2014 5th IEEE RAS & EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
August 12-15, 2014. São Paulo, Brazil

Implementation of 3D visualization aplications based on physical-haptics principles to perform rehabilitation tasks

L.D. LLedo, S. Ezquerro, F.J. Badesa, R. Morales, N. Garcia-Aracil and J.M. Sabater

Abstract— **Nowadays, There are a lot of tools and procedures for the development of computer applications for teaching, entertainment, telecommunications, marketing, design and other more. This paper present a implementation method for developing applications based on virtual reality and procedures physical-haptics, in order to perform rehabilitation tasks, describing the used software tools. The first one is Ogre3D which is used as rendering graphics engine to add realistic 3D visualization features. Then, the physical engine NVIDIA PhysX is used to incorporate accurate physics simulation and to implement collision detection between objects in the virtual environment. The third one is OpenHaptics which is used to generate a force feedback in the haptic device Sensable Phantom. Using the developed applications, the user's immersion sense in the virtual environment is increased and improved, since the user can manipulate virtual objects with realistic physical behaviour. Finally, two examples of implementation in a rehabilitation environment are shown to demonstrate the main features of the developed tool.**

I. INTRODUCTION

Virtual reality(VR) is a powerful tool to develop virtual rehabilitation environments for upper limb which are oriented towards the recovery of functional abilities to perform activities of daily living (ADL), such as, cooking, grasping a glass of water and so on [1]. In short, a recent review about the application of rehabilitation therapies based on virtual reality in stroke patients concluded that there are insufficient evidence of the benefits of VR on grip strength or gait speed and limited evidence about the benefits of VR on recovery arm function and abilities to perform (ADL)[2]. However, the meaning of "limited evidence" should be understood as that the number of subjects enrolled in the reviewed clinical trials are too small to have statically significant outcomes. For instance, a recent study with 376 subjects concluded that "VR rehabilitation in post-stroke patients seems more effective than conventional interventions in restoring upper limb motor impairments and motor related functional abilities" [3]. In short, the authors pointed out that the improvements were greater in the group exposed to virtual rehabilitation therapies than in the group exposed to classical rehabilitation therapies [3].

Kepping in mind these findings, a realistic virtual rehabilitation environment has been developed including: high quality rendering graphics engine to perform realistic 3D environments, accurate physics simulation and collision detection tools and force feed-back in the haptic device Sensable Phantom Omni. This paper proposes a development process of 3D visualization applications based on physical-haptics principles for increasing the levels of trust and realism in the current simulator, and improving the patient's immersion sense in the virtual rehabilitation environment when the patient practices any exercise. Then, a physical calculation system [5] and a haptic system of feedback force [6] have been incorporated. A number of free software tools have been used to design this type of applications, following some established requirements:

- The virtual objects must be displayed with an appearance as much as possible to reality through high density of polygons and quality textures (OGRE3D).
- The elements involved in the physical simulation have to behave in a similar way to reality, when applying force on them (PhysX).
- These systems need a real time interaction [7].
- The simulator elements have to be touchable, so the patient can manipulate the elements with the haptic device [8], receiving a force feedback that indicates the interaction between the patient and another virtual object (Phantom Omni Sensable, OpenHaptics).

This paper is organized as follows: The section 2 describes briefly the tools used. The section 3 introduces the implementation process of the virtual reality applications, combining the 3D visualization, physical simulation and haptic simulation modules. Two examples of implementation and their respective objectives are explained in the section 4. Finally, the conclusions are presented in the section 5.

L.D. LLedo, S. Ezquerro, F.J. Badesa, R. Morales, N. Garcia-Aracil, J.M. Sabater are with Biomedical Neuroengineering Universidad Miguel Hernandez de Elche, Spain {llledo, fbadesa, rmorales, nicolas.garcia, j.sabater}@umh.es

II. LIBRARIES

A. Graphics library - OGRE3D

Ogre3D [9](*Object-oriented Graphics Rendering Engine*) has been used as a graphics engine. It is a software that enables the visualization of highly realistic virtual environments in real time. Its main features are:

- The engine has a high-level interface in C++ and object-oriented.
- It is multiplatform. Supports OpenGL [10] and Direct3D [11].
- Use scene graph to perform efficient object searching and manage the virtual geometry optimizing the graphics load on the GPU [12].
- Architecture based on plugins.
- Manage a interpretation system of scripts to define surfaces materials o implement postprocessing effects.

