Elective in Robotics:
Module of Locomotion and haptic interfaces for VR exploration

Hardware and software description of the Cyberith Virtualizer and its possible applications

Mario Muscio

22 June 2016

Contents

1 Introduction

One of the topics in the haptic and locomotion interfaces field is the motion control. The devices involved in the motion control are usually the treadmills. A special type of treadmill is the omnidirectional treadmill (ODT) that allows a person to run and walk in any direction, as opposed to the traditional treadmills that only goes one way. There exist two ways to create a omnidirectional treadmill:

- *Passive Method.* The human body moves while the device is fixed, i.e. Virtuix Omni, Cyberith Virtualizer.
- *Active Method.* Much more complex than passive one. Using an active unit, the device respond to player's movement by moving a surface below player's feet. The surface moves as player walks by keeping him to the center of the surface itself.

In the passive method the player needs a harness and/or footwear, while in the active method does not. An active treadmill provides better simulation since it is capable of much more than a passive one, but is too expansive and complex.
The treadmills can be used for lots of applications. Most mentioned application is the VR simulation. This device works usually with other devices, in particular visual devices, i.e. Oculus Rift, with decoupled inputs from each device. The conjunction of these devices gives a more natural and fluid immersion in the VR simulation. The special device described in this chapter is the Cyberith Virtualizer (figure 1).

Figure 1: Cyberith Virtualizer

The Cyberith Virtualizer belongs to the passive treadmills and this report will describe the device on hardware and software point of view.

2 The Cyberith Virtualizer

In this chapter, we focus in the description of the Cyberith Virtualizer, in particular: a presentation of the device, the assembling and the interface.

2.1 The device

The Virtualizer is the first product of the Austrian start-up Cyberith, founded in February 2014 by Tuncay Cakmak and Holger Hager. Cyberith is specialized in Virtual Reality Hardware.
As said, the Cyberith Virtualizer is a omnidirectional treadmill and it allows movements of the player in any direction by using a combination of low-friction principle and high precision sensors. The player can walk, jump, crouch, sit and any other possible lower body movement.

Figure 2: Jump and crouch movements on Cyberith

The Cyberith is made up of three main parts: the base plate, the pillars and the ring-construction.

Figure 3: From the right: base plate, pillar and ring-construction

2.2 Hardware and Assembly

To assembly the Cyberith Virtualizer, it needs to follow some steps. These steps are provided in [3]. The first step consists on assembling the flat base plate and the base support for the pillars.

Figure 4: Base support

The plate is composed by 6 optical trackers for the angle of direction and speed of the player's feet. These sensors pick-up the horizontal position. The movement of the feet on the plate leads to a low-friction principle: its material meets a specific friction-coefficient that allows the player to slide over the plate effortlessly. Due to this principle, the player does not need any other device on his feet: just socks are enough, or shoes with appropriate footwear.

Figure 5: Six optical sensors and footwear

Other sensors are available but they are not directly connected with user movements. These sensors have the purpose of enhance the full immersion sensation. In fact the base plate is equipped with vibration units, which can provide haptic feedback linked to virtual environment events, such as in-game actions (explosions, earthquakes, etc).
The next step is to assembly the three pillars. One of the pillars contains a sensor that picks-up the vertical position of the ring-construction. This sensor is attached by a RJ45 cable connected to the angled bar and then connected to the base of the pillar.

The angled bars are connected to the pillars by a system of pulleys and springs.

Figure 6: Right: Angled bars ; Left: Springs

The last step is to assembly the ring-construction. This ring is composed by an inner ring that allows the player to rotate of 360.

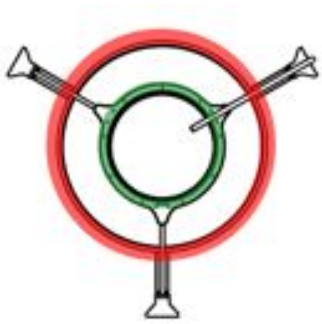

Figure 7: Inner ring highlighted in red ; Ring-Construction highlighted in green

At this inner ring is mounted a harness for player safety and a sensor that picks-up the orientation of the player.

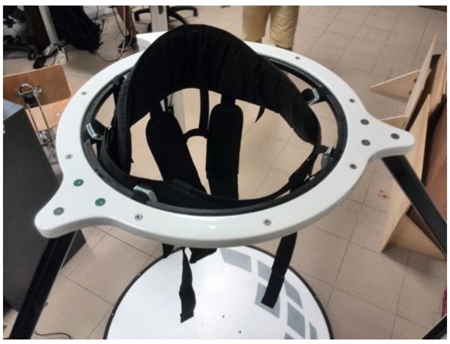

Figure 8: Harness

An additional step is to assembly the support for the visual device on one of the pillars. This support structure allows the player's movement not to be hindered by bulky cables.

