An Immersive and Interactive Visualization System by Integrating Distinct Platforms

Mário Popolin Neto
Federal Institute of São Paulo – IFSP
Registro, São Paulo, Brazil
Email: mariopopolin@ifsp.edu.br

Danilo Medeiros Eler
São Paulo State University – UNESP
Presidente Prudente, São Paulo, Brazil
Email: daniloeler@fct.unesp.br

Alessandro Campanhã de Moraes
and José Remo Ferreira Brega
São Paulo State University – UNESP
Bauru, São Paulo, Brazil
Email: sanfatec@gmail.com, remo@fc.unesp.br

***Abstract*—Visualization applications can be performed on distinct platforms, such as mobile devices and multi-projection systems. Each platform offers specific features to provide further data understanding, and a system that integrates these platforms in a complementary manner is a real challenge. In this paper, we present an immersive and interactive visualization system that aims to explore data from relational databases using 3D graphs representations, where multiple simultaneous users can visualize and interact with the data through a multi-projection system and mobile devices. A single visualization application was created for both platforms using the Unity game engine, and an Unity external package for Virtual Reality applications development, that supports multi-projection system over a PC cluster and passive stereoscopy. Our visualization system aims to provide the users a better data understanding using a 3-screens multi-projection system as data overview, and mobile devices as display and interaction device for navigation and additional information visualization. We also introduce an user case, where the visualization system is used in order to support developers regarding structural problems in a large relational database.**

***Keywords*-Information Visualization; Virtual Reality; Unity game engine; multi-projection system; PC cluster; mobile devices;**

I. Introduction

The CAVE™ (Cave Automatic Virtual Environment) [1] is a multi-projection system created to provide Virtual Environments (VE) on multiple-screens to support not only Virtual Reality (VR) applications, but also Scientific Visualization applications. CAVE™-like systems became popular once the advances and low cost of high-performance graphic cards allowed these multi-projection systems over a PC cluster, where a set of computers are connected via a network for rendering purposes [2].

The multi-projection systems are used to reach high resolution images, wide field of view, and a better immersion into the VE [3]. The immersive experience can be useful and beneficial in the visualization of large datasets [4], and by using the wide field of view in the CAVE™, users performed better on exploration tasks of volume renderings, compared with other two different platforms (Desktop and Fishtank) [5]. Based on Augmented Visualization (AV), which is the process where hypothesis about the data are formed and refined in an interactive environment that allows data manipulation, exploration, and navigation [6], the wide field of view provided by multi-projection systems can also be used to explore large datasets through an overview on the multiple-screens and specific views on mobile devices, since nowadays tablets and smartphones are also able to render complex VEs in real-time [7].

The main contribution of this paper is to integrate distinct platforms in a system, through a visualization application that sits on top of a popular game engine, where multiple simultaneous users can visualize data in a shared immersive space (3-screens multi-projection system over a PC cluster) and obtain detailed information on demand in an individual workspace (mobile device). The multi-projection system provides a wide field of view used as an overview visualization while on the mobile device is possible to navigate through the data requesting additional information, complementing both platforms regarding immersion and touch interaction. The Unity game engine was used to create a visualization application that creates 3D graphs as data representations. Since Unity is a multi-platform game engine, the visualization application can be performed on PCs and mobile devices using the Unity network module for communication and synchronization. A Unity external package for VR applications development was imported into the game engine editor in order to develop the application for our 3-screens multi-projection system.

The remainder of the paper is organized as follows. Section II contains the background work related to this research. Section III describes the material and methods used: Unity game engine, Unity external package, and 3-screens multi-projection system. Section IV explains the immersive and interactive visualization system by describing the data extraction engine, 3D graph creation, and dynamics of visualization and interaction. Section V presents an user case followed by conclusions and future works.

1550-6037/15 $31.00 © 2015 IEEE
DOI 10.1109/iV.2015.74

II. Related Work

Computer science has been used as middleware for several areas supporting simulations, experiments, and data analysis. Data representation as computer-images aims to provide the user knowledge acquisition in an accurate and efficient way, and researchers have explored 3D data representations on immersive platforms for many purposes [4], [8]–[10].

