

Evaluating the Use of Virtual Reality on Professional Robotics Education

Mauro César Charão dos Santos, Vincenzo Abichequer Sangalli, Marcio Sarroglia Pinho

Virtual Reality Group - Computer Science School

PUCRS

Porto Alegre, Brazil

maurocharao@gmail.com, vicenzo.sangalli@acad.pucrs.br, marcio.pinho@pucrs.br

Abstract—This work consists of the study of techniques of robotics and virtual reality to develop a simulator that can be used in robotics schools, having an adequate visualization and a simple and intuitive way of interaction. For this, a 3D virtual environment for robotics was developed. Virtual reality resources have been incorporated to improve the visualization and to facilitate the user interaction with the environment. In order to evaluate the effectivity of the environment, user experiments were carried out on four different hardware configurations. During the simulations, the users had to create trajectories while implicitly defining reference points. From these experiments, automatic reports for the quantitative questions were generated, and questionnaires were filled for the qualitative questions. The results have shown that the use of virtual reality do helps the users in task execution, improving the visualization, reducing the time spent for the tasks and increasing the precision.

Keywords—education, robotics, virtual reality;

I. INTRODUCTION

According to Bell [8], experience is a powerful learning tool because it can provide a detailed understanding of contents, situations or processes. Experimentation in a real environment, however, is not always a viable alternative because of security matters, cost, or even availability of the object of study. In this context, to allow the exposure of learners to situations that represent reality and, with these, enable them to perform tasks, various types of tools have been developed. Among these tools, one of the most used are the simulators, which allows the students to try out the tasks several times, having more chances of learning. In addition, it is possible to conduct a safe training with simulators, which is not always possible in the real world. Simulators can also help lowering costs in cases where actual equipment or replacement materials are too expensive.

Simulators initially used textual interfaces, which usually received an input file or typed data and generated a report with the results [1]. Over the years, with the improvement in the processing capacity of the computers, two-dimensional simulators began to emerge, which presented a symbolic representation of reality [2]. From the 90's, 3D Virtual Environments (VE) started to appear, in which it is possible to perform interactions, simulating a real situation. These VEs

can be immersive when using Virtual Reality (VR) features such as helmets, gloves, trackers, or non-immersive ones, these being the most common to this day.

Although non-immersive simulators are the most common to date, lack of user immersion in VE may not be the best way to create an effective environment with easy-to-interact experiences that properly represent the real world. An example of this are the applications that need to manipulate three-dimensional objects, such as spatial geometry teaching, vehicle conduction training, environment exploration, and robotic arm control. In all these cases, immersive three-dimensional visualization and direct manipulation of objects in space, which are possible in VR environments, can generate a significant gain when compared to traditional applications, since the interaction becomes closer to reality and also more intuitive.

With the use of VR, the senses of human perception can be better utilized, since the stereoscopic visualization, coupled with user movements tracking, allows for a better understanding of the space, a more robust interaction and locomotion around the scene. Besides that, touch features allow the generation of a touch sensation, enhancing the perception of the environment.

In this context, this work conceived, implemented and evaluated the use of VR techniques to develop a simulator of a robotic arm to be used in schools, especially in professional educational environments. The developed environment, called VR Robotics, simulates the Scorbot-ER VII mechanical arm, manufactured by the Israeli company Eshed Robotec [7]. This simulator has a scenario editor in which it is possible to create a problem-situation and a 3D VE that allows navigation in the scenario, as well as the manipulation of all the objects in the scene. The simulator also allows the use of a Teach Pendant (TP), a kind of wired remote control, allowing the user to perform movements with the robotic arm.

To evaluate the various possible configurations of VR devices, and to verify their viability and suitability for the proposed simulation, scenarios with and without stereoscopy, three different types of visualization devices, and systems with and without user movement tracking were tested.

The remaining of this paper is organized as follows: some related works that contributed to the development of this work are presented in Section 2; some background about robotics education is described in Section 3; the user interface architecture is described in Section 4; the details about the

tests that were performed are described in Section 5; the results of the tests are presented in Section 6; and the conclusions are presented in Section 7.

II. RELATED WORK

There are several ways to program robotic arms, some of which require the user to know a programming language and, through it, issue commands to the robot. These procedures are called textual programming. However, a second category called non-textual (or guided) programming allows the robot to be "taught" through the physical movement of the mechanical arm. In this case, the desired path is stored automatically or manually, depending on the type of programming that was used. These two forms of programming are not mutually exclusive and can be combined, using the non-textual form when it is necessary to position the robot in specific locations and the guided form to perform the movements, passing through these positions.

