A Tablet-Based 3D Interaction Tool for Virtual Engineering Environments

Daniel Medeiros
Tecgraf/PUC-Rio
danielps@tecgraf.puc-rio.br

Lucas Teixeira
Tecgraf/PUC-Rio
lucas@tecgraf.puc-rio.br

Felipe Carvalho
Tecgraf/PUC-Rio
kamel@tecgraf.puc-rio.br

Ismael Santos
Petrobras
ismaelh@petrobras.com.br

Alberto Raposo
Tecgraf/PUC-Rio
abraposo@tecgraf.puc-rio.br

Abstract

Three-dimensional computer-aided design (3D CAD) modeling and reviewing is one of the most common engineering project tools. Interaction in these environments is characterized by the need for a high precision level to execute specific tasks. Generally this kind of task uses specific interaction devices with 4 or more degrees of freedom, such as 3D mice. Currently applications involving 3D interaction use interaction devices for object modeling or for the implementation of navigation, selection and manipulation techniques in a virtual environment. A related problem is the need to control naturally non-immersive tasks, such as symbolic input (e.g., text, photos). In addition, the steep learning curve to handle such non-conventional devices is a recurring problem. The addition of sensors and the popularization of smart-phones and tablets, allowed the use of such devices in virtual engineering environments. These devices, differs to other devices by the possibility of including additional information and performing naturally non-immersive tasks.

This work presents a 3D interaction tablet-based tool, which allows the aggregation of all major 3D interaction topics, such as navigation, selection, manipulation, system control and symbolic input. To validate the proposed tool,the SimUEP-Ambsim application was chosen, an oil and gas simulator that has the complexity needed and which allows the use of all techniques implemented. Then, the tool was tested in another application, a photo-voltaic solar plant simulator, in order to evaluate the generality of this work concept.

Keywords: 3D Interaction, Virtual Reality, Mobile Devices, Virtual Engineering Environments

1 Introduction

Immersive 3D Interaction in virtual engineering environments sometimes is characterized by the use of non conventional devices required for some specific tasks, e.g., 3D modeling and design review. Generally, these devices have 4 or more DOFs (degrees of freedom) and are very expensive, which discourages sporadic users from having devices like these.

Keyboard and mouse are often used for WIMP (Windows, Icons, Menu, Pointers) interfaces and normally have only two degrees of freedom. For that reason they are not the most suitable for 3D Interaction. Another problem is that they can only be used on a table, thus becoming unfeasible for use in immersive environments such as CAVE [Cruz-Neira et al. 1992].

Multitouch devices have become increasingly powerful and are now practically portable computers. A major advantage of the tablet is its portability and diversity of embedded sensors. These sensors capture information about the device and the environment, being even capable of measuring the atmospheric pressure, position and acceleration. The use of such sensors makes the tablet an interesting alternative to interaction devices used in Virtual Environments (VEs). The relatively low cost and easy access to mobile devices (such as tablets with sensors and multi-touch screens) is an incentive to the use in three-dimensional interaction techniques.

Sensors like accelerometers and magnetometers can inform the spatial orientation of the device and enable it to track the user position. The touch screen provides greater flexibility for this type of device, allowing the inclusion of additional features in the interface such as menus, buttons and check boxes. Such elements could be used to perform certain actions on the application, enabling the user to control certain parts of an application or to display additional information.

A current research issue in the 3D interaction area is related to the use of new devices and the better use of techniques of navigation, selection and manipulation of objects in a virtual environment. Two related problems are: the need to control naturally non-immersive tasks, such as symbolic input (e.g., text, pictures); and the learning curve required for handling such devices.

In this work we present a tablet-based virtual reality tool that addresses all 3D interaction major topics such as navigation, selection, manipulation, system control and symbolic input.

This paper is organized as follows, section 2 presents some related works and section 3 describes the concepts used in the design of the proposed tool. Section 4 is dedicated to the development issues and section 5 presents the study cases. Finally conclusions and future work are presented in section 6.

2 Related Work

Ruiz et al. [Ruiz et al. 2011] present different techniques for the use of the sensors found in smartphones and tablets in everyday tasks such as answering a call or performing a search for contacts. Several works of 3D interaction have mapped interaction tasks in virtual environments using mobile devices. This interaction can occur directly, when the user uses the tablet or smartphone screen to interact with the virtual environment and indirectly, when the interaction made by the user is captured by the device and is sent to a remote application where the interaction is made.

A problem of direct interaction on mobile devices is the occlusion caused by the user's fingers to interact with the VE, often hiding the object of interest. The work proposed by Telkernaroglu et al. [Telkenaroglu and Capin 2012] solves this problem by proposing selection and manipulation techniques using one or two fingers to perform transformations on the object so that the object of interest is not occluded. This work also proposes a form of navigation using the gesture known as pinch.

VRCAI 2013, November 17 – 19, 2013, Hong Kong.