Figure 9: Support structure for a visual device

At the end the complete device is like this:

Figure 10: Cyberith Virtualizer after assembly

2.3 Software Interface

To interface the Cyberith Visrtualizer with a computer, the devise uses a simple USB-Plug&Play device.With the easy-to-use software interface it is possible ”to choose between emulating either a controller or a keyboard, making the Virtualizer compatible with any software that allows for controller and/or keyboard input”[1].
There no exists a dedicated software to work with the device. The company just provides some API available for C++, C# and Python programming languages. A great disadvantage is that these API are available only on Windows O.S., they have not created API (or at least that we know) that allow other O.S. to work with the Cyberith Virtualizer. These API are contained in a script within a Software Development Kit (SDK) and they are described also in [4]:

- *GetPlayerHeight*: this call function gives back the current height (float) of the ring-construction in centimeters. The value is 0 for default height (upon reset) , less than 0 when the player is getting lower (crouching) and more than 0 when upper (jumping).

- *ResetPlayerHeight* : this function resets the default height of the ring-construction.

- *GetPlayerOrientation*: this call function gives back the orientation (Vector3 type) as being the forward vector of the body (ring).

- *GetMovementSpeed*: this function gives back the speed (float) of the player's movement in meter per second.

- *GetMovementDirection*: this function gives back the Movement Direction (Vector3 type) as a normalized vector of structure: (right, 0, forward).

There is another function available called *ResetPlayerOrientation* that in the current Windows-SDK provided by the company is disabled.
The first step necessary to work with the device is the calibration. The company provides a GUI for the calibration operation:

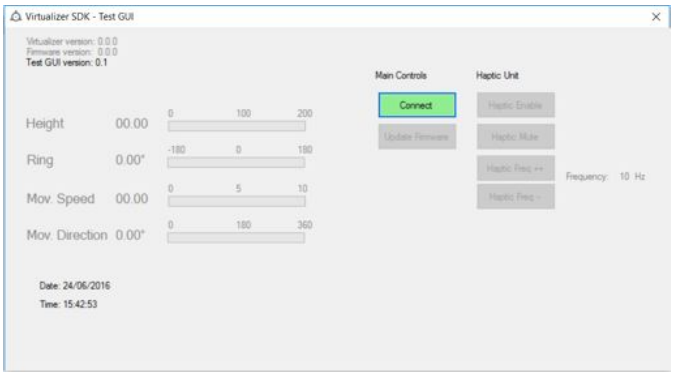

Figure 11: The Test GUI before calibration

After starting the GUI, the next step is to connect the USB-Plug&Play and perform the calibration steps: first move the ring-construction on its minimal and maximal height and then rotate the inner ring of more than 360. After these steps, the player just have to press the CONNECT button in the GUI and it will give the current values from the sensors:

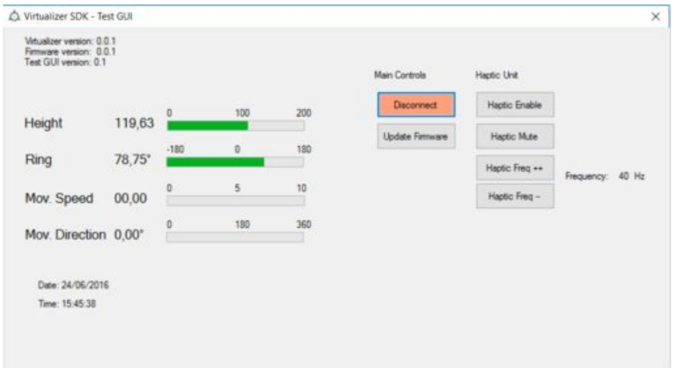

Figure 12: The Test GUI after calibration

After the calibration is possible to disconnect the device from Test GUI and use it. It is important to disconnect the Cyberith Virtualizer from the calibration software because it has the priority on the device: if the user does not disconnect the device and perform a Unity scene, Unity will never use the resource because the listener of Unity find the resource in the busy state. The device can be finally used. In the next image it is possible to see that the work of the virtualizer leads only on the values retrieved by the sensors.

