

## Proposal and evaluation of a tablet-based tool for 3D virtual environments

Daniel Medeiros, Felipe Carvalho, Lucas Teixeira, Priscilla Braz, Alberto Raposo  
*Tecgraf - Computer Graphics Group*  
*PUC-Rio (Pontifícia Universidade Católica do Rio de Janeiro)*  
*Rio de Janeiro, Brasil*  
 {danielps,kamel,lucas}@tecgraf.puc-rio.br; {pbraz,abraposo}@inf.puc-rio.br

Ismael Santos  
*CENPES*  
*Petrobras*  
*Rio de Janeiro, Brasil*  
 ismaelh@petrobras.com.br

**Abstract**—The introduction of embedded sensors in smartphones and tablets allowed the use of these devices to interact with virtual environments. These devices also have 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 tasks, such as navigation, selection, manipulation, system control and symbolic input. This tool is for general-purpose systems, as well as, engineering applications. Generally this kind of application uses specific interaction devices with four or more degrees of freedom and a common keyboard and mouse for tasks that are naturally non-immersive, such as symbolic input (e.g., text or number input). This article proposes a new tablet-based device that can perform all these major tasks in an immersive environment. It also presents a study case of the use of the device and some user tests.

**Keywords**-3D interaction; virtual reality; mobile devices; virtual environments;

### I. INTRODUCTION

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 suitable for 3D Interaction. Another problem is that they can only be used on a table, thus becoming unfeasible to use in environments such as CAVEs [1].

The current availability of mobile devices such as tablets with sensors and multi-touch screens is an incentive to their use in three-dimensional interaction techniques. A major advantage of the tablet is its portability and diversity of embedded sensors. Sensors like accelerometers and magnetometers inform the spatial orientation of the device and enable to track the user position. The touch screen provides greater flexibility for this type of device, allowing the inclusion of additional features to 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.

There is a large amount of research in 3D interaction area related to the use of devices for creating better techniques for navigation, selection and manipulation of objects in a virtual environment. Two prominent problems in this field are the need to naturally control non-immersive tasks, such

as symbolic input, and the learning curve required for handling devices specifically created for 3D interaction.

In this work we present a tablet-based virtual reality tool that provides all major categories of 3D interaction tasks described in Bowman et al. [2], namely, navigation, selection, manipulation, system control, and symbolic input. Despite the great representation potential of mobile devices in virtual environments, we did not find a tool that had all that interaction tasks. Thus, this work addresses the study of the incorporation of these interaction tasks into a single tablet-based tool for immersive virtual environments.

The above mentioned categories of 3D interaction tasks are discussed in the following section. Then, in Section III, we discuss some work related to the use of mobile devices in 3D interaction. In Section IV we describe the concepts used in the design of the tool, and in Section V we discuss its implementation. Sections VI and VII present, respectively, user tests and their results. Finally, Section VIII concludes the paper.

### II. 3D INTERACTION TECHNIQUES

This section describes the categories of 3D interaction tasks defined by Bowman et al. [2].

#### A. Selection and Manipulation

Selection techniques are those that allow the user to choose one or more objects in the virtual environment. Manipulation techniques modifies the properties, normally the position and orientation, of a selected object. Some manipulation and selection techniques described in Bowman et al. [2] and Poupyrev et al. [3] show different approaches according to the device chosen and by their representations on the virtual environments. These techniques are often classified as exocentric and egocentric and are normally divided into two subtasks: the indication of the object to be selected and the subtask of confirming its selection. Exocentric techniques use an external view of the virtual environment. An example of this type of techniques is the World in Miniature [4], which allows the user to interact with the objects in a miniature version of the environment.

Egocentric techniques are the ones that use a direct interaction metaphor. Raycasting [5], which can be used

by pointing devices such as the Flystick (Figure 1), is a technique in this category. In this technique the user uses such devices to point to a specific object in the scene, controlling a ray that is used to intersect the desired object. After the selection, the user can manipulate such object with transformations such as rotation and translation.

A problem related to this technique is the difficulty caused by the sensitivity of some devices. Some variations of this technique such as the Flashlight [6] and Aperture [7] solve this issue by using a cone object instead of a ray to select multiple objects. The cone aperture is modified dynamically to control the choice of the objects.

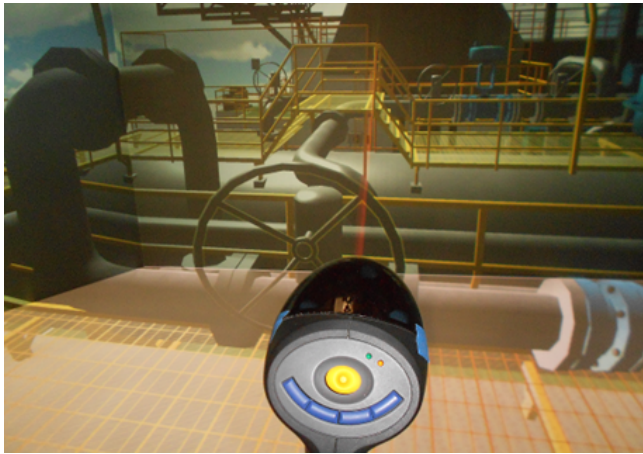


Figure 1. Raycasting using a Flystick device.

In the Image Plane Technique [8] the user selects and manipulates objects through gestures in a 2D projection located in front of him/her. One use of this technique is with the vGlove [9], that uses a virtual touch plane located in front of the user to map selection events. The vGlove also has an embedded system that gives force feedback when the user selects an object.

A problem of raycasting-based techniques is the undesired selection of multiple objects, which sometimes requires a second phase called disambiguation. To solve this problem, there are Progressive Refinement selection techniques, a way to gradually reduce the amount of selectable objects. The progressive refinement process can be done by many discrete tasks or as a continuous process. Furthermore, several ways can be used to select a group of objects. One possibility is to use the objects' characteristics such as color, size and shape, for example. Another way is to use solids or plane figures to determine selection areas where the target is. An example of this technique is the Bubble Selection [10], which uses scalable spheres to improve the selection.