Among the works that discuss the design and evaluation of indirect navigation techniques for three-dimensional virtual environments we may cite [Noronha et al. 2012], [Katzakis et al. 2011], [Radhakrishnan et al. 2013] and [Benzina et al. 2011]. In Sensor Fusion [Kim et al. 2012], the Walk In Place navigation technique is proposed. This technique uses the accelerometer and magnetometer sensors present in smartphones coupled to the foot of the user. These sensors are able to faithfully capture the user's walk and allow the navigation even in complex environments using visualization systems like CAVE, for example. However as shown in Medeiros et al. [Medeiros et al. 2012] these works do not take into account precision tests of devices, a characteristic considered important in virtual engineering.

Besides navigation tasks, mobiles have been used for selection tasks in two-dimensional environments. The ARC Pad [McCallum and Irani 2009] maps the user touches on the mobile device into a remote computer, emulating movements of a mouse in a WIMP environment. The work proposed by Ruan et al. [Ruan et al. 2010] expands this idea and, besides mapping the user touches, uses a mobile device with a physical keyboard to perform various tasks such as entering addresses into web browsers. Boring et al. [Boring et al. 2010] proposes techniques for selection and manipulation of 2D objects in displays located away from the user using image processing on the video captured by the device's camera. Nancel et al. [Nancel et al. 2011] expands the ARC Pad, mapping the user's touches in pointing and zooming tasks on a display wall system. However, a related problem is the lack of precision in the mapping of these touches into a 3D environment, a feature considered important for interaction with virtual engineering environments. Debarba [Debarba et al. 2012] proposes a system using smartphones for multi-resolution environments that utilizes a virtual window scheme similar to the Image Plane Technique [Pierce et al. 1997], that is used to improve the accuracy of object selection in this type of environment. However, it is noted that tests have not been conducted in more complex virtual environments, and because of the size of the device used it does not include additional functionality in the solution proposed. Furthermore, the idea of virtual window increases the accuracy of selection, and was used in the design of the proposed tool.

The use of mobile devices in virtual environments is not limited to navigation and selection tasks. Platforms such as Microsoft SmartGlass and Nintendo Wii U [Pace 2013] were recently announced and bring to the user the so called Second Screen Experience. In these systems the tablet acts as a complementary tool to the game and may contain additional information related to it, such as maps and menus, and it could be also used as a joystick.

Another interesting use of 3D interaction in mobile devices is the ARDrone[Krajník et al. 2011]. The ARDrone is a quad rotor helicopter, i.e., a helicopter toy with four helices that is able to glide in the air. The control of the drone is made through a mobile device. An application is installed on the device and with it the user can view the contents captured by the cameras present in the airplane model, one on the front and one at the bottom, and sent by a dedicated wireless network.

3 Concept

Despite the potential visualization provided by the screen of mobile devices during interaction in a virtual environment, tools that use all the categories of techniques proposed by Bowman et al. [Bowman et al. 2004] were not found. From this starting point, a mobile-based solution was designed to use and expand the interaction tools found in training procedures in virtual engineering.

In this work we present a tablet-based virtual reality tool that aggregates all major topics of 3D interaction such as navigation, selection, manipulation, system control and symbolic input. In the design of the proposed tablet-based interaction tool, one of the major requirements was the maximum decoupling between the mobile device and the graphic application, enabling its use in different applications.

One of the techniques used is a selection by progressive refinement [Kopper et al. 2011] adapted to use a virtual camera metaphor. This technique uses a tracked tablet that controls the position and orientation of the virtual camera in an immersive environment. For this, optical trackers such as ARTracker and BraTracker [Pinto et al. 2008] are used to ensure accuracy for the proposed technique.

This virtual camera is represented in the virtual environment through a truncated semi-transparent pyramid, rendered along with the virtual scene. It is used in a similar way to that used in Bubble Selection [Vanacken et al. 2007] and Cone Casting [Forsberg et al. 1996], where a solid or a plane is used to define the group of interest. Figure 1 illustrates the concept of the proposed selection tool.

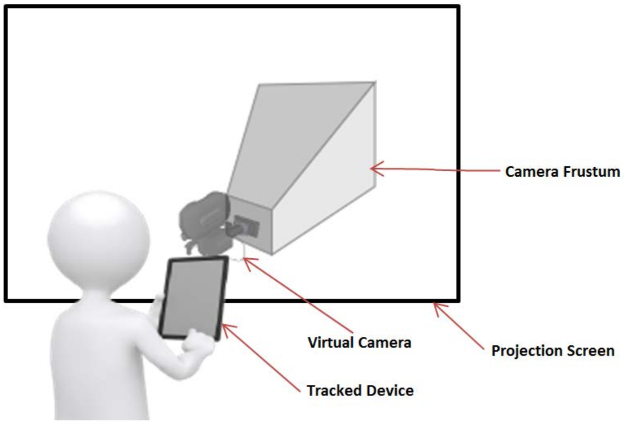


Figure 1: *Scheme of the proposed selection tool.*

In the second phase of the technique, the disambiguation step, the selection and manipulation are mapped into the image that is rendered by this virtual camera. This image is sent to the tablet, which draws it on the touch surface (touchpad). All user touches are normalized and sent to the graphic application that maps the touches made on this surface according to their use, performing all the necessary calculations.

