

Immersive Analytics of Large Dynamic Networks via Overview and Detail Navigation

Johannes Sorger
Complexity Science Hub
Vienna, Austria
sorger@csh.ac.at

Manuela Waldner
TU Wien
Vienna, Austria
waldner@cg.tuwien.ac.at

Wolfgang Knecht
Complexity Science Hub
Vienna, Austria
knecht@csh.ac.at

Alessio Arleo
TU Wien
Vienna, Austria
alessio.arleo@tuwien.ac.at

Abstract—Analysis of large dynamic networks is a thriving research field, typically relying on 2D graph representations. The advent of affordable head mounted displays however, sparked new interest in the potential of 3D visualization for immersive network analytics. Nevertheless, most solutions do not scale well with the number of nodes and edges and rely on conventional fly- or walk-through navigation. In this paper, we present a novel approach for the exploration of large dynamic graphs in virtual reality that interweaves two navigation metaphors: overview exploration and immersive detail analysis. We thereby use the potential of state-of-the-art VR headsets, coupled with a web-based 3D rendering engine that supports heterogeneous input modalities to enable ad-hoc immersive network analytics. We validate our approach through a performance evaluation and a case study with experts analyzing a co-morbidity network.

Index Terms—Immersive Network Analytics, Web-Based Visualization, Dynamic Graph Visualization

I. INTRODUCTION

Virtual and augmented reality (VR and AR) are by no means new concepts: already in 1968, Sutherland presented a first, three-dimensional head mounted display [1]. In 1993, thanks to the technical advancements of 3D hardware, the IEEE Conference on Virtual Reality (IEEE VR) was held for the first time. As such, research on graph visualization in virtual reality has already been conducted in the 90ies [2]. Since then, several studies have shown that stereoscopy can improve the users’ understanding of graphs as compared to monoscopic 3D and 2D graph drawing [3]–[9]. Raja et al. [10] showed that increasing immersion in a CAVE, by supplying more display space and supporting head tracking, had a positive effect on the users’ analysis of 3D scatterplots. In comparison, VR experienced through a head-mounted display (HMD) is even more effective than the immersive environment of CAVE-like solutions [11]. For instance, users could collaboratively analyze graph connectivity significantly faster using HMDs than in a CAVE-like environment [12]. Similar benefits have been shown for 3D scatterplots: an immersive VR setup could increase classification accuracy and led to higher engagement by the users compared to 2D and 3D desktop rendering [13], [14]. Bowman and McMahan [15] argue that increased immersion can reduce visual clutter and strengthen the comprehension of the displayed scene. Despite these encouraging findings and the increasing availability of commercial HMDs, many visualization researchers kept a conservative attitude

towards immersive environments and their potential data-mining applications [16]. In recent network visualizations, the use of 3D techniques is still rare, and little research has investigated novel graph drawing or interaction techniques for immersive network analytics.

Considering these promising previous results and the availability of affordable state-of-the-art VR headset solutions, in this paper we investigate the challenge of enabling immersive analytics for large dynamic networks using modern VR headsets. We explore the possibilities of “immersion” into the network for detailed analysis while being provided with the context of a well-known overview perspective on the network to stay oriented. Our proposed exploration technique takes advantage of modern immersive technologies using a flexible, expandable and browser-based software infrastructure natively compatible with VR headsets but also supporting traditional input methods to support ad-hoc immersive analytics also for users without programming experience. Our contributions are listed in the following:

- A novel approach for the immersive exploration of networks in VR via overview&detail navigation;
- A fast web-based render engine for large dynamic graphs offering VR controller support for ad-hoc immersive network analytics;
- First encouraging results of a case study with domain experts analyzing medical data (a co-morbidity network), showing the potential of VR for network analytics.

II. RELATED WORK

Adapting visualizations to new modalities, such as VR, is not a trivial task, as it may or may not require new ways to represent and interact with the data [17]. Researchers therefore have explored ways of how to interactively *manipulate*, *navigate*, and *represent* networks in VR. Already in an early CAVE-like environment, Osawa et al. [18] introduced hand gestures to manipulate distant nodes of a graph. They also introduced a variation of a “spotlight” [19] that distorts the graph around the focus region for exocentric navigation around the graph. More recently, Huang et al. [20] as well as Erra et al. [21] presented a natural graph exploration and manipulation interface for VR using a Leap Motion sensor. Drogemuller et al. [22] introduced a virtual “filter cube” to filter and highlight nodes in the graph.