SQUAD [11] is another example of progressive refinement technique (Figure 2). This technique uses a first phase where the user casts a sphere that specifies a volume containing the target object. The user may then disambiguate the selection

in one or more phases on a menu that distributes the objects randomly in four quadrants and allows to select between them. Another option of this technique is to use raycasting to select objects in environments which don't require a high degree of precision.

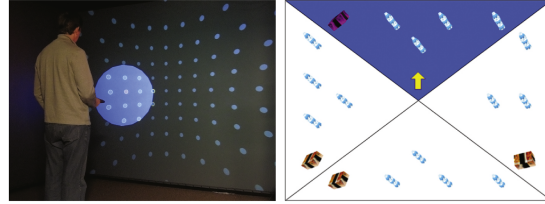


Figure 2. The two phases of SQUAD technique: at left, *sphere-casting*, at right, the QUAD menu [11].

Another related technique is the Discrete Zoom [12] — Figure 3. This technique divides the screen into four quadrants, which enables the user to choose a specific portion of the screen. When it happens, it is shown an expanded view of the chosen quadrant, decreasing the number of objects that can be selected. This operation may be done several times until the desired object is selected.

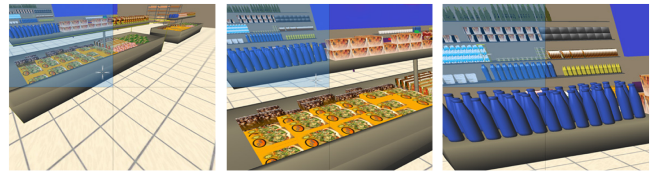


Figure 3. *Discrete Zoom Technique* [12].

Another category of selection and manipulation technique is known as Direct User Control [13]. In this kind of technique, the user selects and manipulates objects directly, using the hands. A classic example of this category is the Virtual Hand, where the user's hands is tracked and mapped directly into the virtual environment. A problem of this approach is the inability to interact with objects that are far from the user. This limitation is solved by the Go-Go Technique [14], where the reach of the virtual hand is interactively modified when the user goes beyond a certain threshold distance, allowing the selection of far-located objects on the scene.

### B. Navigation

Navigation is the act of taking the user from one place to another in a given direction in the virtual environment. Navigation techniques can be of Exploration, when the user has no specific goal to navigate through the environment and Search, when the user has a goal and may or may not rely on additional information (wayfinding) to assist him to get to his goal [15]. Navigation may also be classified as a form

of manipulation, because it is the manipulation of a virtual camera or a view of the user within the virtual environment. An example of this is the World in Miniature, already mentioned, where the user navigates and manipulates objects on a reduced version of the virtual environment. Other navigational techniques consist in indicating an object in the virtual environment, usually through a pointing. Then the technique is responsible for positioning the virtual camera at a certain distance from the object pointed. For example, the technique Zoomback [16] this distance is two feet, while in LaserGrab technique [16] this distance corresponds to the relative distance between the head and the hand of the user. In addition, some techniques assume manipulation of the virtual camera based on metaphors, for example, “Eyeball-in-Hand”, “Flying” and “Scene-in-Hand” [17] [18].

There are many classifications for navigation techniques in the literature [2] [19] [5]. In one of those classifications, the techniques are divided in active navigation tasks, where the user directly controls the locomotion movements inside the virtual environment; and passive, where the movements are controlled by the system. In that classification some authors include a third category called Route-Planning [20], where the user indicates a route or the object to be searched and the system performs the necessary movements to reach a certain goal [21]. Another classification divides the techniques by the way the navigation occurs in the virtual environment [5], in a physical or virtual way. In physical navigation techniques the user controls rotation and translation in the virtual environment using movements with its own body, that is normally tracked by a dedicated tracking system with six degrees of freedom (DOFs); in virtual techniques the body of the user remains stationary while the movement is done, normally by a specific interaction device. As stated by Bowman et al. [2] both classifications are complementary, making possible the combination of different techniques of both categories in one system.

The physical navigation category intends to emulate natural movements of the human body. One of the first uses of this type of technique was the walking metaphor. Although this is the most natural form of navigation, it presents some problems such as the limitation of space, since most of the trackers used do not operate well at a great distance from the user. One way to solve this problem is called Walk In Place [22]. In this technique the user emulates the gesture of walking without moving, decreasing the limited physical space needed, but compromising the realism of interaction [2]. Some uses of physical navigation techniques are those that use the Microsoft Kinect. This device is able to track user movements and use them to make interactions with the virtual environment [23] [24].

There are works that attempt to mix physical and virtual navigation techniques in a virtual environment. A very common way to mix the two forms is to use users’ movements to control the point of view on the virtual environment, de-

forming the projection matrix of the environment according to the user’s position within the immersive environment, using specific devices such as the Flystick for navigation, manipulation and selection tasks. In one of these works, by Cirio et al. [25], a technique called Real Walking is presented. This technique keeps the user in a secure position in relation to the immersive environment, where a tracked device can be used to perform the navigation task.

### C. System Control

Besides navigation, selection and manipulation techniques it is still possible to incorporate different elements in virtual environments, such as buttons, menus and check boxes. These elements, when used to send commands to perform a certain action in the virtual environment, are called elements of System Control.

Bowman et al. [26] [27] used Pinch Gloves (Figure 4) together with Head Mounted Displays (HMDs) for system control. They proposed a system called TULIP to associate menu items to the fingers, which are selected by pressing the thumb with the finger that contains the corresponding item (Figure 5). Mine et al. [28] proposes a menu system that is triggered by the movement of the user’s head and uses a physical device to confirm the selection of a menu item. Other studies in the literature propose the use of mappings elements of WIMP interfaces in virtual three dimensional environments, including: VEWL that abstracts concepts of windows and buttons [29]; Liang et al. introduces the concept of ring menus [6] and Gerber et al., spin menus [30].