In the representation of the virtual camera in the touchpad, the user can select the object of interest with a tap above it (Figure 2 - item A). After selection, the object is marked and the user can confirm the selection with an extra tap on the object. Once the user confirms the selection, the object can then be manipulated. The manipulation can be done through gestures known as scale and rotation (Figure 2 - items B and C, respectively). For an even more precise selection, zoom in and zoom out events were also developed on the touchpad image using the scale gesture.

For the navigation task, we proposed the incorporation of directionals on the interface of the mobile device to control rotation and translation in a first person view. The use of directionals is justified by the high precision of the sensors embedded in the device, that need some element to filter or adjust their precision [Medeiros et al. 2012].

In addition to the use of mobile devices with sensors for selection,

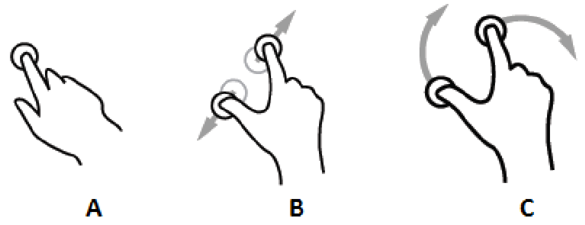


Figure 2: *Selection/Manipulation gestures. (a) Tap (b) Scale (c) Rotation.*

manipulation and navigation techniques, there is also the possibility of incorporating different features in the tool using interface elements such as buttons, menus and check boxes. These elements, when used to send commands to perform an action in the application such as opening a door or confirming the selection of an object are called elements of system control.

By the nature of the selection and manipulation techniques proposed in this paper, we see the possibility of using 3D widgets due to the fact that the proposed technique maps the virtual environment on the tablets 2D interface. If the camera is positioned in front of a 3D widget, for example, it is mapped in the interface of the tablet, and it becomes a 2D widget, which facilitates the selection of these elements.

Symbolic input is another related task which can be incorporated. This kind of task allows the user to communicate symbolic information, such as text, numbers and other symbols. One possibility for users is adding annotations to a previously selected object, for example. The integration of symbolic input in the proposed tool is possible by a virtual keyboard that is triggered via buttons in the interface.

4 Development

After the definition of the interaction techniques and their applications in immersive virtual environments, requirements were elicited to build the mobile application. One of the main requirements was the maximum decoupling between the mobile application and the graphical application to be used. To guarantee that, we used the approach of a remote controlled device that sends data captured by the device over the network.

In the development of the mobile application we used the Android open source operating system, which provides all the functionality needed for the use of the sensors available on the device and allows the use of the communication platform VRPN, a virtual reality framework that optimizes the data transmission for virtual reality devices over the network.

Just after the definition of the development tools, an interface was sketched in a way that would be able to use each one of the proposed techniques effectively. This interface (Figure 3) contains a touch pad that is used to draw the image received by the camera and allows the user to select the desired object. Thus, the (x,y) normalized position of the touch input is sent to the graphical application that performs the calculation to select and then manipulate an already selected object by the gestures defined (Figure 2). In this interface the user can also activate the virtual keyboard to tag objects, for example.

For the navigation task, the user has joystick controls similar to those used in the ARDrone application, which when pressed allow translation and rotation (Figure 3 - items B and C, respectively). To spatially locate the tablet in relation to the screen, a marker is used as 3D tracking point together with an ARTracker optical tracking system.

In this interface the user has controls to add annotations or tags, to a selected object (Figure 3 - item G), to switch the selection mode (Figure 3 - item D), to lock the virtual camera or the ray in order to start the selection/manipulation mode (or unlock it to activate the navigation mode) (Figure 3 - item E), and controls which can configure some application features such as the IP Address of the graphic application used (Figure 3 - item A). The interface has also a textual label that can represent some feedback of the graphic application, such as the name of the selected object or the amount of rotation of a selected object.

The decoupling between the mobile application and the desktop application is guaranteed by the use of the VRINPUT module of the LVRL 3D interaction framework [Teixeira et al. 2012] that receives the data generated by the tablet and transforms them into events, which will then be interpreted and used by the chosen graphics application. The VRINPUT is used in the application through a dynamic library (dll) to receive data sent by the tracker and the tablet.

Because the VRPN tool is totally focused on virtual reality devices and these are not traditionally used to perform input operations, it doesn't have a specific message for this purpose. To this end we used UDP sockets to communicate regardless of VRPN connection. An overview of the system developed is found in Figure 4.

5 Study Case

To validate the developed tool we used the SimUEP-AmbSim application (Figure 5), a training simulator for oil platforms developed on the Unity3D Engine [Creighton 2010]. The application has the structure of a game, where users browse an oil platform to achieve predetermined goals. The choice of SimUEP-AmbSim is justified by its complexity and because it enables the use of all the features proposed by the developed tool.

