Development of a VR Simulator for Medical Training Using Free Tools: A Case Study

Daniel F. L. Souza[1],Ícaro L. L. Cunha[1], Leandro C. Souza[1],
Ronei M. Moraes[2] and Liliane S. Machado[1]

[1] Departament of Informatics and [2] Departament of Statistics
Universidade Federal da Paraíba
Cidade Universitária s/n – 58051-900 – João Pessoa/PB – Brazil

danieltidus@gmail.com, ivellius@yahoo.com.br, leokarllos@yahoo.com.br,
ronei@de.ufpb.br, liliane@di.ufpb.br

ABSTRACT

The simulation of medical procedures through virtual reality systems can offer realistic training environments for new professionals. This paper presents the development of a simulator for medical training, based in a previous study case. The application was completely developed using a free system called CyberMed which provides high-level access to several functionalities expected in a virtual reality simulator. Details of the application development will be presented. The results obtained will be compared with the previous case study.

Keywords

Virtual Reality Systems, Haptics, 3D Visualization, Medical Training.

1. INTRODUCTION

In the last decades, virtual reality (VR) applications have been developed and used to enrich the learning process of students and professionals in medical fields. The technological advances have allowed the manufacturing of graphic cards used to provide high-level realistic visualizations at low-cost. More recently, the development of haptic devices has allowed the enhancement of VR applications and now users can touch and feel the material properties of virtual objects. Nowadays, sight, touch and hearing can be explored together in a VR system. In medicine, the presence of touch and 3D visualization can provide immersive and interactive simulations [4, 15, 19].

One of the advantages of VR simulators for medicine is the possibility of training without risks [1]. Additionally, these applications avoid the damage of materials and also can present models and situations hard to be trained in real conditions [2]. Then, virtual models can replace plastic models, guinea pigs and cadavers because they cannot replicate the ideal properties of a live human body: plastic models are not available for all body parts and do not present the same properties of a live human body; guinea-pigs present tissue properties different from the human body tissues; and cadavers have their tissues properties degraded if compared to live tissues [10]. However, VR simulators for medicine must provide realistic and immersive environments which answer to the users movements. Once these systems require several computational resources to process all information in real-time, the development of them is not a simple task.

This paper presents the development of a VR simulator for medical training using free tools. The goal is to show how a free system, called CyberMed, can be used to simplify the development and reduce the cost of this kind of application. The features and results of this application are compared with a previous work.

2. VR AND MEDICAL TRAINING

VR applications for training in medicine have been developed for different procedures. Ocular surgery simulators use stereoscopic visualization to allow observing eye structures and their manipulation through haptic devices [18, 20]. Endoscopic procedures [4], laparoscopy [19] and arthroscopy [9] can be learned through VR system. There are several other medical application based on VR, dedicated to tumor detection [5] and bone marrow harvest [11], as example. Those systems use techniques and concepts chosen according their goal, as collision detection, haptics, interactive deformation, 3D visualization, volumetric reconstruction, assessment and ergonomy, among others.

Nowadays, a visualization system (monoscopic or stereoscopic), an interaction device and physical models compose a typical VR simulator for medicine [4]. Once VR system can provide interaction and immersion in realistic environments, medical simulators can offer more than a reproduction of real training by the recreation of the real circumstances of a procedure.

Besides the advances, the technology available still do not allow the creation of a realistic simulator to explore the whole human body using volumetric models, 3D visualization, multiple haptics, deformation, interactive cut and assessment. The use of all these features requires a lot of processing and can compromise real-time performance. In haptics, for example, it is commonly used a single point of contact for touch and force feedback due to the processing required for multiple points of contact . Then, VR simulators for medicine are defined and developed to deal with specific parts of the human body and present some limitations: according to the procedure requirements, some VR techniques are

more explored than others. These limitations do not impede the development of good and useful training simulators [4].