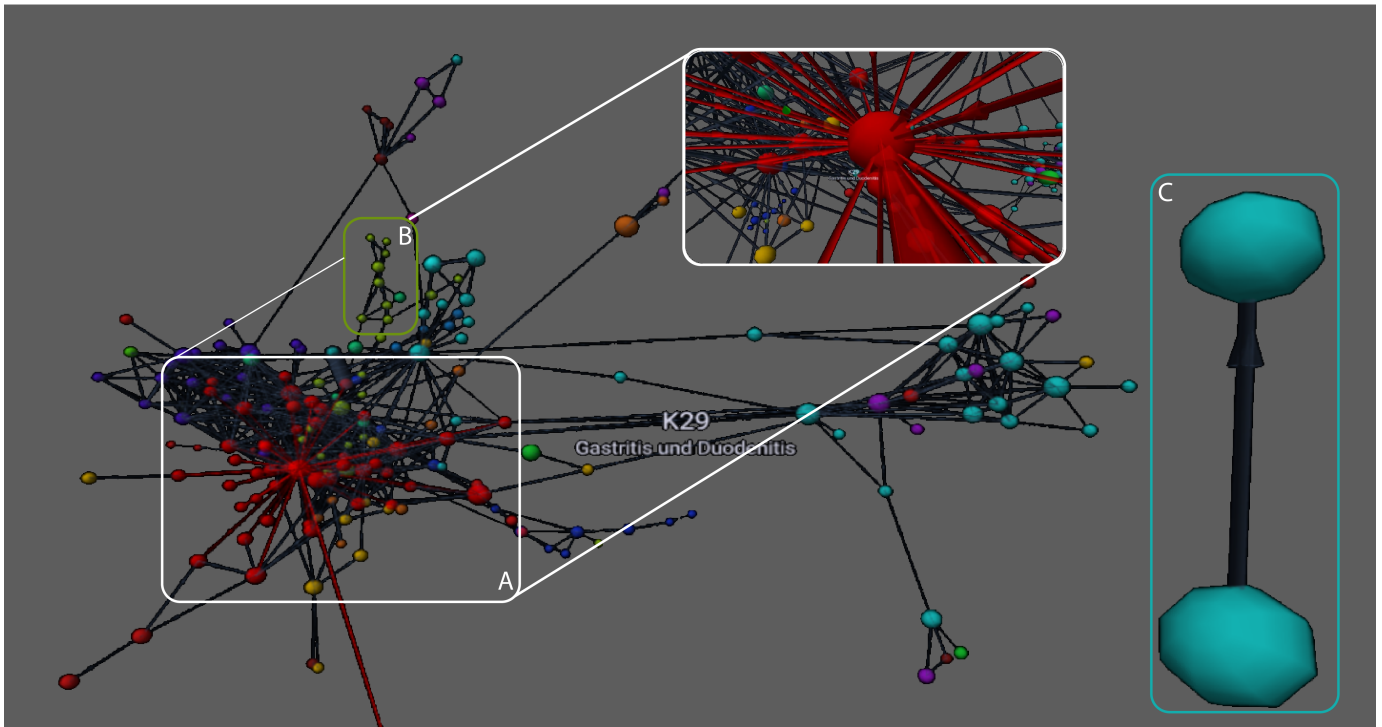


Fig. 1. The overview perspective of a co-morbidity network. A) K29, a “hub” node with a high degree; the inset shows the same node from the detail perspective. The node is selected, therefore its neighbors and out-edges are highlighted in red and the node label is displayed. B) A cluster. C) An isolated component of the graph. (For better readability on printed paper the background color and contrast of the image have been adapted.)

Besides manipulation, navigation is considered to be one of the key aspects to be solved for information visualization in 3D [23]. Modern graph visualization in HMD-based VR usually provides a fly-through interface using a 6 DOF camera control [14], [21] or the ability to walk through the graph [22]. Drogemuller et al. [24] compared four VR navigation techniques for graph exploration using room-sized VR systems. They found that two-handed flying [25] using two controllers was most efficient for finding single nodes or paths between two nodes in the graph. While this work compared the appropriateness of established VR navigation techniques, our goal was to explore a new navigation method to facilitate immersive network analytics in particular.

Most prior immersive network analytics systems show networks as conventional node-link diagrams based on a force-directed 3D graph layout. Some also experimented with other ways of how to represent networks in VR or AR. Büschel et al. [26] evaluated different edge styles for AR graphs and found that curved edges are harder to interpret. Halpin et al. [7] initialized their VR scene with a 2D graph layout. Users could interactively “extrude” nodes to the third dimension to explore the social network. Kwon et al. [27] suggested to render graphs using a spherical layout in VR, optimized towards usage while seated. Interaction was carried out via mouse and keyboard. Compared to a 2D representation of the graph in VR, immersive visualization allowed users to successfully complete more complex tasks on larger graphs than before [8]. Marriot et al. [9] argue that egocentric data

views, such as proposed by Kwon et al. [8], [27], may increase user engagement. However, a study by Yang et al. [28] showed that maps explored from an egocentric perspective led to lower performance than any other representation in VR (an exocentric view, a flat map view, and a curved map view). We share the belief that egocentric data views are a unique strength of virtual environments to provide a new perspective on the data that cannot be easily obtained on a desktop screen. In contrast to Kwon et al. [8], [27], however, we also believe that a conventional exocentric view on a network provides a visual anchor for orientation and overview. In contrast to these prior works, we therefore present and explore a new immersive network analytics approach, where users can flexibly switch between an exocentric overview and an egocentric detail view.

III. DESIGN AND REQUIREMENTS

Our goal is to provide new insights in the course of network analytics, such as social network analytics [29] or network medicine [30], by providing a new visual perspective of the data, enabled through VR. To design our system, we make use of the “Data-Users-Tasks” design triangle by Miksch and Aigner [31]. This technique requires the definition of three crucial elements that will steer the design process: the **Data** the system will process, the **Users** that will get in contact with the system, and the **Tasks** the system is expected to perform. In this section, we will discuss each of these aspects.

A. Data

To face the challenges of modern network analytics tasks [29], [30], our system is expected to handle large, dense, dynamic network data. Nodes and edges can be associated with additional attributes, such as a cluster membership for nodes or edge weights that rank a connection among the others. In the scope of this paper, we assume temporal changes of the graph to be known beforehand, and to be discrete, i.e., they are grouped into distinct time “frames”.

B. Users

Large network exploration is a difficult task to perform, in 2D and 3D alike. Our target users are researchers who wish to explore large networked domain specific data using an intuitive and life-like approach, in which they can manipulate, move around, and interact with the data as if it would be physically present with them. We aim to support users who are not necessarily experts or knowledgeable in network visualization, graph drawing, or even data science and visualization, i.e., from disciplines such as healthcare, finance, or social science.