Studies pointed a better user performance, regarding information visualization tasks on 3D scatter plots, in the immersive VE provided by CaveDataView [4]. CAVE-SOM [10] provides natural and intuitive interaction on 3D representations in immersive VEs, allowing information preservation concerning dimensionality reduction for 2D representations. Both visualization systems [4], [10] use a wand as interaction device and do not allow multiple simultaneous users as a head-track rendering matter, since the perspective correction of the VE requires the viewpoint of only one user. Selecting part of the 3D structure in order to obtain additional information can be troublesome using devices such as wands, and losing the representation overview on the navigation and information request can be a major issue, since disorientation and free movement can lead users to get lost in the VE [9].

The immersive experience in our system is achieved through a wide field of view and stereoscopy, allowing an overview of the 3D data representation to multiple simultaneous users in the multi-projection system. The overview visualization can serve as navigation orientation to users on mobile devices, where they can navigate on a particular display and access additional information on demand by touching on the screen at a point of interest.

The visualization system XIM [8] uses a 3-screens multi-projection system with different views of the data on each screen. The 3D graphs created by the Unity game engine are displayed using head-track rendering on the center screen, an overview of the 3D graphs are presented on the right screen, and real-time activity plots are displayed on the left screen. Besides not featuring multiple simultaneous users like in [4], [10], this visualization solution [8] does not allow requests for additional information, the user can only navigate using devices such as Nintendo Wii Remote™ and Microsoft Kinect™ having the overview on the right screen as navigation orientation.

We also used the Unity to create 3D graphs as data representations. Our solution uses all 3-screens from the multi-projection system as overview, not only for navigation orientation, but also to provide an overall visualization to identify clustering, patterns, trends, and outliers in the 3D representation, whereas on mobile devices users can navigate and get additional information on demand based on AV. The visualization and navigation dynamics applied in our system are based on the experiment [9], which evaluated the benefits that immersive VEs may have when presenting 3D graphs as data representations.

Differently of the systems mentioned above [4], [8]–[10], our solution results in an heterogeneous cluster with PCs (6 rendering nodes and 1 server node), from the 3-screens multi-projection system, and mobile devices (mobile nodes). All nodes performed the same visualization application developed using the Unity game engine and a Unity external package for VR applications development.

III. Materials and Methods

Unity game engine was chosen aiming our heterogeneous cluster (PCs and mobile devices), since it is a multi-platform game engine with intuitive workflows and custom rendering engine (Direct X and OpenGL). Unity [1] is a commercial game engine created and maintained by Unity Technologies, and supports game development for over than 10 platforms (Windows, Linux, Mac OS, iOS, Android, web browsers, and consoles).

Unity belongs to the Framework Modular Engine group [11], providing distinct modules as subsystems for the game engine, offering a good relation between customization and complexity. There are seven common modules in current game engines [12]: Graphics, Physics, Collision Detection, I/O, Sound, AI and Network. Our visualization application uses the Unity network module for communication and synchronization among PCs and mobile nodes. The Unity network module implements the Client-Server communication model, where Unity Clients connected to Unity Server can communicate according to the application dynamic using the Networkview component (Figure 1), which enables Unity Client applications to perform RPC (*Remote Procedure Call*) functions and synchronization of scene objects properties.

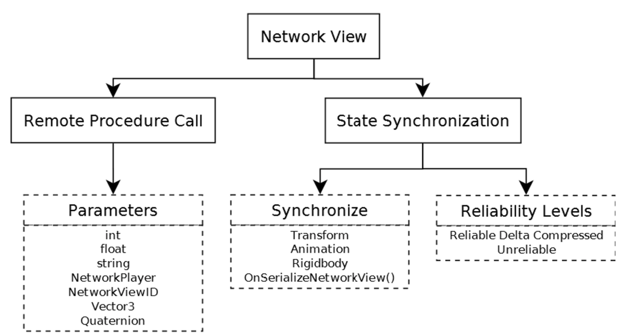

Figure 1. Networkview component features.

The Unity network module is used as base for the Unity Cluster Package, a Unity package for VR applications development that allows Unity applications on multi-projection systems over a PC cluster, since current game engines (including Unity) originally do not provide support to develop applications for such immersive platforms.