978-1-4799-3127-9/6/14/$31.00

- It has hardware acceleration, thanks to the shaders. It also allows working with hardware buffers.

B. Physical library - PhysX

PhysX [13] is physical simulation engine with hardware acceleration developed by NVIDIA. It is able to perform collision detection and interaction among the simulation elements in real time and generates virtual elements response to the influence of forces, determining the motion. His physical modules solve the dynamic behaviors of rigid bodies, deformable bodies [5] or tissues and fluid particle systems [14], [15], releasing the CPU to perform a high load of physical calculations, so much increases the speed of the simulation by programming the CPU [12].

This engine allows to simulate objects with a high degree of realism, create physical effect such as gravity or friction, calculate collisions and design objects with complex shapes articulately. It can be run on many platform and its SDK is free marketing, but closed source. It has an ANSI C++ programming interface, implemented as a hierarchy of classes that contain accessible features and calculations specific to the hardware use.

C. Haptic library - OpenHaptics

The OpenHaptics tool [16] allows incorporate haptic functionalities to the simulation applications in order to create a sense of immersion in the virtual space and establish interaction between the user and the environment from contact or force feedback. In this way, the user can touch, feel and manipulate objects simulated in 3D environments allowing the realization of perceptual or motor tasks and cover all possible areas of the scene. OpenHaptics is a set of free software libraries for the programming language C++.

III. SYSTEM ARCHITECTURE

Developed applications are virtual environments that project visual images generated by computer where the user can interact to perform rehabilitation tasks. Thanks to the incorporation of a haptic system, the user can manipulate virtual tools that allow the transmission of the touch sense and the sensation of objects. As shown in figure 1, the user controls the haptic device (Phantom Omni Sensable model, whose specifications can be seen in [8], [17]) to manage the position and orientation of a virtual tool that interacts with the simulated environment. The appearance of the virtual tool depends on the application being executed. The movement of the tool would cause contact between the virtual scene elements generating a collision detection which will result in a force feedback.

The physical engine ensures that the elements behave in a similar way to reality, then it detecs collisions between the virtual tool and the environment elements, and estimates the interaction force approaching the contact force that would be generated from real objects. The magnitude and direction of the forces generated will depend on the type of simulated tool. Through control algorithms haptic device, and depending on the pose of the virtual tool and the forces

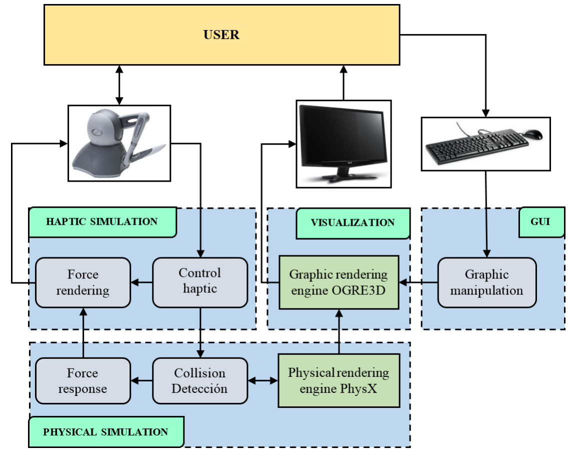

Fig. 1. Basic system architecture

generated by the physical engine, a rendering of forces to the user is produced, maintaining the transfer of forces in both directions in real time to maintain a stable performance.

This type of applications require a series of graphics models responsible for representing three-dimensional object. There are 3D modelling tools like Blender [18] or 3D Studio to get these objects. In addition, each graphic object requires a physical model for the physical engine to perform the calculations of collisions. These physical models are a simplified version of its corresponding graphical model. Thus, during the execution of Ogre3D, each graphical model replicates the movement performed by the corresponding physical model within the physics simulation, using the interaction forces to calculate their effect in the virtual environment objects. The user is at all time a visual and haptic feedback to use the application. Also, the user has an option to use the keyboard and mouse to change the position and orientation of the camera, change the rendering mode of the scene or add information panels with frames per second and other properties.