Figure 4. Pinch Gloves [27]

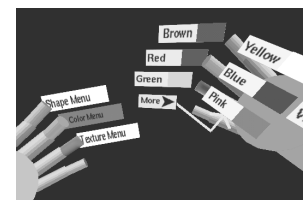


Figure 5. Tulip System [27]

### D. Symbolic Input

Another task which could also be incorporated to a Virtual Reality system is the Symbolic Input. Symbolic Input allows the user to communicate symbolic information, such as text, numbers and other symbols. Tasks of this type are commonly found in graphical user interfaces (GUIs), on WIMP environments through the use of applications for editing text, email and spreadsheets. These tasks are still under-explored in virtual environments. Kitty (Figures 6 and 7) is an example of device used for symbolic input [31]. It is similar to a glove, developed for typing text without the use of a keyboard. The basic idea is to associate letters

to a combination of fingers positions. Another technique is proposed by Bowman et al. [26] [32], where Pinch Gloves are used for symbolic input tasks.



Figure 6. *KITTY* [31]



Figure 7. *KITTY*  
*example of use* [31]

### III. 3D INTERACTION USING MOBILE DEVICES

In the literature, there are several uses of smartphones and tablets for 3D interaction. 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 sent to a remote application.

A problem of direct interaction with mobile devices is the occlusion caused by the user's fingers to interact with the VE, often hiding the object of interest. Telkernaroglu et al. [33] propose a solution to this problem by means of selection and manipulation techniques that use 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.

Among the works that discuss the design and evaluation of indirect navigation techniques for three-dimensional virtual environments we may cite [34], [35], [36], and [37]. In Sensor Fusion [38], a Walk In Place navigation technique is proposed. This technique uses the accelerometer and magnetometer sensors present in smartphones coupled to the feet of the user. These sensors are able to faithfully capture the user's walk movement and allow the navigation even in complex environments using visualization systems like a CAVE, for example. However, as shown in Medeiros et al. [39] these works do not take into account precision tests of devices, a characteristic considered important in virtual engineering environments.

Mobiles have also been used for selection tasks in two-dimensional environments. The ARC Pad [40] maps the user touches in the mobile device to a remote computer, emulating the movements of a mouse in a WIMP environment. The work proposed by Ruan et al. [41] 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. [42] proposes techniques for selection and manipulation of 2D objects in displays located away from the user using image processing in the video captured by the device's camera. Nancel et al. [43] 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 in a 3D environment, a feature considered important for interaction with virtual engineering environments. Debarba [44] proposes a system using smartphones for multi-resolution environments that utilizes a virtual window scheme similar to the Image Plane Technique [8], used to improve the accuracy of object selection in this type of environment (Figure 8). However, it is noted that because of the size of the device used it does not include additional functionality in the proposed solution. Anyway, the idea of virtual window increases the accuracy of selection, and was used in the design of the tool we are proposing in the present paper.



Figure 8. LOP-Cursor [44]

The use of mobile devices in virtual environments is not limited to tasks of navigation and selection. Platforms such as Microsoft SmartGlass and Nintendo Wii U [45] 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 [46], a quad rotor helicopter, i.e. an helicopter toy with four helices that is able to glide in the air. The control of the drone is made through a mobile device. Using an application installed in the device, the user can view the contents captured by the cameras present in the airplane, one on the front and one at the bottom, and sent by a dedicated wireless network created between the device and the drone.

### IV. CONCEPT

In spite of the potential provided by the screen of mobile devices during interaction in a virtual environment, we did not find tools that use all the categories of techniques proposed by Bowman et al. [2]. From this start 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 virtual reality tool based on tablets that aggregates all 3D interaction major tasks. For navigation, an in-screen joystick is used. For Selection, a progressive refinement technique adapted to use a “Eyeball-in-Hand” (EiH) metaphor was chosen [17] [18]. The EiH is an extra virtual camera, called EiHCam, that provides a visualization from the viewpoint of the user hand, in our case, from the viewpoint of the tablet. This allows a 3D correct view from the world and the EiHCam can be slight sloped in relation with the screen, what is intuitively more comfortable than a virtual window. We use a 6DoF tracker fixed in the tablet to measure its position and the orientation. For this, optical trackers such as ARTracker and BraTracker[47] are used to ensure accuracy to the proposed technique. The tablet’s touch screen is used for manipulation, system control and symbolic input tasks.

This EiHCam viewpoint is represented in the virtual environment through a truncated semi-transparent pyramid (frustum), rendered along with the virtual scene. It is used in a similar way to that used in Bubble Selection [10] and Cone Casting [7], where a solid or a plane is used to define the group of interest. Figures 10 and 11 illustrate the concept of the frustum proposed. The user can also increase or decrease the size of the frustum, with the scale gesture (Figure 9- item B).

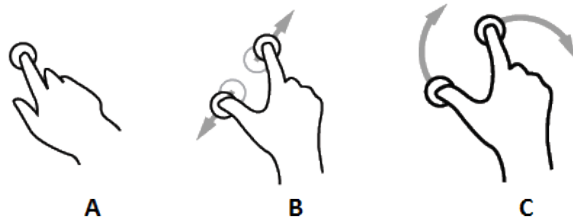


Figure 9. Selection/Manipulation gestures. (a) Tap (b) Scale (c) Rotation

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

In the image of the EiHCam in the Touchpad, the user can select the object of interest with a tap above it (Figure 9 - item A). After selection, the object is marked and the user can confirm his/her selection with an extra tap on the object in Touchpad. 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