[1] http://unity3d.com/unity

A. *Unity Cluster Package*

The Unity Cluster Package [2] is a drag-and-drop component set that aims to support VR applications development using the Unity game engine. This package supports Unity applications on multi-projection systems over a PC cluster, passive stereoscopy, and access to VRPN (*Virtual Reality Peripheral Network* [13]) servers featuring device-independent interaction. Unity Cluster Package components arrange scene objects and C# scripts according to their functionality. Two components were used in order to develop our visualization application:

- *Multi Projection Camera*: it is a custom virtual camera for multi-projection applications using a generalized perspective projection matrix [14], which encodes the size, aspect ratio, position, and orientation of the screen according to the three points from the screen corners (Pa at the lower left, Pb at the lower right, and Pc at the upper left); and
- *Node Manager*: it uses the Unity network module to implement the Master-Slave [2] rendering model, where each slave node (Unity Client) and the master node (Unity Server) perform the same application with different virtual camera configurations. This component initializes and manages the cluster node application obtaining information from a configuration file, which contains the node type, screen specification (Pa, Pb, and Pc), and the IP address and connection port to the Unity Server.

The Unity Cluster Package is a good option for immersive applications development allowing to extend Unity applications for different multi-projection systems regarding the size, aspect ratio, position, and orientation of the screens. This package was used in order to develop the visualization application for a 3-screens multi-projection system called MiniCAVE.

B. *MiniCAVE*

MiniCAVE [15], [16] is a low-cost multi-projection system that arranges three screens (2.5m x 1.5m each) at an angle of 30 degrees to each other. MiniCAVE supports passive stereoscopy through polarized lenses, where for each screen there are two projectors (BenQ W1000 HD) connected to two dedicated Linux computers (Intel Core i7 8GB RAM) equipped with NVIDIA FX 1800 graphic cards. A gigabit switch is used to connect the six rendering computers to a server computer.

Since our visualization application runs in the MiniCAVE and on mobiles devices with the Android operating system, integrating both immersive and mobile platforms, a 300Mbps wireless router is connected to the MiniCAVE's gigabit switch allowing the connection of mobile devices.

[2]https://sourceforge.net/projects/unityclusterpackage/

IV. Immersive and Interactive Visualization System

The visualization system presented in this paper uses a visualization application, developed using both Unity and Unity Cluster Package, to create 3D graphs representations of data from relational databases. The system integrates distinct platforms (MiniCAVE and mobile devices) through the visualization application to provide a better data understanding offering immersion (wide field of view and stereoscopy) and interaction (particular display and touch sensitive screen).

An immersive overview of the 3D graph is provided in the MiniCAVE, where multiple simultaneous users can have navigation orientation and also spatial knowledge to identify clustering, patterns, trends, and outliers in the 3D representation. Mobile devices are used as interaction device, providing an individual workspace where the user can navigate through the 3D graph and request more information of a specific graph node using the touch sensitive screen.

The MiniCAVE and mobile devices integration results in a heterogeneous cluster. All cluster nodes (PCs and mobile devices) perform the same visualization application, where slave and mobile nodes are Unity Clients connected to the Unity Server on the master node, as shown in Figure 2. The Unity Cluster Package provides a coherent, seamless, and contiguous view from the slave node applications through replications of the Multi Projection Camera instantiated in the master node application. Since Multi Projection Camera uses the Networkview component from the Unity network module, any change in its position in the master node application is synchronized to the slave node applications.

The NodeManager component from the Unity Cluster Package enables, through configuration file, the specification of the cluster node type, screen (Pa, Pb, and Pc), and for which eye the application addresses the projection. The Slave #3 and Slave #4 applications (Figure 2) are assigned to the same screen (front), but for different eyes (Slave #3 for the left eye, and Slave #4 for the right eye). Figure 3 shows the configuration file of the Slave #4. The angle of 30 degrees at the MiniCAVE side screens is not considered in order to provide a better overview of the 3D graph. In proper proportions, at the start of the mobile node application is displayed as same as the front screen in the MiniCAVE.

```
<node type="slave">
  <server ip="192.168.1.13" port="25000"/>
  <screen stereo="true" eye="right">
    <pa x="-8"  y="-5"  z="0"/>
    <pb x="8"   y="-5"  z="0"/>
    <pc x="-8"  y="5"   z="0"/>
    <pe x="0"   y="0"   z="5"/>
  </screen>
</node>
```

Figure 3. Slave #4 configuration file.