Another important point in the use of SimUEP-AmbSim is the existent support for different interaction devices and 3D visualization systems. In some of the specific goals of the application, the user needs to manipulate certain objects within the platform (valves for example) and when selected they can be rotated in order to open or close them. To achieve the proposed objective there is the possibility to mark certain objects, inserting information or tags on them that help in the execution of the specified tasks.

The integration with the developed mobile tool was made using a VRINPUT reader developed with the library, as already mentioned, and imported by the application SimUEP-AmbSim. These data are received in the form of events as they happen.

Once obtained, these data are processed and routed to specific parts of the application. The directional controls are used for navigation; tracker data are used for positioning the virtual window, and the touch pad (Figure 3 - item F) is used for selection (single touch) and object manipulation (double touch) on selected objects. The keyboard events are handled separately as described above, using UDP sockets. More details about the development of the proposed tool will be explained in the following subsections, grouped by their respective categories.

5.1 Selection and Manipulation

For selection and manipulation tasks, selectable objects (already available in SimUEP-AmbSim) were used. There are objects that are manipulated by a touch, as is the case of doors and automatic

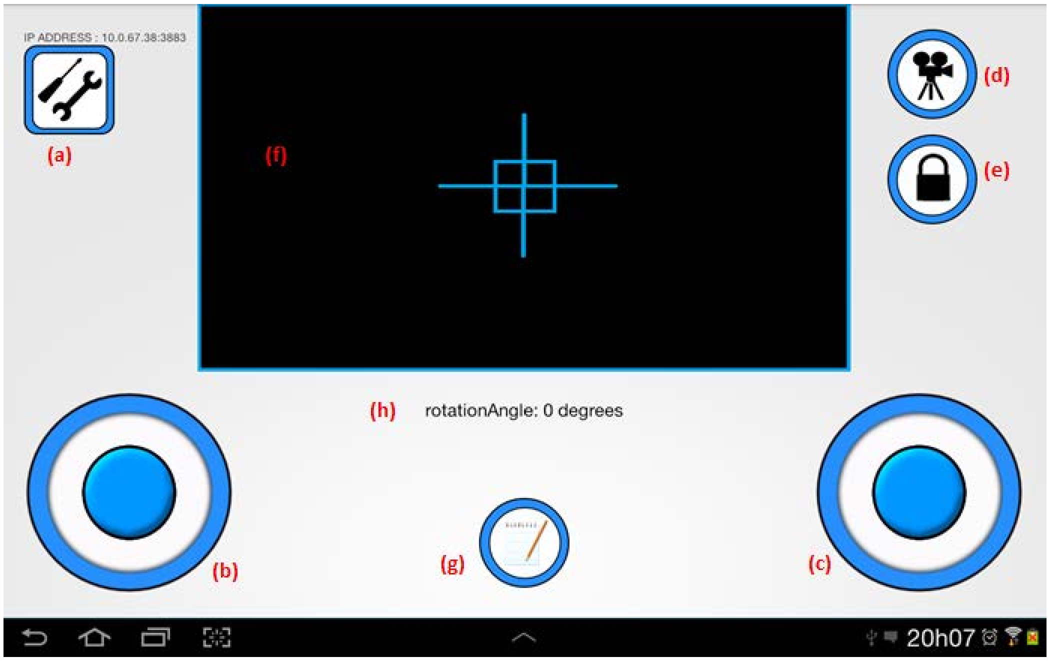


Figure 3: *Mobile application interface.*

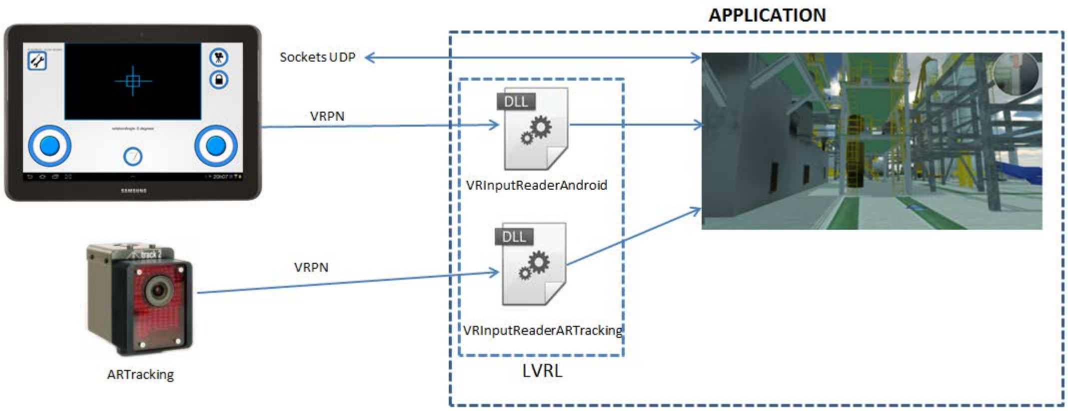


Figure 4: *System architecture overview.*

valves. Other objects require more precise gestures, as is the case with manual valves.