The use of real robots in the teaching of robotics is unfeasible for several schools, due to the high costs of the equipment. According to Jason [9], there are four viable options to teach robotics in schools: using low-cost educational robots; using low-cost educational simulators; using VR simulators; using simulators along with one or more real robots to execute the programs created in the simulators.

Educational robots have low reliability and low-cost simulators usually have several limitations, besides making it difficult to use, since it generally does not have an adequate visualization. For this reason, VR simulators are becoming an interesting teaching alternative.

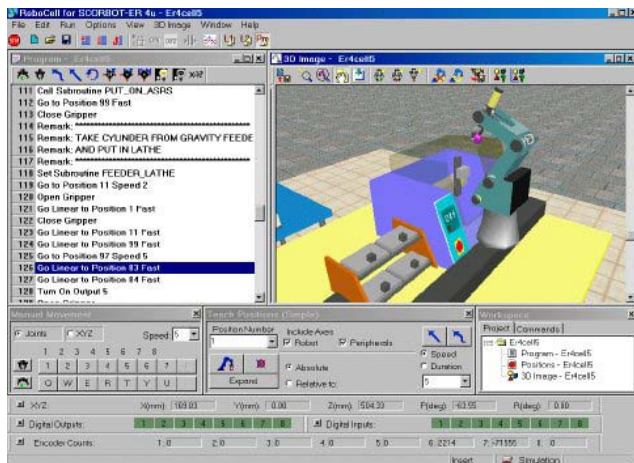


Fig. 1. RoboCell Program ([10])

For the Scorbot robot models manufactured by the Israeli company Eshed Robotec [7], there is the RoboCell simulator, created by the American company Intelitek [6], seen in Figure 1. This simulator integrates an old program of the same company, called Scorbase [11], with a 3D graphic module, in order to allow the creation of programs, the simulation of the execution and the collection of the points from the desired trajectory without the need of a real robot. The simulator

allows the sending of programs created to the real robot, as well as executing a synchronized simulation with the real robot. In this simulator, a cell consists of the description of the scenario in which the robot is in, placing a representation of the objects close to the robot, such as a table, parts, boxes, etc. In this way, the same cell can be used by several created programs.

Considering that a Scorbot robot costs approximately US\$6,500 and each license of the program costs approximately US\$1,000, the use of this simulator considerably reduces the cost of assembling a laboratory, although it still represents a high value, making it difficult to create laboratories for large classes.

A cheaper software alternative is the Denford's VR Robot [4], seen on Figure 2, which costs between US\$200 and US\$600 per station, depending on the options. Compatible with the Mitsubishi RV-M1 robot, this program features a 3D VE for simulation and allows an integration with the real robot. In the simulator, it is possible to program both in a non-textual way, through an interface similar to the TP called "VR teachbox", and in a textual way, with the robot's commands, allowing the programs created in the simulator to be sent to the robot. The problem in this case is the cost of the robot, which is approximately US\$16,000.

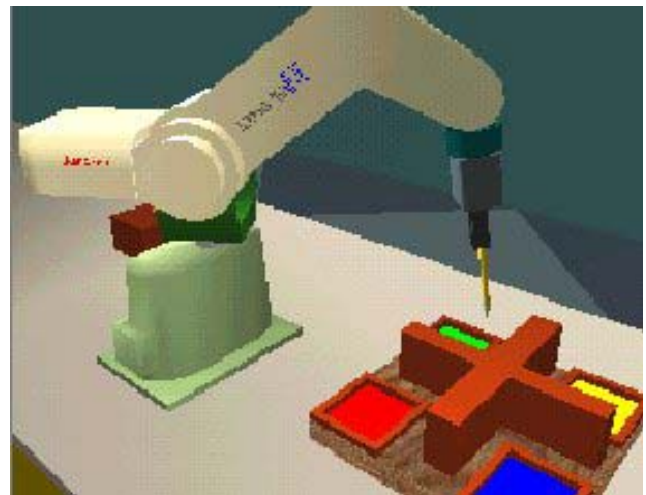


Fig. 2. VR Robot Simulator [4]

For the cases in which it is not necessary to use a real robot, there is the FANUC Robotics' HandlingPRO [5], seen in Figure 3, based on FANUC's robots. This simulator has a 3D environment for creating work cells, which allows both the use of the objects that come with the program and those modeled by the user in CAD tools. With this, it is possible to create environments with more than one robot and with all the objects that stay around the robot in the real environment, allowing the verification of collisions with objects or even other robots during the execution of the simulation. In RoboCell, the editing of programs through the TP is allowed. Moreover, a simulation can be recorded to a video file in the AVI format.