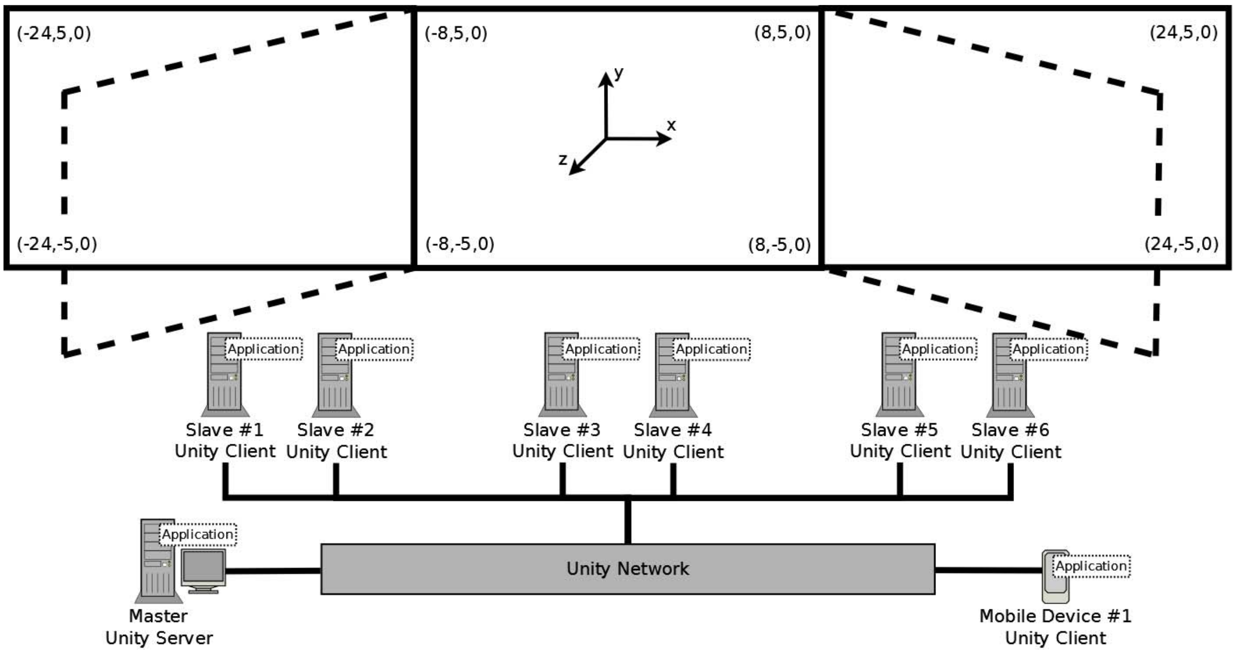

Figure 2. The MiniCAVE and mobile devices integration by a visualization application developed using the Unity game engine and the Unity Cluster Package.

A. *Data Extraction Engine*

The DBVis [17] desktop module is used as data extraction engine, where through an user-friendly interface (UI) it is possible to execute SQL commands over a relational database and, based on the resulting dataset, it allows the mapping of columns to visual attributes like size, colors, shapes and connections.

The UI was developed in Java, which supports connectivity to many relational databases systems through JDBC (Java Database Connectivity) specification, and it uses the Information Visualization (IV) library JUNG [18], providing 2D graphs representations from the resulting dataset. The data attributes not initially mapped to visual properties can be used as third dimension (coordinate z for 3D graphs) or additional information in text format. After the visual mapping, it is possible to export an XML file containing the data representation and attributes.

We use the desktop module to export 3D graphs as data representations from relational databases. The exported XML file are fetched by the visualization application from an Apache server. The data workflow is shown in Figure 4.