IV. VIRTUAL REHABILITATION APPLICATIONS

This section shows some examples of 3D visualization applications developed using physical principles and incorporating force feedback, through the combination of the tools presented previously.

A. Rehabilitation simulator - Cooking tasks

The first example is a simulator for the rehabilitation of upper limbs, whose stage displays a kitchen that contain and run the simulations of a number of elements, such as, a pan, potatoes, various shakers, a table, a shelf or an hob.

A controllable element represented by a virtual arm have been incorporated to enable user interaction with these elements. Each simulator element consist of a graphical model, designed with Blender and managed by Ogre3D, and a physical model generated by PhysX, which is responsible for assigning shapes to establish the core of the collision detection so that the elements do not intersect each other and simulate gravity to bring more realism. A screenshot of this simulator is shown in figure 2.

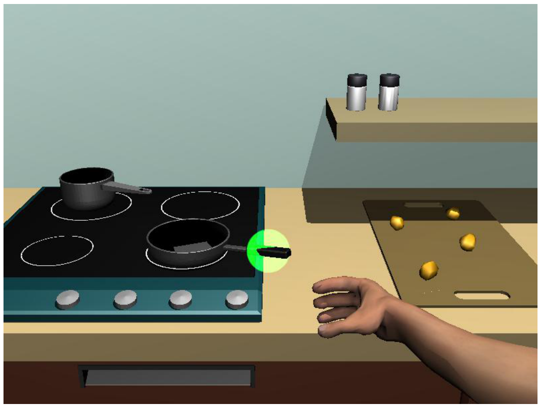

Fig. 2. Screenshot of the kitchen simulator

The position and orientation of the virtual arm is controlled using the haptic system formed by the Phantom device, exploited the features of this device when render a force feedback to provide the user a sense of contact with the simulated elements. Thanks to the control arm, the user can take the various elements that appear in the kitchen to complete the different objectives that can be set in the simulator as daily living tasks: for example, make a recipe by placing the ingredients in the kitchen equipment as pans, change the position of containers or light the fires of the hob.

These targets depend on the type of virtual items displayed on screen, allowing the user to perform any task or a combinations of several tasks. Because of this versatility when adding items, all the tasks that a person could develop in a kitchen can be modelled virtually, setting different difficulty levels depending on the user skill and its evolution.

1) Collision detection: The PhysX physics engine is able to calculate collisions between the controllable element by the Phantom and the other elements of the kitchen, just as calculated collisions between other elements. Thus, contact forces are generated in the simulator from the interaction between objects and gravity force, which allows that the objects to act very similar to reality when they turn to lose balance or when they fall when released from a height. PhysX is also able to generate reports with approximate values of these points of contact. The normal vector of each contact or contact forces can be obtained directly. In figure 3, examples of contact points between the objects are shown.

2) Force haptic rendering: After getting the collision detection data between the arm and virtual environment objects, such as place, time or size, a force vector must be applied to a haptic device to simulate the contact as close to reality as possible, trying to minimize the error between the ideal forces calculated by the physical engine and the forces applicable to the device. OpenHaptics can perform haptic rendering of forces using the values obtained by the physical engine.

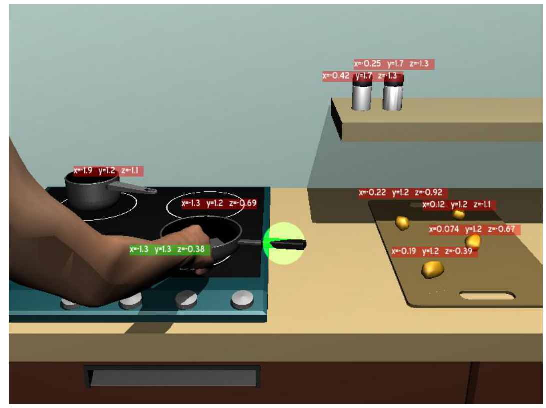

Fig. 3. Screenshot of the kitchen simulator with contacts

B. Rehabilitation simulator - Motor activity of grasping a glass

The second developed example is a simulator for the rehabilitation of the upper limbs as well. The scenario consists of a glass, a table, a coaster and the virtual arm, whose orientation and position is managed with the haptic device. A screenshot of the current simulator is shown in figure 4.