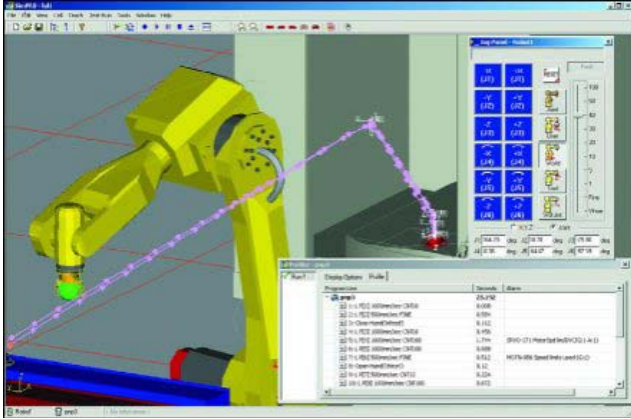


Fig. 3. HandlingPRO Simulator [5]

To try solving the problem of costs and damages related to teaching students with real robots, the work from Hurtado et al. [12] proposes a virtual environment to train on robotics systems, coupled with haptic and 3D visual feedback. The system was tested using a haptic lever device to control a virtual TP and by using direct programming through RAPL-3 commands. According to the authors, after completing the virtual environment tasks and training on their system, all the users were able to successfully operate the real robot.

On Leong et al [13] work, the fusion of the information from a laser range finder and a camera was used to create a system that teaches how to make robot paths in an augmented reality environment for robot laser welding. It includes an image processing step that automatically finds the paths for the welding process and calibrates the distances with a laser range finder, resulting in a positioning accuracy of $\pm 0.5\text{mm}$ and a liftoff distance accuracy of $\pm 1\text{mm}$. The path is then shown to the user in a VE and the user can be taught by interacting with it through a virtual tool in an augmented reality environment.

III. BACKGROUND

For the development of the present work, three schools that teach robot programming were studied: (1) Faculty of Technology (FATEC) SENAI POA, Industrial Automation undergraduate course and Electronics Technician course, (2) CIM Laboratory – Mechatronic Engineering School - PUCRS, and (3) SENAI Ney Damasceno Ferreira, Professional Education School, Mechatronics Technical course, Gravataí-RS. In these schools, concepts such as direct kinematics, inverse kinematics and the definition of reference points for the creation of programs are worked on.

The programming methodology used at the surveyed schools is mainly based on the definition of these reference points to create robot trajectories. According to this methodology, the robot must first be positioned with the TP at the desired point, which is then recorded in a memory position in the robot. In programs, when it is necessary to move the robot to this same position, only the address of the memory in which the point was recorded is used.

Even though programming is taught in the classes after teaching how to use the TP, from the beginning of the use of this device it will always be necessary to use the TP to make programming feasible. This happens because the exercises require the manipulation of parts and the control of peripherals connected to the robot, given that the position to fetch the pieces is discovered with the TP.

As seen, most robotics schools need an efficient and inexpensive graphic simulator. In order to try to solve this problem, a simulator for the Scorbot-ER VII robotic arm was developed, presented in the following sections.

IV. USER INTERFACE ARCHITECTURE

A. Virtual Environment

The VR Robotics VE, seen in Figure 4, was developed with the objective of creating a simulator that could be used in robotics schools, having an adequate visualization and a simple and intuitive form of interaction. Simulators available in schools presented visualization and navigation problems, which lead us to create this simulator using techniques to improve the visualization and navigation through the environment, using several different devices that made it possible to test the best way to visualize and track, allowing the evaluation of which set of devices were best fit for each problem.



Fig. 4. VR Robotics program with its three windows

In VR Robotics, the student can basically find the reference points by moving the robot with the TP until it reaches the desired position, which is then recorded in a memory position of the robot controller and can be used later in the ACL programs.

In order to have a large 3D visualization area with three-dimensional perception, stereoscopy was used, which can be utilized with three different visualization devices.

On the other hand, the issue of locomotion and user orientation within the VE was developed to allow the interaction with a mouse and a keyboard or a joystick. Any of

these devices have access to the main user positioning functions and robot manipulation.

The following sections introduce the VR Robotics interface and the devices used for the tracking and visualization.

