Spatial Big Data Visualization and Manipulation

Maria Gliga-Hambet
Technical University of Cluj-Napoca
Str. Memorandumului 28
gligaa.maria@gmail.com

Dorian Gorgan
Technical University of Cluj-Napoca
Str. Memorandumului 28
dorian.gorgan@cs.utcluj.ro

ABSTRACT
This paper explores the visualization and manipulation of spatial big data in a Virtual Reality (VR) environment. The solution allows users to interactively visualize, manipulate and understand complex datasets. It aims to improve interpretability of data by leveraging the immersive qualities of VR. The tasks involve describing sequences of operators that work together to achieve a specific user goal. Experimental validation evaluates these operators in terms of functionality, usability, performance, and scalability.

Author Keywords
Multidimensional data; data visualization; virtual reality (VR); data manipulation; user experience (UX).

ACM Classification Keywords
H.5.m. Information interfaces and presentation.

General Terms
Human Factors; Design; Virtual Reality.
DOI: 10.37789/icusi.2024.13

INTRODUCTION
With the increased use of data-driven approaches in the decision-making process, big data analysis and manipulation has played a vital role in how various industries have transformed in the past couple of years. Big data refers to large volumes of data, which are complex in their volume, velocity, and variety.

Even though running such large datasets through machine learning models can reveal various patterns, there is also the need for human involvement to make the process even more efficient, as presented in [9] and [14]. This is because computers do not understand the meaning behind data and numbers. Here is where the concept of Human-in-The-Loop (HITL) becomes valuable by introducing people's input and expertise into the decision-making process of artificial intelligence systems.

As it is presented in [11], humans best understand data when there is a visual representation of it. Thorpe's work [12] presents that the human brain can recognize and process visual information in as little as 150 milliseconds, much faster than it can process text or numbers. This phenomenon is called pre-attentive processing, and it allows us to immediately recognize and understand visual cues, giving us a quicker comprehension of complex data sets. Visual representations make it possible for us to rapidly detect patterns, trends, and anomalies. Attributes such as shape, size, color, and position offer valuable insights about objects and the relationships between them.

One manner of creating such a visual representation is by means of a Virtual Reality (VR) application. Their popularity has increased over the past years, offering the user a fully immersive experience. In the context of data analysis and manipulation, such an environment is particularly useful for the exploration of three-dimensional or more complex data, as shown by previous works [6], [10]. Immersive environments offer significant benefits over traditional 2D visualization methods.

The research aims to design and experiment a Unity VR application for Oculus Quest 2 and 3, which allows the user to perform various tasks relating to the visualization and manipulation of data. Defining these tasks involves outlining sequences of operators that work together to achieve a specific user goal. These operators will be tested against functionality, performance, scalability, and efficiency. The research makes use of the immersive capabilities of virtual reality to provide efficient data exploration and analysis.

RELATED WORK
The virtual reality environment for big data visualization and analysis can be highly efficient. Our research explores and evaluates solutions which are reported in the scientific literature.

iViz
One such solution is reported in [3], in which the authors explore the usage of VR environments for visualizing and analyzing multi-dimensional data. Their Unity 3D based prototype is called iViz and allows the user to observe up to 10^6 data points and customize how they are plotted based on the attributes. On top of this, it provides a collaborative feature through which multiple users can examine the data in a shared view. The authors tested the prototype in an experiment conducted with geologists, which were presented with data from Mars. The group using the immersive solution performed better in measuring distances and orienting in space.

LookVR
LookVR is a virtual reality tool, available on Steam [7]. It allows the exploration and interaction with data from Looker, which is a business analytics platform. The platform itself allows customers to gather data from multiple databases, which can be of more than 50 types. LookVR enables the visualization of this data through VR-specific visualizations, as well as their exploration to derive insights. It connects to the Looker API, which ensures real-time access to the data. The interactive part of the system relies on

choosing the plot type and its scale, as well as filtering options.

