

3D VIRTUAL ENVIRONMENTS FOR MANUFACTURING AUTOMATION

Henrique Borsatti Lisboa
Luiz Arthur Ramos de Oliveira Santos
Eder Rafael Miyashiro
Karen Juliana Sugawara
Paulo Eigi Miyagi
Fabício Junqueira

University of São Paulo, Brazil

henriquelisboa@outlook.com, luizarthur.santos@gmail.com, edermiyashiro@gmail.com, karen.sug@gmail.com, pemiyagi@usp.br, fabri@usp.br

Abstract. *The increase in the use of automation and robotics in the industries, aiming to increase productivity and lower costs, makes it important the development of tools to mitigate the difficulties found in this new reality and optimize the already existing processes. Using virtual reality concepts, virtual manufacturing software allow 3D graphical visualizations of a product or a manufacturing process, to identify problems in the sequence of commands of industrial equipments by means of simulations without interruptions in the machines of the shop floor to perform tests. The aim of this work is to design and implement a virtual manufacturing environment more immersive, intuitive and interactive, providing an experience as close as possible to the real. For this, it was used a virtual manufacturing environment consisting of an industrial robot (Kuka KR15/2), and the conventional input devices (mouse, keyboard or joystick) were partially eliminated and replaced by non-intrusive sound and motion sensors (Microsoft Kinect). It was used the arm movement of the operator to control the robot and the camera, and voice commands to activate other auxiliary functions. The main features of the software are: detection of motion of the operator arm, motion of the robot joints, inverse kinematics of the robot, collision detection, and generation and interpretation of robot programming code. As a result, this virtual manufacturing environment can be used in engineering courses as well as specialized employee training without the need for supervision of use or disrupt the actual machine on the shop floor or in the teaching unit. The recording motion function may be used in passive learning programming. Here, the programmer moves the virtual robot through the desired trajectory and record the positions. This programming method is useful, and even indispensable in some instances having the advantage of being easy to learn and use. Then, the generated code can be loaded and used in real robot.*

Keywords: *Virtual reality, Manufacture, Industrial robots*

1. INTRODUCTION

The Virtual Reality (VR) is first of all an “advanced user interface” to access applications running on the computer. It has features like visualization of three-dimensional environments, moving these environments in real time, and interact with elements of the environment (Tori; Kirner, 2006). VR provides a new way to represent the man-machine interaction and explore the relationship between these in the real world, as well as interactions between users (Li, Sun, 2009). Using different man senses, such as vision, hearing, touch, etc. for perception, as well as gestures, postures, signs, speech, for interacting with the equipment, a virtual environment can produce effects of immersion.

Among all areas, due to the low cost, the not immersive virtual reality (with the use of conventional displays) is the most widely used. To make the experience more efficient and profitable, new techniques have been sought to make it more intuitive and interactive (easy interaction). Thus, the trend of the VR industry is to eliminate or restrict the use of conventional input devices such as mouse, keyboard and joystick (Gutierrez; Vexo; Thalmann, 2008).

The use of VR has expanded gradually in various areas of knowledge, ranging from entertainment to real-time applications. This is due to many factors, such as research advances, people's access to computational resources thanks to the cheapness of the equipment, components and interfaces (Curzel; Hounsell; Leal, 2007).

On the engineering, according Mujber; Szécsi; Hashmi (2004), in recent years advances in VR provided a boost to its application in different activities, such as product design, modeling, control shop floor simulation processes, production planning, training, testing and verification. It helps to reduce product development time and increases the quality of production (Kim et al., 2006). It has great potential to solve problems in industrial applications, still in the design phase, thus avoiding unwanted expenses during deployment (Mujber; Szécsi; Hashmi, 2004). Designers can control the virtual machines within the virtual factory and evaluate digital product in advance. Thus, we seek to eliminate unexpected errors as well as reduce the costs and process times. It also enables the creation of realistic prototypes, much cheaper and quicker to elaborate than physical prototypes, which can be applied during the development of a production process. In the automotive and shipbuilding industries, the combined use of VR systems with virtual manufacturing has promoted competitiveness increase. According Kadir et al. (2011), this environment can

be studied best security conditions. Data management of manufacturing and life cycle of the product can be easily documented more systematically.

Its application in the manufacturing area is recent. It is possible to check and verify various manufacturing processes in a virtual way, reducing the cost and time required when compared to technologies traditionally used (Banerjee; Zetu, 2001).

Due to the benefits of VR for the manufacturing area, numerous virtual systems have been developed, among them (Kadir et al., 2011): Virtual Manufacturing (VMfg), Virtual Machine Tool (VMT), Virtual Machining (VMach), Virtual Assembly (VA), Virtual Tooling (VTO), and Virtual Prototyping (VP).

In Brazil, few companies use the VR in the study and development of products. Among them, we can mention: Petrobrás¹, Embraer³, Vale⁴, GM⁵, e Volkswagen⁶. In common they are large companies with ample resources to heavily invest in this type of technology.