2. DEVELOPMENT OF SIMULATORS

One of the difficulties related to the development of VR applications, which explore more than one human sense, is the integration of the tasks related to each sense and their synchronization. Most of the simulators found in literature were developed through the implementation and integration of all routines necessary. Because there are several devices and specific APIs (Application Programming Interface) dedicated to their programming, a programmer with experience to deal with each device or API is necessary to develop the VR simulators [10, 11]. Methods for collision detection, deformation and stereoscopic visualization also need to be completely implemented each time an application is developed. Besides, the several methods need to deal with the same models and interaction devices. Then, all this information and all application tasks must be synchronized to avoid delays or inconsistencies which can compromise the realism.

Recently, some systems were developed to the design and implementation of VR applications, as the ReachIn API [14]. However this kind of development system is not free and demands specific non-free platforms [10]. To solve this, two free systems have been developed to allow the integration and synchronization of tasks and tools in VR systems: The CyberMed [6, 7] and the CHAI3D [5]. The main difference between these systems is the level of abstraction offered to the programmer. In this case, the CyberMed presents a higher level of abstraction than the CHAI3D. However, the CHAI3D is more generic and can be used to develop any kind of application. On the other hand, the CyberMed is dedicated to the development of medical applications and was not tested to other ends. Due to its specific objective, the CyberMed presents methods to on-line assess user's performance. This feature is not found in other development systems. Finally, the CyberMed is based on a free platform (Linux) while the CHAI3D is available only for the Windows™ operational system.

2.1 The CyberMed System

The CyberMed system has been developed to facilitate the process of development of VR simulators for medical purposes [6]. It is composed by a set of classes which provides several facilities and support for several devices. This feature gives to the programmer a high level system which do not demand specific knowledge about devices and their programming API. The first version of the system offers support to four different methods for visualization (monoscopic, spectral multiplexing, polarization multiplexing and temporal multiplexing [17]), support to haptics, collision detection, deformation of tissues and assessment of user's training [7, 12]. Moreover, all the functionalities available in the CyberMed are synchronized. The implementation details are completely transparent for the programmer. However, the system is flexible enough to allow low-level access to expert programmers. Additionally, the system presents interfaces to the addition of new methods and devices support.

In an application developed with the CyberMed, the topology of the objects is stored in a data structure named OF [1]. It will be used by collision detection methods of the CybCollision and by deformation methods of the CybDeformation. A copy of this information is also stored in a class named CybParameters and is used to haptic rendering routines. The class CybInterator contains data related to the interator used in the application. Because the CyberMed is ready to the use of 2D and 3D devices, the CybInterator class communicates with the CybHaptics and CybMouse. All information about the visual and haptic scene is also stored in the CybParameters and used to synchronize them. The CybView and CybHaptics classes manage these scenes, respectively. The CybView was developed through calls to the OpenGL functions and does not depends on any window system. Figure 1 shows a diagram of classes relationships.

The CyberMed can be used in any Linux distribution with a C++ compiler. The use of specific haptic devices only demands the installation of the device API. Once the class which supports the device is present in the CyberMed, no knowledge of the device API will be required to develop an application with the CyberMed.

An assessment class, called CybAssess, is available in the CyberMed and can be used to collect user actions during a simulation [7]. This class uses data from the CybParameter and CybInterator classes to assess the user performance. Figure 2 shows the integration of the CybAssess class in the CyberMed.

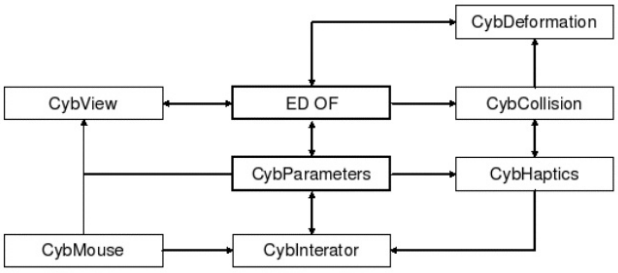

Figure 1. Classes of CyberMed and their relationships.