C. Tasks

The purpose of our system is to provide an immersive exploration and interaction technique to obtain (expected and unexpected) insights from dynamic networked data. Typical network analytics tasks across domains have been identified and summarized by Lee et al. [32]. Driven by the challenges of modern network analytics approaches in different domains [29], [30], we focus here mainly on *topology-based tasks* [32], including:

- finding *hubs*, such as highly connected nodes in a comorbidity network or the most influential members of a social network,
- detection of *communities*, such as tight interaction between proteins or highly connected social communities,
- locating the shortest molecular *paths*, or
- characterizing the *dynamics* of social interaction.

In addition, we speculate that it is important to first obtain an *overview* of the network, as stipulated in the widely accepted information seeking mantra [33]. Gaining an overview is essential to infer global insights out of the network’s topology, allowing users to identify interesting spots and areas suitable for more in-depth evaluation. Finally, the system should provide *details-on-demand* to inspect attributes of single nodes and edges to derive detailed information from local structures.

We believe that immersive network analytics is an appropriate method for supporting topology-based exploration and browsing through large, dense graphs as it can provide a new, complementary perspective of the data. Using VR, the user can inspect parts of the network in detail, i.e., from a first person perspective, e.g., by following a path of edges through the network, or by inspecting local connectivity from an angle with better visibility; thus enabling browsing and topology-based tasks [32].

IV. SYSTEM

In this section, we describe the system in detail, both in terms of visual design and interactions.

A. Network Visualization

Our goal is to provide expert users with a new perspective on their data in VR. It is therefore important to show the data as closely as possible to their conventional representation form to support quick orientation. Hence, we render the network using the most wide-spread graph rendering method: node-link diagrams. Nodes are rendered as spheres of different diameters. Edges are represented as tubes. In case of a directed graph, the direction of each edge is represented by an arrow, close to and directed towards the target node (see Figure 1C). The color and diameter of the nodes as well as the color and girth of the edges can be changed according to user-defined attribute values in the graph input file. The 3D layout of the graph is delegated to a 3D extension of the D3 force-layout [34], [35].

B. User Perspectives

A special design consideration of our system is the notion to let users explore the network and its evolution from two different “perspectives”: **overview** and **detail**. In the overview, users regard the graph from “outside”, meaning the graph is positioned so that it completely fits within the user’s view. This enables the user to look at the entire network (from different angles), in order to get an overview of interesting node/edge formations and clusters – i.e., to spot interesting parts of the graph that they would like to explore in more detail. The overview thereby shares similarities to a conventional 2D graph layout, which expert users analyzing networks are usually very familiar with.

If the users want to inspect a part of the network in more detail, they can select a specific node to immerse themselves into the detail perspective. From this perspective, users can effectively explore local neighborhoods even in very dense networks.

After teleportation between perspectives, users may have difficulties staying oriented. To improve orientation, the previous point of view is therefore represented within the context of the graph as a camera shaped object, informing the user on the position from where the exploration started. To further aid orientation and to facilitate switching perspectives, in the detail perspective, the overview camera’s position is indicated as a green arrow in the top of the user’s view. The arrow guides the user towards the overview camera’s position by pointing to its direction (see Fig.3).

C. Navigation and Interaction

Input: In the current version of our system, we designed interaction techniques specifically for the HTC Vive. However, the controls can be adapted to any input device that features the same set of input modalities. The HTC Vive (akin to other state of the art solutions) features one wireless controller for each hand, to allow users to use their hands freely and

independently from each other - as opposed to conventional game-pads that are held with both hands together. Since we aimed to make interaction with the graph intuitive also for users who are not yet familiar with VR technology, the system is designed to work with only one controller as input device. The user can thus single-handedly interact with and navigate within the graph. In the same vein, the controls do not need to be specifically adjusted for left- or right-hand users.

Required input modalities (as supported by the HTC Vive controller) are the position and orientation of the controller, a two-dimensional track-pad enabling input on an x- and y-axis (i.e., directional pad, or D-pad), two additional buttons (a trigger button and an input modifier button). Further, the VR headset position and rotation have to be traceable. The button layout on the HTC Vive controller is depicted in Figure 2.

In general, input can be differentiated into graph interactions and graph navigation.

Graph interactions: Graph interactions are carried out with the “laser-pointer”, i.e., a visible ray that is cast from the controller into the scene, according to the controller’s position and orientation (see Fig.2). Upon hovering with the laser-pointer on a graph element, the user can inspect details, such as a node label or an edge weight. Such details on node or edge attributes are displayed as a text label in the center of the user’s view (see Fig.1, center), in order to achieve the best readability. Due to the curvature of the magnifying lenses in the headset, text in the periphery can be distorted and blurred, and would thus be hard to read. Hovering on a node also reveals additional details of the network, by allowing a user to inspect the direct neighborhood of the hovered node (see Fig.1-A). Direct neighbors of the hovered node are thereby highlighted in red, while all other graph elements that are not part of this neighborhood are lowlighted so that they blend into the background. This is achieved by reducing their opacity and giving them a uniform dark hue. Graph interactions function the same way independent of the chosen perspective.

Graph navigation: Graph navigation on the other hand, offers a different navigation method for each perspective (i.e., overview and detail) that is designed to complement the respective task. In both modes, the D-pad can be used for direct navigation. In the overview, input on the D-pad rotates the graph around its local axes, enabling the user to quickly get an impression of the network from various angles. The entire graph is thereby enclosed within a (non-visible) bounding geometry that is used to adjust the camera to a suitable viewing distance when rotating the network representation. The laser-pointer can thereby be used to explore local connectivity from the distance by high/low-lighting individual network neighborhoods.