In addition, virtual manufacturing systems may also be used in learning environments (Cheng, Wang, 2011). Potkonjak et al. (2010) reported the growth of distance education and e-learning in the last two decades due to the growth of the Internet and improvements in communication technologies. In particular, they highlight the courses in electrical and mechanical engineering, which require laboratory activities carried out with the aid of physical (real) equipments. Therefore, to allow a full experience of e-learning in this area of knowledge, Potkonjak, et al. highlight the VR as a useful tool in creating virtual labs that allow distance learning, where mathematical models can be used to simulate the dynamic behavior of the main real entities. In the specific case of a robotics lab, an entity that uses both concepts of mechanical and electronics, and the robot itself, it is necessary to set the environment where it is inserted. In the virtual environment, the robot is replaced by a graphic model and a mathematical model that simulates its dynamics. For the environment, this could be more or less complex, elements may have only static or have elements that are associated with the movement of the robot. The environment can also involve other robots, computer numerical control (CNC) machines, transporters (AGVs – automated guided vehicles), etc. Potkonjak et al. also indicate the advantages of using virtual learning environments. In particular, in the virtual robot example, all parameters (physical or load applied to it) can be changed so that different tests can be performed before, for example, students perform an improper command in a real robot which can cause an accident or damage.

Another trend that has been explored in many areas such as medicine, education, engineering, among others, is the use of games, or “serious games”. Serious games have been explored for their positive effect on learning, their leisure and enjoyment factor (Kosmadoudi et al., 2013). Serious games encourage active and critical learning through game environments where users enjoy the process of pursuing challenging tasks, achieving goals by making decisions, formulating strategies, construct knowledge, try different alternatives and succeed or fail without worrying about the consequences in real life. Moreover these environments involve some aspect of competition where the user tries to influence the final outcome of the game process and at the same time feels attracted to the outcome. Such behavior can be described as intrinsic motivation, which has been recognized to be one of the mechanisms of exploration and curiosity.

The aim of this work is to design and implement a virtual manufacturing environment more immersive, intuitive, interactive, and low cost, providing an experience as close as possible to the real.

This work is organized as follows: section 2 presents materials and methods adopted for this work; section 3 presents a application example; and in section 4 are presented the conclusions.

2. MATERIALS AND METHODS

2.1 Hardware

For the development and execution of the virtual environment, it was used a computer containing: 1 Motherboard Intel DX58OG; 1 Intel Core i7 950 3.06GHz, 8Gb DDR3 - PC1333 memory, 1 HD SATA II 1TB 7200RPM, 1 Video Card GeForce GTX 580 1.5 Gb - DDR3 - PCI-Xpress; 1 Nvidia 3D Vision Glasses Kit with data transmitter; 1 BenQ XL2410T LED Monitor Full HD 23.6" 3D Ready; Keyboard; Optical Mouse; and 1 Microsoft Kinect. This configuration has cost approximately R\$7.000,00, which makes it accessible to small and medium enterprises.

Microsoft Kinect is becoming an important 3D sensor. This system combine structure light with two classic computer vision techniques: depth of focus and depth of stereo. It uses infrared laser lighting with a speckle pattern. The depth map is constructed by analyzing a speckle pattern of infrared laser light. It is receiving a lot of attention thanks to a rapid human pose recognition system developed on top of 3D measurement. The low cost, reliability and speed of the measurement promises to make Kinect one of the primary 3D measuring devices for indoor robotics and mapping, 3D

¹ revistagalileu.globo.com/Galileu/0,6993,ECT803835-1938-1,00.html

² www.revistapetrobras.com.br/revistapetrobras/files/pdfs/Materia%20de%20Capa.pdf

³ www.embraer.com/pt-BR/ConhecaEmbraer/QualidadeTecnologia/Paginas/Home.aspx

⁴ www.usinagem-brasil.com.br/3531-poli-usp-desenvolve-simulador-de-trem-para-a-vale/

⁵ computerworld.uol.com.br/carreira/2010/09/24/pioneirismo-na-veia

⁶ www.motorclube.com.br/materias/volkswagen-inaugura-centro-de-realidade-virtual.aspx

scene reconstruction and object recognition (Gonzalez-Jorge et al., 2013). Yet according to Gonzalez-Jorge et al. (2013), it shows a decrease of precision with an increase of length from 1m to 2m. Precision changes between 2mm and 6mm for 1m range and between 4mm and 12mm for 2m range.

The Nvidia 3D Vision is a high definition stereoscopic vision equipment. It generates images from two different perspectives. It has an infrared emitter that transmits data to a wireless glasses. There is also the need for a monitor with a screen refresh rate of at least 120Hz. Stereoscopy is obtained when the 3D model data are sent to the video card driver. It renders the scene twice, with a small horizontal shift of the camera, forming two images, one for each eye. The monitor should then display the images for the left and right eyes in different frames. rates. The infrared emitter then sends data to the glasses which closes alternately each of the lenses. Thus the final frequency for each eye is half the refresh rate of the monitor, i.e. 60Hz (Gateau, 2009).

