environments.

2 RELATED WORK

We review related research, focusing on object selection in virtual 3D environments, especially when the space is cluttered with a high level of occlusion (Section 2.1), and research targeting the selection of network nodes in existing 2D and 3D applications (Section 2.2).

2.1 Object Selection in 3D

Selecting objects in cluttered virtual 3D environments presents significant challenges due to occlusion and limited accessibility. Over the last years, VR applications started to incorporate head gaze [2], eye gaze [37], or speech recognition [33] to address these challenges. While the majority of environments still rely on handheld controllers or camera-tracked hands for interaction, modern HMD technology allows the combination of gaze and hands [35]. Controllers or tracked hands are often represented virtually, aiding users in understanding the mapping between physical and virtual space, anticipating actions, and imitating a physical touch.

Ray-Based Interaction — One common technique is using a laser ray controlled by hand-held devices to select objects through pointing [1]. Similarly, the users' eye gaze can be incorporated as a selection ray pointing at the first intersected or the closest item (snapping) [35], a technique that is also used for hand-based pointing [28]. Ray-based selection is particularly useful for selecting objects beyond the user's physical reach and is usually combined with pressing controller buttons or hand gestures to trigger actions. However, in cluttered spaces, accurately targeting small and distant objects with a ray (or direct touch) can be challenging [50].

Improving Selection Accuracy in Cluttered Spaces — Several methods have been presented to facilitate selection when clutter and occlusion become a challenge [1]. One option is to **rearrange elements** in a grid or other non-occluding structures [8, 21, 41, 56]. In a second step, users apply ray-based interaction to select the target element. While effective and guaranteeing an occlusion-free selection of the elements, this approach can disrupt the original spatial relationships between elements. Another approach is to **filter out objects**, for instance, based on the distance to the user, incorporating a virtual cursor, or applying a filter volume to hide contained objects [8, 9, 41, 56]. Moreover, **diambiguating** close objects by slightly changing the positions of elements around the laser ray to avoid occlusion [17, 56] or applying a local force-directed layout [21] can facilitate the selection. Furthermore, dynamic **selection volumes** that adjust in size can help to ensure that only one element is selected [48]. This approach has been extended to support multiple elements, mapping them to individual fingers for selection [10]. Furthermore, Montano-Murillo et al. [31] proposed using a 3D volume to select a slice and project the content to a 2D plane for easier selection. Zhu et al. [58] incorporate a hand-controlled volume and use a control-display gain, dampening the hand movement to allow for a precise selection within the volume. Techniques like **sphere-casting and progressive refinement** allow users to progressively narrow down their selection until the target object is selected [23]. Molina and Vázques [30] allow users to correct their initial selection of atoms in a virtual molecule by hopping to the neighboring objects using the controller touchpad. In addition, various **bimanual techniques** have been proposed, such as controlling aspects of the selection process with a user-controlled plane and a ray intersection method. These methods involve steering each component with different hands and determining the closest object to the intersection of plane and ray [55, 57]. Wu et al. [54] compare point- and volume-based interaction techniques together with (de-) selection mechanisms, finding point-based selection advantageous for unstructured data and volume-based beneficial for structured, grid-like object alignments.

Besides methods targeting accuracy improvement for hand-based selection, interesting approaches have been proposed for **eye gaze interaction**. Sidenmark et al. [43] identify the object to select by comparing the shapes of objects within the gaze cone with the gaze path of a user and selecting the most similar object. Another method by Sidenmark et al. [42] also uses a gaze cone to retrieve candidate objects but applies movements to the candidate objects and matches these with the vergence of the user's eyes to detect the target object. The Cone&Bubble approach is based on cone selection to retrieve a candidate set from which the target node is selected using a bubble cursor [45]. Instead of relying exclusively on eye or hand tracking for selection, the weighted pointer approach calculates the error rate of eye gaze and includes a fallback modality, combining both signals to allow for a more precise selection [44]. Despite these advancements, most research has focused on selecting unstructured elements–mostly spheres or general objects–with no specific application use case. Moreover, previous studies, such as structure-based neighborhood refinement [30], only compared variations rather than comparing the applicability of fundamentally different methods against each other. In our work, we focus on node selection in 3D networks, which come with specific characteristics, such as the position of nodes and their topology, which are essential for the understanding of users and their ability to select desired nodes (e.g., the node with the highest degree). Therefore, some previously presented techniques, such as grid alignment of candidates, which incorrectly alter the structural representation of the underlying data, are not applicable for our application. We address this gap by designing multiple selection techniques tailored to the specific requirements of 3D node-link representations and comparing these fundamentally different selection concepts in a comprehensive user study.

2.2 Network Node Selection in 2D and 3D

The *selection* of elements is a fundamental technique in visual analytics systems and common 2D visual network exploration frameworks. These frameworks typically use two-dimensional graph layout algorithms to avoid node occlusions or respect semantic information encoded in the data. Therefore, common tools support the selection of nodes through multiple techniques, including direct clicking, rectangular or lasso brush, textual search, or the selection from a list panel [4, 40]. When node representations are too small for selection, e.g., in a large network, users can typically change the **zoom level** to interact with them accurately. Apart from navigation-oriented interaction for effective node selection, **focus+context methods** have been introduced. These approaches typically distort the geometric space around the cursor in such a way that nodes with a high distance to the focus move closer together and nodes around the focus move apart, allowing for easier selection in the focus region [13, 32, 38]. To cope with geometric distortions, ignoring the graph topology and changing the structure, topology-preserving methods have been developed [15, 51, 52].

Existing applications for immersive network exploration support node selection by using **ray-based pointing** [12, 36, 46], directly **touching** them in the virtual space [6, 47], placing a three-dimensional **selection volume** [26], or checking node **names in a list** displayed within the virtual environment [36]. While immersive 3D environments support standard techniques, such as scaling or navigation, to facilitate the selection of nodes, more advanced concepts have not been investigated yet. Whereas multiple approaches have been developed to facilitate the selection of network nodes in 2D setups by adapting their positions based on the focus point, none of them have been applied to 3D networks. Currently, node selection in immersive setups is mainly based on direct ray- or hand-based interaction without additional support. We fill this gap by presenting 3D node selection techniques incorporating the graph structure and applying insights from general 3D object selection research.

3 NODE SELECTION CONCEPTS

Many immersive network analysis environments currently rely on generic interaction methods, mostly direct ray-casting, to allow the selection of a node, as shown in Section 2. Advanced methods, such as hiding elements, refining the initial selection, or changing positions, have not yet been applied for this use case. In contrast to generic objects, node-link network representations come with an additional challenge: This type of data visualization is especially used when the topological relation between data entities, i.e., nodes, is crucial and must be perceivable. Therefore, approaches, such as 2D grid alignments, as used in some 3D selection concepts, are not desirable for node-link representations as they would cause users to lose track of the network topology. Moreover, the position of nodes often has a semantic meaning, e.g., geospatial location,

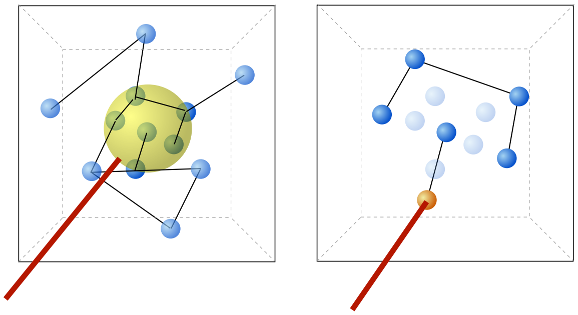

Figure 2: With the ■ Volume technique, users place a selection volume into the target node region (left). After confirmation, only the contained nodes remain visible and can be moved apart using the joystick (right, faded nodes: original positions), and the target node (orange) can be selected.