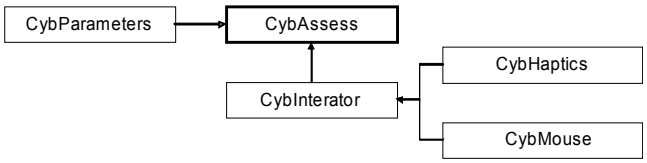

Figure 2. CybAssess and its relationship with other CyberMed classes.

3. CASE STUDY

This work used the CyberMed to develop a simulator for the training of bone marrow harvest, based on a previous work [10]. The bone marrow harvest in one of the stages of the bone marrow transplant and demands dexterity from the physician who performs it. Basically, two steps compose it: palpation and harvest [13]. In the palpation step, the physician must identify the iliac crest under the skin of the pelvic region. The iliac crest is the

region used to perform the second step: the harvest of bone marrow.

The simulator developed presents the two steps of the harvest procedure. Since the simulator is concerned to training [11], an extra step was added to allow the anatomy study of the pelvic region. Thus, three modules compose the final application: Study, Palpation and Harvest. These modules were implemented to have the same modules and functions described in the previous work [10].

The final application used the CyberMed classes: CybView, CybMouse, CybParameters, OF, CybInterator, CybHaptics and CybAssess. These classes provided to the final application the follow functionalities, respectively: visualization, mouse interaction, storage and management of 3D models, interaction control, haptic control and on-line user's assessment.

The CybView class allowed choosing the visualization mode of the application. In spite of CyberMed support four different view modes, the simulator was defined to provide only the monoscopic and temporal multiplexing visualization modes. With this class two light sources were defined and an illumination model was enable to the visual scene and its objects. The menu entries were also made in the CybView. The objects used in the visualization are obtained from the OF and CybParameters classes which store the models topology, the model used to represent the interaction device and the transformation matrixes of the scene. The CybInterator class was used to choose the interaction object. The CybAssess will access in real-time the transformation matrixes and CybInterator to the user's assessment. In order to reach that, it was used the CybAssess with an assessment method based on Maximum Likelihood [7], already available.

The CybHaptics class allowed setting the material properties of the models and enabling the haptic device. This class provided an abstract layer for several API used to programming haptic devices and did not demand knowledge about the device API. The device used was the Phantom Omni [16], presented in Figure 3. However, the application also works with the Phantom Desktop device [16].

Figure 3. The haptic device Phantom Omni.

Five models compose the application: three for the human body structures and two for the interaction object. The interaction objects, finger and a needle, were modeled with the Blender package [3] and saved in VRML format supported by the CyberMed. The models of human body structure used were the same of the previous work [10] and represent the skin of the pelvic region, the iliac bone and the bone marrow. Figure 4 shows the models used to represent the interaction object in the visualization. The CybHaptics allowed relating the contact point to a vertex of the objects: a point in the tip of finger and tip of the needle.

The menu was designed to offer to the user three modules for the training. This menu is dynamically modified according to the user choices. Only a visual exploration is enable in the Study module and the user can choose which structures he wants to see and set their transparency. Figure 5 shows the Study module and the menu options available. The user can also modify the position and orientation of the structures through mouse interaction. If shutter glasses were available, it is possible to start the stereoscopy visualization.

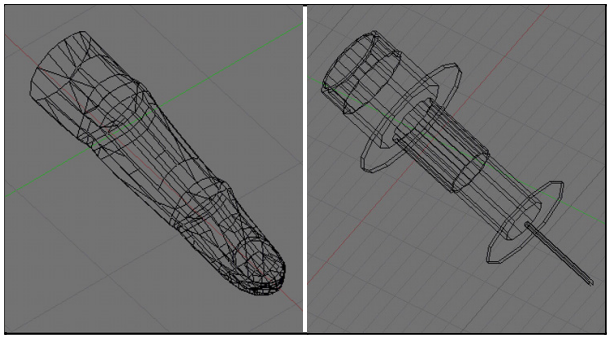

Figure 4. Models used to represent the interaction object in the visual scene: a finger and a needle.