2.2 Softwares

The software used for this project were: Windows 7 Professional 64bit, Visual Studio 2010, XNA Game Studio 4; Kinect SDK 1.5, and Blender. With the exception of Windows, the other can be obtained free of charge.

Visual Studio 2010 was chosen to allow the integration of other technologies used in this project, i.e., the XNA Game Studio 4 and Kinect SDK 1.5. Moreover, it allows a more natural integration with industrial automation equipment via OPC (OLE for Process Control). The OPC specification was based on the OLE (Object Linking and Embedding), COM (Component Object Model), and DCOM (Distributed COM) technologies developed by Microsoft for the Microsoft Windows operating system family. The specification defined a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability. The aim was to reduce the amount of duplicated effort required from hardware manufacturers and their software partners, and from the SCADA (Supervisory Control And Data Acquisition) and other HMI (Human-Machine Interface) producers in order to interface the two^{7,8}.

The XNA was adopted for managing the 3D environment. According Pathak and Joshi (2009), XNA is an excellent framework for developing 3D applications. It is based on Microsoft .NET Framework and supports coding in C# (C-Sharp) and Visual Studio 2010 development environment.

According Paneels et al. (2010), there are several tools for creating 3D models in the context of VR such as Blender⁹, 3D Studio Max¹⁰, Maya¹¹ and Rhino¹². Among these, Blender is distributed free of charge (Amici et al., 2010).

2.3 Method

For the design and implementation of virtual manufacturing environment, using the software mentioned above, it is suggested the following steps:

- a) Define the elements that should make up the virtual manufacturing environment.
- b) 3D graphical modeling of each element, separately, using Blender.
- c) Export the elements in FBX format^{13 14}, which is recognized by XNA.
- d) Import the graphics for the XNA project. Each element must be rotated 90° around the X and 180° around Z axis because Blender and XNA adopt different coordinate systems.
- e) Program the kinematics and dynamics of each element including bounding boxes for collision detection, as well as other methods and functions specific to each element.
- f) Associate the input devices, such as keyboard, mouse, joystick, and even Kinect (gestures and/or voice commands), to each element.

3. APPLICATION EXAMPLE

A robot Kuka KR15/2 was adopted as application example. It is used in robotics classes in the mechatronics engineering course at Escola Politécnica of USP. It was designed and implemented a virtual model of the learning environment, including this robot.

⁷ www.opcfoundation.org/

⁸ en.wikipedia.org/wiki/OLE_for_process_control

⁹ www.blender.org

¹⁰ www.autodesk.com/3dsmax

¹¹ www.autodesk.com/maya

¹² www.rhino3d.com

¹³ www.autodesk.com.br/adsk/servlet/index?siteID=1003425&id=11158213

¹⁴ en.wikipedia.org/wiki/FBX

The movements of the right wrist of the operator are used to interact with the environment (move the end-effector of the robot and the camera). Voice commands are used secondarily, so as to assist in the use of implemented functions such as motion detection and recording.

The main features of the software are: detection of motion of the operator arm, motion of the robot joints and inverse kinematics of the robot, generation and interpretation of robot programming code, among others.

The following are some of the main aspects of graphical modeling and implementation of the virtual manufacturing environment chosen.

3.1 3D graphical modeling using Blender

The modeling in Blender can and should be done in two ways: in “object mode” and “edit mode”. In the “object mode” (Fig. 1.a), to ensure compatibility with XNA, the elements must be positioned at the origin of the coordinate system, i.e., $(x, y, z) = (0, 0, 0)$, no rotations must be made, and the scale 1.0 must be used. In the “edit mode” (Fig. 1.b), vertices, edges, faces or even the whole element can be deformed (translated, rotated or have their dimensions altered) freely.