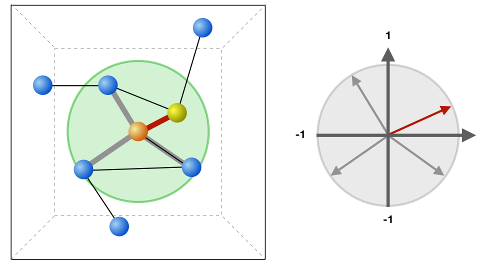

Figure 3: With ■ Neighborhood, a node of a graph (left) is selected (orange), and the nodes within its neighborhood in a fixed radius (green) can be selected using the joystick. The joystick coordinate system (right) is mapped to the node directions within the camera plane.

which would be lost in grid alignments. On the other hand side, the structure of networks offers further opportunities for developing selection techniques tailored to the specifics of this data type.

In the following, we present six interaction concepts for selecting network nodes in a three-dimensional VR environment. Our goal is to find techniques that help users to *efficiently* and *intuitively* select intended nodes without requiring a high level of *physical effort*.

3.1 Ray

As a baseline, we included the ■ Ray technique (see Figure 1a). This approach matches the most common selection technique in current immersive network exploration tools (see Section 2.2). One of the VR controllers is used to control a visual ray emanating from the virtual hand. The ray ends at the first interactable object that is intersected. A color change indicates the currently focused element–in our case, only nodes–, which can be selected by pulling the trigger of the controller. While this approach can be expected to be intuitive and fast, increasing node density, causing occlusion and clutter, may affect the accuracy and efficiency.

3.2 Touch

The 3D tracking and representation of the VR controllers also allow for more direct interaction with the virtual elements through the ■ Touch technique (see Figure 1b). With this technique, users directly touch a node with the virtual, controller-operated hand. Similar to ■ Ray, a touched object is highlighted by color and can be selected by pushing a controller button. While we expect this approach to be intuitive and align well with the immersive nature of the environment, it could require a high level of physical effort and lead to sensitivity issues with closely spaced nodes.

3.3 Filter

As seen in Section 2.1, one technique to facilitate object selection in occluded 3D spaces is to hide occluding elements through a filter technique (see Figure 1c). For instance, a cursor is applied to the visual ray (e.g., a sphere object) and the user moves it along the ray (e.g., by using a joystick) [56]. Objects between the user and the cursor are hidden, while objects behind the cursor remain visible. This approach comes with the downside that moving the ray to focus the intended node can change the visibility of objects. Moreover, there is no clear visual indication of the boundary between visualized and hidden objects. With the ■ Filter method, users can see a transparent, glass-like plane object that can be moved within the environment using the hand or ray. To use the ray to move the plane, users target the gray stripe at the side of the plane, hold the controller trigger, and use the joystick to adjust the depth while the rotation matches the hand rotation. Objects between the user and the plane are hidden, while elements behind stay visible. The blue part of the filter plane allows the ray to pass so that objects behind the plane can be selected. This approach enables users to select occluded objects, clearly visualizes the clipping boundary, and retains node visibility when users move the ray toward the target node. Moreover, it can be adapted to Augmented Reality setups incorporating a physical plane [25]. We expect this technique to provide an intuitive opportunity to remove objects covering relevant elements while placing the filter plane using hand or ray interaction should be possible without taking much time. In addition, the option to move the plane remotely using the ray should reduce the physical demand required for this method.

3.4 Volume

Multiple existing 3D object selection approaches incorporate a virtual selection volume that can often be resized and moved within the virtual space [8, 23, 56]. Typically, the user places this selection volume so that the target object is–among other objects–contained in the volume. Then, an action like pressing a controller button is triggered, and the contained elements are rearranged in a non-occluding way (most commonly, a grid) before the target object can be selected using ray-casting. However, this approach is difficult for node-link structures, where the appearance of node objects is typically highly similar, and the target object is usually determined by its position and role in the topology. Hence, we included the ■ Volume technique (see Figure 1d). With this technique, a user creates a scalable sphere that is attached to the visual ray. By moving the ray, the position of the sphere can be adjusted while its depth can be modified using the controller joystick (see Figure 2 (left)), and users can position the selection volume around the target node and press the controller trigger to fix the selection. All objects that

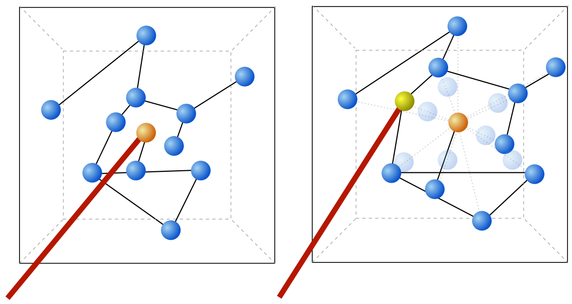

Figure 4: With the ■ Fisheye technique, users select a node (left, orange) close to the target node and continuously apply the disambiguation algorithm by using the joystick. This moves the other nodes closer to the boundary (depending on the distance) and increases the space between the nodes, allowing users to select the target node (right).

are not included in the volume are then rendered semi-transparent, indicating the potential selection set. After fixing the selection, the part of the network not contained within the sphere is hidden and the ray is used for selection. To disambiguate the nodes within the selection volume, users can move the joystick up (or down to undo the effect), which initiates a continuous, force-based change of the node positions (similar to [21]). Thus, nodes slowly drift apart (while the joystick is held upwards), making it easier for users to select the desired node while the network structure is preserved (see Figure 2 (right)). While this method induces some effort, as users need to scale and place a sphere, disambiguate the initial selection, and then select the actual target nodes, we expect it to provide comparably high accuracy, especially with increasing density. The required physical interaction is limited to controlling the ray in combination with button and joystick interaction. Hence, we see potential for this technique, particularly when the degree of occlusion is high and direct ray-casting is not accurate enough.

3.5 Neighborhood

Ray-based selection is often incorporated into VR applications, as it allows direct object selection from a remote position without any additional overhead. However, with increasing density and distance to the objects, the number of undesired selections, where neighbors of the target are selected, tends to increase. To cope with this issue, we included the ■ **Neighborhood** technique (see Figure 1e). The idea of this approach is to combine ray-based selection with the option to correct the selection within the current neighborhood (similar to [30]). Originally, we implemented this method incorporating the network topology. After selecting a node (i.e., a wrong node close to the target node), users could move the controller joystick, highlighting the attached network edge closest to the joystick angle, and press a button to follow this edge, selecting the attached node. However, as there is no guarantee that physically close nodes are connected with an edge, we changed the approach, adding virtual edges between nodes within a certain radius of the currently selected node and allowing users to follow these edges with the joystick. For the mapping, all nodes within the selection neighborhood are projected onto the camera plane with the selection node as the center. Then, each node position corresponds to an angle that can be represented with the joystick (see Figure 3). We expect this technique to be a good compromise for selecting nodes in sparse and dense networks. For sparse ones, users can directly select the target node using the ray without needing the neighborhood correction. For more dense networks, users select the node closest to the target that they can reach and use the joystick to move to the target node. Therefore, we estimate that the ■ **Neighborhood** technique does not require a high level of physical effort without issues regarding efficiency or accuracy.