B. *3D Graph Creation*

The visualization application fetches the exported XML file from an Apache server using the Unity class WWW, which can retrieve the contents of URLs. All graph nodes are created as new instances of a customize Unity object (GraphNode) that arranges the assets: SphereMesh, TextMesh, and LineRenderer.

According to the mandatory attributes *x*, *y*, *z*, *id*, *label*, *size* and *color*, for each graph node from the exported XML file a new GraphNode is created, where the Unity SphereMesh (Unity primitive object for spheres) has position as (*x*, *y*, *z*), *id* and *label* as identifiers, scale as *size*, and color of the render material as *color*.

The remaining attributes of the graph node are placed into the Unity TextMesh (rendered text) in the respective GraphNode, concerning the possibility of additional information request. Unity TextMesh allows RichText, enabling the usage of a tag set to incorporate multiple font styles and sizes. The Unity TextMesh in the GraphNode is hidden by default, aiming to explore this additional information on demand.

Graph edges are represented using Unity LineRenderers, which takes an array of two or more tridimensional points and draws a straight line between them. The Unity LineRenderer must have the same color as the GraphNode source in order to represent a directed graph and, in case of undirected graph, it has both GraphNodes (source and target) colors. The usage of Unity LineRenderers as graph edges decreases the rendering complexity of the scene, since this asset is not rendered as 3D object but as billboard lines that have width, texture, and color.

The custom object GraphNode arranges C# scripts not only to build the 3D graph using the Unity assets described above, but also to implement the visualization and interaction dynamics.

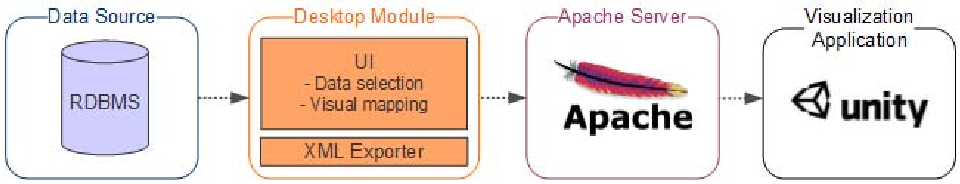

Figure 4. Data workFlow.

C. Visualization and Interaction

The visualization and interaction dynamics are based on the AV process [6], and on the experiment [9] to point out the effects of immersion and navigation on the acquisition of spatial knowledge of data represented as 3D graphs. By four separated conditions, the authors [9] provided the participants different immersion and navigation options, that could be tested by questions aiming the participant's knowledge of the graph being displayed.

In the experiment [9], the interaction was performed on the multiple-screens and it was observed the strong tendency for users to maintain focus on the front wall, where the side and floor walls could be a distraction, and the users navigated away from the graph in order to attain an overview. In our system, the immersive overview of the 3D graphs are provided in the MiniCAVE (wide field of view and passive stereoscopy), while interaction on mobile devices offers an individual workspace, where the user can navigate on a single and mobile display without losing the overview context. By separating the overview and interaction displays on the MiniCAVE and mobile devices, we aim to reach intuitive interaction and a better usage of the muliple-screens.

Since on single display users usually have a better performance navigating in an egocentric way (flying camera) [9], on mobile devices users can navigate with 3-DOF (Degree-Of-Freedom) using the touch sensitive screen to move the virtual camera: up/down (sliding one finger up/down), left/right (sliding one finger left/right), and forward/backward (sliding two fingers apart/together). Based on the AV process [6], mobile devices are also used to provide exploration and exhibition of additional information of a specific graph node by touching it on the screen turning visible the GraphNode's TextMesh with additional attributes.

Selection feedback can serve as a visual reference while navigating [9]. Hence, a Unity PacticleEmitter are attached on the GraphNode object to render a selection effect on the graph node in the MiniCAVE when the user touches it on the mobile device screen. The selection procedure is a Unity RPC function performed in the slave node applications requested by a mobile node application. By touching a graph node for the second time, the TextMesh and PacticleEmitter are hidden, disabling the additional attributes view on the mobile device and the selection effect in the MiniCAVE.