In the detail perspective, the user has the option to “fly” through the network using free flying controls for immersive exploration. Input on the D-pad directly controls the user’s position relative to his or her current viewing vector (pressing forward/backward on the D-pad lets users fly to/from where they are looking; pressing left/right allows the user to strafe perpendicular to the viewing vector of the VR headset). Since

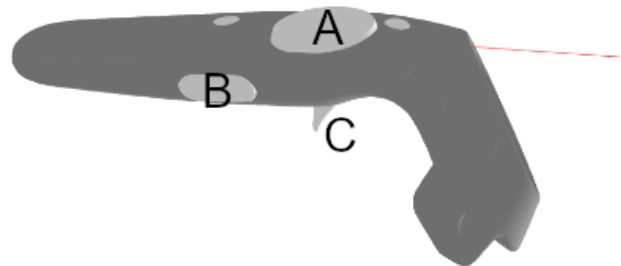


Fig. 2. A depiction of the virtual representation of the controller: A) Directional pad; B) Side button for D-pad input-modification (temporal navigation); C) Trigger for node/edge selection. The vertical axis of the controller determines the laser pointer orientation.

6 degrees of freedom (DOF) navigation in VR can cause discomfort for some users [36], [37], our system also offers a second navigation mode in the detail perspective: by selecting a hovered edge or node with the trigger button, an automatic flight transition to the selected graph element is triggered. The velocity is thereby increased in comparison to the free flying controls to enable faster navigation within the graph. Arrival at a node/edge is smoothed with an ease-out curve with the aim to further reduce motion sickness. Using this mode, the user can follow paths in the network from a first person perspective.

Switching between overview and detail: To switch to the detail perspective from the overview, users must select a node of interest as their detail-exploration starting position. The camera representing the detail perspective thus moves to the selected entry point. By confirming their selection on the node containing the camera, the user instantly teleports to the selected node position. As teleportation may have a negative effect on the user’s orientation when changing both the position *and* the rotation after a teleportation [38], we only translate the user’s position, but keep the orientation of the user’s starting perspective. To switch back to the overview perspective, the user must simply click on the green overview-indicator arrow in the top of the view – thus sparing users from tracking down their initial overview position before being able to switch perspectives. Special consideration is thereby taken to avoid disorientation: when switching back to the overview perspective, the entire graph is placed directly in-front of the user’s current viewing vector, on a height that matches the user’s head’s position in space. This way, the user is able to instantly see the graph at eye-level when teleporting back to the overview, thus sparing them from unpleasant viewing angles and from searching the location of the graph.

Aside from rotation, free and automatic flying, and teleportation, our system also supports room-scale navigation, i.e., depending on the scale of the depicted network and the size of the allotted physical VR interaction space, users are able to physically walk around within the network, simultaneously changing their relative position and orientation in the scene.

Temporal Navigation: The final navigation modality allows users to scroll through the temporal evolution of a dynamic network, both from an overview and a detail perspective. Temporal navigation is carried out via the D-pad’s horizontal input axis while pressing the input-modification button. A time axis will appear on top of the laser pointer, indicating the current time step in relation to the time axis (see Figure 3). When the time frame changes, the nodes and edges of the current time step fade in, while graph elements not present in the current step are faded out, as depicted in first two images in the sequence of Figure 3. The smooth fading thereby facilitates observation of the temporal change. The layout of the graph (i.e., node and edge positions) is kept static to facilitate the comparison between different time steps and to avoid disorientation of the user. The temporal navigation mode also lends itself to filtering graph elements by sliding through discrete attribute ranges instead of through time. Additionally, this navigation mode can be used for the comparison of two different (but related) networks by navigating (switching) from one set of nodes/edges to the other one.

D. Implementation

The system is implemented as a client-only web application, i.e., no server-side scripts need to be executed at runtime. The code is written completely in JavaScript using three.js [39] and A-Frame [40] – a layer on top of three.js that supports browser based VR applications by handling the stereoscopic rendering as well as controller and head-tracking input from VR hardware. The foundation of our system is based on an open source library for viewing graphs in VR [41]. This library handles the loading, layout, and rendering of the graph, and offers simple navigation via flying by mouse and keyboard outside of VR. In VR, the core does not offer any navigation or interaction capabilities (only looking and pointing via the headset’s viewing vector). In order to support the navigation and interaction methods that form part of our contribution, we thus extended this core to accept input via VR controllers (specifically, the HTC Vive). The interface between the VR application (i.e., the web-browser) and the VR hardware is handled by SteamVR [42] that natively supports the HTC Vive.

Rendering large node-link diagrams in 3D can have a big impact on performance as all graph elements are rendered as geometries. To improve performance, edges can alternatively be rendered as simple lines. However, this reduces their visibility, makes them harder to select, and removes the support for encoding edge weights in the girth. To avoid such compromises in usability and readability, we further extended the core library’s rendering capabilities to support instancing of node and edge geometry via three.js *InstancedBufferGeometry*. The improvement almost doubles the achieved frame rate on the largest graph of the performance evaluation (2000 nodes, 6000 edges subsection V-A), from 13 to 22 frames per second.