3.6 Fisheye

Previous work has shown that applying focus+context visualization methods to 2D graphs can help to disambiguate a region of interest while still showing the rest of the network and preserving its topology [15, 51, 52]. To evaluate the applicability of such distortion-based disambiguation techniques, we included the ■ **Fisheye** technique (see Figure 1f and Figure 4). The main idea is that users can select any object close to the target node (if the target is not directly selectable) as a focus point and apply the technique, which increases the node distances around the focus and moves objects to the boundary that are not close to the focus. After selecting a focus node, users can move up the joystick to continuously apply the technique and move it downwards to revert it. Selecting a new focus point resets the graph to the original layout, making sure that the original appearance can always be restored. We implemented this approach analogously to the 2D *graphical fisheye* definition by Wang et al. [52]. The set of positions of the graph nodes is denoted as $P = \{p_1, ..., p_n\}$, where $p_i \in \mathbb{R}^3$. In contrast to a 2D screen, the virtual 3D environment has no natural boundaries. Thus, we use the graph bounding box as a virtual boundary, resulting in six boundary planes $\{B_1, ..., B_6\}$. Given the current focus position p_f, the new position p'_i of node i with the original position, p_i can be calculated by

$$p'_i = p_f + (b_i - p_f)\beta'_i, \text{ with } \beta'_i = \frac{(m+1)\beta_i}{m\beta_i + 1} \text{ and } \beta_i = \frac{||p_i - p_f||}{||b_i - p_f||}$$

The boundary position b_i is determined by casting a ray from p_f through p_i and calculating the intersection with the closest plane in this direction. Hence, each node is moved away from the focus point towards the boundary considering the ratio of the distances to the focus and boundary point, and including a variable factor m, regulating the strength of the fisheye effect. We expect this technique to be mostly beneficial when the target node can not be directly reached by the ray but by nearby nodes. Moreover, it could require some practice, as comparable techniques are rare in visual representation for daily use. We further estimate that the effect will be more helpful in cases where the boundary points for the individual nodes are distributed across multiple different boundary planes instead of a single one, as nodes are then moved apart from each other and not only in a common direction. Nevertheless, we included this technique as a straightforward modification of the 2D version to evaluate the usefulness of fisheye distortion in general for this use case and to determine factors that might be improved in future versions (see Section 5).

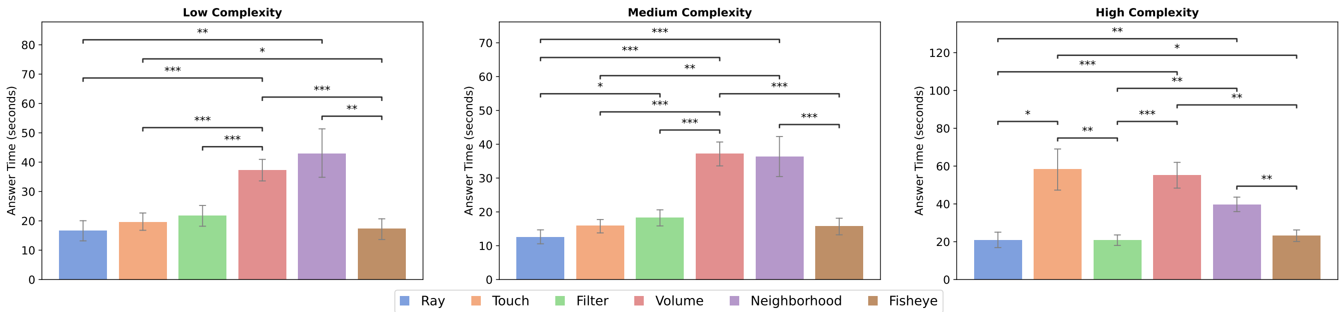

Figure 5: The answer time results with significant differences indicated for our six selection methods regarding *low*, *medium*, and *high* data complexity (see Section 4.2). Error bars show the standard error.

4 EVALUATION

To evaluate the performance of the six presented node selection techniques for immersive VR environments, we conducted a user study with 18 participants. Our setup is comparable to common visual network exploration toolkits, where a user aims to select an individual node. The perfect selection technique lets users select a desired node *fast*, without requiring much *physical effort* (as this can become tiring and exhausting) and matches the *user preference*. This should still be the case for increasingly complex networks. Therefore, we investigate the following key questions:

- How *efficient* are the selection techniques and how is this affected by the network complexity?
- Which degree of *physical effort* do the selection techniques require and how is this affected by the network complexity?
- What are the *preferences* regarding the selection techniques?

While *accuracy* is typically an essential factor for comparison studies, we do not explicitly investigate this variable, as in our case, users were allowed for multiple methods to also select close-by nodes, to iteratively approach the target node, for instance, by hopping to the desired node or disambiguating the region. Therefore, the initial selection of nodes other than the target, theoretically leading to lower accuracy, is not necessarily considered to negatively influence the outcome, which is why the accuracy variable itself would be misleading in our case. However, when users were not able to select desired nodes, the indirect impact of this could still be measured by increasing times required to solve a task and by reduced user preference. Similarly, we do not investigate *correctness*, as the correct solution was always clearly identifiable (due to the color coding) to users, and they worked on a task until the desired node was finally selected.

4.1 Apparatus

We implemented our techniques in a Unity application including SteamVR, perceived by a Valve Index head-mounted display (HMD) and operated by the original VR controllers. The VR environment consisted of a virtual room in which three-dimensional graph structures were visualized in front of the participants. As in comparable 3D network analysis platforms, we use colored sphere primitives to represent nodes and cylinder tubes to visualize edges. To ensure that participants could see all nodes, the transparency of the individual nodes was dynamically adjusted, such that nodes occluding other nodes (based on the current perspective) were rendered slightly transparent, ensuring that nodes behind it shine through. The controller of the dominant hand was used to control a visual ray and apply the different actions associated with interaction techniques, as described in Section 3. Participants were seated during the user study, therefore, the joystick of the other controller was used for navigation. However, for all techniques, except for **Touch**, we incorporated an invisible boundary that could not be crossed, ensuring that users could not move close to the graph. Without this restriction, users could navigate into the graph and use the ray to directly select the target node, ignoring the concepts we implemented, which contradicts the use case of distant node selection with occlusion factors we investigate. Aside from the network visualization and the interaction concepts, we included a study system, dynamically loading the data and configurations, and guiding users through the study with text labels floating in the scene.

4.2 Data

To test the performance of our selection concepts in relation to the network complexity, we created graphs of three different sizes, matching common complexities in graph analysis [22]. *Low* graphs contained 50 nodes, *medium* 120 nodes, and *high* graphs included 200 nodes. We randomly added edges, ensuring that each node was connected to one to five nodes, leading to a connected graph with the number of edges increasing together with the number of nodes. For the node positions, we applied the spring layout from NetworkX [18]. While we automatically selected a random node as the target, a manual inspection was applied to check that all target nodes could be seen from the initial perspective without unfair differences between the networks.