3Data

3Data, previously known as DatavizVR, is another virtual reality data visualization software, available at [4] The platform currently has a beta version available for free on Steam and Meta and it allows users to plot, manipulate and analyze data from previously loaded datasets. The platform also supports WebVR, which means that the platform can run not only on VR headsets, but also on mobile devices and personal computers. For the beta version, there is only the single user mode, but a collaborative feature is provided with the 3data license.

Virtualitics

Virtualitics [13] is a more advanced analytics platform in comparison to the previous options. It facilitates the integration of artificial intelligence (AI) in the context of VR visualizations. Insights are derived from complex data by means of multi-dimensional visualizations and AI-powered analytics. The platform has a Windows application which allows operations on data either from a desktop or VR mode. On top of that, Virtualitics offers a Python API, which enables developers to integrate it into their projects. In comparison to LookVR, Virtualitics is not as strict regarding the source of the data, which can be either a local file or a remote database. The platform has two view options: a spreadsheet view, which offers a tabular representation of the data, and a mapping view, which enables users to map attributes onto dimensions.

VISUALIZATION PIPELINE

We considered the visualization pipeline presented by M. Kraus et al. in [8] a suitable approach for our solution as well, as it focused on the way the user understands the data and acts accordingly. Below there is a visual reinterpretation of their proposed pipeline in Figure 1, together with an explanation of each of the stages.

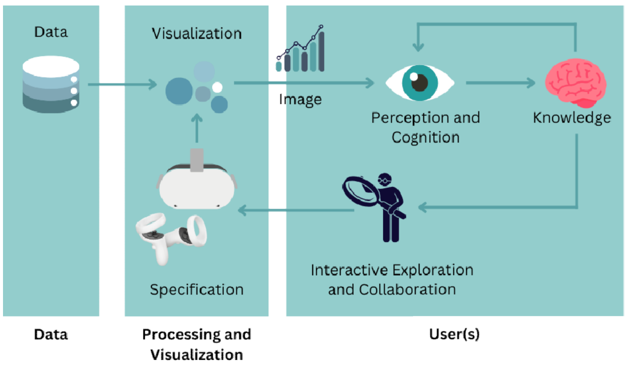

Figure 1: Abstract Visualization Pipeline

- The Data phase represents the input for the visualization pipeline. The choice of data types and data storage methods regarding the visualization can affect the performance of the system. Therefore, finding the storage method that is aligned with the system's needs is important. Some data, such as spatial data or 3D data is displayed more naturally in a VR environment, as compared to traditional options.
- Visualization is the stage at which data is transformed and mapped into visual representations that can be better understood by the user. The output of this stage is the "image" which will be observed in the VR environment.
- The image is the result of the visualization stage, and it is most often represented by a 2D, or 3D image, but other feedback methods such as audio and haptic are also used. This transformation is essential for a later proper understanding of the data.
- The Perception and Cognition step involves how the user perceives and processes the visualization. One user may perceive the" image" differently from another due to individual differences such as interpretation and physical sensory capabilities. The immersive nature of the environment has positive effects on analysis tasks.
- The Knowledge is the outcome of the pipeline and is derived from the visualization. It depends on the user's prior experience and understanding on the visualization's domain and therefore can vary from one person to another.
- The Interactive Exploration and Collaboration describes the user's interaction with the system and the visualizations, as well as the collaboration with others (which is not the case for our solution in the current implementation).
- Specifications refers to the visualization type, dataset considerations and mapping of data attributes. At the same time, they include hardware and environmental conditions.

DATA MAPPING IN 3D SPACE

Data mapping in 3D space involves transforming datasets into three-dimensional representations, adding depth to traditional 2D visualizations. Adding this third dimension, the users can see the data from multiple perspectives which improves their understanding of it. Increasing the number of attributes that are represented in relation to one another also makes trends more visible and the user can make more informed decisions.

There are multiple techniques of plotting the data such as scatter plots, bar charts and line charts, volume rendering and heatmaps. Choosing the exact approach depends on various factors including data types, dimensions, and analysis tasks. Because our data is numerical, we decided that the graphs generated by the application will be scatter plots, which can easily be transformed later into heat maps as well. The latter allows for 4 dimensions to be represented at the same time: 3 on the axes and the fourth one by color value.