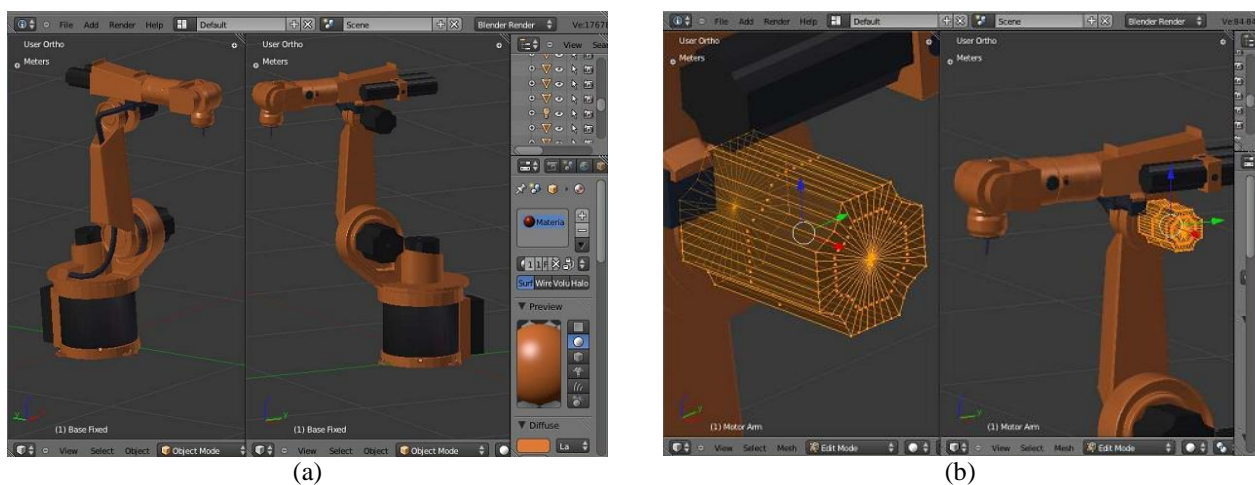


Figure 1. (a) Modeling in (a) “object mode” and (b) “edit mode” on Blender.

To allow movement of elements in XNA, it is necessary to create an armature involving each model. Then every moving part is associated with a bone (Fig. 2a). The armature is similar to a skeleton, formed by bones interconnected that, when moving, move and deform the parts associated with them (Gumster, 2009).

Then each model must be exported in FBX format, as shown in Figure 2.b.

3.2 Detection of motion of the operator arm

For the reconstruction of the human skeleton, there are studies that make the reconstruction using custom algorithms which can delimit the resultant errors in different dynamic situations (Straka et al., 2011). In general, for tracking applications, data must be acquired between 1 and 3 meters from the sensor.

We opted for drawing, in real time, a circle at the position where the joint of the right wrist is located, and write in the screen the depth detected in order to determine the accurate detection of the motion. The depth of the joints is given in meters. The maximum and minimum values of x and y depend on the distance from user to camera. They can vary approximately between -0.8 and 0.8, close to the camera (1m), and -1.2 to 1.2, far from it (4m) (Fig. 3). The origin of the coordinate system is approximately the center of the image obtained. The value of x increases to the right of the user (left of Kinect) and y increases upward. The two planes shown in exaggerated form, exemplify the variation of the maximum and minimum values in relation to the distance from the camera (Fig. 4). The plan is considered by Kinect as a reference, with which it compares the pattern obtained by the infrared camera with a stored reference and whose differences determines the joints depth.

This knowledge was used to convert data from the joints for the coordinates of the image, in pixels. The results are shown in Figure 3. Moreover, due to fluctuations in the position of the joints (provided by Kinect) observed throughout the project and also in (Gonzalez-Jorge et al., 2013), a hysteresis filter was implement.

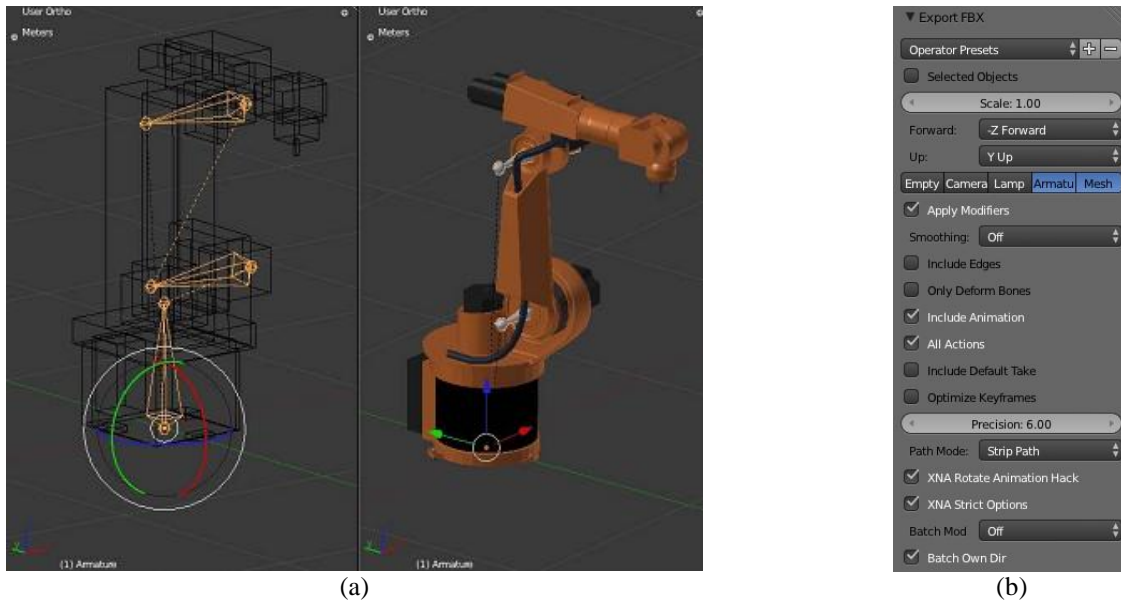


Figure 2. (A) Armour and bones on Blender; and (b) menu for exporting the model using the FBX format to be used in XNA.