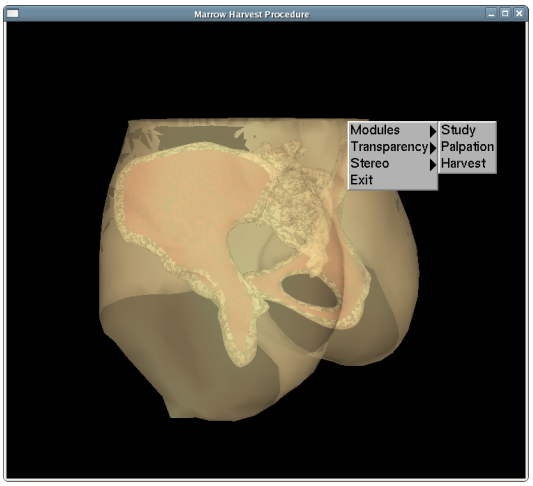

Figure 5. Screenshot of the Study module and the menu options available.

The second module available is the Palpation module. For this step of the training, all interaction with the objects was disabled and the position of the objects was fixed. In fact, the only visualization possible is from the backside of the skin model.

Because the visualization of the bone and bone marrow models is not allowed, these models were disable and will not be rendered. However, they can be identified by touch since the user starts the haptic interaction (Figure 6). With the haptic device the user will be able to feel the different material properties throughout the skin and identify a hardest area, located under the iliac crest (not visible). In this module, the finger model was related to the haptic device and the point of contact is located on its tip.

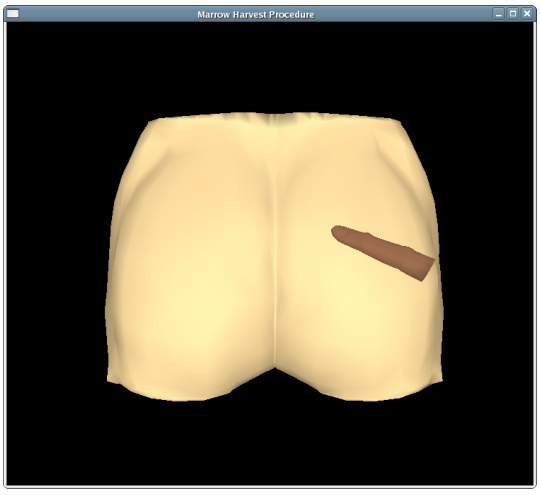

Figure 6. Palpation module and the representation of the haptic device by a finger.

In the last module, the Harvest, the user can practice how to harvest the bone marrow. As in the Palpation module, movements with the body models are not allowed and a needle represents the haptic device. In this module is possible to penetrate into the models with the haptic device and all body structures - skin, bone and bone marrow - are haptically displayed. It allows reaching the bone marrow under the skin and inside the bone.

The code in Figure 7 shows parts of the implementation with the CyberMed used to read the models and set the material properties for the haptic interaction.

To the on-line user's assessment the simulation had to be executed several times to calibrate the assessment tool. This stage allows acquiring the assessment parameters to be used in the on-line assessment. In order to reach that, an expert executes several times the training and at the end of that, he/she labels each one into M classes of performance (M=5 in the Figure 8: "Well Qualified", "Qualified", "Need Some Traing", "Need More Training" and "Novice"). This procedure is necessary to measure the statistical variability for each class of performance and improve the accuracy of the assessment method. Several methods for training assessment were proposed in the literature. The CyberMed uses a Maximum Likelihood based method [8]. Figure 8 shows the calibration stage of the simulator in the harvest module.