Special consideration also is needed for the implementation of the dual camera perspectives and associated navigation and interaction modalities. The camera orientation, and VR controls are child objects of a parent container, the *active*

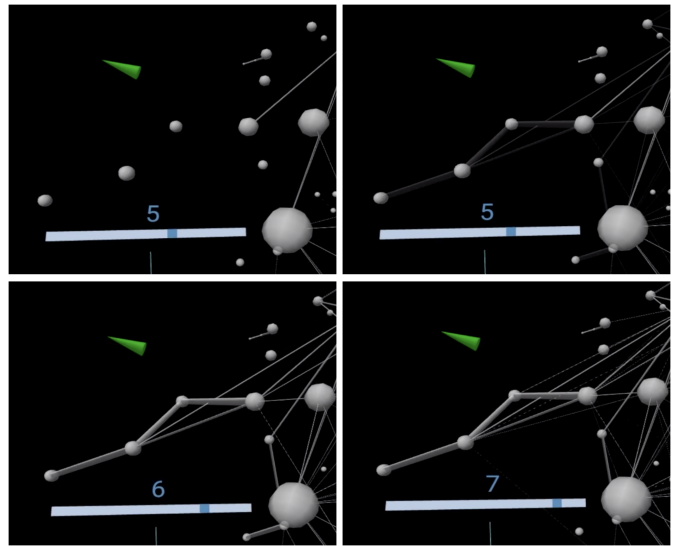


Fig. 3. A sequence captured from a temporal transition between three time frames. The time-bar on top of the laser-pointer indicates the current time step. The top right image shows the transitional state between steps 5 and 6. The images are captured in detail perspective. The green arrow points to the position of the overview camera.

rig that represents the user’s position in the VR scene. To maintain two perspectives, a second rig, the *passive rig*, serves as a placeholder, storing the other perspective’s position and orientation. When switching between perspectives, position and orientation of the two rigs are simply interchanged.

V. EVALUATION

We evaluate our approach by means of a quantitative and qualitative evaluation. The former is carried out as a performance analysis of the tool, in terms of a frame rate benchmark with networks of increasing size (see subsection V-A). The latter is carried out through a case study with two domain experts who were asked to perform a walk-through in a real healthcare dataset to evaluate how our design choices reflect on their use of the system (see subsection V-B). Both experiments are run on the same test setup, on a desktop PC equipped with an AMD Threadripper 1900-X CPU with 32GB of RAM and a Geforce GTX 1080 Ti graphics card. The headset used is an HTC Vive head-mounted display and controllers.

A. Performance and Frame Rate

To enable immersion and avoid motion sickness, it is necessary to render stereoscopic images with a fluid frame rate. For that matter, SteamVR automatically stops sending updates to the headset when performance drops below 20 frames per second (fps). To assess the performance of our system, we simulate a typical use case scenario on increasingly large graphs and record the frame rate in three configurations: a static *overview*, *rotation* of the network in the overview, and navigation in the *detail* perspective. As a benchmark, we created three random graphs (using the Erdős-Reyny [43] model), with 500, 1000, and 2000 nodes, and an average

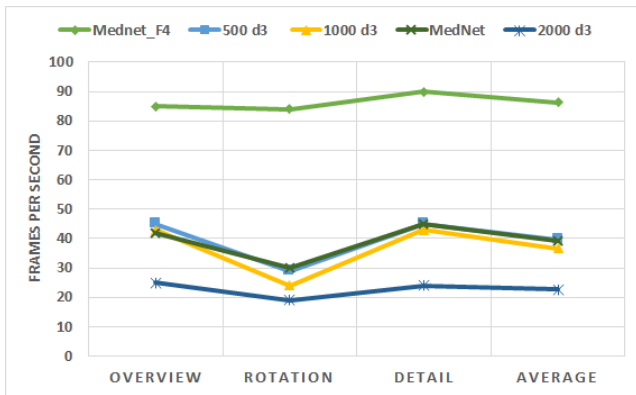


Fig. 4. System performance comparison. On the y-axis the average frames per second; on the x-axis the perspective in which the measurement was taken.

node degree of 3 (i.e., with 1500, 3000, and 6000 edges respectively). Additionally, we use 2 real-world datasets that include the one used in our case study: *MedNet F4* with 199 nodes and 593 edges (see subsection V-B) and *MedNet* with 692 nodes and 3047 edges, for a total of five graphs. The results are reported in Figure 4. The chart shows that with the smallest graph the average fps rate is about 85, with no major difference on the two perspectives. As the size increases, the average frame rate drops down to 35-40 fps with a low of about 22 fps for the largest graph. The rotation in the overview perspective requires the most computing power, since the entire graph is on screen while being animated. In the detail perspective instead, only a portion of the graph is visible thus reducing the load on the graphics hardware. It is worth mentioning that on real-world datasets, performance was satisfactory: the *MedNet F4* graph, used during the case study (see subsection V-B) rendered with ~ 85 fps for smooth navigation and interaction. When pushing the system with the last graph (2000 nodes, 6000 edges), the large number of elements on screen had a significant impact on rendering, with frequent stutter especially during ray-casting with the laser-pointer.

B. Case Study

The goal of the case study was to gain first realistic insights about the usefulness of the two network perspectives. We were interested to find out whether users see the necessity to switch their perspective and which kinds of discoveries they make in either perspective. This helps us to provide a first characterization of the potential mutual benefits of the two perspectives for immersive network analytics. In addition, we asked users to report symptoms of motion sickness, disorientation, or other navigation problems to identify potential issues after switching views.

