A LOW COST APPROACH FOR TEACHING ROBOTICS BASED ON VIRTUAL REALITY MULTI-PROJECTION SYSTEM

Heleno Murilo Campeão Vale, heleno@sc.usp.br
David Custódio de Sena, davidc@sc.usp.br
Claiton de Oliveira, claiton@sc.usp.br
Arthur José Vieira Porto, ajvporto@sc.usp.br
School of Engineering of Sao Carlos (EESC) – University of Sao Paulo (USP)
Av. Trabalhador São-carlense, 400, Pq. Arnold Schimidt, São Carlos - SP, CEP 13566-590, Brazil.

Abstract. *Robotics is an important area in science and technology and traditionally the teaching of robotics requires very expensive equipment that may be damaged if it is used improperly. In order to overcome these problems, computer systems can be designed based on virtual reality that is known as one of the most advanced form of interaction between man and computer. When virtual reality is used with multi-screen projection systems, it provides a significant improvement to the perception of realism and accuracy of a synthetic world. Therefore, it can be used to minimize the costs in teaching robotics and the risk of damage as well. This work proposes an educational system intended to provide an authentic context and to generate high levels of motivation in learning robotics. The kinematics simulation including forward kinematics and inverse kinematics was implemented on the virtual robots. The construction of all virtual objects and environments was based on Extensible 3D (X3D) technology, which is the successor specification to the VRML97 format. The utilization of multi-projection systems is addressed in detail throughout this paper.*

Keywords*: robotics education, multi-projection system, x3d*

1. INTRODUCTION

The needs of the actual economy and industry have made robotics an important subject lectured at the universities. Practical classes are necessary in order to teach robotics, but the equipment is normally very expensive and voluminous and the misuse of the same can damage the equipment or even harm the user.

Virtual Reality (VR) is one of the most advanced existing interfaces between man and machine. In the last years it has been disseminated among the researchers of several areas due to the cost of the devices and the increasing capacity of the personal computers. Some of the benefits of the use of VR are that the tests can be performed in a safer way than the tests done on equipment and the high level of realism provides reliable results. In order to improve the level of realism of these simulations, multi-projections devices like CAVE can be used.

In this work a Virtual Environment (VE), applied in a multi-projection system, was created to aid the teaching of robotics. As an educational system, it was created to motivate the students and to teach, in an interactive way, the introduction to the technology of robotics in general. However, no specific robot was simulated.

The Second Chapter of this work introduces the principles of Forward and Inverse Kinematics.

The Third Chapter presents the tools used to develop the VR and to design tridimensional objects as well as the specific tools to develop multi-projection systems.

Chapter Four talks about multi-projection devices for VR and explains the structure of the device used in the Multi-User Immersive Visualization Laboratory of the School of Engineering of Sao Carlos - University of São Paulo.

The geometrical model of the used robot and its immersive virtual environment are described in the Fifth Chapter as well as the basic characteristics of VRML and X3D file formats.

Chapter Six presents the use of the system for teaching robotics.

Finally the conclusions and the suggestions for future works are presented.

2. FORWARD AND INVERSE KINEMATICS

Kinematics is the science of motion that treats motion without regards to the forces which cause it. Within the science of Kinematics, one studies positions, velocity, acceleration and all higher order derivatives of the position variables (with respect to time of any other variables (s)). Hence, the study of the kinematics refers to all the geometrical and time-based properties of the motion (Craig 2005).

2.1. Forward kinematics

Forward kinematics is the static geometrical problem of computing the position and orientation of the end-effector (hand) of the manipulator. Specifically, given a set of joint angles, the forward kinematics problem is to compute the position and orientation of the tool frame relative to the base frame.