Fig. 4. Screenshot of the motor activity simulator

The user controls the arm movements aimed grasping the virtual glass is the simulator as main objective. When holding the glass, the user attempts to simulate drinking task, making the necessary moves to bring the glass to his/her mouth. The glass is secured by surrounding it with the hand, so that can tilt if the arm does not approach correctly, due to the strength if the simulated gravity. Finally, the user has to place the glass on the coaster.

1) Fluid dynamics: PhysX has a particle system capable of performing a dynamic simulation of fluids and gases, to enhance the sense of physical immersion of the simulator. This physical feature, with subsequent graphics rendering with Ogre3D, increases the potential of the system when simulating situations of daily life, such as liquid spill into a glass.

This feature has been used in the simulator to check weight changes that the user has to bear when the fluid is poured out, so that the force feedback of the haptic device is modified in real time to simulate the gravity effect. A series of simulations of filling and pouring liquid into a glass have been performed to check the consistency of the fluid dynamic algorithm. First time is used to fill the glass, then wait a few seconds until the glass being to turn.

All these simulations are gathered in figure 5, which shows a graph with the dynamic behaviuor of the liquid to pour out the content with different velocities of inclination or tilt speeds expressed in radians per second, and the force exerted by the set glass-fluid on the user. The graph shows that increasing the tilt speed when turning the glass, the weight is reduced faster due to faster liquid spill.

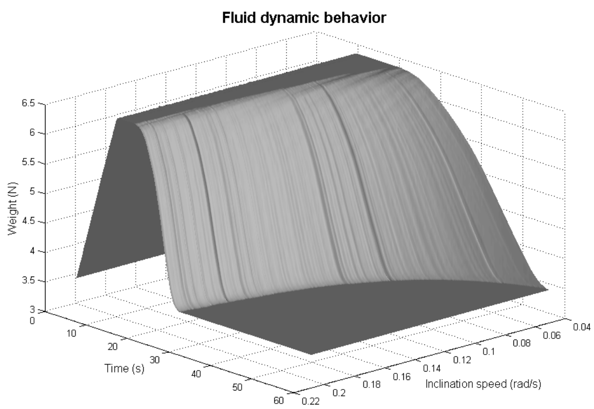

Fig. 5. Graphic with pouring fluid simulations

The PhysX fluids are particle systems. Therefore, a graphical mesh is created for rendering fluid with the graphic engine Ogre3D, whose vertices are located in the positions of the particles and a material is applied to represent each vertex as a sphere.

To better analyze the graph above, a simulation sample is extracted with its fluid dynamic behaviour of the spill and its corresponding variation in weight, with a particular tilt speed. The extracted sample is observed in figure 6. This figure shows a drecrease of weight generated by the vessel and the fluid, as a function of time due to the escape of liquid.

Some simulator screenshots have been added in the graph to show the visual state of the vessel and the fluid during the pouring process, for example when it starts to pour, when it has half the fluid and when it is spilled. These pictures shows how each fluid particle is represented as a white sphere, and how they interact with each other.

V. CONCLUSIONS

In this paper, the needed tools to develop rehabilitation applications of virtual reality have been presented. Ogre3d has been used for the 3D visualization of the scene elements successfully. Also, it has been chosen as graphics engine thanks to its simplicity and efficiency when programming in C++, its architecture based on plugins and the existence of a large number of free and open source libraries, which facilitate the creation of applications whithout expense.