By double-tapping a graph node on the mobile device, a secondary overview is achieved in the MiniCAVE having focus on the graph node selected. The focus selection is a Unity RPC function performed in the master node application requested by a mobile node application. Double-tapping on the background returns to the original overview.

V. User Case

A dataset containing structural information about the corporate database of the university is used to validate our immersive and interactive visualization system. This relational database runs on PostgreSQL 9.0 and has 700 tables, 1100 foreign keys and 77 millions records, approximately.

All university corporate systems store their data in this database, and many of these systems are still in development, being necessary a special attention over the database structure. In this way, visualizations to aid in the identification of possible structural problems, such as high coupling between tables of different systems, can be created enabling developers to explore the database structure and have insights about problems that can be avoided.

In the DBVis [17] desktop module, a dataset resulting from SQL command to fetch structural information about the database is mapped with graph nodes as tables and edges as foreign keys. The graph node color represents the schema where the table is located, and the size represents how many relations the table has. The 3D graph is exported as XML file, that besides the mapped attributes, contains for each graph node additional attributes (Table I) as additional information, where "Estimated Rows" is used as graph node depth (third dimension).

The exported 3D graph visualized in our system allows the developers to quickly identify important tables in terms of relationships and high usage, through the graph nodes and edges overview and sense of depth from the passive stereoscopy in the MiniCAVE (Figure 5-(a)), since graph edges represent foreign keys and the number of data rows are the graph node depth. The additional information about tables of interest are obtained for each user on his/her individual workspace on the mobile device by touching the graph node on the screen (Figure 5-(b)), where the additional attributes (Table I) are displayed generating the selection effect on the respective graph node in the MiniCAVE (Figure 5-(a)).

Table I
ADDITIONAL ATTRIBUTES.

Attribute	Description
Estimated Rows	Number of data rows
Table Schema	The schema where the table is located
Total Size Megabytes	Total table (data and indexes) size in megabytes
Table Catalog	The database where the table is located
Relations	Number of relations with other tables
Table Name	The table name
Size Megabytes	Table data size in megabytes