4.3 Procedure

We invited participants for individual sessions taking up to one hour. The study took place in a controlled environment at our university. After arriving, participants signed the consent form and received standardized explanations of the study subject and the different interaction techniques. Then, the HMD was mounted and adjusted while the participant's dominant hand was entered into the system. The practical study part started with a graph shown in the environment, for which all six selection techniques could be

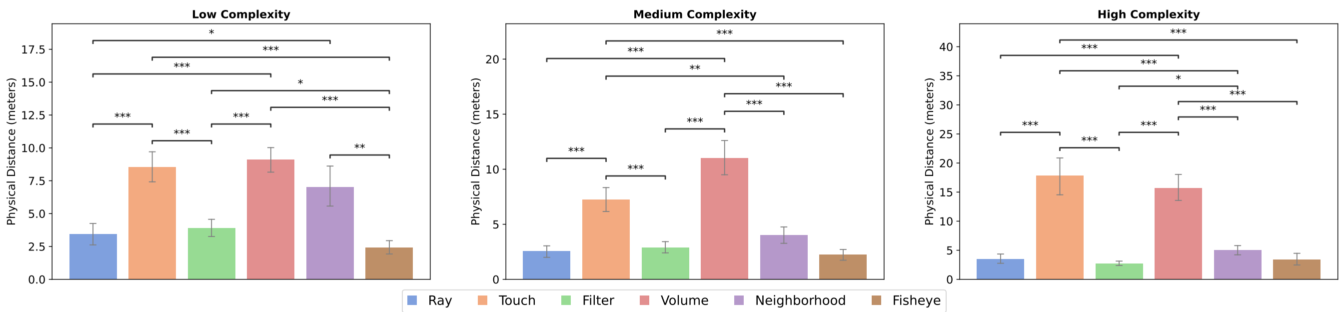

Figure 6: The results of the required physical effort with significant differences indicated for our six selection methods regarding *low*, *medium*, and *high* data complexity (see Section 4.2). Error bars show the standard error.

tried out. Textual guidance and advice by the study supervisor made sure that all techniques were understood and could be used by the participants. After the trial session, the actual study part started. For each interaction concept, three complexities (low, medium, high) were tested and repeated three times, resulting in $6 \times 3 \times 3 = 54$ individual tasks. For all users, the same networks were used, but the order of the selection techniques was randomized. The data complexity increased from low to medium to high within each task and for each selection technique. Prior to solving tasks, users received an explanation of the next technique before the network was visualized in the VR environment with one highlighted target node. This node had to be selected using the corresponding technique before the study continued with the next task. Questions could be asked at any time and comments were noted for the qualitative evaluation. After completing the practical session, a questionnaire with demographic and content-related questions was filled and the participants received a compensation of 10 EUR before leaving.

4.4 Participants

We invited 18 participants (nine female, nine male), of which 16 were right-handed. The age of the participants varied between 20 and 32 (M=26.1, SD=3.21). All participants reported normal (eight people) or correct-to-normal (ten people) vision with no known color deficiencies. As we did not require any preknowledge or experience, we asked our participants for their familiarity with *VR* and *network analysis*. Ten people reported *no*, four *little*, four *some*, and one *expert-level* VR experience. Regarding network analysis, eight people reported having *no*, three *little*, two *some*, two *high-level*, and three *expert-level* experience. Thus, the participants covered the entire range of experience levels with the study-related concepts with a focus on novice users.

4.5 Results

In the following, we report the results of our user study. We applied a significance level of $\alpha = 0.05$ and denote three classes of statistical significance: $p < 0.001 : ***,\ p < 0.01 : **,\ p < 0.05 : *$.

4.5.1 Efficiency. The distributions of the answer time results (see Table 1 and Figure 5) significantly differ from normal distributions (Shapiro-Wilk Test) and can not be transformed. Thus, we rely on non-parametric tests for the statistical evaluation. Friedman tests show significant differences among the six conditions regarding answer time for *low* ($p < 0.001,\ Q = 81.81$), *medium* ($p < 0.001,\ Q = 75.04$), and *high* ($p < 0.001,\ Q = 57.29$) data complexities. Post-hoc pairwise Wilcoxon signed-rank tests with Holm-Bonferroni correction indicate significant differences between individual selection methods, visualized in Figure 5 and summarized in more detail in Table 2. Across all conditions, **Ray** was the fastest selection technique, significantly outperforming **Neighborhood** and **Volume** (also **Filter** for medium- and **Touch** for high-complex data). These techniques were also outperformed by **Fisheye**, which also outperformed **Touch** for low and high complexities. While for low and medium complexities **Filter** was only significantly faster than **Volume**, it additionally outperformed **Touch**, and **Neighborhood** for high-complex networks. **Touch** was only faster than **Volume** (only for low and medium complex networks) and **Neighborhood** (only medium complexity). Overall, **Ray**, **Filter**, and **Fisheye** showed the best results regarding efficiency, with **Filter** having the most stable average answer times across the complexities.

4.5.2 Physical Effort. Working for a long time in an immersive setup can be exhausting, especially when the physical effort is high. To evaluate how physically demanding the six selection techniques

Table 1: The average *efficiency*, *physical effort*, and *user preference* for all six selection methods regarding low (L), medium (M), and high (H) complex data. User preference is measured on a scale from 0 to 10 (lowest to highest preference) for non-cluttered (NC) and cluttered data (C).

	Efficiency			*Physical Effort*			*Preference*	
	L	M	H	L	M	H	NC	C
■	16.60s	12.48s	20.84s	3.44m	2.55m	3.52m	8.11	5.33
■	19.60s	15.84s	58.33s	8.54m	7.23m	17.82m	7.61	5.50
■	21.83s	18.22s	20.86s	3.91m	2.89m	2.74m	7.00	8.11
■	37.23s	37.27s	55.11s	9.10m	11.02m	15.72m	6.89	7.88
■	42.95s	36.30s	39.54s	7.01m	4.03m	5.03m	5.72	5.11
■	17.32s	15.77s	23.05s	2.41m	2.23m	3.40m	4.33	4.89

■ **Ray** ■ **Touch** ■ **Filter** ■ **Volume** ■ **Neighborhood** ■ **Fisheye**

Table 2: P-values of the pairwise tests with Holm-Bonferroni correction. The upper-right matrix triangle depicts the values regarding *efficiency*, the bottom-left triangle the *physical effort*. Each cell contains the p-values for *low*, *medium*, and *high* data complexity for the six methods:
■ Ray ■ Touch ■ Filter ■ Volume ■ Neighborhood ■ Fisheye