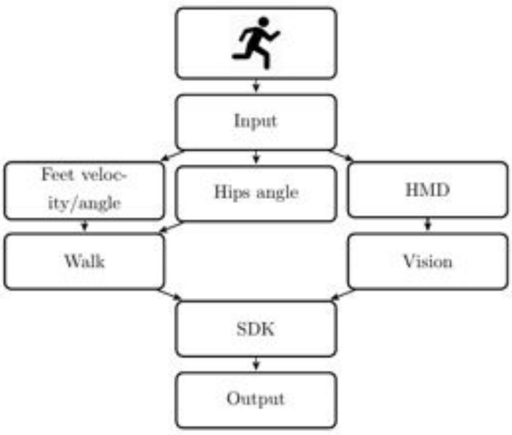

Figure 13: Operating principle of the Cyberth Virtualizer

3 Testing the device and Application

3.1 Tests

To test the device we have performed a Unity 3D scene of the city of Osaka (Japain).

Figure 14: 3D representation of Osaka

To perform a 3D walking in the city using Unity, it needs just the environment and an object in the scene that represents the player. An object in a Unity scene can be composed by other objects creating a hierarchy. The hierarchy our object is composed by the physical object and a camera that performs the same transformations in the scene of the primary object.

Figure 15: Sphere (CVirtPlayerController) representing the player

These transformations are controlled by the scripts written in C# and added to the structure of the object. In this test the most important scripts added to CVirtPlayer-Controller are CVirtPlayerController.cs and CVirtDeviceController.cs. The first script allows to the object to move in the scene, in particular to apply rotations and move forward or backward according to the values of the sensors of the Cyberith Virtualizer. The last script allows to choose the device controller: first the Virtualizer, then a gamepad and last mouse and keyboard.

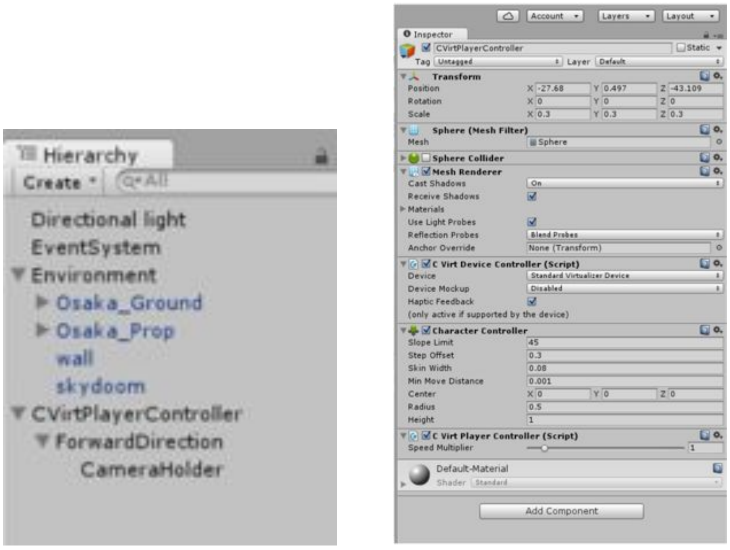

Figure 16: CVirtPlayerController scripts

The position of the camera in the space is such that the player can see the scene on the screen in first person. So the player is almost immersed in the 3D environment.

Figure 17: First person test

After the calibration of the device and the loading of the 3D scene, the last step is just to run the program and enjoy the walking in Osaka.
The other test performed on this device is the extraction of the values of the sensors. This is done by executing the Windows-SDK and using the API mentioned before to extract these value of the sensors of the device in real time. The output will be something like this:

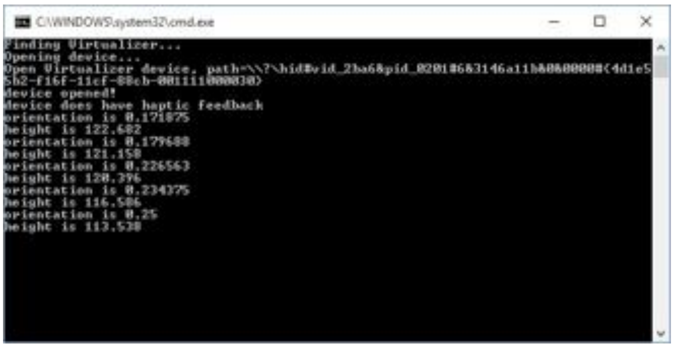

Figure 18: Height and angle of the ring-construction

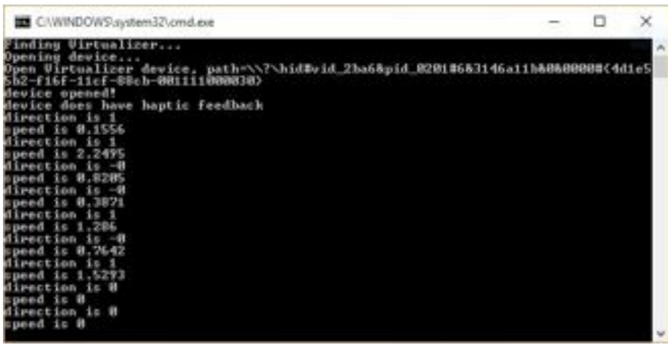

Figure 19: Movement speed and direction of the player

3.2 Applications

In this section possible applications of the Cyberith Virtualizer, but also of the other device within the omnidirectional treadmill class, are considered. The most usually application of this device, and also the one used for all the demonstrations available on Internet, is the VR simulation, i.e games. This can be done in conjunction with another device which is the Oculus Rift: as said in the introduction this conjunction can perform a better simulation. Another application of the device can be in the field of cinema: together with a green screen this type of application would allows limitless movement for the actors.

Another important application can be in the medical field: "the study of people's walk in order to correct its defects or more in general in physiotherapy"[1].

Another possible application of the Virtualizer is, as in the tests described before, a simple walk, but in this case not in a modern city. It is possible to have a walk in the ancient city, like Pompei, reconstructed in 3D and enjoy the charm of history. There is a Italian start-up, called Oniride, which creates 3D reconstruction of various architectural sites. These reconstructions are available on the stores of the smartphones, so just with using a very cheap Virtual Reality Headset and a smartphone is possible to "really" live these sites. Other companies which do the same thing enrich their reconstructions by adding a simulation of peoples living in those particular historical periods.

An application that can be interesting is in Robotic field. The device can be used to allow a humanoid robot to perform the same movements of the player in real time using the values of sensors for controlling the robot.
There are also plans to enrich the structure of the Cyberth Virtualizer to perform a better simulation, giving some feedbacks to the player. This was presented by [5]:

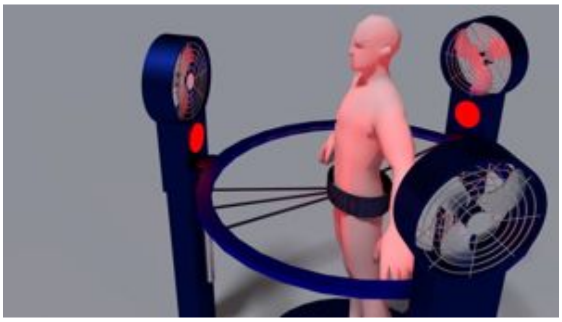

Figure 20: Wind and heat sensors added to the Cyberith Virtualizer

As shown in the figure, the device is enriched with other sensors, such as infrared lamps to generate heat and ventilators for the wind. They are supposed to be assembled with the pillars in order to have a better virtual reality experience.

4 Conclusion and final considerations

In this report a new type of omnidirectional treadmill for VR applications was presented. A detailed description of the device, starting from the hardware structure and assembly description and ending talking about how the software interface works, was given. Then some tests were described and finally some possible applications of the device were explained. However, it is not possible to say much more about the Virtualizer, since it is still a developing device.
Some final and personal considerations are now given. A first consideration is about the walk of the user on the device. From the simulations, it was highlighted that the walk is not so natural for the player; a possible solution can be changing completely the principle of the low-friction for the base. Maybe it is possible to implement for the base plate a magnetic system between the surface of the plate and the footwear. This can perform a better walk, but the cost will be too much expensive and there will be no feedback on the player feet.
Another issue found working with this device, and also mentioned before, is the absence of libraries available for other O.S. This can be a problem in the Robotic application mentioned before. Most of the robots are controlled by programs available on the Linux O.S. (see NAO Robot). So it would be a good idea to create the same API actually available on Windows also for Linux O.S. in order to facilitate the interface between the device and the robot.

References

[1] Philipp Van Belrlekom,
"The Virtualizer - Immersive Virtual Reality Gaming",
Vienna 08.07.2014

[2] A.Spada,
"Cyberith Virtualizer - Working principle and applications of a new omnidirectional treadmill"

[3] T. Cakmak and H.Hager,
"Assembly Instructions"

[4] *"Cyberith SDK - Unity Project - Integration Guideline"*
https://developer.cyberith.com/

[5] ZolarEnergy.net,
"Walking, Wind & Heat in VR",
goo.gl/CkY1W2

[6] http://www.oniride.com/public/it/homepage-ita

[7] http://treadmill-review.toptenreviews.com/omnidirectional-treadmills-the-future-of-mo
html

[8] http://www.virtualrealitytimes.com/2015/04/09/
list-of-omnidirectional-treadmills-under-development/

[9] http://cyberith.com/product/