Despite the benefits of simplifying complex information and helping with the identification of patterns, 3D visualization can also lead to visual clutter and distortions. In the context of VR environments, data representations can be distorted and lead to errors due to problems with depth perception. Depth perception refers to the ability to visualize objects in three dimensions, as well as to decide how far away that object is.

DESIGN CONSIDERATIONS IN VR ENVIRONMENTS

As previously stated, Virtual Reality environments give the user a sense of immersion inside the digital world. This is achieved by simulating the real world by providing visual, audio, and tactile feedback, which must align perfectly with what our brains are usually accustomed to. In order to achieve this, a number of design considerations must be taken into account, in comparison to visualizations aimed for a normal display. Not following them could not just misguide the user's behavior and reduce the immersion level, but also cause unwanted perception issues and VR sickness.

VR Sickness

Even though the technology has advanced a lot, it is still common for users to experience various sickness symptoms when using a VR headset. As it is described in Chapter 11 of [5], VR sickness is a general term for motion sickness, cybersickness and simulator sickness. Even though they are similar, the terms are not completely synonyms:

- Motion sickness is related to the symptoms and observable signs that appear after the exposure to real (physical or visual) and/or apparent motion, as it is presented in [2].
- Cybersickness refers to motion sickness caused by immersion in a digital world, where the user experiences the sensation of movement through visual stimuli despite remaining physically stationary.
- Simulator sickness refers to sickness resulting from apparent motion that does not match physical motion but can also be caused by accommodation-vergence conflict and flicker.

VR INTERACTION PATTERNS

As presented by Jerald in [5], an interaction pattern is a high-level, generalized interaction concept that may be applied in many contexts to achieve similar user goals. It is important to differentiate them from software design patterns, with which people might confuse them. Interaction patterns are described from the point of view of the user and are not dependent on the implementation. An interaction technique, on the other hand, is more specific and more related to the implementation part. Techniques which are similar can be grouped under the same pattern.

The distinction between the two is important as higher-level organization of the concepts makes analysis and comparison easier and when a certain technique fails, others from the same pattern may be a better fit.

Selection implies choosing out of a set of objects on which one or ones you want to apply an operator on, to mark the start of a manipulation task. Selection can create confusion in a virtual reality environment, especially when objects are located far from the user. As a selection pattern we are employing the pointing pattern, which is suitable for selecting objects in the distance or when there is the need for a precise movement.

Manipulation implies changing attributes for one or more objects. These attributes can be anything from position to color and texture. Normally, manipulation is next after selection - in order to do something with an object, you first need to have it in your possession. The manipulation pattern used is the proxy pattern, in which the selected object acts as if it is directly in the user's hand. This can also make the pattern more difficult to use, however, because of mismatches in position and orientation.

Viewpoint control implies changing the user's perspective. It can include moving, rotating and scaling the world, as well as traveling. When controlling the viewpoint in a virtual reality environment, it is important to consider motion sickness and manner by which it can be reduced. There were two patterns used regarding this, in order to make the application comfortable for everyone and also inclusive:

- Walking Pattern, which provides real walking movements to control the virtual camera. It is best suited for smaller navigating worlds because even though it offers an immersive experience, it can also cause fatigue.
- Steering Pattern, which makes use of the controls such as hand movements or the controllers to direct the movement. Large distances can be navigated easily, and the lack of physical space is not an issue anymore.

In comparison to the previous patterns, indirect control is more abstract. It gives control to an intermediary to modify the environment or system when clear spatial mapping is difficult to be obtained. Such an example is the Widgets & Panels Pattern. It uses 2D desktop widgets (buttons, sliders, and labels) and panels (containers for multiple widgets) in VR. They are useful when direct interaction is too complex and imprecise. Their placement, orientation and transparency must be carefully designed in order not to obscure the user's view.

VR SYSTEM

Currently, there are several headsets which are similar regarding their features and functionalities. Some of the most popular ones include Meta Quest Pro, Meta Quest 2, HTC Vive Focus 3, and Valve Index. Meta Quest 2, previously known as Oculus Quest 2, was the chosen virtual reality headset for which the platform has been primarily created and tested upon.