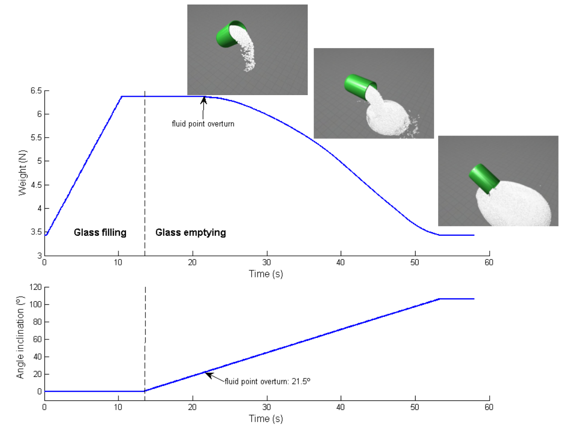

Fig. 6. Graphic with a extracted sample

The physical engine NVIDIA PhysX has been used to simulate virtual scenes with a high degree of dynamic realism. This library performs calculations of physical behavior and collision detection between virtual objects, effectively and fast enough for a right performance.

This provides a lot of advanced simulation features and allows hardware acceleration to prevent that the CPU implements all the physical load, which helps to improve the 3D visualization. The OpenHaptics software and the Sensable Phantom device have been used as haptic device, due to its simplicity to incorporate force feedback in this virtual simulators. The combined use of these tools offer a great versatility and high potential for developing any type of simulator.

As future work, the incorporation of news virtual elements and objectives are proposed to represent more tasks of everyday life in the rehabilitation applications. Design and development of new simulators for rehabilitation of lower limb and/or hands have also been raised.

ACKNOWLEDGMENT

This work has been supported by the Spanish Government through the project ”Interpretation of the human intention and performance through biomedical signals and analysis of kinematic and dynamic motion” (DPI2011-29660-C04-04).

REFERENCES

[1] Stephen R. Ellis. What Are Virtual Environments?, IEEE Computer Graphics & Applications, Julio, 1994.
[2] K. Laver, S. George, S. Thomas, J. Deutsch, M. Crotty: Virtual reality for stroke rehabilitation. Cochrane Database Syst Rev 2011, 9:CD008349
[3] A. Turolla, M. Dam, L. Ventura, P. Tonin, M. Agostini, C. Zucconi, P. Kiper, A. Cagnin, L. Piron, Virtual reality for the rehabilitation of the upper limb motor function after stroke: a prospective controlled trial Journal of NeuroEngineering and Rehabilitation 2013, 10:85
[4] S. E. Kober, G. Wood, D. Hofer, W. Kreuzig, M. Kiefer and C. Neuper, Virtual reality in neurologic rehabilitation of spatial disorientation, Journal of NeuroEngineering and Rehabilitation 2013, 10:17.
[5] A. Maciel, T. Halic, Z. Lu, L.P. Nedel, S. De, Using the PhysX engine for Physics-based Virtual Surgery with Force Feedback, Int J Med Robot 2009, 341:353.
[6] K. Salisbury, F. Conti y F. Barbagli. Haptic Rendering: Introductory Concepts, IEEE Computer Society, marzo/abril 2004.

[7] E. Akenine-Mller, T. Haines and N. Hoffman. Real-Time Rendering. AK Peters, Ltd., 2008
[8] PHANTOM Tecnhologies, http://www.dentsable.com/haptic-phantom-omni.htm/
[9] Ogre 3D, http://www.ogre3d.org/
[10] D. Shreiner. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1 (7th Edition). Addison-Wesley, 2009
[11] DirectX 9 - http://www.microsoft.com/es-es/download/details.aspx?id =34429.
[12] R. Fernando, GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics, Pearson Higher Education, 2004.
[13] PhysX library 2.8.4, https://developer.nvidia.com/physx-sdk/
[14] W. J. van der Laan, S. Green, M. Sainz, Screen space fluid rendering with curvature flow, Proceedings of the 2009 symposium on Interactive 3D graphics and games, 91:98.
[15] R.G. Winkler, M. Ripoll, K. Mussawisade, G. Gompper, Simulation of complex fluids by multi-particle-collision dynamics, Computer Physics Communications, Volume 169, Issues 1:3, 1 July 2005, Pages 326-330.
[16] SensAble OpenHaptics Toolkit 3.0: Programmer's Guide, http://www.geomagic.com/download_file/view/2370/6392/
[17] SensAble Technologies, http://www.sensable.com/
[18] Blender, http://www.blender.org/