1) *VR Robotics Interface*: VR Robotics has a 3D scenario as its main window, along with two other windows, one for the Scenario Editor, in the upper left window and another for the 2D TP in the upper right window. In the following sections the functions of these windows will be explained.

2) *3D Environment*: the 3D environment can be displayed in a window or in full screen. In the environment, it is possible to navigate using a joystick, mouse or keyboard, with locomotion and rotation commands and exchange of predefined views.

3) *Scene Editor*: to allow the creation of new environments, or the editing of existing environments in VR Robotics, a Scenario Editor was created, which allows you to add/remove/position elements such as a robot, a part, a conveyor belt, a rotating table, a bin, an experience table and a CNC lathe, whose 3D representations can be seen in Figure 5, where each object is identified in the figure.

In order to allow a correct simulation, each environment must have only one robot, which can be associated, in this editor, to peripherals such as the conveyor belt and the rotating table, and then triggered through the TP.

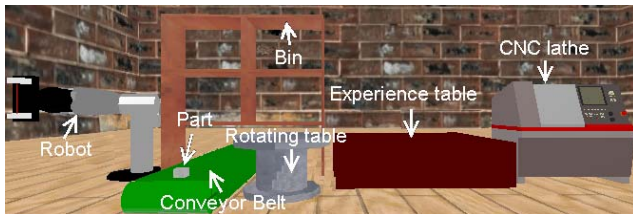


Fig. 5. 3D representation of the Scenario Editor objects

4) *2D Teach Pendant*: this control, seen in Figure 6, was created to allow the robot's manipulation and collection of the reference points. The TP is displayed in a 2D window, and allows the movement of the robot axes, tables and conveyor belts, also allowing a change in the movement speed of these devices. Its image is identical to the actual TP. With this control, it is also possible to record the current position of the robot in a memory position, as well as sending the robot to a previously recorded position. Furthermore, it is possible to change the way the robot moves, which can be by joints or by the X, Y and Z Cartesian axes.

B. Tracking

The tracking of the user was used as a tool to make the interaction with a VE more intuitive. With the tracking of the user's head, one can change its orientation and even its positioning in the VE just by moving his head, without the need to press any key, helping in the visualization of the environment in a way similar to the real one. As for the

tracking of the user's hand, it is possible to move a pointer, which can be used, for example, to select a button on the TP.

To test which of the tracking devices yields the best result, magnetic and optical tracking were used in VR Robotics, as seen in the following sections.

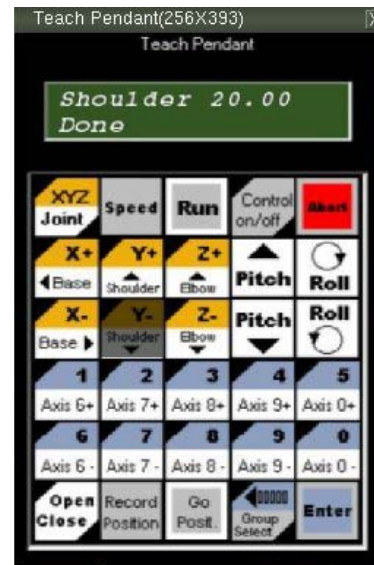


Fig. 6. Teach Pendant

1) *Magnetic Trackers*: the Polhemus Isotrack II tracker has two sensors, one of which can be used to track the user's head and another one to the hand, as seen in Figure 7. Each sensor works with six degrees of freedom, providing position and orientation in the three coordinate axes, but some movements were limited. The translation was disabled on the vertical axis for the user's head, while only the translation on the vertical and horizontal axes were used for his hand, preventing users from performing movements that would cause them to lose orientation in the VE.



Fig. 7. Isotrack II (2 sensors) tracking

2) *Optical Tracker*: the optical tracking is done with the use of a simple camera, which does not use a wide angle lens, as seen in Figure 8, and was implemented using the ARToolkit library [3]. This library uses the optical recognition of printed markers, which must be fixated on the points that should be tracked. In this case, two labels were used, one for the user's head movement and another one for the hand movement. Each sensor operates with six degrees of freedom, but if the sensor leaves the focus of the camera, the tracking information is lost. Therefore, when using this form of tracking, the movement of users outside the camera's field of view must be avoided.

C. Visualization

In order to test which type of visualization device has better results when obtaining the reference points in VR Robotics, three different types of devices were used, all of them supporting the use of stereoscopy, to improve the three-dimensional perception of the environment.