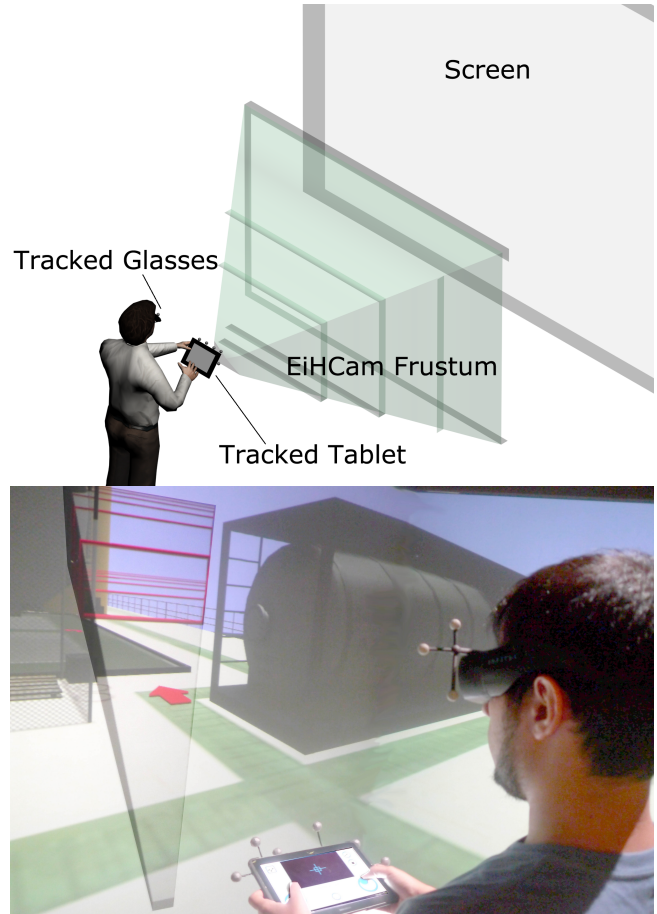


Figure 10. The frustum drawn on screen based on the tablet’s position.

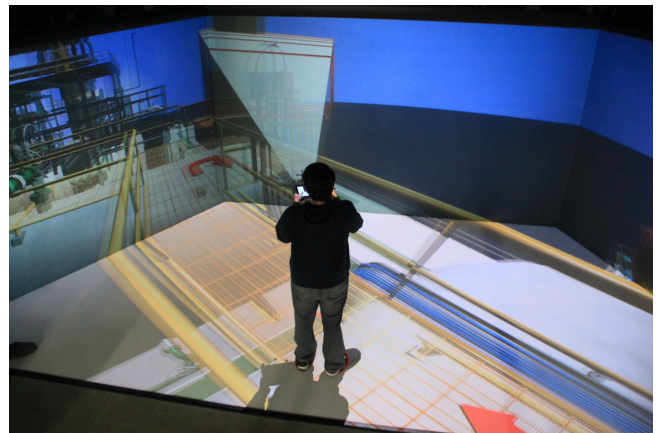


Figure 11. Testing the frustum.

9 - items B and C, respectively). For an even more precise selection zoom in and zoom out events on the touchpad image can be executed 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 movement in a first person view. The use of directionals is justified by the low precision of the sensors embedded on the device, that need some element to filter or adjust its precision [39].

By the nature of selection and manipulation technique proposed in this paper, we saw the possibility of using 3D widgets for system control. This is made possible because the proposed technique maps the virtual environment in the two-dimensional interface of the tablet. 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 this elements.

Finally, the integration of symbolic input on the proposed tool is possible by a virtual keyboard that can be triggered via buttons interface. A possibility for users of this type of technique is to add annotations to a determined previously selected object, for example.

## V. DEVELOPMENT

One of the main requirements of our proposal was the maximum decoupling between the mobile device 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 in 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 12) 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 defined gestures (Figure 9). 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 rotation (Figure 12 - item B and C, respectively). To spatially locate the tablet in relation to the screen a marker is used as 3D tracking point together with a ARTracker optical tracking system [48].

In this interface the user also has controls to add an annotation, or tag, to a selected object (Figure 12 - item G), switch the selection mode (Figure 12 - item D), lock the EiHCam image or the ray to start the selection/manipulation mode (or unlock to activate the navigation mode) (Figure 12 - item E), and can configure some application features such

as the IP Address of the graphic application used (Figure 12 - item A). The interface has also a textual label to provide feedback of the graphic application, such as the name of the selected object or rotation value of a selected object (Figure 12).

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 [49] 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. Figure 13 shows an overview of the system.

### A. Study Case

To validate the developed tool we chose the SimUEP-AmbSim application (Figure 14), a training simulator for oil platforms developed with the Unity3D Engine [50]. 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. In some of the specific goals of the application the user needs to manipulate certain objects within the platform, valves for example, that when selected can be rotated in order to open or close them (Figure 15). To achieve the proposed objective there is the possibility to mark certain objects, inserting information or tags on them that helps on the execution of the specified tasks.

The integration with the developed mobile tool was made using the dynamic template library (dll) from the VRINPUT readers 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.

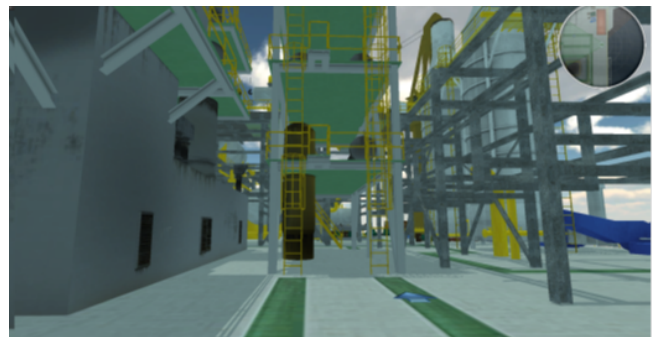


Figure 14. Screenshot of SimUEP- AmbSim application.