The only difference between the calibration procedure for training assessment and the final training application is the presence of calibration options in the menu. In the calibration procedure, these options will be selected by an expert to label the class of performance for each calibration procedure. In the final application this menu option is replaced by an assessment option which can be selected by the user to receive his assessment report at the end of the simulation.

```
1  int numLayer = 5;
2
3  // read the models
4  char *arqname[30] = {"1.wrl","2.wrl","3.wrl","4.wrl", "5.wrl"};
5  ofMesh<cybTraits> malha[numLayer];
6  CybDataObtainer<cybTraits> data(numLayer);
7  ofWrlReader<cybTraits> entrada;
8  CybParameters cybCore;
9
10 // set haptic material properties for the skin
11 interator.hapticDevice.createHapticLayers(0,true);
12 interator.hapticDevice.createMaterialPropertyContext(numLayer, true);
13 interator.hapticDevice.setMaterialPropertyValue(0, POPTHROUGH, 1.0f);
14 interator.hapticDevice.setMaterialFace(0, POPTHROUGH, FRONT);
15 interator.hapticDevice.setMaterialPropertyValue(0, DYNAMIC_FRICTION, 0.6f);
16 interator.hapticDevice.setMaterialFace(0, DYNAMIC_FRICTION, FRONT);
17
18 // set haptic material properties for the bone
19 interator.hapticDevice.createHapticLayers(1,true);
20
21 // association of the contact point to a model point
22 interator.setObjectType(0, -0.020913, 0.078050, -10.034132);
23
24 //  enable material property for haptics
25 interator.hapticDevice.enableHapticMaterialProperty();
```

Figure 7. Example code of the CybBMH using CyberMed.

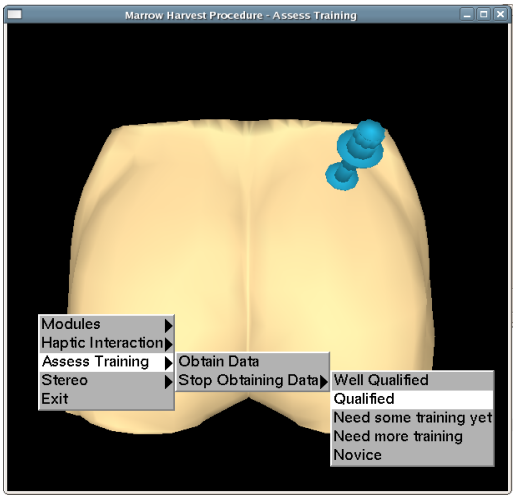

Figure 8. Assess training: acquisition of the assessment parameters.

4. COMPARISON WITH PREVIOUS RESULTS

The bone marrow harvest simulator was designed and developed by [10, 11]. The present work used the same models and material property parameters to develop a new application, with the same training goal but using only free tools in its development. To compare the applications, the previous simulator will be called BMHSimulator and the new application CybBMH. The Table I shows the main differences between BMHSimulator and CybBMH.

The CybBMH was developed only using free tools: GNU C++ language and the CyberMed system. The modeling package used to model the needle and the finger was the Blender [20], also free. Because the CyberMed integrates and synchronizes visual and haptic routines, it was not necessary the use of other tools for the development of the CybBMH. In opposite, the BMHSimulator required the use of several tools that must to be integrated, besides the necessity of synchronization of the visual and haptic tasks [2].

The use of a single class to store the models topology allows utilizing the same three objects for visual and haptic scenes in the CybBMH. Additionally, two models were used to represent the haptic device in the visual scene. The illumination treatment offered by the CyberMed also allows providing a better visual quality. The visual quality of the simulators can be observed in the Figure 9. The BMHSimulator used three models for the visual scene and other three models for the haptic scene. Due to performance reasons, the haptic models were simplified and the number of vertexes was reduced. Then visual and haptic scenes did not share the same object and was necessary to use six different models in the simulator. The representation of the haptic device was done through textures applied on crossed plans. In spite of optimizing the rendering process, this approach presented graphic imperfections.

Table I. Main differences between the simulator previously developed and the new simulator.

	BMHSimulator [10, 11]	CybBMH
Operational system	Windows™	Linux
programming tools	- Visual C++ - OpenGL - Ghost API - Microsoft Foundation Classes	- GNU C++ - CyberMed
Haptic device	Phantom Desktop [20]	Any Phantom family device [20]
Stereoscopy method	temporal multiplexing	temporal multiplexing
Number of models	6	5
Time for application development	~1 year	~2 months
Hardware	800MHz CPU 1Gb RAM 128Mb graphics card	3.4MHz CPU 2Gb RAM 512Mb graphics card