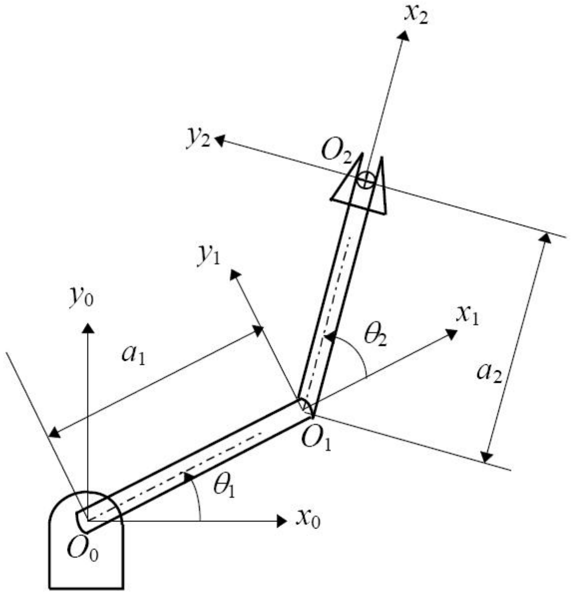


Figure 1 – An example of a robot with two revolute joints

A commonly used convention for selecting frames of reference in robotic applications is the Denavit-Hartenberg, or D-H convention. In this convention, each homogeneous transformation *Ai* is represented as a product of four basic transformations. The D-H parameters for the example of Figure 1 are shown in Table 1:

Table 1. Denavit-Hartenberg parameters for a two revolute joints robot.

Joint	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2

With the D-H parameters of Tab. 1 it is possible to define the transformation matrixes for the two revolute joints robot as shown in Eq. (1):

$$A_0^1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } A_1^2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

In the matrixes of the Eq. (1), S_1 and C_1 refer respectively to the Sine and Cosine of θ_1. S_2 and C_2 refer to the Sine and Cosine of θ_2. The product of these two matrixes gives the transformation of the base to the end-effector (hand) of the two revolute joints robot as shown in Eq. (2):

$$A_0^2 = A_0^1 A_1^2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_1 + a_2 C_{12} \\ S_{12} & C_{12} & 0 & a_1 S_1 + a_2 S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

In the matrix Eq.(2), S_{12} and C_{12} refer respectively to the Sine and Cosine of $\theta_1 + \theta_2$. The first two elements of the fourth column of Eq. (2) refer to *X* and *Y* of the point O_2 (hand). They are the coordinates of the end-effector relatively to the base of the robot: O_0 $(X_0\ Y_0\ Z_0)$. The orientation of the end-effector is given by a rotation of $\theta_1 + \theta_2$ degrees about Z_0.

2.2. Inverse kinematics

The problem of inverse kinematics is posed as follows: given the position and orientation of the end-effector of the manipulator, calculate all possible sets of joint angles which could be used to attain this given position and orientation. This is a fundamental problem in the practical use of manipulators. The inverse kinematics problem is not as simple as the forward one. Because the kinematics equations are nonlinear, their solution is not always easy or even possible in a closed form. Also, the questions of existence of a solution, and of multiple solutions, arise. Solving this problem is a least requirement for most robot control systems.

3. VIRTUAL REALITY TOOLS

Most VR applications have several specific features in common. They use input devices for the user to interact with the system and they use display devices for the user to see the virtual world. Images and three-dimensional (3D) model files often need to be loaded into the VR system. During the development of three-dimensional graphics systems, the developer needs to work only with high level constructors to create, manipulate and visualize geometrical objects, with the visualization details being automatically handled by a scene graph that uses software libraries such as OpenGL and DirectX to access the graphics hardware.

A scene graph is composed by hierarchy classes used as an interface for 3D programming. The 3D program creates graphical object instances that are placed in a scene graph that can also have lights, sound and other elements that allow the creation of virtual environments. A scene graph is composed by a group of 3D objects combined in a tree structure that determines the contents of the virtual universe and how it should be visualized. Among the several available scene graph tools we considered three options: Java3D, OpenSceneGraph and OpenSG.

Java3D is the Java-based portable scene graph, object-oriented and platform independent. One of the main features of this API is the easy exportation through the Internet via Java applets embedded in HTML pages. The developers would like to position this API at the highest programming level without loosing any performance due to the increasing distance to the hardware layer. As any Java application (run on a virtual machine) is slower than a C++ equivalent binary, it is hard to achieve such performance.