	■	■	■	■	■	■
■		0.1305 0.3961 0.0117	0.1305 0.0187 0.8810	< 0.0001 < 0.0001 0.0001	0.0016 0.0001 0.0035	1.0 0.6538 1.0
■	< 0.0001 0.0001 < 0.0001		1.0 0.7567 0.0062	< 0.0001 < 0.0001 1.0	0.0528 0.0038 1.0	0.0295 0.7567 0.0355
■	0.4065 0.5318 1.0	< 0.0001 < 0.0001 < 0.0001		0.0001 < 0.0001 < 0.0001	0.0660 0.1101 0.0039	0.1305 0.7567 1.0
■	< 0.0001 < 0.0001 < 0.0001	0.6186 0.5318 1.0	< 0.0001 < 0.0001 < 0.0001		1.0 0.7567 0.8810	< 0.0001 0.0001 0.0016
■	0.0378 0.0697 0.0692	0.1413 0.0033 < 0.0001	0.3361 0.6651 0.0124	0.1273 0.0009 0.0001		0.0056 0.0010 0.0070
■	0.4065 0.6651 1.0	< 0.0001 < 0.0001 < 0.0001	0.0137 0.1449 1.0	< 0.0001 < 0.0001 < 0.0001	0.0027 0.1240 0.0692	

are, we gathered the movement of the VR controllers and the headset during the user study, combined these values into a single measure, representing the physical effort, and present the results in the following (see Table 1 and Figure 6). The distributions of the physical effort do not follow normal distributions (Shapiro-Wilk Test) and can not be transformed. Friedman tests reveal significant differences in physical effort for *low* ($p < 0.001$, $Q = 88.17$), *medium* ($p < 0.001$, $Q = 75.34$), and *high* ($p < 0.001$, $Q = 96.87$) data complexities. Hence, we applied pairwise Wilcoxon signed-rank tests with Holm-Bonferroni correction (see Figure 6 and Table 2). ■ **Fisheye** significantly outperformed ■ **Touch** and ■ **Volume** for all complexities, as well as, ■ **Neighborhood** and ■ **Filter** for networks of low complexity. ■ **Ray** required significantly less physical interaction than ■ **Touch** and ■ **Volume** across all complexities and less than ■ **Neighborhood** for low-complex networks. Moreover, ■ **Filter** outperformed ■ **Touch** and ■ **Volume** (all complexities), as well as, ■ **Neighborhood** (only highest complexity). For medium- and high-complex networks, ■ **Neighborhood** outperformed ■ **Touch** and ■ **Volume**. In summary, ■ **Ray**, ■ **Filter**, and ■ **Fisheye** had the least physical demand (with similar results), while interactions requiring direct touch or the placement of a selection volume in the graph led to significantly more physical effort.

4.5.3 Qualitative Feedback. Aside from the quantitative data, we gathered qualitative feedback using a questionnaire and the participants' comments during the study. As one part of the questionnaire, we asked the participants for their *personal preference* regarding the selection techniques (see Figure 7 and Table 1). In the case of non-cluttered data, on average, participants mostly favored ■ **Ray**, followed by ■ **Touch**, ■ **Filter**, ■ **Volume**, ■ **Neighborhood**, and ■ **Fisheye**. In the case of cluttered data, the preferences shifted,

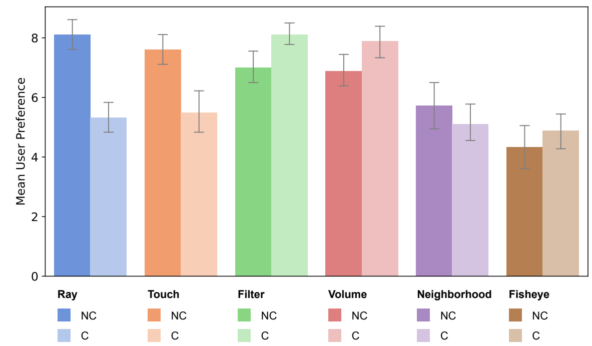

Figure 7: Average user preferences (0-10) for the six selection techniques considering non-cluttered data (NC) and cluttered data (C). Error bars show the standard error.

preferring ■ **Filter**, followed by ■ **Volume**, ■ **Touch**, ■ **Ray**, ■ **Neighborhood**, and ■ **Fisheye**. For non-cluttered data, the method indicated by most participants as their favorite was ■ **Ray** ($n = 9$), and the least preferred method was ■ **Fisheye** ($n = 10$). In the cluttered case, ■ **Filter** and ■ **Volume** ($n = 9$) were mostly preferred, ■ **Neighborhood** was the least favored technique ($n = 7$).

This trend is also indicated by the comments of the participants. Regarding the ■ **Ray** technique, users mentioned that it was "easy to use" ($n = 6$), and "good for non-cluttered data" ($n = 3$), while five participants argued that the technique was "hard to get precise selection". Individual participants also found it "intuitive" ($n = 1$), but also "training-intense" ($n = 1$). Seven participants "liked" the ■ **Touch** technique, finding it "intuitive" ($n = 3$), "realistic" ($n = 1$), and appreciated the ability to walk into the graph ($n = 2$). However, seven participants also found it "hard for cluttered data". In contrast to that, the ■ **Filter** selection was considered "helpful", especially for cluttered data ($n = 13$). Nevertheless, individual users found it "hard to rotate" ($n = 1$), argued that it requires some involvement ($n = 1$), and suggested making the filter plane scalable ($n = 1$). The ■ **Volume** technique was highly appreciated by the participants, finding it "very useful" ($n = 8$), "easy to use" ($n = 3$), especially helpful for cluttered data ($n = 7$), and appreciating the disambiguation after the initial selection ($n = 2$). While seven participants denoted the ■ **Neighborhood** technique as "useful" and "fun to use" ($n = 1$), it was also considered to be "hard to control" ($n = 3$), "difficult", and challenging to anticipate the next selected nodes, especially when the clutter level was high ($n = 1$). As possible improvements, participants suggested to use only edge-connected nodes instead of having virtual edges to all nodes in a certain radius ($n = 1$) and to hide nodes not contained in the neighborhood and thus, not selectable with the joystick ($n = 1$). The overall least preferred technique ■ **Fisheye** was only considered "useful" by three participants, especially for cluttered data, and two participants found the disambiguation mechanism helpful. This was mostly the case, as participants found this method "difficult" and "complex" ($n = 3$), and thus, "not useful" ($n = 6$). In general, users were able to solve the tasks using all methods, and the study setup worked as intended. When asked for other selection approaches not used in the study, participants mentioned that they would like to combine the ■ FILTER technique with ■ NEIGHBORHOOD ($n = 2$) or ■ VOLUME ($n = 2$) to first reduce the clutter, and then select a node with the option to correct it using the joystick.

5 DISCUSSION

Following, we discuss the results and impact of our user study.

5.1 Result Discussion and Findings

The **efficiency** evaluation revealed that the ■ RAY technique was the fastest across all network complexities, particularly excelling in low and medium complexities. This finding aligns with its widespread use in current VR applications, where its direct pointing mechanism is straightforward and quick for non-cluttered environments. However, as network complexity increased, the performance of ■ FILTER and ■ FISHEYE became comparable to the ■ RAY technique, which became more challenging with increasing occlusion and clutter. While the ■ RAY technique was still the fastest method for high-complex data, the comments of the participants and their reported preferences suggest that for even denser networks and with more training with the more complex filter and disambiguation techniques, other techniques might outperform ■ RAY. Especially the ■ FILTER technique seems promising, as its efficiency was constant between the different complexities, showing its usefulness in dense environments and suggesting that techniques incorporating occlusion management can significantly enhance selection efficiency in cluttered VR setups. While not preferred by most users, the ■ FISHEYE technique also performed well in terms of efficiency across all complexities. The focus+context approach might have helped users to disambiguate regions of interest, although it required some training and familiarity. As also suggested by the participants' comments, ■ TOUCH works mostly for small and non-cluttered networks, while the ability to accurately select a node highly decreases, leading to a drastic increase in required time. The ■ NEIGHBORHOOD method required a similar amount of time across all complexities, suggesting that this method requires some training and refining the initial selection by joystick navigation can be challenging, especially when the target is in the center of the graph and occlusion limits the vision. This impression is also supported by the preference values and the comments. Hence, combining it with a technique like ■ FILTER–as suggested by participants–might help to make use of the capabilities of this technique. Similarly, ■ VOLUME required more time than other techniques, but the user preference was much higher, especially for cluttered networks. Together with the comments, this raises the impression that participants were willing to invest time in placing the selection volume and moving the nodes apart before finally selecting the target, reducing unintended selections and increasing confidence. Notably, for most methods, the answer time decreased between the low and the medium data complexity conditions. This could indicate that the difficulty between those conditions did not drastically increase while the experience from the low-complex data helped users to solve the medium-complex tasks.