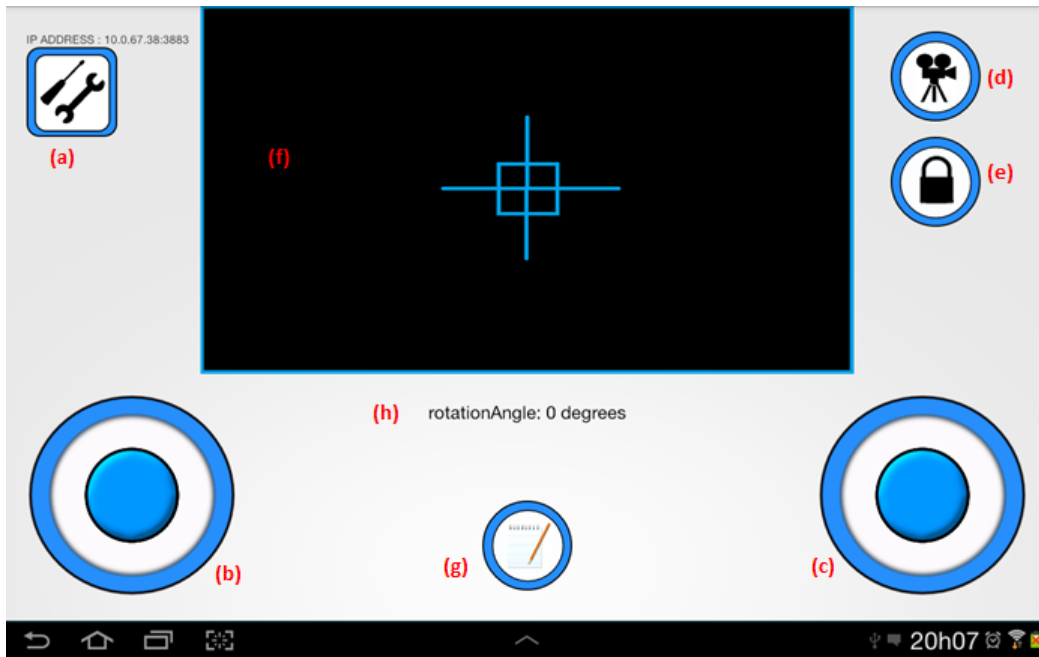


Figure 12. Mobile application interface.

]

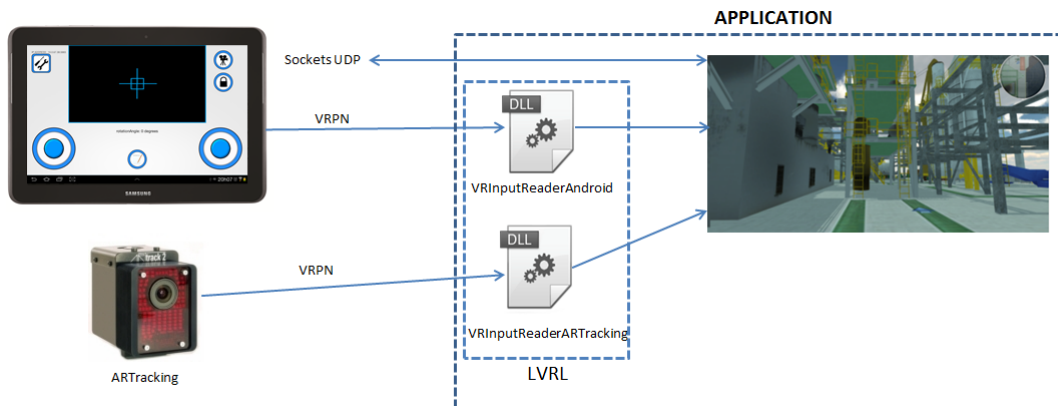


Figure 13. System architecture overview.

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, the Touch Pad (Figure 12 - item F) is used for selection (single touch) and object manipulation (double touch) on selected objects. Keyboard events are handled separately as described above, using UDP sockets. More details will be explained in the following subsections, grouped by its respective category.

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 valves. Other objects require more precise gestures, as is the case of manual valves. These gestures are mapped in real-time on the graphical application.

The object selection is performed when the user approaches a selectable object, position the EiHCam in a way that the camera frustum contains the desired object. Right after it, the user presses a button on the interface (Figure 12 - item D) then a rendered image from the EiHCam is sent through the network and it is drew on the Touch Pad (Figure 12 - item F). After that the user can touch the desired object directly causing its selection. If the object is properly

selected, it will be highlighted and with an extra touch on the same object, it can be manipulated. The objects such as valves can be manipulated using the rotation gesture (Figure 9 - item B) that performs the rotation around the axis of the valve clockwise or counterclockwise, depending on the gesture made by the user.

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 EiHCam position and intercepts the point mapped on the near plane (Equation 1). The mapping of the touch on the device is showed on the Figure 16.

$$r_{dir} = ||\vec{p}_{wc} - cam\vec{m}_{pos}|| \quad (1)$$

The positioning of the EiHCam is made by the direct use 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. In the SimUEP-AmbSim, because it was developed on Unity3D, a correction on the Z coordinate was necessary.

For a more precise selection of objects within the SimUEP-AmbSim a zoom technique was implemented on the rendered image of the EiHCam. This event occurs where there is no selected (or highlighted) object and two fingers are detected by the system. This functionality was



Figure 15. Selection/manipulation of a valve.

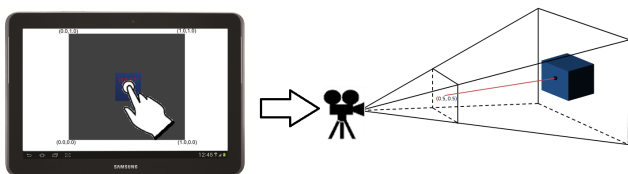


Figure 16. Touch Interface.

implemented using the scale gesture used, that measures the zoom in and zoom out by the euclidean distance between the two fingers x,y coordinates (Equation 2). This gesture is then mapped on the camera field of view. These two magnitudes are directly proportional since the higher the distance of the two fingers, the higher the fov will be calculated (Equation 3).

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

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

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, 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 could be manipulated using the gestures used in the previous technique.

2) *Image Transmission*: To send the rendered images on Unity, the EiHCam renders the scene and saves it in 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, that allows to reduce the size of the final file but preserving 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 amount 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 [51] [52].

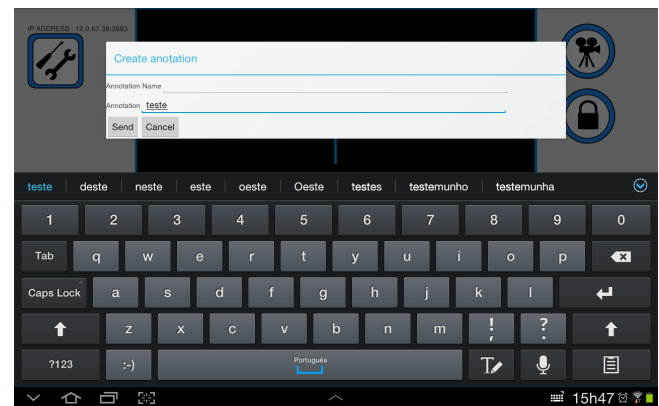


Figure 17. Symbolic Input: Annotation Creation.