The OpenSceneGraph project provides an open-source scene-graph. OpenSceneGraph uses multi-threading to be responsive while providing high rendering performance. As an extension to the thread safety, clustering support is also included. In the clustering configuration, remote systems are given synchronized copies of the scene for culling and drawing. The remote systems are treated as pipe-lines. OpenSceneGraph provides read-only access to the graphics database and is classified as a distributed scene definition system.

OpenSG is a portable scene graph system to create real-time graphics programs, e.g. for virtual reality applications. Like OpenSceneGraph, OpenSG sup-ports both multithreading and clustering. OpenSG also allows thread safe write access to the scene graph. This feature, however, does not currently apply to the clustering support. The cluster configuration retains the traditional topology where only the master node has write access and the other nodes act as graphics pipelines. OpenSG is classified as a distributed scene definition data.

There is a solution called Blender for 3D objects to be modeled without programming them. This tool is an open source 3D modeling program which allows one to create textured meshes for still rendering or animation.

To develop virtual reality applications and run them in a VR system, a VR toolkit is needed. One of the most known VR toolkits is the CAVELib. This tool is an API designed for creating immersive, interactive 3D applications. The software is completely configurable at run-time and can support up to thirty-two arbitrarily placed projection screens. It is composed by a set of libraries used as a base for developing virtual reality applications. These libraries manage input devices, inter-process communication, and display parameters.

An open-source VR toolkit solution is VR Juggler, which is a C++ based, cross platform set of virtual reality (VR) libraries that provides an abstraction layer between the hardware of a VR system and the virtual world created in a software. VR Juggler pro-vides a platform for virtual reality application development which allows a user to run an application on almost any VR system. VR Juggler is scalable from simple desktop systems like PCs to complex multi-screen systems running on high-end work stations and super computers. The current development environment supports many VR configurations including desktop VR, HMD, CAVE™-like devices, and Powerwall™-like devices.

4. VIRTUAL REALITY MULTI-PROJECTION DEVICES

The modern research in virtual reality is directed in four different ways, based on visualization displays: the Cathode Ray Tube (CRT), the Head-Mounted Display (HMD), the Binocular Omni-Oriented Monitor (BOOM) and the multi-projection systems like CAVE (Cruz-Neira, 1992). The characteristics of each device are divided into two categories: immersion fields and visualization fields. The immersion fields are divided into: view field, viewpoint, perspective, physical representation of the body and intrusion. And, the visualization fields are presented in: visual acuity, linearity, spatial view, progressive refinement and collaboration. Another article distinguishes virtual reality from other standard graphics developments made in computer as the following characteristics (Cruz-Neira, 1993):

1. Occlusion;
2. Perspective projection;
3. Binocular disparity (stereo projectors);
4. Motion Parallax (head movement);
5. Convergence (visual search field);
6. Accommodation (visual focus, as a reflection of single-lens in the search field);
7. Atmosphere (environment);
8. Lights and shades

The standard graphics developments provide features number 1, 2, 7 and 8, while the virtual reality adds number 3, 4 and 5.

Multi-projection Devices based on CAVE, have a set of features that gives the user a deep interaction with the virtual reality. According to the authors, the CAVE was developed to be a tool for scientific visualization. Another purpose is the theater of virtual reality, where scientists design the criteria of exposure. The objectives pursued with the development of the first CAVE included:

1. The desire for color images of high resolution and a peripheral vision without geometric distortion;
2. Less sensitivity to head rotation, inducing errors;
3. The ability to merge Virtual Reality with real devices;
4. The need to guide and teach through artificial worlds;
5. The desire to connect supercomputers in net-works and data sources for consecutive refinements.

The first CAVE was developed at the Electronic Visualization Laboratory at the University of Illinois in Chicago, and, presented at SIGGRAPH in 1992 (Cruz-Neira, 1992).

4.1. Multi-user Immersive Visualization Laboratory

This section will describe the Multi-user Laboratory of Immersive Visualization through its hardware and software constitution.

4.1.1. Hardware

The Multi-user Immersive Visualization Laboratory (MIVL) of the University of São Paulo (USP), situated in São Carlos-SP, has a device for multi projection of virtual reality, based in CAVE, assembled by the Absolut Technologies company. This device is composed of three screens of projection measuring: 3,2m x 2,4m for the frontal screen, 2,4m x 2,4m for the right screen and 3,2m x 2,4m for the bottom screen.