For the case study, we asked a female data scientist and a male physicist to present and explore a co-morbidity network [44] that they are currently investigating in the course of their research. They are already working with the dataset for several months (data scientist) to years (physicist) to

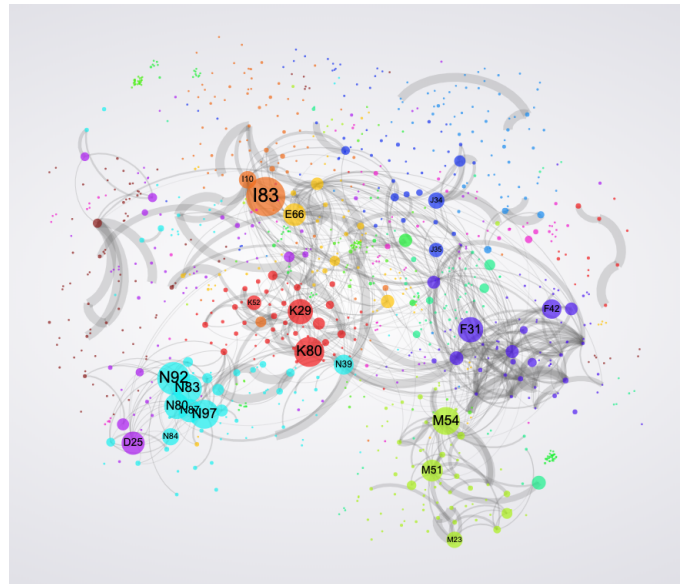


Fig. 5. The semi-automatic 2D layout of the co-morbidity graph created by our domain experts.

analyze patterns of diseases that usually occur together and to compare these patterns across different populations, such as male and female patients. In their standard workflow, they use Gephi [45] for generating 2D visualizations of the entire graph. These visualizations are printed and presented to collaborating medical experts for analysis. The giant component of the network, which was also used for this case study, contains 199 nodes and 593 edges. The domain experts consider their current approach using static 2D graphs as partially insufficient as it is not possible to interactively dig into the network to get more details. They also reported difficulties finding a good graph layout, where groups of strongly connected diseases are clearly visible. To try to overcome those complications, they currently analyze and present their network by computing an initial layout using a force-directed layout and subsequently manually moving certain nodes to separate (known) clusters. Additionally, they only render a subset of all edges to reduce visual clutter. The semi-automatic 2D layout is shown in Figure 5 while the (fully-automatic) 3D layout of the same network is in Figure 1.

After an initial interview about their current workflow, we gave a short introduction to the VR setup and the interaction techniques without revealing what is novel about the navigation approach. We then asked the users to walk through their data set and to orally present the contents as if they would present them to their medical partners, as well as to comment on the usability and appropriateness of the VR setup. We recorded the VR sessions, and transcribed the videos afterwards. We coded all user findings in the transcripts, and also coded if the finding was already known or unexpected, as well as whether it was made in the overview or in the detail perspective. In addition, we extracted all comments and suggestions about the VR setup. The data scientist spent 33

minutes, and the physicist spent seven minutes in the VR environment.

Both users observed and reported a number of known facts about the network during the session. These reported facts covered *clusters* (such as a clearly separate cluster of cancers or the very central cluster of mental diseases, see Figure 1-B), *hub nodes* (such as breast cancer or depressive episodes, see Figure 1-A), *paths* between nodes, and isolated *sub-networks* (such as teeth- or pregnancy-related problems, see Figure 1-C). Most of these known facts were reported while exploring the graph in the overview. This is not surprising, since the users knew the dataset before and are also used to see it laid out in 2D. As the physicist explained, the overview is similar to their well-known 2D view of the graph: “*Overview also works in 2D. If I look at it now, it does not look too different than if looking at an image.*” The data scientist also appreciated that she could get an overview first: “*I can see many many interesting things and then when I decide what will be the focus, then I can go into it and look at it, for example, from perspective of overweight [a node in the network].*”

Both users switched to the detail perspective to explore the network from various nodes at least once in the course of the study. Both users found this perspective particularly interesting. The physicist explained that he “*can browse these hubs more easily than in 2D*”. Indeed, known facts reported while being in the detail perspective were about connections to single nodes. For instance, the data scientist explored the nodes directly connected to obesity: “*I can see what will happen when I eat too much. So I will get diabetes [...], I will be depressed. Of course. Ok, I will have sleep disorder. So these things are quite known. Its super interesting to see them from this point of view.*”

While the physicist did not report any unexpected findings, the data scientist reported a few previously unknown discoveries. Around half of these discoveries thereby were made in the detail perspective. For instance, she found that some cancers were separated from the main cancer cluster and that endocrine diseases are quite separated from other diseases. From the overview perspective, she was surprised to see that mental diseases seem to be the most connected cluster in the network and that some diseases are far away from other diseases, like tonsillitis.

Both users reported they could not see all edges and their encoded weights when observing the graph from a node point of view in the detail perspective. Also, they both expressed the wish to be able to easily traverse the network specifically along directed edges in the network to be able to follow a “*patient ‘career’ from the perspective of the diseases.*” What both users appreciated was the ability to easily obtain the labels of the nodes. The data scientist explained that only in VR, it is possible to render all the edges and still to be able to read all the associated labels.

From our case study it seems that users are more prone to receive symptoms of motion sickness in the detail perspective than in the overview. During her case study, the data scientist reported three times that her “*brain is confused*” after moving

in the detail perspective. Twice, this happened after free flying, once after an animated teleportation to the neighboring node. She did not report any motion sickness symptoms when rotating the network in the overview perspective. It should be noted that the overall well-being and engagement during the study appeared to be very high, since the data scientist was very surprised to hear that she spent over half an hour exploring her graph in VR. After the study, she reported that to her it seemed that her session had only lasted about five minutes.