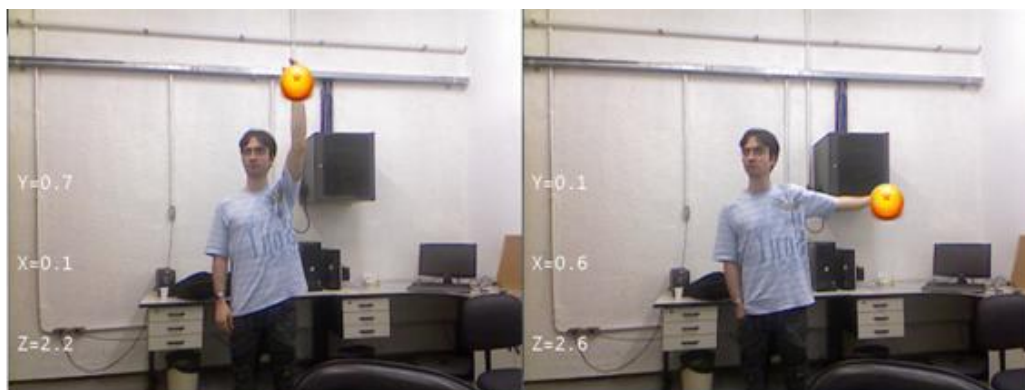


Figure 3. Result obtained in the detection of the right hand joint (the image is mirrored).

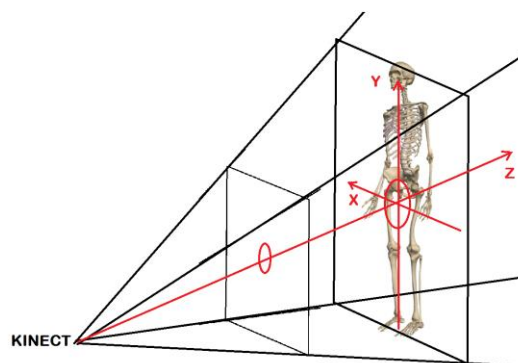


Figure 4. Kinect coordinate system.

3.3 Motion of the robot joints and inverse kinematics of the robot

The equations describing the kinematics of the robot were obtained based on Siciliano et al. (2010) and will be omitted because it is not the focus of the work. However, it is emphasized that the calculation of inverse kinematics, based on the position of the operator right wrist joint needs special attention when it is positioned off the working volume of the robot (Fig. 5.a). This consideration is important to avoid uniqueness edge. To avoid this problem, for each displacement increment, it must be checked whether the new position is contained in the working volume, and then calculate the inverse kinematics. By positioning the right wrist joint out of the working volume, the robot moves as close to it (Fig. 5.b).

Another problem occurs with the robot already in the limit of the working volume (Fig. 5.c) and the operator moves the wrist between two positions outside this volume. In this situation, the robot movement would be locked since there is no path to be traversed. To solve this problem, we have implemented a function which, from a vector located in the center of the coordinate system, if the desired position is outside the working volume, we get the desired point nearest to the intersection of this line with the circle that defines the working volume (Fig. 5.d).