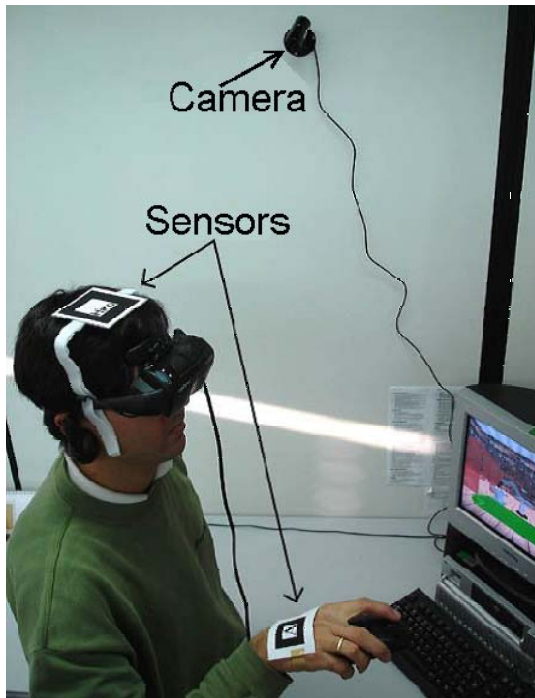


Fig. 8. Tracking with a camera (2 sensors)

1) *Display*: the monitor was used because it is a device that is present in any computer, allowing the use of the simulator without the use of any special equipment. The activation of stereoscopy is done through a Shutter Glasses, as seen in Figure 9.

One advantage of this device is the use of the simulator in an environment where users are already used to working. In addition, the monitor allows the usage of the highest video resolution supported by the computer, which in other devices may not be possible.



Fig. 9. VR Robotics use with a monitor

2) *HMD*: the HMD was used because it is the device that provides a greater user immersion in the VE, once the real world view is blocked. This equipment, seen in Figure 10, allows the visualization to be done directly on it, using or not stereoscopy, since it has a display for each eye. This allows more interaction, considering that the user is free to rotate his head and even perform small movements without losing his view.



Fig. 10. VR Robotics use with an HMD oculus

With the use of the HMD, the visualization of the environment is done in first person. In this way, the user's movements are reflected in the environment as if it were inside the virtual scenario, allowing, depending on the type of tracking, to look at any direction and also getting around.

The advantage of using this configuration is the ease of interaction, being the most intuitive for the user's orientation and locomotion, since the visualization is in the HMD itself and the tracking adapts the user's video according to the movement of his head. The disadvantage of this setting is the image resolution, which may not be too high due to HMD limitations.

3) *Projection Screen*: the projection screen was tested as a display device by allowing the generation of images that can be viewed by several people at the same time, as seen in Figure 11. However, only one user can use a keyboard, mouse or joystick to control the robot. To activate stereoscopy in this device, it is necessary to use two projectors coupled with polarizing lenses, so that the two images are projected on top of each other, on a special screen, that does not undo the polarization when receiving the projection, allowing each eye to receive the image of just one of the projectors with the use of paper glasses with polarized lenses.

D. Implementation Aspects

The VR Robotics simulator was implemented in C++, using the OpenGL library to draw the virtual scenario along with GLUT to create the main window of the program. The SmallVR library was used to manage objects in the virtual landscape, load 3D models and communicate with some of the VR equipment. The ARToolkit [3] library was used for optical tracking, requiring the use of the DirectShow Video Processing Library (DSVL) 2 and DirectX libraries to communicate with the camera. The World Toolkit library (WTK) was used to communicate with the internal tracker. To create the windows displayed in the virtual scenario, a library was developed, since none of the libraries found met the necessary requirements.



Fig. 11. VR Robotics use with a projecting screen. Users to the left and screen to the right

V. EVALUATION

To perform the evaluation tests, a VE that represents a manufacturing cell (Figure 12) was created, in which a user controlling a robotic arm should move a set of parts.

This environment has two rotating tables on the left side, a conveyor belt and an experience table. On the tables, there are five parts, all within the reach of the robot, which is in the center of the VE, on a conveyor belt that was used to position it near any of the pieces. There are also five other rotating tables positioned on the right side of the room. The testing environment was created in a way that required the movement of the robot on the conveyor belt, taking parts at different

heights, depths and distances and releasing them at different distances, causing the robotic arm to be completely stretched in a few moments or closed in others. There was also the concern of placing some parts in places with difficult visualization, so that the user was forced to navigate in the VE to position itself near the part that he wanted to catch.