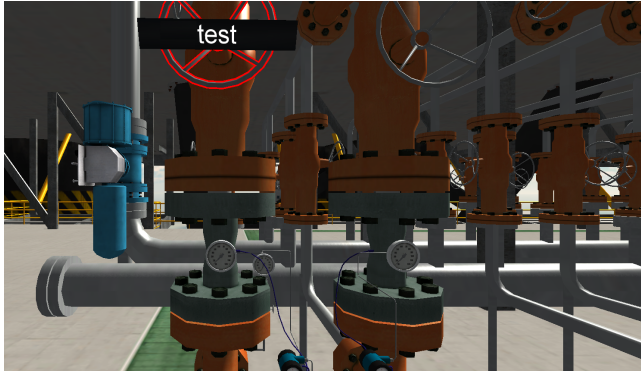


Figure 18. Insertion of tags on valves.

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 12. Then a window appears in the interface of the tablet (Figure 17) and so the user can enter the information he wants and push the button “Send”. Finally, a 3D plane object containing the typed text is positioned in the center of the marked object (Figure 18).

#### VI. USER STUDY: PARTICIPANTS AND PROCEDURES

We conducted a study with users aiming to investigate their perception of the proposed interaction technique for the tablet based solution. For this, users performed a task sequence using such technique in specific scenarios. In addition, we tried to identify their difficulties, ease of learning and suggestions for improvement of the device’s use.

In this study, we recruited ten users and most of them had experience with 3D interaction devices. Just a user had no experience with such devices. However, he had considerable experience in the use of 3D environments for viewing. The average age of users was thirty-six with a standard deviation of eleven. All users are graduated in the computing area and about 90% of them have postgraduate in this area.

The test was divided into six stages: 1) introduction to the study and application of pre-test questionnaire; 2) explanation about the application and the device’s use; 3) training with the device; 4) task execution; 5) application of post-test questionnaire; 6) semi-structured interview.

At first, we explained the test objectives. Then, the users completed a pre-use questionnaire to raise the participants profile regarding the experience with interaction devices.

Subsequently, we showed a brief description of the application and the device. In order to familiarize the user with them, he performed a task in a training scenario, where he could test the proposed technique.

After performing the training task, we have given some instruction and the user performed the test tasks. On the text feedback area of the device, there were also instructions to complete each task (Figure 12 - item H). Then, users

completed post-test questionnaire for each task. This questionnaire consisted of the following statements:

- S1: I am satisfied with the time spent to complete the first task.
- S2: It was easy to select objects in the first task.
- S3: I am satisfied with the time spent to complete the second task.
- S4: It was easy to select objects on the second task.
- S5: It was easy to insert tags on objects in the second task.
- S6: I learned to use the tablet interface quickly.
- S7: It was easy to navigate in the environment using the tablet.
- S8: It was easy to manipulate objects using the tablet.
- S9: I felt comfortable using the tablet.
- S10: This tool has all the functions and features required for interaction in virtual environments.
- S11: I am satisfied with the use of tablet to interact with virtual environments.

Below each of these statements there was a scale from 1 to 5, where 1 means that the user does not agree completely with statement and 5 means he fully agrees with it.

In addition to the questionnaires, we conducted a semi-structured interview in order to capture the participants’ perceptions about the accomplished tasks, clarify about their answers on the post-test questionnaire and get improvement suggestions for the proposed tool.

In the tests, users have performed the tasks in visualization environment with a CAVE (Figure 19). The CAVE used has four projection screens and an optical tracking, which surrounds the user and ensure a higher level of immersion. To improve the selection of the proposed tool we used head tracking that modify the user’s view to match the frustum correctly. The execution of all tasks was recorded to assist in analyzing the results.



Figure 19. The CAVE used in the study.

Two scenarios were defined for the execution of the tasks and evaluation of interaction techniques with the tablet in a virtual platform.

In the first scenario, the user had to select a helmet, open doors and navigate to a point on the platform to interact with a valve.

In the second scenario, the user must navigate to the platform helipad platform to check if there were lights burned and extinguishers lacking. After finding some burned light and fire box without extinguisher, he must select such objects and create tags to indicate the fault.

Due to the complexity of the environment used, waypoints were placed in the environment for the user complete tasks more easily. These waypoints are represented by arrows positioned on the floor of the platform indicating the path for the task.

## VII. RESULTS

In this section, we present the main observations made during the tests as well as the difficulties and suggestions from users about the use of the proposed technique and used device.

An important point about the users who participated in the tests is that beyond the ten participants in previous section, a participant started to perform the tasks, but he did not complete the test. The user felt bad using the device in the CAVE and interrupted the task. Thus, his data were not considered in this analysis.

In Figure 20, we can see the average of the answers from users to statements S1-S11.

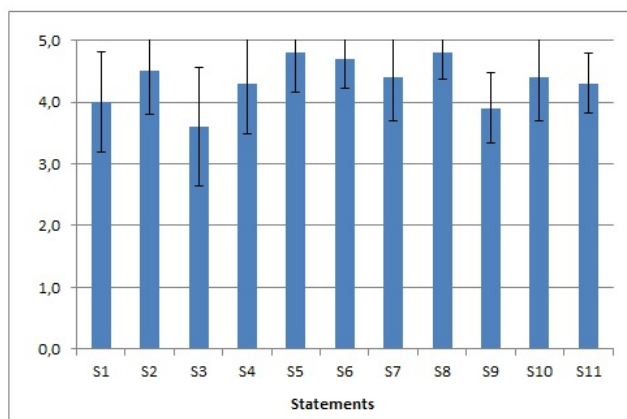


Figure 20. Average responses from users

In general, users liked the tablet as a device for 3D interaction. Regarding the interface, some users emphasized that they found it simple, without unnecessary information, which facilitated interaction in immersive environments. About using the device, users have agreed that it was easy to manipulate objects using the tablet. One user commented that he considered the manipulation with the device more intuitive than with Flystick. Another issue that was noticed in the analysis of the data was the dissatisfaction of the users

with the time spent on the tasks (as seen in figure 20). One of the reasons for this dissatisfaction reported by the users was the small amount of time that was spent in the training procedure. Users reported that if they could spend more time on the training they could have completed the objectives more easily and in a smaller amount of time.