For the integration with Unity, which is the engine used for developing the solution, the following options were considered:

- Oculus SDK is created specifically for the development of applications that will run on Oculus VR devices. Therefore, its libraries and tools are tailored to the Oculus hardware. This way, the hardware's capabilities are used in the best way possible.
- SteamVR is a runtime for games and other applications that are using various VR platforms, including the Oculus ones. It communicates with OpenVR - an API and SDK by Valve that allows hardware access from multiple vendors without knowing specific details. OpenVR acts as a mediator between applications and SteamVR and helps in handling different VR devices.

Since the Oculus SDK is optimized for Oculus devices, providing extensive support and integration for them, we decided to opt for this option since the experiment's purpose is rather focused on finding great visualization and interaction techniques rather than creating a solution that fits various headsets.

In order to integrate the Oculus SDK, we made use of Unity's XR Interaction Toolkit and XR Plugin Management. Installing the two plugins can be done in the Unity Package Manager. Once the installation is complete, the next step must be completed from the project settings section. In the XR Plugin Management page, the Oculus option must be the one enabled. This imports a set of files to the project that allow us to use the Oculus SDK.

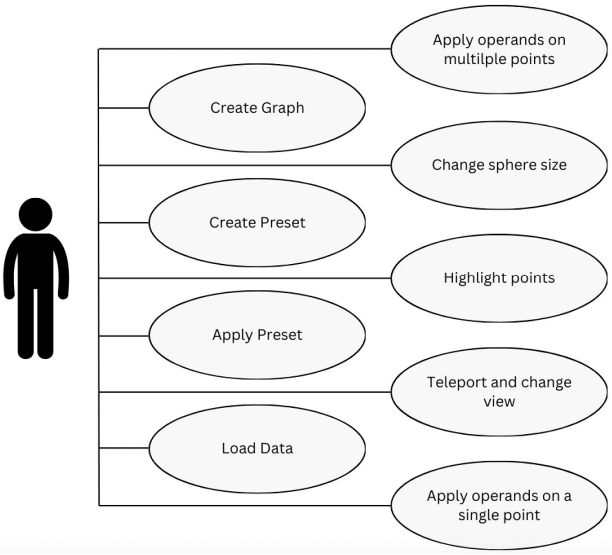

Figure 2. Application use case diagram

CONSIDERATION ON DATA INTERACTION

The proposed solution focuses on implementing the use cases presented in Figure 2. In order for a smooth implementation, certain considerations regarding data management, selection and manipulation had to be taken into account.

Data Management

The first step in the user's interaction with the application is selecting the input files for the data and its limits, as well as an output file where the results are stored at the end of the session. This is done by means of drop-down menus, which are populated with all the .CSV file names from the path previously assigned in the application. Since the input data set is considered to be complex, the users can select only the attributes they are interested in analyzing. The user can opt to apply additional constraints for each attribute, which will replace the limits read from the file and filter out the points. The limits file provides the minimum and maximum values for each of the attributes. They are used to normalize the attributes and plot them inside the virtual world, with reference to the graph's origin and limits.

Data Visualization

After selecting the subset of the dataset on which they want to operate, the user is presented with the main menu, which can be seen in Figure 3. This menu remains accessible through the whole session as it contains important operands, such as creating a new graph and changing the height from

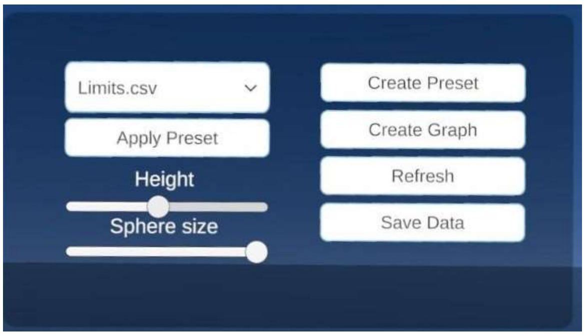

Figure 3. User interface of the main menu

which the user sees the environment. Since the menu should be always available, but also not interfere with the visualization, we have opted from enabling its on and off states by the rotation angle of the left controller. When the user rotates the left controller by an angle of at least 30 degrees in any direction, the menu will become visible. Its position in the world is calculated regarding the user's view, placing it directly in front of them, but at a distance that makes selection possible and also comfortable to the eyes.