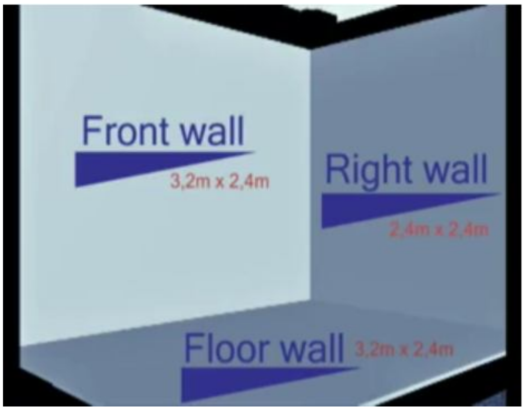


Figure 2 – Dimensions of the screens of the MIVL multi projection system

For each screen of projection there are two Christie projectors, 3000 DS+60/DW30/Matrix model and a computer equipped with Nvidia video card FX4500 series and 4GB of RAM memory. The connections between the computers and the projectors are made through DVI cables. There is a main computer where it is possible to access and to control the three other computers responsible for the screens. The system allows passive stereo visualization, where each projector is responsible for the referring image of one of the user's eyes. The referring images are overlapped in the screen so that it is possible to get the sensation of immersion in any virtual environments, through the use of stereoscopic eyeglasses. The stereoscopic eyeglasses used in the MVIL are: INFITEC filters. This type of system makes the simultaneous immersion of few users in the same virtual environment possible, all of them equipped with stereoscopic eyeglasses.

4.1.2. Software

The operational system in the four computers of the MIVL is Microsoft Windows XP Professional. For the modeling of objects and virtual scenes, amongst others, Blender version 2.48a and OpenGL libraries for C++ are used. The programming platform is the Visual C++ 2005 Express Edition with VRJuggler 2.2.1 to allow the virtual environments display in the multi-projection system. VRJuggler allows the use of tools that improve the development of virtual worlds for multi-projection, while the creation of the scene graph and the importation of 3D models are allowed through the use of OpenSG 1.6.0.

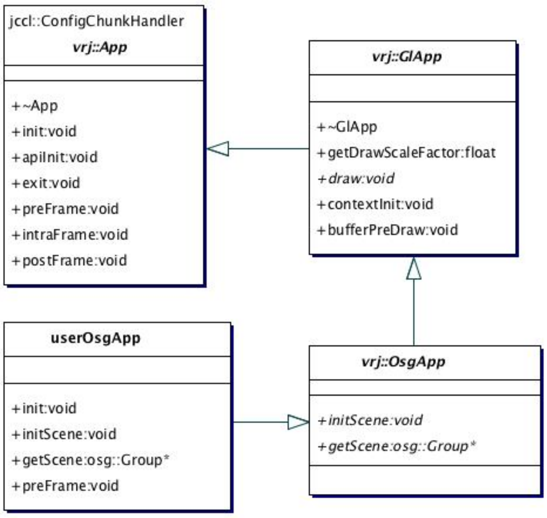


Figure 3 – VRJuggler classes working with OpenSG

5. VIRTUAL ENVIRONMENT FOR TEACHING ROBOTS

The development of a virtual environment is usually divided into two steps: the modeling of the environment and the programming of the same. The concepts involved with the modeling of the robots are described below.

5.1. Robot Geometrical Model

The free software Blender was used for the construction of a generic robot geometrical model for teaching robotics. Two main file types were used: VRML and X3D. The Virtual Reality Modeling Language (VRML) is a standard file format for representing 3D graphics that was created mainly for the use over the internet. Although VRML allows a simple text file to keep vertices, edges, polygons or even animations, it was superseded by Extensible 3D (X3D) file format. X3D is the ISO standard XML-based file format for representing 3D computer graphics. This format has some features extensions to VRML like humanoid animations, interactive 3D, NURBS and the ability to encode the scene using XML syntax.