Among the major difficulties encountered with the tablet use, we can highlight the device's weight and the lack of tactile feedback on use of directional. Some users suggested using a smaller tablet in the environment interaction. About the use of directional, users suggested using a tactile feedback, so they do not need to look at the tablet for navigating and rotate the camera. In addition, some users pointed out that they had difficulties with the frustum in navigation and they have suggested the possibility to enable/disable its use.

Concerning the transition from 2D to 3D, a user has reported that this change detracted somewhat immersion. However, like all other users, he considered that the use of the device made the selection of objects much easier.

## VIII. CONCLUSION

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

The use of virtual reality simulators has been an efficient way to reduce costs in staff training at 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 by 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 application in another areas of interest to validate the proposed tool as the one presented by Noronha et al. [34].

## ACKNOWLEDGEMENTS

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

## REFERENCES

- [1] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The cave: audio visual experience automatic virtual environment," *Commun. ACM*, vol. 35, no. 6, pp. 64–72, Jun. 1992. [Online]. Available: <http://doi.acm.org/10.1145/129888.129892>
- [2] D. A. Bowman, E. Kruijff, J. J. LaViola, and I. Poupyrev, *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

- [3] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst, "Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques," *Computer Graphics Forum*, vol. 17, no. 3, pp. 41–52, 1998. [Online]. Available: <http://dx.doi.org/10.1111/1467-8659.00252>
- [4] R. Stoakley, M. J. Conway, and R. Pausch, "Virtual reality on a wim: interactive worlds in miniature," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 265–272. [Online]. Available: <http://dx.doi.org/10.1145/223904.223938>
- [5] M. Mine *et al.*, "Virtual environment interaction techniques," *UNC Chapel Hill computer science technical report TR95-018*, pp. 507 248–2, 1995.
- [6] J. Liang and M. Green, "Jdcad: A highly interactive 3d modeling system," *Computers and Graphics*, vol. 18, no. 4, pp. 499–506, 1994.
- [7] A. Forsberg, K. Herndon, and R. Zeleznik, "Aperture based selection for immersive virtual environments," in *UIST '96: Proceedings of the 9th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, 1996, pp. 95–96.
- [8] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine, "Image plane interaction techniques in 3d immersive environments," in *Proceedings of the 1997 symposium on Interactive 3D graphics*, ser. I3D '97. New York, NY, USA: ACM, 1997, pp. 39–ff. [Online]. Available: <http://doi.acm.org/10.1145/253284.253303>
- [9] P. Gallotti, A. Raposo, and L. Soares, "v-glove: A 3d virtual touch interface," in *Virtual Reality (SVR), 2011 XIII Symposium on*, 2011, pp. 242–251.
- [10] L. Vanacken, T. Grossman, and K. Coninx, "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*, 2007, pp. –.
- [11] R. Kopper, F. Bacim, and D. Bowman, "Rapid and accurate 3d selection by progressive refinement," in *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, 2011, pp. 67–74.
- [12] F. Bacim, R. Kopper, and D. A. Bowman, "Design and evaluation of 3d selection techniques based on progressive refinement," *International Journal of Human-Computer Studies*, vol. 71, no. 78, pp. 785 – 802, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1071581913000384>
- [13] W. R. Sherman and A. B. Craig, "Understanding virtual reality-interface, application, and design," *Presence: Teleoperators & Virtual Environments*, vol. 12, no. 4, pp. 441–442, 2003.
- [14] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa, "The go-go interaction technique: non-linear mapping for direct manipulation in vr," in *Proceedings of the 9th annual ACM symposium on User interface software and technology*. ACM, 1996, pp. 79–80.
- [15] 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>
- [16] R. Zeleznik, J. LaViola, D. Acevedo, and D. Keefe, "Pop through buttons for virtual environment navigation and interaction," in *Proceedings of Virtual Reality 2002*, March 2002, pp. 127–134.
- [17] C. Ware and D. R. Jessome, "Using the bat: A six-dimensional mouse for object placement," *IEEE Comput. Graph. Appl.*, vol. 8, no. 6, pp. 65–70, 1988.
- [18] C. Ware and S. Osborne, "Exploration and virtual camera control in virtual three dimensional environments," in *SIG3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*. New York, NY, USA: ACM, 1990, pp. 175–183.
- [19] R. Ball, C. North, and D. A. Bowman, "Move to improve: promoting physical navigation to increase user performance with large displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '07. New York, NY, USA: ACM, 2007, pp. 191–200. [Online]. Available: <http://doi.acm.org/10.1145/1240624.1240656>
- [20] D. A. Bowman, E. T. Davis, L. F. Hodges, and A. N. Badre, "Maintaining spatial orientation during travel in an immersive virtual environment," *Presence*, vol. 8, no. 6, pp. 618–631, 1999.
- [21] H. Fang, S. Ong, and A. Nee, "Interactive robot trajectory planning and simulation using augmented reality," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 227 – 237, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0736584511001116>
- [22] M. Slater, M. Usoh, and A. Steed, "Taking steps: the influence of a walking technique on presence in virtual reality," *ACM Trans. Comput.-Hum. Interact.*, vol. 2, no. 3, pp. 201–219, Sep. 1995. [Online]. Available: <http://doi.acm.org/10.1145/210079.210084>
- [23] G. Ren, C. Li, E. O'Neill, and P. Willis, "3d freehand gestural navigation for interactive public displays," *Computer Graphics and Applications, IEEE*, vol. 33, no. 2, pp. 47–55, 2013.
- [24] P. Dam, P. Braz, and A. Raposo, "A study of navigation and selection techniques in virtual environments using microsoft kinect®," in *Virtual Augmented and Mixed Reality. Designing and Developing Augmented and Virtual Environments*, ser. Lecture Notes in Computer Science, R. Shumaker, Ed. Springer Berlin Heidelberg, 2013, vol. 8021, pp. 139–148. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-39405-8\\_17](http://dx.doi.org/10.1007/978-3-642-39405-8_17)
- [25] G. Cirio, P. Vangorp, E. Chapoulie, M. Marchal, A. Lecuyer, and G. Drettakis, "Walking in a cube: Novel metaphors for safely navigating large virtual environments in restricted real workspaces," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 18, no. 4, pp. 546–554, 2012.