From the user's perspective, creating a new graph implies selecting an attribute for each of the axes. For this to be possible, the system firstly populates the dropdowns' options dynamically with the attributes selected by the user. Upon pressing the "Create Graph" button, the method that implements this in the background contains the following steps: creating the graph itself and associating it with its menu. The tasks performed on each graph (be that filtering, modifying, or deleting) are initiated from the individual menu of each graph. When instantiating it, we made sure to take the angle from which the user is looking at the graph into account. This is important for them to use the menu without having to move around. Figure 4 shows an example of a plotted graph with and without a highlight operation applied on it.

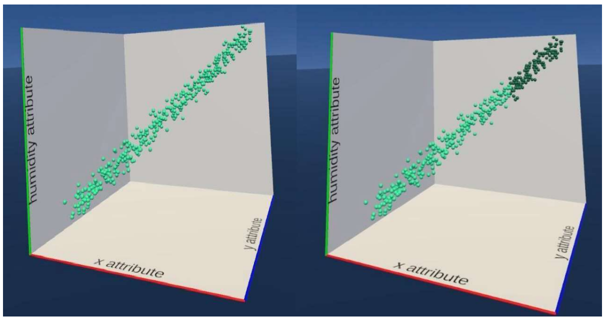

Figure 4. Plotted graph without and with highlight operation

The individual menu of each graph allows operations within that graph. For each operation, the menu has a different page, which also allows the return to the main one. The transition from one state to another is made with methods placed on the buttons to activate or deactivate certain pages. For the rendering in the scene of the menu, we are using a previously created prefab, which is referenced through the Inspector and instantiated. The graph menu also allows the user to delete the representation in case they do not find it useful anymore. For implementing this, we are creating an empty game object which is the parent of all the objects relating to one graph - walls, axes, data points and menu. In order to be able to access the empty game object, we are saving a reference to it in the graph instance. Upon clicking the "Delete Graph" menu, the object is destroyed and so are all its children in the hierarchy. The position of the other graphs remains unchanged and the slot where the delete graph was remains empty from there on.

Data Selection

Data selection is performed previously to any manipulation task. In order to give the user as much freedom as possible, they can perform selection both by clicking manually on each data point game object and by applying a filtering task on the whole graph. Of course, the two methods of selection can also be combined. Initially, the data points were designed to be green, and the selection was marked by changing their color to red. We realized that this could cause color blind people issues, especially if the shades of the colors are similar, which they were. To solve this issue, we decide to darken the hue already applied on the data points. This solves the issue, as the problematic part comes from hues that appear like one another.

Data Manipulation

The operations performed on a graph can either be made on a single data point or on multiple data points simultaneously. For the latter, the change with which the user can modify the values is given by the following rules, which ensures that the value with which each point is changed is the same and the limits of the graph are still respected:

- Move upwards with a maximum of maxAxis - maxPoints
- Move downwards with a maximum of minPoints - minAxis

In order to improve the user's experience, the changes made with the help of the sliders are seen in real time. This gives the user a preview of how the graph would look like, before saving the changes. On top of that, the changing value of the slider can also be seen by the user in the text box associated with the slider so that the precise value is always known.

Applying and Creating Presets

A preset is an ordered set of operations, which can be applied on any data set which contains at least the three attributes on which the preset is defined. This sort of functionality becomes useful when the user observes a combination of operands which yields great visualizations. By creating a preset with them, they do not have to manually perform each operation every time, which is not only time-consuming, but also prompts errors. Both the creation of new presets and using already existent presets can be performed from the main menu. The main mechanism used to save the presets in between sessions is storing the information outside the application, in .CSV files. Each collection of operands and their exact order is saved in an individual file. For creating a preset, the user firstly selects the attribute for each axis. After that, they are presented with a drop-down menu with the possible operations. Upon selection, a second window opens for inserting the required parameters. After providing the parameters, the operation is added to a scroll-able list so that the user can keep track of what has been added so far.