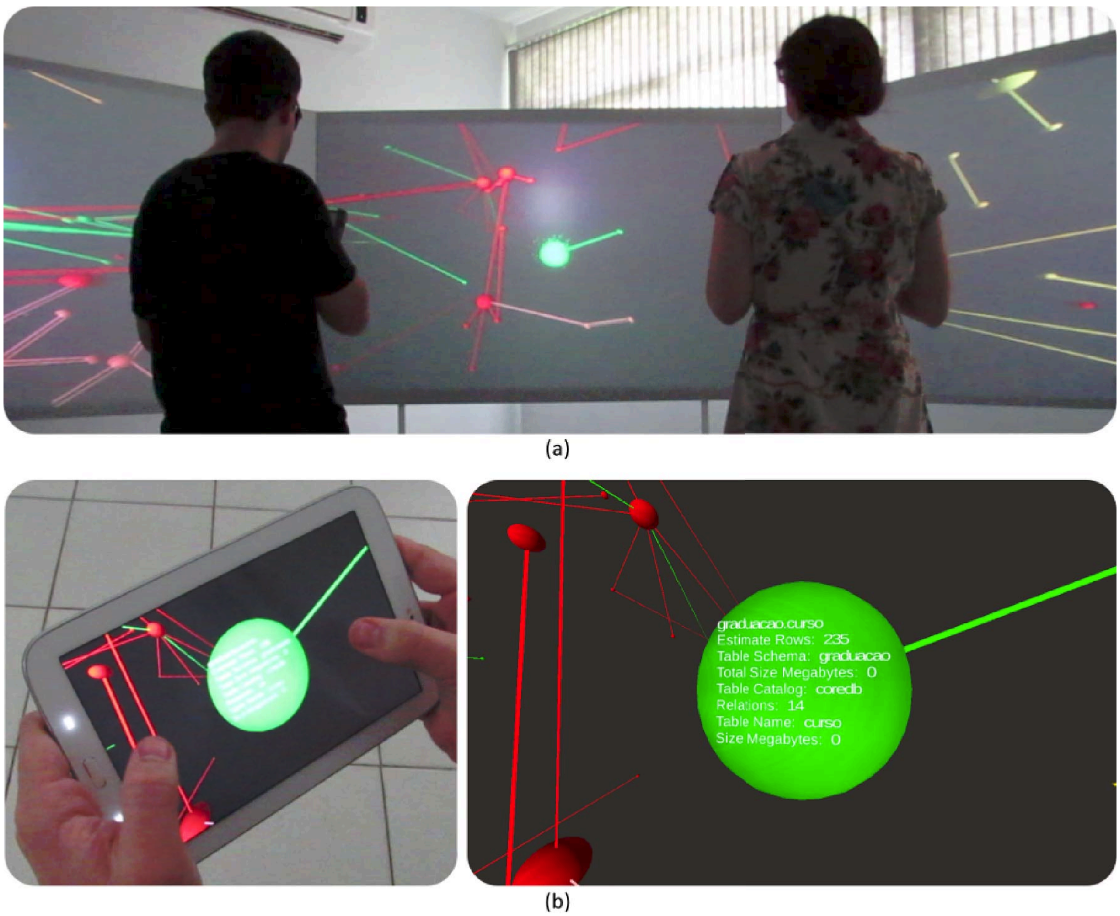

Figure 5. Immersive overview in the MiniCAVE (multiple simultaneous users) (a). Navigation and additional information on the mobile device (individual workspace) (b).

CONCLUSIONS AND FUTURE WORKS

This paper presented an immersive and interactive visualization system, which integrates distinct platforms (MiniCAVE and mobile devices) in order to explore data from relational databases. The visualization system combines both platforms features providing an immersive overview of 3D graphs used as data representations in the MiniCAVE, and user interaction on mobile devices, regarding navigation and additional information request. The visualization application was developed using the Unity game engine and Unity Cluster Package, concerning our heterogeneous cluster (MiniCAVE's PC cluster and mobile devices).

Some considerations can be made compared to other visualization systems (Table II). Our system provides multiple simultaneous users, since head-track rendering is not used and mobile devices have a particular display. Mobile devices can be intuitive interaction device, perhaps more than Wands and/or Console Devices, by means of the touch sensitive

Table II
COMPARISON WITH OTHER VISUALIZATION SYSTEMS.

System	Wide Field of View	Stereoscopy	Head-Track Rendering	Data Representation	Interaction Device	Simultaneous Users
Our system	3-screens	Passive	No	3D Graph	Mobile Device	1-6
CaveDataView [4]	4-screens	Active	Yes	3D Scatterplots	Wand	1
CAVE-SOM [10]	4-screens	Active	Yes	3D Cubes	Wand	1
XIM [8]	3-screens	–	Yes	3D Graph	Console Device	1

screen. By separating the overview and interaction displays on the MiniCAVE and mobile devices, our system enables user navigation and additional information view without losing the overview context. The user is able to fetch and map any kind of data from relational databases to 3D graphs through the desktop module, and we presented as user case database structural data visualized as 3D graph.

As future works, we plan to explore data from relational databases in the immersive and interactive visualization system, and to investigate other 3D representations, an experiment for comparison between 2D and 3D graphs representations, and the visualization of data on the web since Unity provides the WWW class. We will also look into applications integrating MiniCAVE and mobile devices, using the Unity game engine and the Unity Cluster Package, for other purposes than data visualization.

ACKNOWLEDGMENT

The authors acknowledge the financial support of the Brazilian financial agency São Paulo Research Foundation (FAPESP) – grant # 2013/03452-0. The authors also would like to thank to CAPES Foundation, a body of the Brazilian Ministry of Education. Mário Popolin Neto was recipient of scholarship from CAPES.

REFERENCES

[1] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: the design and implementation of the cave," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '93. New York, NY, USA: ACM, 1993, pp. 135–142. [Online]. Available: http://doi.acm.org/10.1145/166117.166134

[2] O. G. Staadt, J. Walker, C. Nuber, and B. Hamann, "A survey and performance analysis of software platforms for interactive cluster-based multi-screen rendering," in *Proceedings of the Workshop on Virtual Environments 2003*, ser. EGVE '03. New York, NY, USA: ACM, 2003, pp. 261–270. [Online]. Available: http://doi.acm.org/10.1145/769953.769984

[3] D. Schikore, R. Fischer, R. Frank, R. Gaunt, J. Hobson, and B. Whitlock, "High-resolution multiprojector display walls," *Computer Graphics and Applications, IEEE*, vol. 20, no. 4, pp. 38–44, Jul 2000.