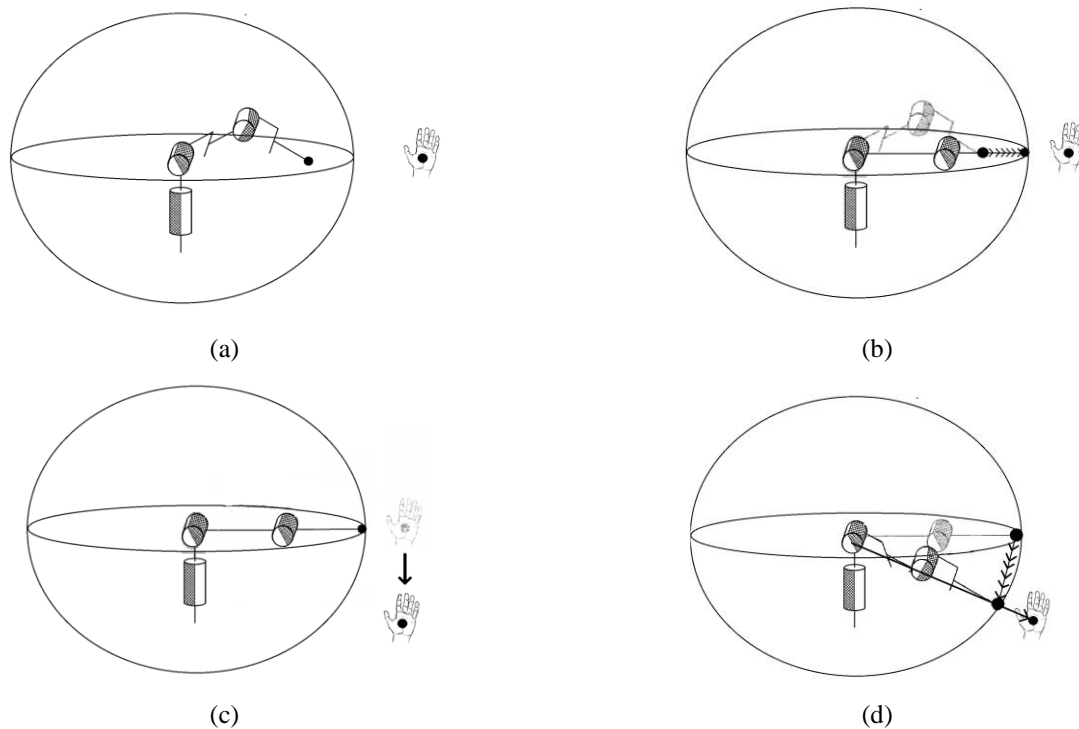


Figure 5. (a) desired position outside the working volume; (b) robot moving until the limit of the working volume; (c) motion starting in the limit of the working volume; and (d) linear motion of the robot limited by the working volume.

3.4 Generation and interpretation of robot programming code

For future application of this project, it was decided to store the data generated by the virtual robot movement in order to control the real robot by body movements. For this, we used the programming language KRL (Kuka Roboter, 2010), created by the Kuka manufacturer. Using the ANTLR¹⁵ (Another Tool for Language Recognition) and a grammar¹⁶ based on the KRL language, it was generated two classes, the KRLParser and the KRLLexer, to assist in the interpretation of the code KRL.

Three commands to the robot end-effector displacement were implemented: LIN, PTP and LIN CONT. LIN is the command for linear motion where the robot moves continuously through the points with small increments, which determine a straight path between the start and end points. PTP (Point to Point) is the command in which the intermediate positions are not entered in the schedule of the trajectory that does not actually exist. All motors start at the maximum (limited by the programmed speed) so that the movement between two points is the fastest possible in each axis. LIN CONT is the continuous movement in which the target point is not exact but approximate. Thus the robot does not need to be slowed down to a stop at the destination point, causing less wear of the real robot and a lower cycle time.

3.5 Results

This section presents some results obtained with the virtual manufacturing environment implemented:

- a) Stereoscopic vision (Fig. 6).
- b) Use of traditional interfaces (keyboard and mouse) to move the robot (Fig. 7). Also in this figure, there is the possibility to move each joint of the robot separately (forward kinematic) or impose one position to the end-effector (inverse kinematic).

¹⁵ www.antlr3.org

¹⁶ www.antlr3.org/grammar/1317637876413/krl.g

- c) Robot motion based on a KRL code file (Fig. 8).
d) Interaction with the robot through the Kinect sensor (Fig. 9).

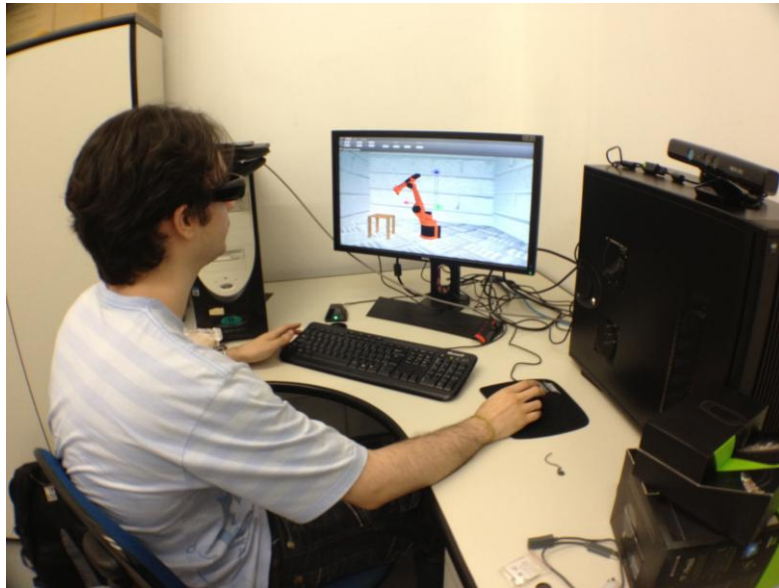


Figure 6. 3D view of the virtual manufacturing environment.