The object selection is performed when the user approaches a selectable object, positioning the virtual camera in a way that the camera frustum contains the desired object (Figure 6). Right after this approach, the user presses a button on the interface (Figure 3 - item D) and then a rendered image from the virtual camera is sent through the network and it is drawn on the touch pad (Figure 3 - item F). After that the user can touch the desired object directly causing the selection of it. If the object is properly selected, it will be highlighted and with an extra touch on the same object, it could be manipulated. The objects such as valves can be manipulated using the rotation gesture (Figure 2 - item B) that performs the rotation around the axis of the valve clockwise or counterclockwise, depending on the gesture made by the user. Figure 7 shows all those steps inside a CAVE environment.

The mapping of the user's touch on the touch pad is done as follows: when the user performs a tap on the touch pad, the normalized position is sent through the network and then mapped by the application on the near plane, that transforms to a position in world coordinates and then turns it into a ray that is cast in a direction that starts at the virtual camera position and intercepts the point mapped on the near plane (Equation 1). The mapping of the touch on the device is shown in Figure 8.

$$\vec{r_{dir}} = ||\vec{p_{wc}} - \vec{cam_{pos}}|| \tag{1}$$

The positioning of the virtual camera is made by the direct use of the values of the position and rotation received by the tracking system. According to the application, there may be differences between the coordinate system of the tracking and the coordinate system of the application. On the SimUEP-AmbSim, because it was developed in Unity3D a correction to the Z coordinate was necessary.

For a more precise selection of objects within the SimUEP-AmbSim, a zoom technique was implemented on the rendered im-

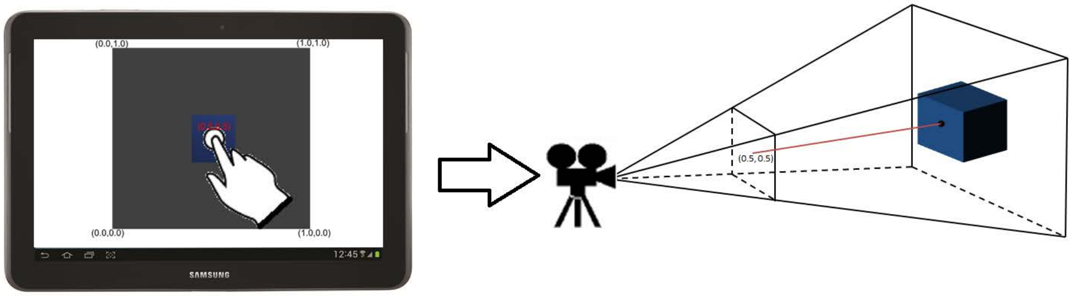


Figure 8: *Mapping of user touch in the virtual scene.*

age of the virtual camera. This event occurs when there is no selected (or highlighted) object and two fingers are detected by the system. This functionality was implemented using the scale gesture, that measures the zoom in and zoom out by the euclidean distance between the two fingers x,y coordinates (Equation 2). This gesture was then mapped on the camera field of view (FOV). These two magnitudes are directly proportional since the higher the distance of the two fingers, the higher the FOV which will be calculated and the smaller the distance between the two fingers, the smaller the FOV calculated (Equation 3).