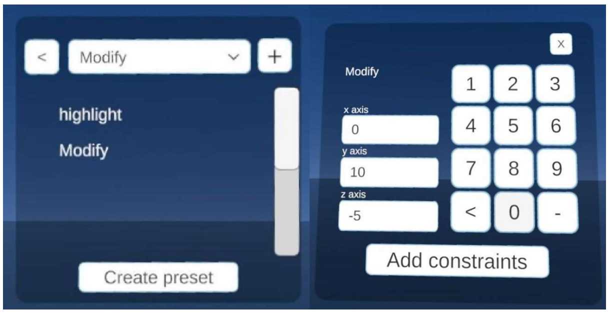

Figure 5. User interfaces for creating a preset

EXPERIMENTAL VALIDATION

The evaluation has been performed by two computer science graduates, both with little prior VR experience. One of them is a member of the development team, having work experience in the UI/UX domain as well. For the evaluation, we have decided on a summative evaluation, having the following criteria in mind: functionality, usability, performance, and scalability. As it is presented in [1], a summative evaluation provides the big picture and assesses the general experience of the finished product.

For each of the four directions in which we are performing the evaluation, we decided on a set of measurable attributes, which are essential for providing a qualitative experience. The attributes were chosen based on the proposed functioning of the system, combined with factors that offer a smooth user experience in a digital world. They are presented in Tables 1 to 4, together with the average score obtained from the two evaluators. On top of considering the absolute values of 0 and 100 as benchmarks, there are certain attributes which can be also partially fulfilled. Because of that, another interpretation for acceptability levels is required:

- Unsatisfactory: $0\% \leq EQ < 40\%$
- Medium: $40\% \leq EQ < 60\%$
- Satisfactory: $60\% \leq EQ \leq 100\%$

In order to obtain accurate results, we are evaluating the system in multiple scenarios, in which the complexity of the data varies in terms of the number of columns and rows:

- Scenario 1: 100 data points, with 4 variables
- Scenario 2: 100 data points, with 100 variables
- Scenario 3: 1,000 data points, with 4 variables
- Scenario 4: 1,000 data points, with 100 variables
- Scenario 5: 5,000 data points, with 4 variables
- Scenario 6: 5,000 data points, with 100 variables

Functionality attribute	Score
1.1.1. Supports dynamic creation and rendering of multiple graphs based on selected data attributes.	100
1.1.2. Features precise graph customization and axis labeling according to normalized data values.	75
1.1.3. Incorporates real-time updates to ensure that changes in data are immediately reflected in visualizations.	85
1.2.1. Enables users to modify, remove or filter data points directly within the virtual reality environment.	75
1.2.2. Supports easy application and removal of data transformations for flexible data analysis.	85
1.2.3. Provides mechanisms for immediate reflection of data manipulations in visual outputs.	95
1.3.1. Offers advanced data filtering capabilities, allowing users to apply conditional expressions on graph axes.	70
1.3.2. Allows users to create, save, and reuse custom filter settings for enhanced data exploration.	85

Table 1. Evaluated attributes for Functionality

Usability attribute	Score
2.1.1. Provides an intuitive interface for VR navigation and data interaction, tested for user ease and efficiency.	90
2.1.2. Offers improved control mechanisms based on user feedback to enhance navigation within the VR environment.	85
2.1.3. Includes effective navigation aids to assist users in orienting and moving within the virtual space.	0
2.2.1. Ensures high readability of graphs from various viewpoints and under different user conditions.	85
2.2.2. Adapts visualizations for accessibility, including support for color blindness.	100
2.3.1. Focuses on user comfort during extended VR sessions to minimize VR fatigue and discomfort.	100
2.3.2. Adjusts interaction dynamics to mitigate VR sickness, enhancing user experience.	90

Table 2. Evaluated attributes for Usability

Performance attribute	Score
3.1.1. Optimizes graph rendering times to handle large datasets swiftly while maintaining high performance.	70
3.1.2. Ensures responsive interactions, with system reactions within 50 milliseconds for all user actions.	80