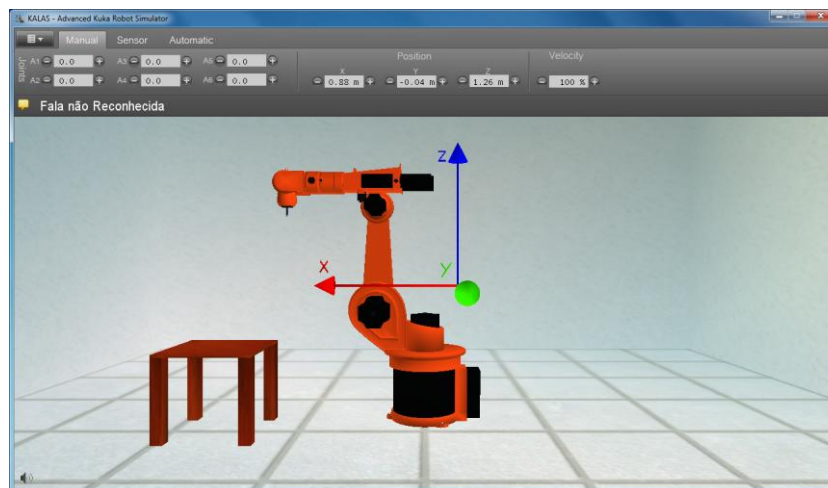


Figure 7. Motion of the robot through the activation of its joints (forward kinematic) or by the position of the robot end-effector (inverse kinematic).

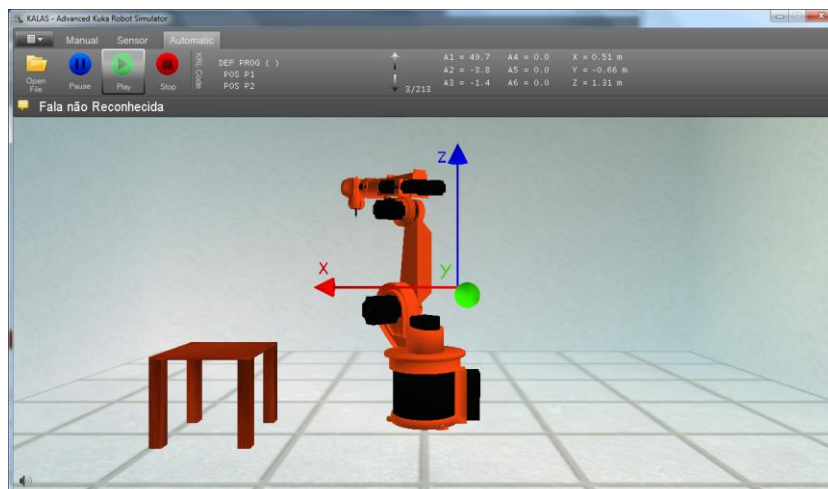


Figure 8. Robot motion based on a KRL code file.

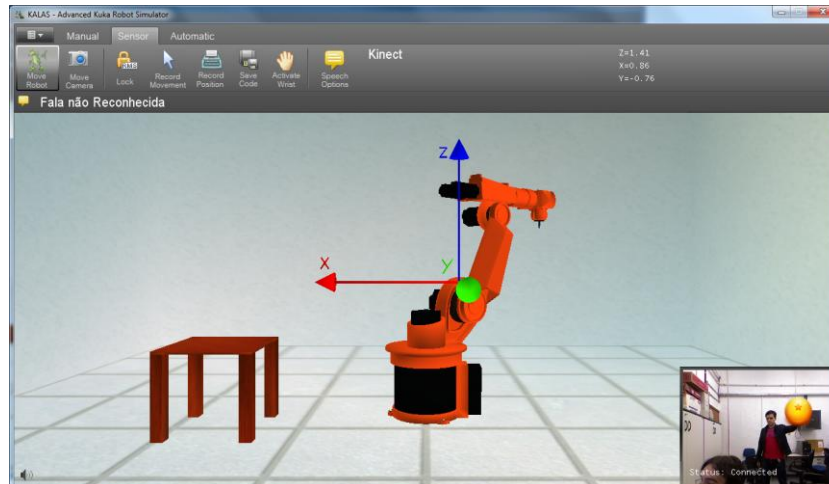


Figure 9. Interaction with the robot through the Kinect sensor.

4. CONCLUSIONS

The aim of this project was to build a virtual manufacturing environment more immersive, interactive, intuitive, and low cost, with applications in the areas of education and training. For this purpose it was designed a simulation software of a virtual manufacturing environment where conventional input devices were partially replaced. In this sense the project was successful, including the use of motion sensors (Kinect) and stereoscopy technology.

As a result, this virtual manufacturing environment can be used in engineering courses as well as specialized employee training without the need for supervision of use or disrupt the actual machine on the shop floor or in the teaching unit. The recording motion function may be used in passive learning programming. Here, the programmer moves the virtual robot through desired trajectory and record the positions. This programming method is useful, and even indispensable in some instances having the advantage of being easy to learn and use. Then the generated code can be loaded and used in real robot.

More information about this project can be obtained from (Santos, Lisbon, 2012) and (Miyashiro, Sugawara, 2011).

The next steps are to use OPC to promote interaction between virtual models and real entities, allowing models to be used as SCADA (Supervisory Control And Data Acquisition) systems for monitoring and/or remote operation; promote the interaction of various users through a virtual environment; and the modeling of other equipment.