In this work two generic revolute joint robot models, one with just one joint and the other with two joints, were created in Blender in order to export them with interactions of X3D file format. It works well in a desktop based system, but it can’t be used in a multi-projection device. The model had to be exported in VRML format because OpenSG can’t import X3D format yet. There is some research on using X3D with multi-projection systems.

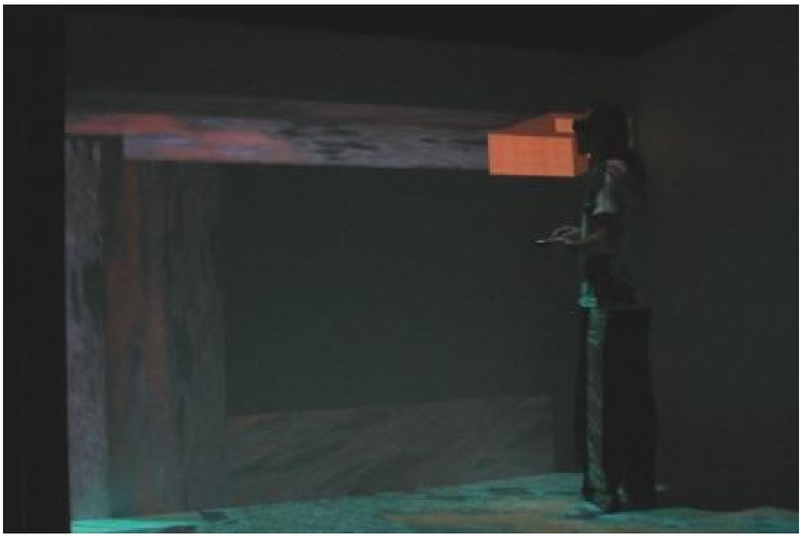


Figure 4 – A simple revolute joint robot model for teaching robotics

5.2. Programming The Multi-Projection System

In this work a VE for teaching robotics was developed for the use in the MIVL multi-projection system. The VRJuggler manages all the I/O interfaces running as an operational system, while the OpenSG creates, controls and keeps the scene graph. For the three-dimensional object importation a scene graph, like shown in Fig. (5), was created:

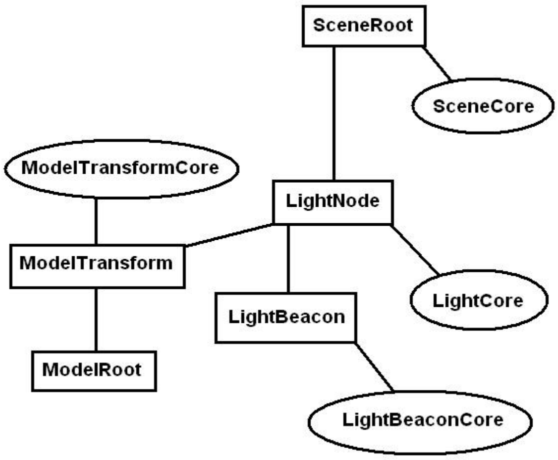


Figure 5 – Example of an OpenSG scene graph.

The scene graph is composed of nodes, represented by rectangles, and cores, represented by ellipses. The cores apply transformations, such as rotation, scale and translation, to the node and all its children nodes, due to the hierarchic characteristic of the scene graphs. The LightNode is the node responsible for the illumination of the scene. It has two children, the beacon node that defines the position and orientation of the light source, illuminating all its children nodes and the ModelTransform node that contains the transformations of its child node ModelRoot. The Model Root node is the imported three-dimensional geometric model. In this work VRML file format was used to enable the use of the multi-projection system.