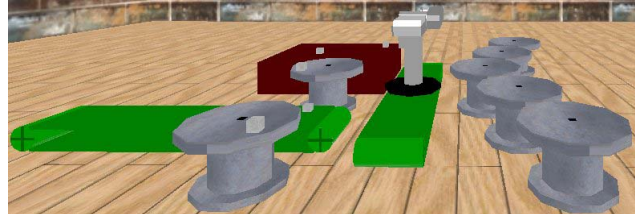


Fig. 12. Testing environment

In order for a task to be performed with the greatest possible precision, users were advised to adopt the following guidelines: close the robot's claw only when he desired to pick up a part; drop the parts as close to the center of the tables as possible (highlighted with a black rectangle); open the robot's claw only when the attempt to pick up a part failed or to drop a part on a table on the right side of the scene; do not pick up a part that has been released, even if the final positioning has not been satisfactory; perform the task as fast as possible, but without interfering with the previous rules.

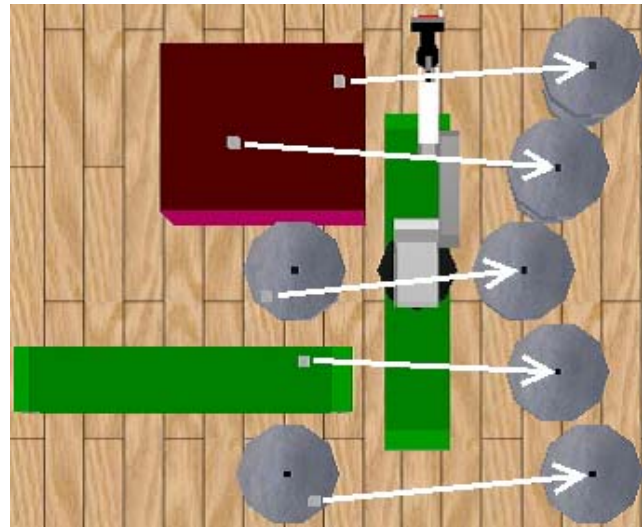


Fig. 13. Upper view of the testing environment in VR Robotics

The task required the creation of trajectories to pick up and drop the parts, which is the same procedure utilized to obtain reference points and is adopted in school to teach robotics, leading the user to perform this activity in an implicit way.

While performing the tests, quantitative and qualitative data was collected through an automatically generated report and through the pre-test and post-test questionnaires, filled out by the users.

In order to evaluate the interaction tools used in VR Robotics, pilot tests were first performed using four different configurations for these tests: a monitor with and without stereoscopy, a HMD with stereoscopy and tracking, and a projection screen with stereoscopy. Each configuration has been tested with at least three users.

In these first tests, there were two types of TP: the 3D one, which was inside the VE (as a virtual object attached to the user) and the 2D one, which was in an application window, not susceptible to modifications according to the user's movements.

In VR versions, users manipulated the 3D TP using a joystick, which controlled the movement of a cursor over the TP, as seen in Figure 14. To select a command, it was first necessary to position the pointer over the desired button and trigger it with a joystick button.

This form of interaction, although simple, showed that users who performed it took 80% more time to complete the task than those who used the fixed TP version, on average. In addition, some users answered in the questionnaires that the joystick caused confusion because it required the use of many buttons. Other users responded that eventually the robot hid part of the 3D TP, impairing the view of it.

In the non-stereoscopic version, users manipulated the 2D TP, as seen in Figure 6, by clicking with the mouse on the buttons without worrying about the selected button.

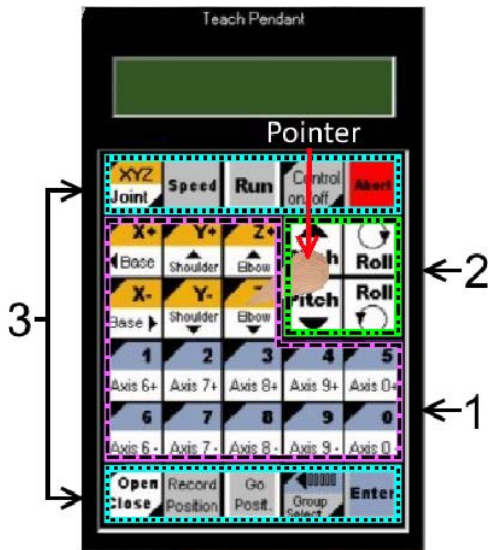


Fig. 14. 3D Teach Pendant used on pilot-tests. The area number 1 refers to the translation options, while the second one refers to the rotation. The area 3 comprises the auxiliary functions

Time was considered an important factor, since it can indicate greater ease of interaction and comfort in the use of the equipment. In addition, it is important to perform the task more quickly in cases where there are fewer computers than the number of students. Considering the results presented and the importance of the shorter execution time, modifications were made to the TP interface, described in the following section.