Physical effort is a critical factor in VR applications, where prolonged use can lead to fatigue. Our results indicate that the ■ FISHEYE technique required the least physical effort across all complexity levels, likely due to its joystick-based disambiguation mechanism, which minimized extensive movements. The ■ FILTER and ■ RAY techniques also demonstrated low physical effort, particularly in high-complexity networks, emphasizing their suitability for extended use. Conversely, the ■ TOUCH and ■ VOLUME techniques demanded the highest physical effort. The direct hand interaction in ■ TOUCH required substantial movement, especially in cluttered environments. Similarly, the ■ VOLUME technique's sphere placement and subsequent node separation, while effective in reducing occlusion, involved considerable physical manipulation, making it less favorable for long sessions. The ■ NEIGHBORHOOD technique did not reveal notable physical demand patterns.

The **qualitative feedback** provided additional insights. The ■ RAY technique was preferred for non-cluttered data due to its simplicity and speed. However, in cluttered environments, the ■ FILTER and ■ VOLUME techniques emerged as favorites. The ability to reduce occlusion and disambiguate nodes without losing their topology made these techniques more appealing and useful for complex networks. Surprisingly, despite its intuitiveness and low complexity, ■ TOUCH did not convince the participants, especially with increasing occlusion, since touching nodes to efficiently select them requires space around the target and increases the physical effort. The preferences regarding the ■ NEIGHBORHOOD technique were mixed. While it enables users to refine their initial selection within its neighborhood, occlusion made it hard to see the effects, and with increasing neighborhood members, the joystick-based interaction required more precise operation. Thus, we consider this approach still useful, but in combination with other disambiguation or filtering techniques. The ■ FISHEYE technique, despite its efficiency and low physical effort, was less favored due to its perceived complexity and learning curve. Moreover, we see our implementation as an initial attempt to transfer the benefits of 2D fisheye-disambiguation into 3D. Investigating implementations that are tailored to the characteristics of 3D setups and evaluating different designs could greatly improve the results of this technique.

5.2 Research Impact and Outlook

Our findings have significant implications for the design of VR-based network analysis tools. The ■ FILTER and ■ VOLUME techniques, with their strong performance in cluttered environments, should be **considered for larger graphs** in the future. Integrating **occlusion management** and **node disambiguation** features can enhance the usability and effectiveness of these tools, particularly for large and dense networks. Especially the ■ FILTER approach comes with high potential, not only for selecting nodes but also for exploration of the network structure. Our techniques preserve the network topology, allowing for node selection without losing track of the structure and characteristics. The study results suggest that **combining techniques** could further improve performance. For instance, integrating the occlusion management of ■ FILTER with the correction mechanism of ■ NEIGHBORHOOD or ■ VOLUME could provide a robust solution for various network complexities. This is also the case for ■ FISHEYE, our disambiguation technique that can be coupled with the other methods. The results of our study also suggest that different data complexities require different selection techniques. Therefore, network analysis tools could incorporate multiple different selection techniques and apply them based on the data complexity or individual user preference. While our work focused on hand-based interaction, it would be of high relevance to investigate the applicability of the concepts for gaze interaction. As discussed in Section 2.1, gaze interaction has already been integrated for selection in immersive settings, showing high potential. Some of our methods, namely ■ **Ray** (directly selecting focused nodes), ■ **Fisheye** (disambiguating the area around the focus point), and ■ **Neighborhood** (hopping to nodes relative to the gaze direction), can be directly adapted to gaze interaction. For ■ **Filter** and ■ **Volume**, depth is essential for placing the objects, which is more difficult with eye gaze. However, by integrating a second modality, such as a controller or hand gestures, these methods could also be used with gaze interaction.

Despite careful consideration, our study comes with **limitations**. The participant pool primarily included *novice users* with varying VR and network analysis experience, potentially influencing results. Future studies should involve a broader range of expertise. The *controlled laboratory setting* may not fully reflect real-world complexities. The study focused only on *individual node selection* tasks, so results may differ for other interactions like multi-node selection or dynamic graph exploration. The *network sizes* tested may not represent the full spectrum encountered in practice. Additionally, some techniques, like the fisheye effect, were initial adaptations from 2D to 3D and could benefit from further refinement. Future research should address these limitations by including diverse participants, real-world settings, and optimized technique implementations.

For **future research**, hybrid approaches, such as combining ■ **Filter** and ■ **Neighborhood**, should be evaluated in real-world applications. Additionally, the fisheye's potential in VR environments warrants further investigation. Refining its implementation to reduce complexity and improve intuitiveness could make it a valuable technique for immersive network analysis. Further, adapting and evaluating the presented techniques for gaze interaction (or gaze with a second modality) is of high interest. Moreover, testing the approach with even larger and denser networks might lead to further insights. Similarly, investigating the applicability of our approaches or deviations for multi-node selection could be of high interest. Understanding user training and adaptation to this technique would also provide insights into enhancing its usability.

6 CONCLUSION

This work evaluated six structure-preserving node selection techniques in an immersive VR environment, focusing on efficiency, physical effort, and user preferences. We conducted a user study with 18 participants, testing each technique on graphs of three different complexities. Our findings show that ray-casting is efficient for non-cluttered environments, while filter plane and selection volume techniques perform better in cluttered settings by managing occlusion and disambiguating nodes. Other techniques, such as fisheye disambiguation or neighborhood refinement, were promising but less preferred by users or led to increased physical or temporal effort. These insights inform and validate the design of VR-based network analysis tools, highlighting the need for hybrid approaches combining various techniques. Future research can optimize these methods and explore their applicability in further domains.

ACKNOWLEDGMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2117 – 422037984 and Project-ID 251654672 – TRR 161.

REFERENCES

[1] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[2] Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. 2016. Interaction techniques using head gaze for virtual reality. In *2016 IEEE Region 10 Symposium (TENSYMP)*. 110–114. https://doi.org/10.1109/TENCONSpring.2016.7519387

[3] Danielle S. Bassett and Edward T. Bullmore. 2009. Human brain networks in health and disease. *Current opinion in neurology* 22, 4 (2009), 340–347.

[4] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. 2009. Gephi: an open source software for exploring and manipulating networks. In *Proceedings of the international AAAI conference on web and social media*, Vol. 3. 361–362.