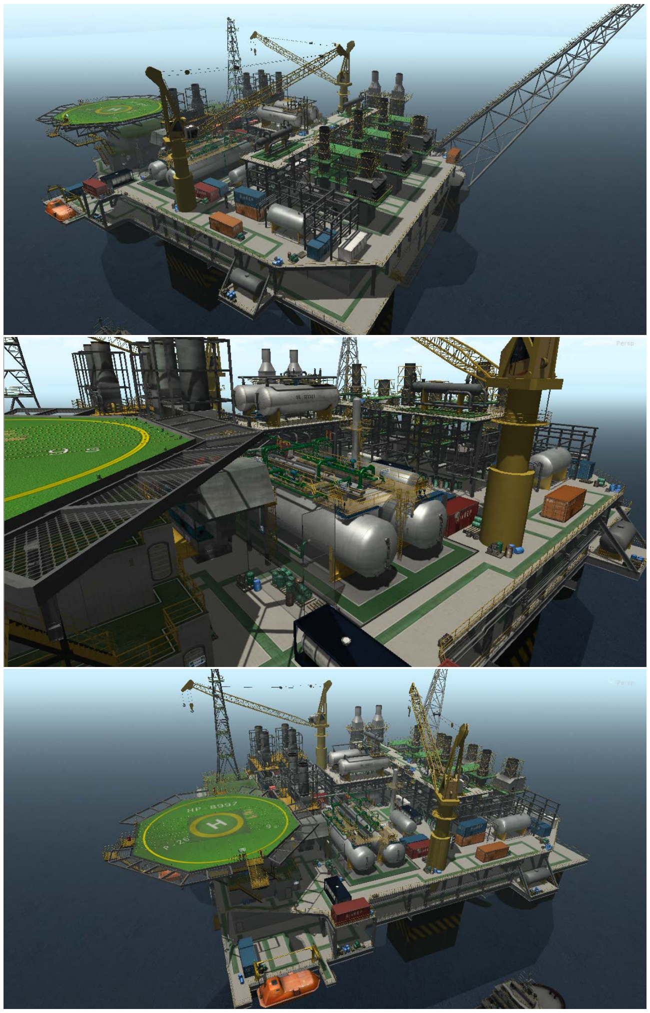


Figure 5: *Screenshots of SimUEP- AmbSim application.*

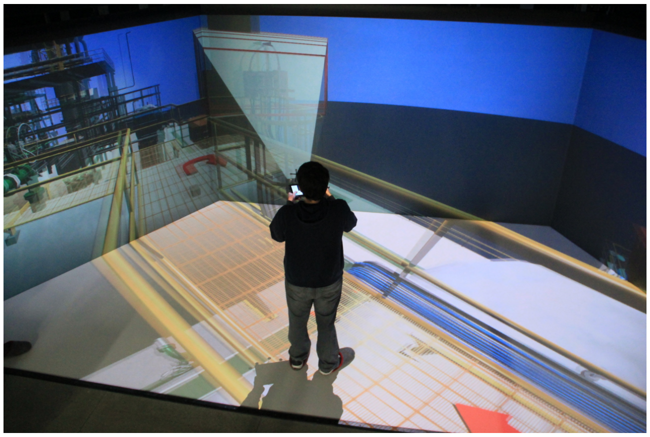


Figure 6: *Implementation of the proposed selection tool using virtual camera metaphor.*

$$dist = \sqrt{(x_1^2 - x_2^2) + (y_1^2 - y_2^2)} \tag{2}$$

$$Fov_{new} = Fov_{old} + (dist - previousDist) * Fov_{old} \tag{3}$$

It was also noticed that certain situations don't require a high selection precision level, as in the case of the selection of big objects, such as doors and automatic valves. For that type of object the user can switch to a raycasting-based technique, in that, once the ray intercepts the object, the object is then highlighted and with an extra touch on the touch pad the user confirms the selection of that object. Once selected, the object can be manipulated using the gestures used in the previous technique.

5.2 Image Transmission

To send the rendered images on Unity, the virtual camera renders the scene and saves it as a texture, which contains the image pixels. To increase the performance of the image transmission and decrease the required bandwidth the JPEG format was used to compress the textures saved by the camera. This format was chosen for the high degree of compression, which allows to a reduction the size of the final file but preserves the details of the rendered image.

The images are sent through the network only if an important modification is done in the virtual environment, such as the highlight or selection of an object. This is justified by the large number of frames generated (30 frames per second, approximately) and a high degree of processing power needed for compression and transmission of images through the network, requiring more complex procedures of compression and transmission [Shi 2011] [Lamberti and Sanna 2007].

5.3 Symbolic Input

Once an object is selected, the user can also enter notes associated with it. This procedure is done using the arrow at the bottom center, as seen in Figure 3. Then a window appears in the interface of the tablet (Figure 9) and so the user can enter the information he or she wants and push the button Send. Finally, a 3D plane object containing the typed text will be positioned in the center of the marked object (Figure 10).