VI. TEST RESULTS

During the tests we collected data about the time spent to complete the task, the precision while moving parts and number of attempts to grab an object. The following sections present and discuss the results of these metrics.

1) *Time spent*: Figure 15 shows a comparison on the average time consumed to complete the task in every tested configuration.

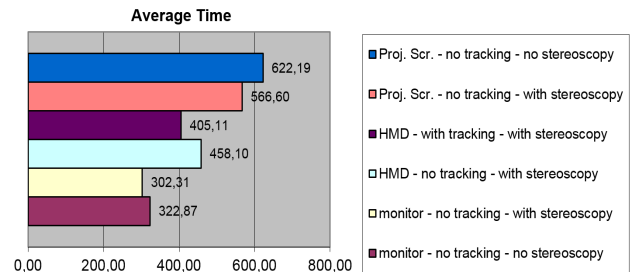


Fig. 15. Average time for task completion

Analyzing the graph it is possible to notice that the use of stereoscopy reduced the time to perform the task using both a monitor and a projection screen. The user's head tracking, used only with the HMD, also decreased the time to perform the task. The projection screen was the device in which the users took more time to perform the task, but this was also the most precise device, as can be seen in the next section.

The average time to pick up the parts showed the same behavior as the average total time of the task, as seen in Figure 16. This indicates that the monitor with stereoscopy is the configuration that enables the task to be performed more quickly.

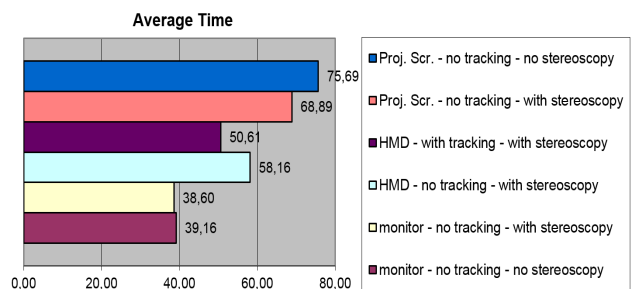


Fig. 16 Average time to pick up parts

2) *Precision while moving parts*: one of the objectives of the construction of VR Robotics was the attempt to reduce the error in the positioning of the robotic arm. As can be seen in Figure 17, the configuration that presented the highest precision was the projection screen with stereoscopy. The HMD with user head tracking was the configuration with the highest error, which may indicate that even if the time of completion of the task with tracking was lower, when the interaction is performed with the mouse this device does not help, since the user is standing in front of the monitor.

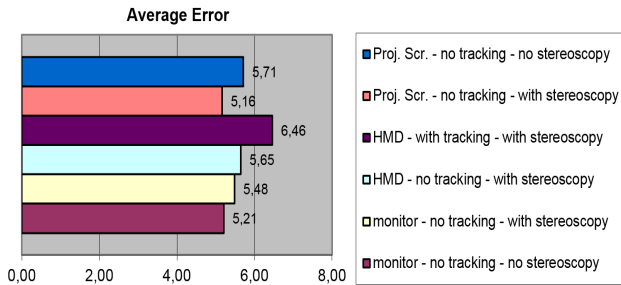


Fig. 17. Average error on the final parts position

Another factor that should be considered when analyzing the quality of the visualization is the distance from the claw to the part when trying to pick up a part. Analyzing Figure 18, we can see that the device that presented the worst result was the projection screen without stereoscopy, but on this same device, when stereoscopy was used, there was practically no error in picking up the parts, indicating that the stereoscopy aided in performing the task, which is confirmed by the users' responses in the post-test questionnaires.