C. Lessons Learned and System Limitations

Our case study has shown the potential of combining a well-known overview of a network with a first person detail perspective, which is only possible in an immersive environment. The overview thereby represents a known reference frame, while the detail provides a new, egocentric perspective into the network. We found that the detail perspective facilitates the analysis of local neighborhoods, especially when the nodes of interest lie within a dense part of the network. Users may discover unexpected information in known networks from such an unknown perspective, such as nodes separated from their main cluster or unexpected connections between nodes. User feedback also indicates room for improvement for such detailed network perspectives: the weight of the edges should be better visible and flying to adjacent nodes can lead to motion sickness symptoms.

The presented case study has to be considered as early feedback. The limitations are, besides the limited number of users and the single data set, that both users knew the data set before. However, we believe that this is a common scenario in the case of network analytics, where VR will most likely represent a complementary tool rather than the only way of how to visually inspect a network. To evaluate the role of VR in data analysts’ workflows and to improve a fluid transition between desktop and VR analysis, longer-term field studies will be necessary. In the future, it will also be necessary to formally compare the types of insights users gain from exploring networks solely from a detail, solely from an overview, or from a switchable overview&detail navigation in a controlled user study. Our expectation is that being able to switch between those two views will strengthen the users’ understanding of the network topology. The fact that our case study users switched between those views and reported facts both, from the overview and the detail perspective, are first indications of the strength of the approach.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a new approach to facilitate network analytics by combining well-known exocentric overviews with an immersive detail perspective in VR. This immersive perspective provides insights into the network that are not possible in a conventional desktop setting. Our case study has shown that users find this new detail perspective interesting and that it supports them in gaining new, detailed insights into known data. The case study also indicated that the

initial overview is important to stay oriented. Our web-based implementation thereby enables fluid exploration of networks wherever an HMD-based VR setup is available, which is an important step to make immersive network analytics accessible to a broad spectrum of users. Our results open up new exciting research opportunities for graph drawing in VR – especially in the context of immersive detail inspection of networks.

ACKNOWLEDGMENTS

This work was supported by the FFG Project 857136 at CSH Vienna, as well as the Research Cluster “Smart Communities and Technologies (Smart CT)” at TU Wien.