[5] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. 2017. A Taxonomy and Survey of Dynamic Graph Visualization. *Computer Graphics Forum* 36, 1 (2017), 133–159. https://doi.org/10.1111/cgf.12791

[6] Alberto Betella, Enrique Martínez Bueno, Wipawee Kongsantad, Riccardo Zucca, Xerxes D. Arsiwalla, Pedro Omedas, and Paul F. M. J. Verschure. 2014. Understanding large network datasets through embodied interaction in virtual reality. In *Proceedings of the 2014 Virtual Reality International Conference*. ACM, Article 23, 7 pages. https://doi.org/10.1145/2617841.2620711

[7] Matthew Brehmer and Tamara Munzner. 2013. A Multi-Level Typology of Abstract Visualization Tasks. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2376–2385. https://doi.org/10.1109/TVCG.2013.124

[8] Jeffrey Cashion, Chadwick Wingrave, and Joseph J. LaViola Jr. 2012. Dense and Dynamic 3D Selection for Game-Based Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (2012), 634–642. https://doi.org/10.1109/TVCG.2012.40

[9] Yuan Chen, Junwei Sun, Qiang Xu, Edward Lank, Pourang Irani, and Wei Li. 2021. Global Scene Filtering, Exploration, and Pointing in Occluded Virtual Space. In *Human-Computer-Interaction – INTERACT 2021*. Springer, 156–176.

[10] William Delamare, Maxime Daniel, and Khalad Hasan. 2022. MultiFingerBubble: A 3D Bubble Cursor Variation for Dense Environments. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*. ACM, Article 453, 6 pages. https://doi.org/10.1145/3491101.3519692

[11] Walter Didimo, Luca Grilli, Giuseppe Liotta, Lorenzo Menconi, Fabrizio Montecchiani, and Daniele Pagliuca. 2020. Combining network visualization and data mining for tax risk assessment. *Ieee Access* 8 (2020), 16073–16086.

[12] Adam Drogemuller, Andrew Cunningham, James Walsh, William Ross, and Bruce H. Thomas. 2017. VRige: exploring social network interactions in immersive virtual environments. In *Proceedings of the international symposium on big data visual analytics (BDVA)*.

[13] Fan Du, Nan Cao, Yu-Ru Lin, Panpan Xu, and Hanghang Tong. 2017. iSphere: Focus+Context Sphere Visualization for Interactive Large Graph Exploration. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2916–2927. https://doi.org/10.1145/3025453.3025628

[14] Kelly R. Finn, Matthew J. Silk, Mason A. Porter, and Noa Pinter-Wollman. 2019. The use of multilayer network analysis in animal behaviour. *Animal Behaviour* 149 (2019), 7–22. https://doi.org/10.1016/j.anbehav.2018.12.016

[15] Emden R. Gansner, Yehuda Koren, and Stephen C. North. 2005. Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics* 11, 4 (2005), 457–468. https://doi.org/10.1109/TVCG.2005.66

[16] Nicolas Greffard, Fabien Picarougne, and Pascale Kuntz. 2012. Visual Community Detection: An Evaluation of 2D, 3D Perspective and 3D Stereoscopic Displays. In *Graph Drawing*. Springer, 215–225.

[17] Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*. ACM, 3–12. https://doi.org/10.1145/1166253.1166257

[18] Aric Hagberg, Pieter Swart, and Daniel S. Chult. 2008. *Exploring network structure, dynamics, and function using NetworkX*. Technical Report. Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

[19] Lucas Joos, Maximilian T. Fischer, Daniel A. Keim, and Johannes Fuchs. 2023. Aesthetic-Driven Navigation for Node-Link Diagrams in VR. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction (SUI '23)*. ACM, Article 25, 10 pages. https://doi.org/10.1145/3607822.3614537

[20] Lucas Joos, Sabrina Jaeger-Honz, Falk Schreiber, Daniel A. Keim, and Karsten Klein. 2022. Visual Comparison of Networks in VR. *IEEE Transactions on Visualization and Computer Graphics* 28, 11 (2022), 3651–3661. https://doi.org/10.1109/TVCG.2022.3203001

[21] Lucas Joos, Karsten Klein, Maximilian T. Fischer, Frederik L. Dennig, Daniel A. Keim, and Michael Krone. 2023. Exploring Trajectory Data in Augmented Reality: A Comparative Study of Interaction Modalities. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 790–799. https://doi.org/10.1109/ISMAR59233.2023.00094

[22] Stephen G. Kobourov, Sergey Pupyrev, and Bahador Saket. 2014. Are Crossings Important for Drawing Large Graphs?. In *Graph Drawing*. Springer, 234–245.

[23] Regis Kopper, Felipe Bacim, and Doug A. Bowman. 2011. Rapid and accurate 3D selection by progressive refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. 67–74. https://doi.org/10.1109/3DUI.2011.5759219

[24] Joseph Kotlarek, Oh-Hyun Kwon, Kwan-Liu Ma, Peter Eades, Andreas Kerren, Karsten Klein, and Falk Schreiber. 2020. A Study of Mental Maps in Immersive Network Visualization. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*. 1–10. https://doi.org/10.1109/PacificVis48177.2020.4722

[25] Katja Krug, Wolfgang Büschel, Konstantin Klamka, and Raimund Dachselt. 2022. CleAR Sight: Exploring the Potential of Interacting with Transparent Tablets in Augmented Reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 196–205. https://doi.org/10.1109/ISMAR55827.2022.00034

[26] Maxim Kuznetsov, Aviv Elor, Sri Kurniawan, Colleen Bosworth, Yohei Rosen, Nicholas Heyer, Mircea Teodorescu, Benedict Paten, and David Haussler. 2021. The Immersive Graph Genome Explorer: Navigating Genomics in Immersive Virtual Reality. In *2021 IEEE 9th International Conference on Serious Games and Applications for Health (SeGAH)*. 8 pages. https://doi.org/10.1109/SEGAH52098.2021.9551857

[27] Joseph J. LaViola, Ernst Kruijff, Ryan P. McMahan, Doug Bowman, and Ivan P. Poupyrev. 2017. *3D user interfaces: theory and practice*. Addison-Wesley Professional.

[28] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. Investigating Bubble Mechanism for Ray-Casting to Improve 3D Target Acquisition in Virtual Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 35–43. https://doi.org/10.1109/VR46266.2020.00021

[29] Michael J. McGuffin, Ryan Servera, and Marie Forest. 2023. Path Tracing in 2D, 3D, and Physicalized Networks. *IEEE Transactions on Visualization and Computer Graphics* (2023), 14 pages. https://doi.org/10.1109/TVCG.2023.3238989

[30] Elena Molina and Pere-Pau Vázquez. 2023. Two-step techniques for accurate selection of small elements in VR environments. *Graphical Models* 128 (2023), 14 pages. https://doi.org/10.1016/j.gmod.2023.101183

[31] Roberto A. Montano-Murillo, Cuong Nguyen, Rubaiat Habib Kazi, Sriram Subramanian, Stephen DiVerdi, and Diego Martinez-Plasencia. 2020. Slicing-Volume: Hybrid 3D/2D Multi-target Selection Technique for Dense Virtual Environments. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 53–62. https://doi.org/10.1109/VR46266.2020.00023

[32] Tamara Munzner. 1998. Exploring large graphs in 3D hyperbolic space. *IEEE Computer Graphics and Applications* 18, 4 (1998), 18–23. https://doi.org/10.1109/38.689657