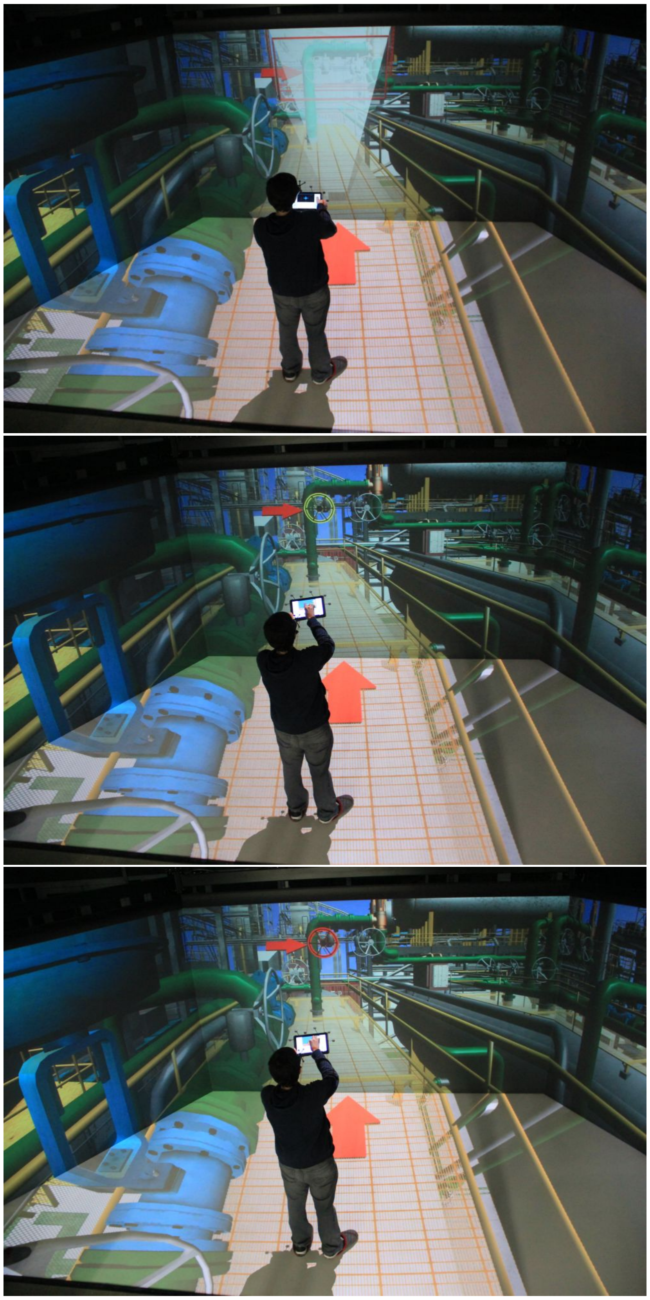
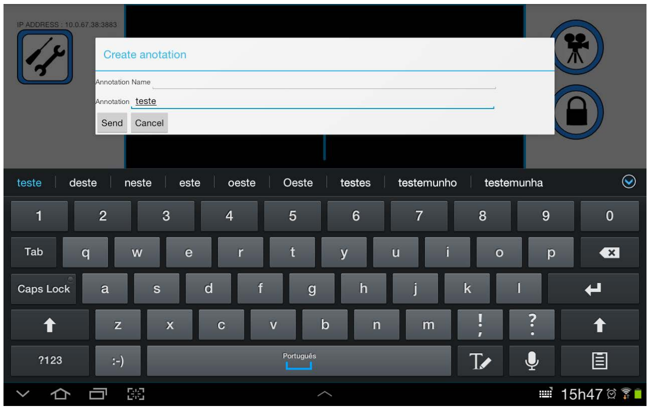


Figure 7: *Selection/manipulation of a valve.*



IP ADDRESS : 10.0.67.38:3883

Create anotation

Annotation Name

Annotation teste

Figure 9: *Symbolic Input: Annotation Creation.*

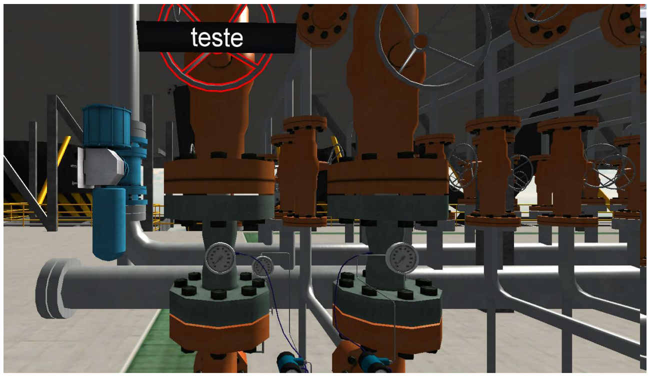


Figure 10: *Insertion of tags on valves.*

5.4 Mobile Application Decoupling Test

The SimUEP is a framework which has many different applications over it. The developed tool was briefly adapted to another application, SimUEP-Solar. The main goal of this was to validate the concern to decouple the mobile application from the graphical application.

5.4.1 SimUEP-Solar

SimUEP-Solar is a first person visualizer of results of photo-voltaic plant simulators. The user can: walk and fly, measure some results, (e.g., voltages and amperages in wires and electrical equipments), and analyze the behavior of the shadows over the solar cells. It can also be used for training operations in a plant. Results of the use of the tool in SimUEP-Solar are shown in Figure 11.

6 Conclusion

Mobile devices have proved to be an option with great potential for use in virtual engineering. It should be emphasized that the proposed tool has the role of adding the main elements of three-dimensional interaction together in a single device, making it a

Figure 11: *Mobile App decoupled testing on SimUEP-Solar.*

complete tool for use in virtual environments.

The use of virtual reality simulators has been an efficient way to reduce costs for staff training on oil platforms. However, even with the possibility of using CAVE-type multi-visualization environments with non-conventional devices such as the flystick, there is still some resistance to the use of such resources. Therefore, the familiarity encountered by users with mobile devices decreases their resistance to immersive virtual reality environments.

As future work we propose the use of different graphic applications in other areas of interest to validate the proposed tool as the one presented by Noronha et al. [Noronha et al. 2012]

Acknowledgements

Tecgraf is a laboratory mainly supported by Petrobras. Alberto Raposo thanks also CNPq for the support granted to this research (470009/2011-0).