REFERENCES

- [1] I. E. Sutherland, “A head-mounted three dimensional display,” in *Proceedings of the December 9-11, 1968, fall joint computer conference, part 1*. ACM, 1968, pp. 757–764.
- [2] L. A. Crutcher, A. A. Lazar, S. K. Feiner, and M. X. Zhou, “Managing networks through a virtual world,” *IEEE Parallel & Distributed Technology: Systems & Applications*, vol. 3, no. 2, pp. 4–13, 1995.
- [3] C. Ware and G. Franck, “Evaluating stereo and motion cues for visualizing information nets in three dimensions,” *ACM Transactions on Graphics (TOG)*, vol. 15, no. 2, pp. 121–140, 1996.
- [4] C. Ware and P. Mitchell, “Visualizing graphs in three dimensions,” *ACM Transactions on Applied Perception (TAP)*, vol. 5, no. 1, p. 2, 2008.
- [5] D. Belcher, M. Billingham, S. Hayes, and R. Stiles, “Using augmented reality for visualizing complex graphs in three dimensions,” in *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings*. IEEE, 2003, pp. 84–93.
- [6] N. Greffard, F. Picarougne, and P. Kuntz, “Visual community detection: An evaluation of 2d, 3d perspective and 3d stereoscopic displays,” in *International Symposium on Graph Drawing*. Springer, 2011, pp. 215–225.
- [7] H. Halpin, D. J. Zielinski, R. Brady, and G. Kelly, “Exploring semantic social networks using virtual reality,” in *International Semantic Web Conference*. Springer, 2008, pp. 599–614.
- [8] O.-H. Kwon, C. Muelder, K. Lee, and K.-L. Ma, “A study of layout, rendering, and interaction methods for immersive graph visualization,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 7, pp. 1802–1815, 2016.
- [9] K. Marriott, J. Chen, M. Hlawatsch, T. Itoh, M. A. Nacenta, G. Reina, and W. Stuerzlinger, “Immersive analytics: Time to reconsider the value of 3d for information visualisation,” in *Immersive Analytics*. Springer, 2018, pp. 25–55.
- [10] D. Raja, D. Bowman, J. Lucas, and C. North, “Exploring the benefits of immersion in abstract information visualization,” in *Proc. Immersive Projection Technology Workshop*, 2004, pp. 61–69.
- [11] C. Demiralp, C. D. Jackson, D. B. Karelitz, S. Zhang, and D. H. Laidlaw, “Cave and fishtank virtual-reality displays: A qualitative and quantitative comparison,” *IEEE Transactions on Visualization & Computer Graphics*, no. 3, pp. 323–330, 2006.
- [12] M. Cordeil, T. Dwyer, K. Klein, B. Laha, K. Marriott, and B. H. Thomas, “Immersive collaborative analysis of network connectivity: Cave-style or head-mounted display?” *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 441–450, 2016.
- [13] J. A. Wagner Filho, M. F. Rey, C. Freitas, and L. Nedel, “Immersive analytics of dimensionally-reduced data scatterplots,” in *2nd Workshop on Immersive Analytics*. IEEE, 2017.
- [14] J. A. Wagner Filho, M. F. Rey, C. M. Freitas, and L. Nedel, “Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2018, pp. 483–490.
- [15] D. A. Bowman and R. P. McMahan, “Virtual reality: how much immersion is enough?” *Computer*, vol. 40, no. 7, pp. 36–43, 2007.
- [16] K. Marriott, F. Schreiber, T. Dwyer, K. Klein, N. H. Riche, T. Itoh, W. Stuerzlinger, and B. H. Thomas, *Immersive Analytics*. Springer, 2018, vol. 11190.
- [17] R. Sadana, V. Setlur, and J. Stasko, “Redefining a contribution for immersive visualization research,” in *Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces*. ACM, 2016, pp. 41–45.
- [18] N. Osawa, K. Asai, and Y. Y. Sugimoto, “Immersive graph navigation using direct manipulation and gestures,” in *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 2000, pp. 147–152.
- [19] A. Forsberg, K. Herndon, and R. Zeleznik, “Aperture based selection for immersive virtual environments,” in *ACM Symposium on User Interface Software and Technology*. Citeseer, 1996, pp. 95–96.
- [20] Y.-J. Huang, T. Fujiwara, Y.-X. Lin, W.-C. Lin, and K.-L. Ma, “A gesture system for graph visualization in virtual reality environments,” in *2017 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2017, pp. 41–45.
- [21] U. Erra, D. Malandrino, and L. Pepe, “Virtual reality interfaces for interacting with three-dimensional graphs,” *International Journal of Human-Computer Interaction*, vol. 35, no. 1, pp. 75–88, 2019.
- [22] A. Drogemuller, A. Cunningham, J. Walsh, W. Ross, and B. H. Thomas, “Vrige: exploring social network interactions in immersive virtual environments,” in *Proceedings of the international symposium on big data visual analytics (BDVA)*. IEEE NJ, USA, 2017.
- [23] R. Brath, “3d infovis is here to stay: Deal with it,” in *2014 IEEE VIS International Workshop on 3DVis (3DVis)*. IEEE, 2014, pp. 25–31.
- [24] A. Drogemuller, A. Cunningham, J. Walsh, M. Cordeil, W. Ross, and B. Thomas, “Evaluating navigation techniques for 3d graph visualizations in virtual reality,” in *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*. IEEE, 2018, pp. 1–10.
- [25] M. R. Mine, F. P. Brooks Jr, and C. H. Sequin, “Moving objects in space: exploiting proprioception in virtual-environment interaction,” in *SIGGRAPH*, vol. 97, 1997, pp. 19–26.
- [26] W. Büschel, S. Vogt, and R. Dachselt, “Augmented reality graph visualizations,” *IEEE computer graphics and applications*, vol. 39, no. 3, pp. 29–40, 2019.
- [27] O.-H. Kwon, C. Muelder, K. Lee, and K.-L. Ma, “Spherical layout and rendering methods for immersive graph visualization,” in *2015 IEEE Pacific Visualization Symposium (PacificVis)*. IEEE, 2015, pp. 63–67.
- [28] Y. Yang, B. Jenny, T. Dwyer, K. Marriott, H. Chen, and M. Cordeil, “Maps and globes in virtual reality,” in *Computer Graphics Forum*, vol. 37, no. 3. Wiley Online Library, 2018, pp. 427–438.
- [29] C. C. Aggarwal, “An introduction to social network data analytics,” in *Social network data analytics*. Springer, 2011, pp. 1–15.
- [30] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, “Network medicine: a network-based approach to human disease,” *Nature reviews genetics*, vol. 12, no. 1, p. 56, 2011.
- [31] S. Miksch and W. Aigner, “A matter of time: Applying a data-users-tasks design triangle to visual analytics of time-oriented data,” *Computers & Graphics*, vol. 38, pp. 286–290, 2014.
- [32] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, “Task taxonomy for graph visualization,” in *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*. ACM, 2006, pp. 1–5.
- [33] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *Proceedings 1996 IEEE symposium on visual languages*. IEEE, 1996, pp. 336–343.
- [34] “d3-force-3d,” <https://github.com/vasturiano/d3-force-3d>, online; accessed 07-September-2019.
- [35] M. Bostock, V. Ogievetsky, and J. Heer, “D3 data-driven documents,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.
- [36] M. Usuh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr, “Walking, walking-in-place, flying, in virtual environments,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 359–364.
- [37] E. Langbehn, P. Lubos, and F. Steinicke, “Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking,” in *Proceedings of the Virtual Reality International Conference-Laval Virtual*. ACM, 2018, p. 4.
- [38] K. R. Moghadam, C. Banigan, and E. D. Ragan, “Scene transitions and teleportation in virtual reality and the implications for spatial awareness and sickness,” *IEEE transactions on visualization and computer graphics*, 2018.

- [39] “three.js javascript 3d library,” <https://store.steampowered.com/steamvr>, online; accessed 06-September-2019.
- [40] “A-frame: Hello webvr,” <https://aframe.io>, online; accessed 10-September-2019.
- [41] “Aframe forcegraph component,” <https://github.com/vasturiano/aframe-forcegraph-component>, online; accessed 07-September-2019.
- [42] “Steamvr,” <https://threejs.org/>, online; accessed 10-September-2019.
- [43] P. Erdős and A. Renyi, “On random graphs,” *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.
- [44] A. Chmiel, P. Klimek, and S. Thurner, “Spreading of diseases through comorbidity networks across life and gender,” *New Journal of Physics*, vol. 16, no. 11, p. 115013, 2014.
- [45] M. Bastian, S. Heymann, and M. Jacomy, “Gephi: an open source software for exploring and manipulating networks,” in *Third international AAAI conference on weblogs and social media*, 2009.