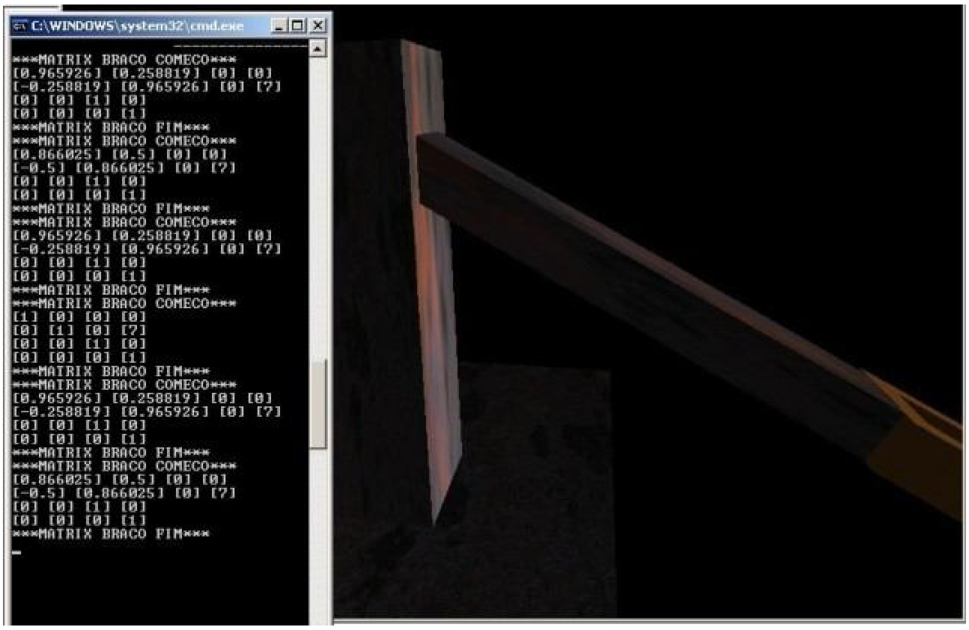
Each part of the robots was imported independently to a specific OpenSG node so the user could apply individual rotations for any robots' joints. Thus, in this work, the ModelRoot node shown in the above Fig. (5) is subdivided in several nodes in a hierarchal way. The closer the arm is to the robot base, the higher is its level of hierarchy. Therefore, the tool, located at the robot last arm, has the lowest hierarchal level in the scene graph. Rotations applied to any joint move the tool proportionally and the rotation of the robot base must rotate all robot arms.

Firstly in this work two simple robots were modeled to enable the learning of forward kinematics, the first and most simple one with only one arm, one joint and a tool; and the second one with two arms, two joints and a tool. The user could apply rotations to any robot joints and watch the animation in the screens of the multi-projection system. The matrixes values of the applied transformations were shown in the prompt screen of the main computer of the cluster. By applying transformations, watching the animations and verifying the matrixes values, the user can learn robotics in a more intuitive way. The interaction between the user and the system was done through a numeric wireless keyboard, in which the direction keys command the joints' rotations in relation to Z axis. The minimum stipulated value for the transformation was fifteen negative or positive degrees.

The values of the transformation matrixes for this work haven't been displayed on the screen of the multi projection device. They are only displayed at the command prompt of the central computer, which commands the device, as shown on Figure 6. This is a complex task, when we deal with multi-projection systems. Therefore, the ideal use of the system requires two users, a teacher using the central computer and a student using the device with stereoscopic glasses and a numerical keyboard, or vice-versa.

The project is ongoing and new functionalities are being implemented in order to make the teaching of robotics for beginners possible.

The inverse kinematics will be implemented in a way that the robotics students can enter position values of the tool a well as verify the values of every matrix of each joint of the robots.



```
***MATRIX BRACO COMECO***
[0.965926] [0.258819] [0] [0]
[-0.258819] [0.965926] [0] [7]
[0] [0] [1] [0]
[0] [0] [0] [1]
***MATRIX BRACO FIM***
***MATRIX BRACO COMECO***
[0.866025] [0.5] [0] [0]
[-0.5] [0.866025] [0] [7]
[0] [0] [1] [0]
[0] [0] [0] [1]
***MATRIX BRACO FIM***
***MATRIX BRACO COMECO***
[0.965926] [0.258819] [0] [0]
[-0.258819] [0.965926] [0] [7]
[0] [0] [1] [0]
[0] [0] [0] [1]
***MATRIX BRACO FIM***
***MATRIX BRACO COMECO***
[1] [0] [0] [0]
[0] [1] [0] [7]
[0] [0] [1] [0]
[0] [0] [0] [1]
***MATRIX BRACO FIM***
***MATRIX BRACO COMECO***
[0.965926] [0.258819] [0] [0]
[-0.258819] [0.965926] [0] [7]
[0] [0] [1] [0]
[0] [0] [0] [1]
***MATRIX BRACO FIM***
***MATRIX BRACO COMECO***
[0.866025] [0.5] [0] [0]
[-0.5] [0.866025] [0] [7]
[0] [0] [1] [0]
[0] [0] [0] [1]
***MATRIX BRACO FIM***
_
```