5. ACKNOWLEDGEMENTS

The authors would like to thank the Brazilian governmental agencies CAPES, CNPq and FAPESP for their partial support.

6. REFERENCES

- Amici, S. de; Sanna, A.; Lamberti, F.; Pralio, B., 2010. "A Wii remote-based infrared-optical tracking system". *Entertainment Computing*, Vol. 1, Issues 3-4, pp. 119-124.
- Banerjee, P.; Zetu, D., 2001. *Virtual manufacturing*. New York: John Wiley & Sons.
- Cheng, Y., Wang, S., 2011. "Applying a 3D virtual learning environment to facilitate student's application ability - The case of marketing". In. *Computers in Human Behavior*, Vol. 27, No. 1, pp. 576-584.
- Curzel, J. L.; Hounsell, S.; Leal, A. B., 2007. "Uso da realidade virtual para ensino de automação da manufatura". *International Conference on Engineering and Computer Education*. pp. 773-777.
- Gateau, S. 3D Vision Technology Develop, Design, Play in 3D Stereo, 2009. Disponível em <http://developer.download.nvidia.com/presentations/2009/SIGGRAPH/3DVision_Develop_Design_Play_in_3D_Stereo.pdf>
- Gonzalez-Jorge, H.; Riveiro, B.; Vazquez-Fernandez, E.; Martínez-Sánchez, J.; Arias, P., 2013. "Metrological evaluation of Microsoft Kinect and Asus Xtion sensors". In. *Measurement*, Vol. 46, pp. 1800-1806.
- Gumster, J. V., 2009. *Blender for Dummies*. Wiley Publishing, Inc.: United States of America, 412 p.
- Gutiérrez, M. A.; Vexo, F.; Thalmann, D., 2008. *Stepping into Virtual Reality*. London, Springer, 2008.

- Kadir, A. A.; Xu, X.; Hammerle, E., 2011. "Virtual Machine Tools and Virtual Machining – A Technological Review". In. *Robotics and Computer-Integrated Manufacturing*, Vol. 27, No. 3, pp. 494-508.
- Kim, Y.-S.; Yang, J.; Han, S., 2006. "A multichannel visualization module for virtual manufacturing". *Computers in Industry*, Vol. 57, N. 7, pp. 653-662.
- Kosmadoudi, Z.; Lim, T.; Ritchie, J.; Louchart, S.; Liu, Y.; Sung, R., 2013. "Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements". In. *Computer-Aided Design*, Vol. 45, pp. 777-795.
- Kuka Roboter, 2010. *Kuka System Software 5.5 - Operating and Programming Instructions for System Integrators*. Disponível em <http://sites.poli.usp.br/d/PMR2560/Manual%20KUKA.pdf>
- Li, S.; Sun, J., 2009. "Application of Virtual Reality Technology in the Field of Sport". *2009 First International Workshop on Education Technology and Computer Science*, pp. 455-458.
- Miyashiro, E. R; Sugawara, K. J., 2011. *Simulação de um robô virtual utilizando manufatura virtual*. Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.
- Mujber, T. S.; Szecsi, T.; Hashmi, M. S. J., 2004. "Virtual reality applications in manufacturing process simulation". *Journal of Materials Processing Technology*, Vol. 155-156, pp. 1834-1838.
- Paneels, S.A.; Roberts, J.C.; Rodgers, P.J., 2010. "HITPROTO: a tool for the rapid prototyping of haptic interactions for haptic data visualization". *Haptics Symposium*, IEEE, pp. 261-268.
- Pathak, R. and Joshi, S., 2009. "Implementation of Virtual Reality and Agents Model Based Nano Domain Simulation an HPC Setup". *First Asian Himalayas International Conference on Internet AH-ICI2009*, 5p.
- Potkonjak, V.; Vukobratovi, M.; Jovanovi, K.; Medenica, M., 2010. "Virtual Mechatronic/Robotic laboratory - A step further in distance learning". *Computers & Education*, Vol. 55, pp. 465-475.
- Santos, L. A. R. O.; Lisboa, H. B., 2012. *Movimentação de robô virtual por meio de sensores de som e movimento*. Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.
- Siciliano B.; Sciavicco L.; Villani L.; Oriolo G., 2010. *Robotics: Modelling, Planning and Control*. London: Springer-Verlang, 632p.
- Straka, M.; Hauswiesner, S.; Ruther, M.; Bischof, H., 2011. "Skeletal Graph Based Human Pose Estimation in Real-Time". In *Proceedings of the British Machine Vision Conference*, BMVA Press, pp. 69.1-69.12.
- Tori, R.; Kirner, C., 2006. "Fundamentos de Realidade Virtual". *VIII Symposium on Virtual Reality. Fundamentos e Tecnologia de Realidade Virtual e Aumentada*. pp. 2-21

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.