- [26] D. A. Bowman, C. J. Rhoton, and M. S. Pinho, "Text input techniques for immersive virtual environments: An empirical comparison," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, no. 26. SAGE Publications, 2002, pp. 2154–2158.
- [27] —, "Text input techniques for immersive virtual environments: An empirical comparison," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, no. 26. SAGE Publications, 2002, pp. 2154–2158.
- [28] M. R. Mine, F. P. Brooks, Jr., and C. H. Sequin, "Moving objects in space: exploiting proprioception in virtual-environment interaction," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 19–26. [Online]. Available: <http://dx.doi.org/10.1145/258734.258747>
- [29] D. Larimer and D. A. Bowman, "Vewl: A framework for building a windowing interface in a virtual environment," in *in Proc. of IFIP TC13 International Conference on Human Computer Interaction - Interact 2003*. IOS Press, 2003, pp. 1–5.
- [30] D. Gerber and D. Bechmann, "The spin menu: a menu system for virtual environments," in *Proceedings of the 2005 IEEE Conference 2005 on Virtual Reality*. IEEE Computer Society, 2005, pp. 271–272.
- [31] F. Kuester, M. Chen, M. E. Phair, and C. Mehring, "Towards keyboard independent touch typing in vr," in *Proceedings of the ACM symposium on Virtual reality software and technology*. ACM, 2005, pp. 86–95.
- [32] D. Bowman, C. Wingrave, J. Campbell, and V. Ly, "Using pinch gloves (tm) for both natural and abstract interaction techniques in virtual environments," 2001.
- [33] C. Telkenaroglu and T. Capin, "Dual-finger 3d interaction techniques for mobile devices," *Personal and Ubiquitous Computing*, pp. 1–22, 2012. [Online]. Available: <http://dx.doi.org/10.1007/s00779-012-0594-2>
- [34] H. Noronha, P. Campos, J. Jorge, B. d. Araujo, L. Soares, and A. Raposo, "Designing a mobile collaborative system for navigating and reviewing oil industry cad models," in *Proceedings of NordiCHI 2012*. ACM, 2012.
- [35] N. Katzakis, M. Hori, K. Kiyokawa, and H. Takemura, "Smartphone game controller," in *Proceedings of the 74th HIS SigVR Workshop*, Tokyo, Japan, 2011.
- [36] S. Radhakrishnan, Y. Lin, I. Zeid, and S. Kamarthi, "Finger-based multitouch interface for performing 3d {CAD} operations," *International Journal of Human-Computer Studies*, vol. 71, no. 3, pp. 261 – 275, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1071581912001279>
- [37] A. Benzina, M. Toennis, G. Klinker, and M. Ashry, "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*, ser. CHI Extended Abstracts (CHI EA) '11. New York, NY, USA: ACM, 2011, pp. 1519–1524. [Online]. Available: <http://doi.acm.org/10.1145/1979742.1979801>
- [38] J. Kim, D. Gracanin, and F. Quek, "Sensor-fusion walking-in-place interaction technique using mobile devices," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, 2012, pp. 39–42.
- [39] D. P. S. Medeiros, L. Teixeira, and A. Raposo, "Navigation methods in engineering models using mobile devices," in *Proceedings of the 13th Symposium on Virtual and Augmented Reality 2012*, Niteroi-RJ/Brazil, 2012.
- [40] D. C. McCallum and P. Irani, "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*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 153–156. [Online]. Available: <http://doi.acm.org/10.1145/1622176.1622205>
- [41] H. Ruan, Y. Qian, Y. Zhang, and M. Zhou, "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*, 2010, pp. 262–265.
- [42] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch, "Touch projector: mobile interaction through video," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 2287–2296. [Online]. Available: <http://doi.acm.org/10.1145/1753326.1753671>
- [43] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, and W. Mackay, "Mid-air pan-and-zoom on wall-sized displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 177–186. [Online]. Available: <http://doi.acm.org/10.1145/1978942.1978969>
- [44] H. Debarba, L. Nedel, and A. Maciel, "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*, march 2012, pp. 125 –132.
- [45] J. Pace, "The ways we play, part 2: Mobile game changers," *Computer*, vol. 46, no. 4, pp. 97–99, 2013.
- [46] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl, "Ar-drone as a platform for robotic research and education," in *Research and Education in Robotics-EUROBOT 2011*. Springer, 2011, pp. 172–186.
- [47] F. Pinto, M. Tecnologia, A. Buaes, D. Francio, A. Binotto, and P. Santos, "Bratrack: a low-cost marker-based optical stereo tracking system," *SIGGRAPH 08: ACM SIGGRAPH 2008 posters*, 2008.
- [48] (2012) Artrack motion capture. [Online]. Available: <http://www.ar-tracking.com/>
- [49] L. Teixeira, D. Trindade, M. Loaiza, F. G. d. Carvalho, A. Raposo, and I. Santos, "A vr framework for desktop applications," in *Proceedings of the 2012 14th Symposium on Virtual and Augmented Reality*, ser. SVR '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 10–17. [Online]. Available: <http://dx.doi.org/10.1109/SVR.2012.27>

- [50] R. H. Creighton, *Unity 3D Game Development by Example Beginner's Guide*. Packt Publishing, 2010.
- [51] S. Shi, "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, 2011, pp. 859–860.
- [52] F. Lamberti and A. Sanna, "A streaming-based solution for remote visualization of 3d graphics on mobile devices," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 2, pp. 247–260, 2007.