Figure 6 – A simple robot arm and its transformation matrix values

6. THE TEACHING OF ROBOTICS THROUGH THE DEVELOPED SYSTEM

The developed virtual environment allows professors from any area related to robotics to show the students, in an interactive way, the mathematical functioning involved in translations and rotations of robots. By using the stereoscopic glasses, the professor as well as the students, inside the multi-projection system, can observe the movements of the robots tridimensionally on the three walls. The transformation matrixes of the movements are always displayed at the screen of the main computer and are updated at each command given by the professor or the student through a wireless keyboard. Because of this, the learning of direct or inverse kinematics can be done in a simple and efficient way.

The virtual environment for teaching robotics was developed exclusively for the use on the MIVL multi-projection system. However, it is possible to adjust it to any multi-projection system. A multi-projection system, as for example CAVE, can make the immersion in a virtual environment considerably more efficient, but the developed system can also be adapted to Desktop systems.

The developed virtual environment provides that programmers are able to enrich the database of types of robots, due to the fact that its construction in C++ is based on object orientation.

7. CONCLUSIONS AND FUTURE WORKS

With the use of multi-projection systems it is possible to considerably extend the sensation of realism and immersion in VEs. Therefore, it can be concluded that the robotics learning can be assisted, in a reliable way, by a multi-projection system. However, practical classes cannot be totally replaced by virtual ones. VR can be an additional tool on teaching robotics.

The developed VE is just an initial project for a complete system of teaching robotics. More advanced concepts, like implementation of inverse kinematics is been incorporated to the system. Some kind of industrial robots can be modeled and adapted to the VE in order to enable an industrial focused robotics' learning. Therefore, as future works our researches intend to develop a WIMP (Windows, Icons, Menus and Pointing devices) interface to control and visualize the robot transformation matrixes, as well as to enable a better way of inputting data. Besides that, Industrial robots, as for example the Puma 562 Manipulator, are been modeled and adapted to the VE of teaching robotics. Inverse kinematics and other types of robots other than revolute joint robots are as well being developed. Finally, another important improvement is to display the transformation matrixes to the users on the screens of the multi-projection system.

8. ACKNOWLEDGEMENTS

We would like to thank the Coordination and Improvement of Personnel with Superior Level (CAPES) and The State of São Paulo Research Foundation (FAPESP) for aid to the good development of this work.

9. REFERENCES

Coiffet, P. and Chirouze, M., 1983, "An Introduction to Robots Technology".

Craig, J.J., 2005, "Introduction to Robotics Mechanics and Control", Third Edition.

Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., 1993, "Surround-screen projection-based virtual reality: The design and implementation of the CAVE", Proc ACM SIGGRAPH 93 Conf Comput Graphics, New York, pp. 135-142

Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., Kenyon, R. V., Hart, J. C., 1992, "Cave. Audio Visual Experience Automatic Virtual Environment", Communications of the ACM, San Diego, Volume 35, Issue 6, Pages 65-72, June.

Oliveira, D.M.; Cao, S.C.; Hermida, X.F.; Rodriguez, F.M., 2007, "Virtual Reality System for Industrial Training", Industrial Electronics. ISIE 2007. IEEE International Symposium on Volume , Issue , 4-7 June 2007 Page(s):1715 - 1720.

Schuemie, M. J., Straaten, P.V.D., Krijn, M., and Mast, C. A. P. G., 2001, "Research on presence in VR: a survey", Cyberpsychology & Behavior Volume 4, Number 2.

Yang, X., Zhao, Y.; Wu, W. and Wang, H., 2008, "Virtual Reality Based Robotics Learning System", Proceedings of IEEE of the International Conference on Automation and Logistics, Qingdao, China.

10. RESPONSIBILITY NOTES

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