3.2.1. Maintains a minimum frame rate of 60 frames per second (FPS) in VR, even under heavy data loads.	75
3.2.2. Stress tests system to ensure stability and optimize frame rates for smooth operations.	80
3.3.1. Implements efficient CPU and GPU usage strategies.	70
3.3.2. Employs advanced data handling and visualization techniques to optimize memory usage and resource allocation.	80

Table 3. Evaluated attributes for Performance

Scalability attribute	Score
4.1.1 Demonstrates robust performance with escalating data volumes, ensuring system scalability.	70
4.1.2 Utilizes optimized data structures and algorithms to handle large datasets efficiently.	70
4.2.1 Handles multiple graphs without performance degradation.	80
4.2.2 Features to manage graph overlap and maintain user focus are effective.	90
4.3.1 System architecture supports easy addition of new features.	85
4.3.2 Integration of new features is seamless and minimally impacts system performance.	90

Table 4. Evaluated attributes for Scalability

All the attributes are considered to have the same influence over the final score of a characteristic so after performing this evaluation, we obtain the following averages, all of them being in the "Satisfactory" category:

- Functionality: 83.75%
- Usability: 78.57%
- Performance: 75.83%
- Scalability: 80.83%

The results point out that the usability characteristic has one of the lowest scores (Figure 6). Its score was, however, influenced by one of the attributes missing altogether from the implementation and obtaining a score of 0. The attribute referred to navigation aids in the virtual world but were not considered a priority in the implementation since the interaction with the entire system is straightforward.

Regarding the performance, for the first 4 scenarios, the project has more than 60 FPS, which means that it functions seamlessly.

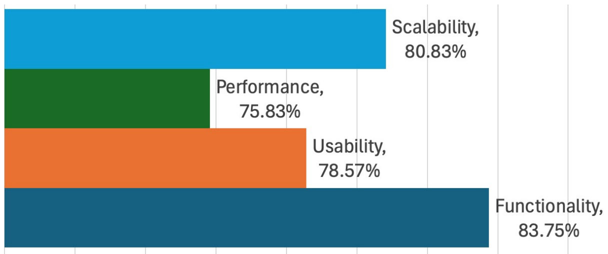

Figure 6. Evaluation criteria

The last 2 scenarios, however, functioned at under 30 FPS, which is starting to cause issues for the user. Even though the scene is not a dynamic one and 30 FPS still provide a good experience, anything below it is not desirable. This is also the main reason why the performance characteristic has a score of only 75.83%.

This analysis points out that the system performs regardless of the number of columns the dataset has but starts to decrease its performance when there are more than 5.000 entries. The reason for this is that currently there is a new object created for each of the points. Even with changing the quality settings, the performance did not improve as the problem is on the CPU side, rather than on the GPU. That is because each point object has more complex logic assigned to it via the scripts.

On top of that, the evaluation highlights the need for more diverse operands that should be applied on the data. Even though the current version does successfully offer a good visualization, having more filtering options would lead to a better analysis. Because the mapping of the n-dimensional data inside the 3D world is achieved via multiple graphs, the user would benefit from being able to move the graphs around and change their position inside the virtual world. What is more, the current solution requires the user to manually refresh the system in order for changes to be seen in all graphs, which represents an extra step that should be avoided. One solution is the system refreshing itself at a certain time interval. The functionality could be increased by allowing the user to see how changing one of a point's attributes would affect the others.

In order to make use of the immersive environment even more, gesture based commands such as zooming in or out on the data by the user moving their hands closer or further apart would be a great addition to improve the user experience.

CONCLUSION

The application successfully plots complex data sets into an immersive environment, which facilitates a comprehensive analysis. VR design patterns were followed in the design and implementation stages, giving a pleasant and fluid experience for the user, which is extremely important in such environments in order to avoid VR sickness. Data analysis can be performed in detail due to the visualization settings, such as sphere size for the plotted points and height from which the graph is seen, as well as due to the large number of plots that can be created. For data manipulation, the user has multiple means to complete their tasks, which offers flexibility and allows the user to pick the best fit.