[33] Johannes Müller, Christian Krapichler, Lam Son Nguyen, Karl-Hans Englmeier, and Manfred Lang. 1998. Speech interaction in virtual reality. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98*, Vol. 6. 3757–3760. https://doi.org/10.1109/ICASSP.1998.679701

[34] Mershack Okoe, Radu Jianu, and Stephen Kobourov. 2019. Node-Link or Adjacency Matrices: Old Question, New Insights. *IEEE Transactions on Visualization and Computer Graphics* 25, 10 (2019), 2940–2952. https://doi.org/10.1109/TVCG.2018.2865940

[35] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + pinch interaction in virtual reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*. ACM, 99–108. https://doi.org/10.1145/3131277.3132180

[36] Sebastian Pirch, Felix Müller, Eugenia Iofinova, Julia Pazmandi, Christiane VR Hütter, Martin Chiettini, Celine Sin, Kaan Boztug, Iana Podkosova, Hannes Kaufmann, et al. 2021. The VRNetzer platform enables interactive network analysis in Virtual Reality. *Nature communications* 12, 1 (2021), 14 pages. https://doi.org/10.1038/s41467-021-22570-w

[37] Alexander Plopski, Teresa Hirzle, Nahal Norouzi, Long Qian, Gerd Bruder, and Tobias Langlotz. 2022. The Eye in Extended Reality: A Survey on Gaze Interaction and Eye Tracking in Head-worn Extended Reality. *Comput. Surveys* 55, 3, Article 53 (2022), 39 pages. https://doi.org/10.1145/3491207

[38] Manojit Sarkar and Marc H. Brown. 1992. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 83–91. https://doi.org/10.1145/142750.142763

[39] Sarah Schöttler, Yalong Yang, Hanspeter Pfister, and Benjamin Bach. 2021. Visualizing and Interacting with Geospatial Networks: A Survey and Design Space. *Computer Graphics Forum* 40, 6 (2021), 5–33. https://doi.org/10.1111/cgf.14198

[40] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research* 13, 11 (2003), 2498–2504.

[41] Rongkai Shi, Jialin Zhang, Yong Yue, Lingyun Yu, and Hai-Ning Liang. 2023. Exploration of Bare-Hand Mid-Air Pointing Selection Techniques for Dense Virtual Reality Environments. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Article 109, 7 pages. https://doi.org/10.1145/3544549.3585615

[42] Ludwig Sidenmark, Christopher Clarke, Joshua Newn, Mathias N. Lystbæk, Ken Pfeuffer, and Hans Gellersen. 2023. Vergence Matching: Inferring Attention to Objects in 3D Environments for Gaze-Assisted Selection. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Article 257, 15 pages. https://doi.org/10.1145/3544548.3580685

[43] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. 2020. Outline Pursuits: Gaze-assisted Selection of Occluded Objects in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, 13 pages. https://doi.org/10.1145/3313831.3376438

[44] Ludwig Sidenmark, Mark Parent, Chi-Hao Wu, Joannes Chan, Michael Glueck, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. 2022. Weighted Pointer: Error-aware Gaze-based Interaction through Fallback Modalities. *IEEE Transactions on Visualization and Computer Graphics* 28, 11 (2022), 3585–3595. https://doi.org/10.1109/TVCG.2022.3203096

[45] Ludwig Sidenmark, Zibo Sun, and Hans Gellersen. 2024. Cone&Bubble: Evaluating Combinations of Gaze, Head and Hand Pointing for Target Selection in Dense 3D Environments. In *2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops*. 642–649. https://doi.org/10.1109/VRW62533.2024.00126

[46] Johannes Sorger, Manuela Waldner, Wolfgang Knecht, and Alessio Arleo. 2019. Immersive Analytics of Large Dynamic Networks via Overview and Detail Navigation. In *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE Computer Society, 144–151. https://doi.org/10.1109/AIVR46125.2019.00030

[47] Delfina Malandrino Ugo Erra and Luca Pepe. 2019. Virtual Reality Interfaces for Interacting with Three-Dimensional Graphs. *International Journal of Human–Computer Interaction* 35, 1 (2019), 75–88. https://doi.org/10.1080/10447318.2018.1429061

[48] Lode Vanacken, Tovi Grossman, and Karin Coninx. 2007. Exploring the Effects of Environment Density and Target Visibility on Object Selection in 3D Virtual Environments. In *2007 IEEE Symposium on 3D User Interfaces*. https://doi.org/10.1109/3DUI.2007.340783

[49] Tatiana von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J. van Wijk, Jean-Daniel Fekete, and Dieter W. Fellner. 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. *Computer Graphics Forum* 30, 6 (2011), 1719–1749. https://doi.org/10.1111/j.1467-8659.2011.01898.x

[50] Jorge Wagner, Wolfgang Stuerzlinger, and Luciana Nedel. 2021. Comparing and Combining Virtual Hand and Virtual Ray Pointer Interactions for Data Manipulation in Immersive Analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 5 (2021), 2513–2523. https://doi.org/10.1109/TVCG.2021.3067759

[51] Weihang Wang, Sriram Karthik Badam, and Niklas Elmqvist. 2022. Topology-aware space distortion for structured visualization spaces. *Information Visualization* 21, 2 (2022), 166–181. https://doi.org/10.1177/14738716211053579

[52] Yunhai Wang, Yanyan Wang, Haifeng Zhang, Yinqi Sun, Chi-Wing Fu, Michael Sedlmair, Baoquan Chen, and Oliver Deussen. 2019. Structure-aware Fisheye Views for Efficient Large Graph Exploration. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (2019), 566–575. https://doi.org/10.1109/TVCG.2018.2864911

[53] Colin Ware and Glenn Franck. 1994. Viewing a graph in a virtual reality display is three times as good as a 2D diagram. In *Proceedings of 1994 IEEE Symposium on Visual Languages*. 182–183. https://doi.org/10.1109/VL.1994.363621

[54] Zhiqing Wu, Difeng Yu, and Jorge Goncalves. 2023. Point- and Volume-Based Multi-object Acquisition in VR. In *Human-Computer Interaction – INTERACT 2023*. Springer, 20–42.

[55] Hans Peter Wyss, Roland Blach, and Matthias Bues. 2006. iSith - Intersection-based Spatial Interaction for Two Hands. In *3D User Interfaces (3DUI'06)*. 59–61. https://doi.org/10.1109/VR.2006.93

[56] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2020. Fully-Occluded Target Selection in Virtual Reality. *IEEE Transactions on Visualization & Computer Graphics* 26, 12 (2020), 3402–3413. https://doi.org/10.1109/TVCG.2020.3023606

[57] Futian Zhang, Keiko Katsuragawa, and Edward Lank. 2022. Conductor: Intersection-Based Bimanual Pointing in Augmented and Virtual Reality. 6, ISS, Article 560 (2022), 15 pages. https://doi.org/10.1145/3567713

[58] Fengyuan Zhu, Ludwig Sidenmark, Mauricio Sousa, and Tovi Grossman. 2023. PinchLens: Applying Spatial Magnification and Adaptive Control-Display Gain for Precise Selection in Virtual Reality. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 1221–1230. https://doi.org/10.1109/ISMAR59233.2023.00139