Both simulators support the use of shutter glasses and haptic devices. However, in the BMHSimulator the use of this hardware required a previous study of OpenGL functions to implement the routines for stereoscopic images generation. The same effort was necessary to integrate the haptic device and its routines to the application through the device API. In that simulator, the application window and menus were defined using the Visual C++ IDE (Integrated Development Environment) and the Microsoft Foundation Classes. The efforts needed to learn how to use all the tools mentioned above and to the implement the application demanded approximately one year for the complete development of the simulator. The CybBMH also demanded efforts to its development. However, it was focused in the use of the CyberMed classes. By the definition of objects and by the access to their of methods, it was possible to use methods, already integrated, ready to deal with stereoscopy, haptic devices and menus. It allowed reducing the time of development of the simulator to approximately two months. Additionally, the CyberMed classes provided an on-line assessment method to assess user's performance. This functionality offers a performance feedback to the user and can be used to know his ability level.

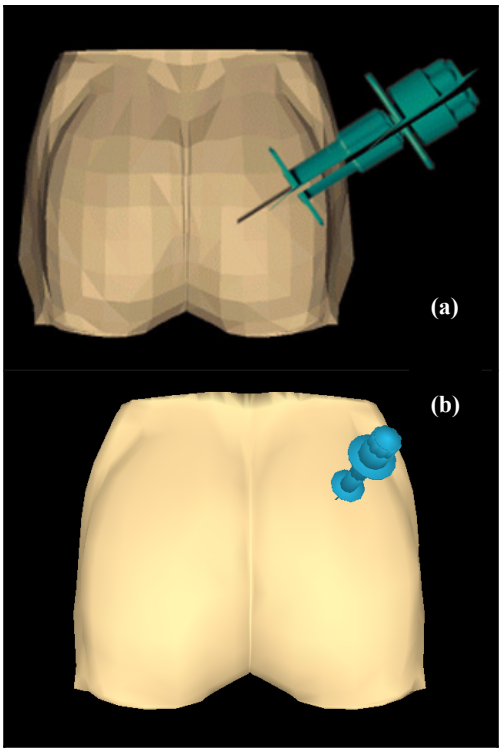

Figure 9. Visual results of a previous application (a) [2] and of the simulator developed with the CyberMed (b).

The sequence of the steps of the simulation is the same in both applications. The dissection and the transparency settings available in the Study module can be equally performed through menus. However the illumination model provided by the CyberMed classes allowed to offer better graphic results in the CybBMH, especially in the Study module, due to the presence of three models composed by thousands of points (skin, bone and bone marrow).

Two haptic devices of the Phantom family [16] were tested and used with the CybBMH: the Omni and the Phantom Desktop. The

exchange of the haptic devices did not require any change in the final code of the application. The main differences between the two devices are the size of the workspace, the precision and the amount of force which can be exerted.

The time consumed in the development of the BMHSimulator and the CybBMH did not include the acquisition of the simulation parameters or the design of models used in the application.

5. CONCLUSIONS

This paper presented the results obtained with the development of a VR simulator for medical training using free tools. The CyberMed system was chosen to develop a simulator based on a previous application for the training of bone marrow harvest [10, 11]. With this work was possible to conclude that the development of simulators for medical training can be done through the use of free tools with satisfactory results.

It was possible to observe that the use of a system dedicated to the development of medical simulators, as the CyberMed, optimized the implementation process and did not require depth knowledge of methods and details related to devices and their API. Besides, the system offered an on-line method to assess the user's performance, a feature very useful in an application dedicated to learning.

The graphic results were considered better than in the previous simulator. However, they can be improved by the use of a global illumination model to project the interaction object shadow in the body. It could be done by the override of the CybView class of the CyberMed. Equally, was observed that new functionalities could be added to the CyberMed, through its abstract classes, to support new methods or devices.

7. ACKNOWLEDGMENTS

This work is supported by the Brazilian Council for Scientific and Technological Development, CNPq (grant CT-INFO 506480/2004-6) and Brazilian Research and Projects Financing, FINEP (Grant 01-04-1054-000).