Regarding future developments of the solution, the following improvements are proposed. First, performance issues should be addressed, particularly for datasets with more than 5,000 entries, where current performance is lacking. Adding multiple plot types would be a simple but effective enhancement, offering diverse insights based on the data type. Customization features, such as scaling axes and moving graphs within the VR environment, would further enhance visualization. On top of that, the evaluation of the solution also points out that the visualization and manipulation of the data set could be further improved by adding more complex operands such as extended filtering options.

Secondly, usability can also be improved by adapting the user interface to the user's skill level and integrating gesture and voice recognition for more intuitive and hands-free interaction. Additionally, introducing a collaborative feature for multi-user interactions in VR would facilitate shared analysis and insights. Finally, incorporating machine learning algorithms could enhance trend identification and predictive analysis, further improving the system's analytical capabilities.

ACKNOWLEDGMENTS

This research was partially supported by the project 38 PFE in the frame of the programme PDI-PFE-CDI 2021.

REFERENCES

1. A. Joyce, Formative vs. Summative Evaluations, 2019. [Online]. Available: https://www.nngroup.com/articles/formative-vs-summative-evaluations/
2. B. Lawson, Motion Sickness Symptomatology and Origins, 09 2014, pp. 531–600.
3. C. Donalek, S. G. Djorgovski, A. Cioc, A. Wang, J. Zhang, E. Lawler, S. Yeh, A. Mahabal, M. Graham, A. Drake, S. Davidoff, J. S. Norris, and G. Longo, Immersive and collaborative data visualization using virtual reality platforms, in *2014 IEEE International Conference on Big Data (Big Data). IEEE*, Oct. 2014. http://dx.doi.org/10.1109/BigData.2014.7004282.
4. DatavizVR Demo on Steam. https://store.steampowered.com/app/551960/DatavizVR_Demo/. J. Jerald, The vr book: Human-centered design for virtual reality, 10 2015.
5. J. J. LaViola, Prabhat, A. S. Forsberg, D. H. Laidlaw, and A. v. Dam, Virtual Reality-Based Interactive Scientific Visualization Environments. *London: Springer London, 2009*, pp. 225–250. https://doi.org/10.1007/978-1-84800-269-2_10
6. LookVR on Steam. https://store.steampowered.com/app/595490/LookVR/.
7. M. Kraus, M. Miller, J. Buchmüller, M. Stein, N. Weiler, D. Keim, and M. El-Assady, Breaking the Curse of Visual Analytics: Accommodating Virtual Reality in the Visualization Pipeline, in *Comput. Vision, Imaging Comput. Graph. Theory Appl. Cham: Springer International Publishing, 2020*, pp. 253–284.
8. Mosqueira-Rey, Eduardo & Hernández-Pereira, Elena & Alonso-Ríos, David & Bobes-Bascarán, José & Fernández-Leal, Ángel. (2022). Human-in-the-loop machine learning: a state of the art. Artificial Intelligence Review. 56. 10.1007/s10462-022-10246-w.
9. P. O'Leary, S. Jhaveri, A. Chaudhary, W. Sherman, K. Martin, D. Lonie, E. Whiting, J. Money, and S. McKenzie, Enhancements to vtk enabling scientific visualization in immersive environments, in *2017 IEEE Virtual Reality (VR)*, 2017, pp. 186–194.
10. Software Freedom Conservancy, *The Psychology behind Data Visualization Techniques*, 2021. https://towardsdatascience.com/the-psychology-behind- data-visualization-techniques-68ef12865720
11. Thorpe, Simon & Fize, Denis & Marlot, Catherine. (1996). Speed of Processing in the Human Visual System. *Nature. 381.* 520-2. 10.1038/381520a0.
12. Virtualitics, Inc. https://virtualitics.com/.
13. Wu, Xingjiao & Xiao, Luwei & Yixuan, Sun & Zhang, Junhang & Ma, Tianlong & He, Liang. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems. 135.* 10.1016/j.future.2022.05.014.