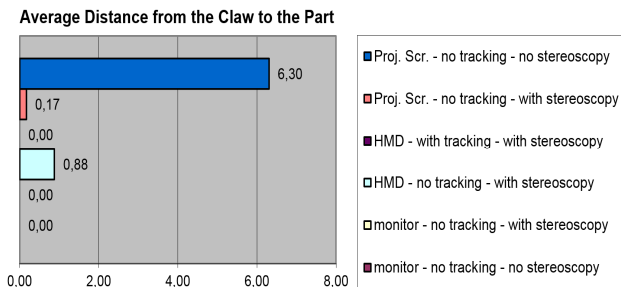


Fig. 18. Average distance from the claw to the part when trying to pick it up

3) *Number of attempts to grab an object*: the number of attempts to pick up the parts may indicate the viewing quality, since wrong attempts indicate that the user could not realize that the claw was not over the part. In Figure 19, we can see that in half of the settings with VR Robotics, only one attempt was made to pick up the parts, but the configuration of the projection screen without stereoscopy and without tracking ended up requiring a greater number of attempts, which indicates that the projection screen must be used with stereoscopy, so that the visualization is adequate.

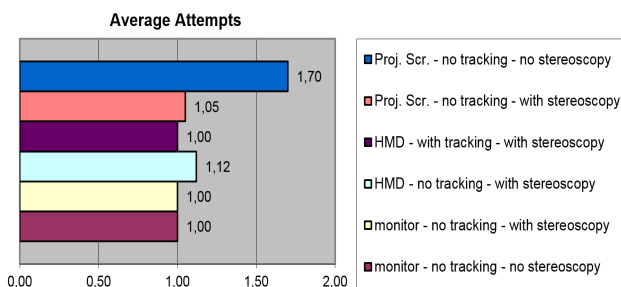


Fig. 19. Graph of the average attempts to catch a part

VII. CONCLUSION

After performing this work, it was verified that the best option to teach robotics in schools, using VR resources, is through the use of a stereoscopic monitor, since this option increased its precision and decreased its total time. In schools that have classes with many students, it may be interesting to use a projection screen with stereoscopy, since only one device can be used by all groups of students. It was found that it is not necessary to use tracking in VR Robotics because of the form of interaction that was used, which mainly uses the mouse as the interaction device. With this, it is also concluded that the HMD does not offer gain in the execution of the task, being more appropriate for other types of virtual environments. In order to obtain more concrete results, it would be necessary to carry out new tests, with a larger group of users and, mainly, using more robotic users, as it was noticed that the performance of some users was not better due to the lack of knowledge about the use of a TP to command a robot. Some users even suggested in the questionnaires that a manual explaining the functions of the TP was presented before the test was performed. As future work, it is suggested to implement an environment for creating and compiling programs in ACL language and, also, the inclusion of a network module that allows multiple users to connect to each other, in the sense that it would enable them to work in a collaborative virtual environment, completing the VR Robotics simulator. It would also be interesting to implement the communication with a real robot, allowing the simulator to send programs and even perform a simulation synchronized with the execution on the real robot. According to the presented results, it is believed that the objective of developing an application that improves the task of obtaining reference points was achieved.

REFERENCES

- [1] ADAMS, Site: <http://www.mscsoftware.com/product/adams>. accessed on 03/12/2015.
- [2] STDR, Site: <http://stdr-simulator-ros-pkg.github.io/>. accessed on 03/15/2015.
- [3] Artoolkit. Site: <http://www.hitl.washington.edu/artoolkit/>. accessed on 07/18/2016.
- [4] Denford. Site: <http://www.denford.co.uk/product.asp?link=119>. accessed on 09/15/2015.
- [5] Fanuc. Site: <http://www.fanurobotics.com>. accessed on 09/15/2015.
- [6] Intelitek. Site: <http://www.intelitek.com>. accessed on 09/15/2015.
- [7] Acl - linguagem de controle avançada - guia de referência e ats - terminal avançado de software. Eshed Robotec Limited, 1992.
- [8] Bell, John T., and H. Scott Fogler. "Vicher: a virtual reality based educational module for chemical reaction engineering." (1996).
- [9] Geissler, Jason, et al. "Virtual reality robotic programming software in the technology classroom." *The technology teacher* 63.6 (2004): 6-9.
- [10] Intelitek Inc. RoboCell User Manual, Nov, '03. Catalog #100346 Rev. D.
- [11] Intelitek Inc. Scorbase User Manual, Nov, '03. Catalog #100342 Rev. E.
- [12] Hurtado, Carlos Vazquez, Alejandro Rojo Valerio, and Luis Ruvalcaba Sanchez. "Virtual reality robotics system for education and training." *CERMA'10*. IEEE, 2010.
- [13] Ng, Chuen Leong, et al. "Intuitive robot tool path teaching using laser and camera in augmented reality environment." *ICARCV'10*. IEEE, 2010.