[4] D. Raja, D. A. Bowman, J. Lucas, and C. North, "Exploring the Benefits of Immersion in Abstract Information Visualization," in *In proceedings of Immersive Projection Technology Workshop*, 2004.

[5] Prabhat, A. Forsberg, M. Katzourin, K. Wharton, and M. Slater, "A comparative study of desktop, fishtank, and cave systems for the exploration of volume rendered confocal data sets," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 3, pp. 551–563, May 2008.

[6] C. Ware, *Information Visualization: Perception for Design*, 3rd ed., ser. Interactive Technologies. Elsevier Science & Technology, 2012.

[7] K. Ryabinin and S. Chuprina, "Adaptive scientific visualization system for desktop computers and mobile devices," *Procedia Computer Science*, vol. 18, no. 0, pp. 722 – 731, 2013, 2013 International Conference on Computational Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913003797

[8] A. Betella, R. Carvalho, J. Sanchez-Palencia, U. Bernardet, and P. F. M. J. Verschure, "Embodied interaction with complex neuronal data in mixed-reality," in *Proceedings of the 2012 Virtual Reality International Conference*, ser. VRIC '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:8. [Online]. Available: http://doi.acm.org/10.1145/2331714.2331718

[9] J. A. Henry and N. F. Polys, "The effects of immersion and navigation on the acquisition of spatial knowledge of abstract data networks," *Procedia Computer Science*, vol. 1, no. 1, pp. 1737 – 1746, 2010, iCCS 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050910001961

[10] D. Wijayasekara, O. Linda, and M. Manic, "Cave-som: Immersive visual data mining using 3d self-organizing maps," in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, July 2011, pp. 2471–2478.

[11] E. F. Anderson, L. McLoughlin, J. Watson, S. Holmes, P. Jones, H. Pallett, and B. Smith, "Choosing the infrastructure for entertainment and serious computer games - a whiteroom benchmark for game engine selection," in *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013 5th International Conference on*, Sept 2013, pp. 1–8.

[12] P. Petridis, I. Dunwell, S. de Freitas, and D. Panzoli, "An engine selection methodology for high fidelity serious games," in *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2010 Second International Conference on*, March 2010, pp. 27–34.

[13] R. M. Taylor, II, T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, "Vrpn: A device-independent, network-transparent vr peripheral system," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '01. New York, NY, USA: ACM, 2001, pp. 55–61. [Online]. Available: http://doi.acm.org/10.1145/505008.505019

[14] R. Kooima, "Generalized Perspective Projection," http://aoeu.snth.net/static/gen-perspective.pdf, [Online; accessed March-2015].

[15] D. R. C. Dias, J. R. F. Brega, L. C. Trevelin, M. Popolin Neto, B. B. Gnecco, and M. de Paiva Guimaraes, "Design and evaluation of an advanced virtual reality system for visualization of dentistry structures," in *Virtual Systems and Multimedia (VSMM), 2012 18th International Conference on*, Sept 2012, pp. 429–435.

[16] D. R. C. Dias, J. R. F. Brega, A. F. Lamarca, M. Popolin Neto, D. J. Suguimoto, I. Agostinho, and A. F. Gouveia, "Chemcave3d: Sistema de visualização imersivo e interativo de moléculas 3d," in *Workshop de Realidade Virtual e Aumentada*. Uberaba, MG, Brasil: WRVA, 2011.

[17] A. C. Moraes, D. M. Eler, and J. R. F. Brega, "Collaborative information visualization using a multi-projection system and mobile devices," in *Information Visualisation (IV), 2014 18th International Conference on*, July 2014, pp. 71–77.

[18] D. Fisher, J. O'Madadhain, and T. Nelson, "JUNG: Java Universal Network/Graph Framework," http://jung.sourceforge.net/, [Online; accessed March-2015].