8. REFERENCES

[1] Al-khalifah, A.; Roberts, D., Survey of modeling approaches for medical simulators, *Proc. 5th Intl. Conf. Disability, Virtual Reality & Assoc. Tech*., Oxford, UK, 2004.

[2] Arbor, A., An Immersive Virtual Reality Platform for Medical Education: Introduction to the Medical Readiness Trainer, *Proc. 33rd Hawaii International Conference on System Sciences*. 2000.

[3] Blender Foundation. Online: http://www.blender.org. Access in December, 2006.

[4] Burdea, G. e Coiffet, P. Virtual Reality Technology, New York: Wiley, 2nd ed, 2003.

[5] Burdea, G.; Patounakis, G.; Popescu, V.; Weiss, R. Virtual Reality Training for the Diagnosis of Prostate Cancer. *Virtual Reality Annual Int. Symp. Proc*. IEEE, 1999, 190-197.

[6] Conti, F.; Barbagli, F.; Balaniuk R.; et al. The CHAI libraries. *Eurohaptics*, 2003. Online: http://www.eurohaptics.vision.ee.ethz.ch/2003/75.pdf

[7] Cunha, Í. L.L.; Moraes, R. M.; Machado, L. S., Unifying Data Structures for Virtual Reality Applications. In: *World Congress on Computer Science, Engineering and Technology Education Proc*. Santos, 2006, 404-408.

[8] Cunha I.L.L, Machado, L.S., Moraes, R.M. Performance Analysis of Assessment Using Maximum Likelihood for Virtual Reality Systems. In*: World Congress on Computer Science, Engineering and Technology Education Proc*, Santos/SP, Brazil, 2006. CD-ROM.

[9] Mabrey, J.; Cannon, W.; Gillogly, S.; Kasser, J.; Sweeney, H.; Zarins, B.; Mevis, H.; Garrett, W.; Poss, R. Development of a Virtual Reality Arthroscopic Knee Simulator. *Studies in Health Technology and Informatics* n.70. IOS Press, 2000, 192-194.

[10] Machado, L.S. *Virtual Reality to Modeling and Simulation of Invasive Procedures for Pediatrics Oncology: A Case Study in Bone Marrow Transplant*. PhD Thesis. Dept. of Electrical Engineering, University of Sao Paulo, 2003. (In Portuguese)

[11] Machado, L.S. e Zuffo, M.K. Development and Evaluation of a Simulator of Invasive Procedures in Pedriatic Bone Marrow Transplant. In: *Studies in Health Tech. and Informatics*, v.94. IOS Press. 2003, 193-195.

[12] Moraes, R. M., and Machado, L. S. Maximum Likelihood for On-line Evaluation of Training Based on Virtual Reality. *Proceedings of Global Congress on Engineering and Technology Education (GCETE'2005).* Santos, Brasil, Março 2005, 99-302.

[13] Pizzo, P. A.; Poplack, D.G., Principles and Practice of Pediatric Oncology. Filadelfia, Lippincott-Raven Publishers, 1997.

[14] Reach In Products - Reach In API. Online: http://www.reachin.se/products/reachinapi/index.asp. Access in December, 2006.

[15] Riva, G., Applications of Virtual Environments in Medicine. *Methods Inf. Med* (42), 2003, 524-534.

[16] Sensable Technologies – Haptic Devices. Online: http://www.sensable.com/products-haptic-devices.htm. Access in Dezember, 2006.

[17] Sherman, W., Craig, A. Understanding Virtual Reality. Morgan Kaufmann, 2003.

[18] Sorid, D.; Moore, S. K., The Virtual Surgeon. *IEEE Spectrum*, v. 37, n. 7, 2000, 26-31.

[19] Székely, G.; Satava, R.M., Virtual Reality in medicine. In: *BMJ*, 319(7220): 1305.

[20] Wagner C,; Schill, M. A.; Mãnner, R, Collision Detection and Tissue Modeling in a VR-Simulator for Eye Surgery, *Proc. Workshop on Virtual Environments*, Barcelona, Spain, 2002, 27-36.