References

Benzina, A., Toennis, M., Klinker, G., and Ashry, M. 2011. Phone-based motion control in vr: analysis of degrees of freedom. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, ACM, New York, NY, USA, CHI Extended Abstracts (CHI EA) '11, 1519–1524.

Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. 2010. Touch projector: mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '10, 2287–2296.

Bowman, D. A., Kruijff, E., LaViola, J. J., and Poupyrev, I. 2004. *3D User Interfaces: Theory and Practice*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Creighton, R. H. 2010. *Unity 3D Game Development by Example Beginner's Guide*. Packt Publishing.

Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., and Hart, J. C. 1992. The cave: audio visual experience automatic virtual environment. *Commun. ACM 35*, 6 (June), 64–72.

Debarba, H., Nedel, L., and Maciel, A. 2012. Lop-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*, 125 –132.

Forsberg, A., Herndon, K., and Zeleznik, R. 1996. Aperture based selection for immersive virtual environments. In *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, 95–96.

Katzakis, N., Hori, M., Kiyokawa, K., and Takemura, H. 2011. Smartphone game controller. In *Proceedings of the 74th HIS SigVR Workshop*.

Kim, J., Gracanin, D., and Quek, F. 2012. Sensor-fusion walking-in-place interaction technique using mobile devices. In *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, 39–42.

Kopper, R., Bacim, F., and Bowman, D. 2011. Rapid and accurate 3d selection by progressive refinement. In *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, 67–74.

Krajník, T., Vonásek, V., Fišer, D., and Faigl, J. 2011. Ar-drone as a platform for robotic research and education. In *Research and Education in Robotics-EUROBOT 2011*. Springer, 172–186.

Lamberti, F., and Sanna, A. 2007. A streaming-based solution for remote visualization of 3d graphics on mobile devices. *Visualization and Computer Graphics, IEEE Transactions on 13*, 2, 247–260.

McCallum, D. C., and Irani, P. 2009. Arc-pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ACM, New York, NY, USA, UIST '09, 153–156.

Medeiros, D. P. S., Teixeira, L., and Raposo, A. 2012. Navigation methods in engineering models using mobile devices. In *Proceedings of the 13th Symposium on Virtual and Augmented Reality 2012*.

NANCEL, M., WAGNER, J., PIETRIGA, E., CHAPUIS, O., AND MACKAY, W. 2011. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, New York, NY, USA, CHI '11, 177–186.

NORONHA, H., CAMPOS, P., JORGE, J., ARAUJO, B. D., SOARES, L., AND RAPOSO, A. 2012. Designing a mobile collaborative system for navigating and reviewing oil industry cad models. In *Proceedings of NordiCHI 2012*, ACM.

PACE, J. 2013. The ways we play, part 2: Mobile game changers. *Computer 46*, 4, 97–99.

PIERCE, J. S., FORSBERG, A. S., CONWAY, M. J., HONG, S., ZELEZNIK, R. C., AND MINE, M. R. 1997. Image plane interaction techniques in 3d immersive environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM, New York, NY, USA, I3D '97, 39–ff.

PINTO, F., TECNOLOGIA, M., BUAES, A., FRANCIO, D., BINOTTO, A., AND SANTOS, P. 2008. Bratrack: a low-cost marker-based optical stereo tracking system. *SIGGRAPH 08: ACM SIGGRAPH 2008 posters*.

RADHAKRISHNAN, S., LIN, Y., ZEID, I., AND KAMARTHI, S. 2013. Finger-based multitouch interface for performing 3d {CAD} operations. *International Journal of Human-Computer Studies 71*, 3, 261 – 275.

RUAN, H., QIAN, Y., ZHANG, Y., AND ZHOU, M. 2010. Touchinteract: An interaction technique with large displays using touchscreen-phone. In *Ubiquitous Intelligence Computing and 7th International Conference on Autonomic Trusted Computing (UIC/ATC), 2010 7th International Conference on*, 262–265.

RUIZ, J., LI, Y., AND LANK, E. 2011. User-defined motion gestures for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM, New York, NY, USA, CHI '11, 197–206.

SHI, S. 2011. Building low-latency remote rendering systems for interactive 3d graphics rendering on mobile devices. In *Proceedings of the 19th ACM international conference on Multimedia*, ACM, 859–860.

TEIXEIRA, L., TRINDADE, D., LOAIZA, M., CARVALHO, F. G. D., RAPOSO, A., AND SANTOS, I. 2012. A vr framework for desktop applications. In *Proceedings of the 2012 14th Symposium on Virtual and Augmented Reality*, IEEE Computer Society, Washington, DC, USA, SVR '12, 10–17.

TELKENAROGLU, C., AND CAPIN, T. 2012. Dual-finger 3d interaction techniques for mobile devices. *Personal and Ubiquitous Computing*, 1–22.

VANACKEN, L., GROSSMAN, T., AND CONINX, K. 2007. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *3D User Interfaces, 2007. 3DUI '07. IEEE Symposium on*, –.