Immersive Virtual Reality application using Google Cardboard and Leap Motion technologies

Mihai Chifor
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
chifor.mihai21@gmail.com

Teodor Stefanut
Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
teodor.stefanut@cs.utcluj.ro

ABSTRACT

Virtual Reality applications have known an accelerated development in the past few years stimulated by the advancements in the hardware capabilities and also by the benefits from the similarity with the real world. However, the available solutions for immersive VR are still very expensive and often lack the friendliness expected on user interaction techniques. In this paper we will present a low cost, immersive, VR capable setup based on two new technologies: Google Cardboard and Leap Motion. The proposed solution is showcased using an educational application that allows its users to virtually visit Elmwood Park Zoo, Pennsylvania and learn more on the wild animals in the process. Most of the user interactions are performed through hand described gestures inspired from the real world, helping the users to accommodate more easily and perform naturally in the virtual environment.

Author Keywords

Virtual Reality; Google Cardboard; Leap Motion; immersive; educational

ACM Classification Keywords

H.5.1. Information interfaces and presentation (e.g., HCI): Multimedia Information Systems.

INTRODUCTION

Virtual reality (VR) represents today a promising media platform that allows the users to experience different computer-generated worlds in a very similar approach to the real one. One key aspect in feeling these worlds as real is the user’s ability to interact with and influence the virtual environment. Therefore, a simulation of the interaction between a human being and the real environment has to be implemented into the virtual one, giving to the user the illusion of reality.

Up until recent years, one of the main issues concerning VR has been related with the very limited possibilities of the devices that allow the users to interact with the scene. Currently, there are two main categories of haptic interfaces: the *on-body* (attached to the user’s body) and *off-body* (positioned independently of the user, somewhere in the real environment – on the floor, walls, desk, etc.).

Nowadays, most commercially available devices are off-body, allowing the user to move freely in the real environment. A few devices inspired from game-optimized ones have been used as alternative inputs for the VR environments, but only with limited success as they proved to feel unnatural. Due to the similitude with reality, one of the first user instincts when immersed in a virtual world is to use his/her body to move around and interact with the elements of the environment. While walking around the virtual scene using user’s physical body is currently a popular research direction that is complementary to the one presented in this paper, using the hands-free motion in virtual interactions has proved to be a more natural approach and represents our choice of interaction.

Continuous development of the virtual simulations and their integration with VR environments has encouraged the implementation of learning scenarios that could benefit from the advantages of this technology. Through the use of detailed 3D models and virtual reality environments as part of the training process, one can experience an entirely new side of training. Some of the main benefits would be:

- the learning process has a higher level of interactivity being able to transform the user experience from passive to active
- while immersed into the virtual environment, the user is protected from distractions from surrounding elements
- VR based learning can represent a great solution for situations where the teaching materials are very expensive or the conditions of training are very hard to reproduce in real world; a scenario that has been modeled with a high level of detail allows learners to interact with it, understand and follow best practice procedures or carry out complex scenarios with very reduced cost / trainee (as all the resources can be reused as many times as needed).
- the interactive VR scenarios help the user gain knowledge, test his/her reactions in dangerous situations and test possible harmful scenarios in a safe environment, without being put at risk.

The purpose of this article is to describe the user experience into a 3D immersive environment created with the use of 2 particular pieces of hardware, namely Google Cardboard [4] as a head mounted display (HMD) and Leap Motion Controller [10] for hand tracking recognition. We developed our application based on a real life zoo model, namely Elmwood Park Zoo, Pennsylvania [14]. The virtual world was developed using a 1:1 scale to increase the realism by maintaining the proportions of the objects, animals and even user's virtual hands.

RELATED WORKS

Continuous development of mobile phones has allowed in the last few years the development of more complex 3D mobile applications than ever before. Joining this trend, Google has proposed a new visualization technique that allows these applications to immerse their users into Virtual Reality environments, providing spectacular results at very low costs. This new approach is named Google Cardboard, and is actually composed out of a headset to support the display and two optical lenses, a mobile device (which is the actual display), a magnet (for user interactions) and specific display settings of the 3D application.

Google Cardboard has been showcased in applications like:

- Earth: fly where your fancy takes you on Google Earth
- My Videos: Watch your videos on a massive screen
- Exhibit: Examine cultural artifacts from every angle
- Photo Sphere: Look around the photo spheres you've captured
- Windy Day: animated short story

Probably the most popular application that engages the user in a virtual tour using Google Cardboard is Google Maps for Android [5]. The application has been extended to allow its users to visit different places (like Versailles) in virtual reality (VR) mode and with a virtual guide [4]. It also provides the user with directions, interactive maps, and satellite/aerial imagery of many countries. However, the available user interaction techniques are limited to the ones based on the phone's magnetometer, which only allows the user to confirm specific actions or options into the virtual environment.

Similar to Google Maps for Android, in our application we have focused on a particular place and we have created a detailed 3D model of the environment. The main goal has been to enable the user to interact in different manners with the elements of the VR medium, using natural gestures described with his/her own hands. For movement tracking and gesture recognition we have used the Leap Motion device, presented in detail in the next section of the paper.

Leap Motion is a promising input device that, for now, is under heavy development. Its capabilities have been showcased in several demo application, one of the most complex being "Form and Function 3D" app. In this educational tool the users have the possibility to compare hearts of different animals and learn anatomy in the process. The 3D accurate anatomical models have been created based on real-world specimens and can be manipulated by the user through hand gestures through Leap motion device. The application functionalities include surface and internal views of the organs, blood flow simulations and quiz mode for knowledge evaluation.

Currently, only a few applications that combine both virtual reality and the hand tracking capabilities of the leap motion controller are available on the Leap Motion App Store [9]. Most of these applications are actually games, while our example is more concerned with educational aspects.

HARDWARE USED IN IMPLEMENTATION

Leap Motion

The Leap Motion Controller represents a new and important improvement in consumer finger/object and gesture tracking input technology. The device has been made available to the public in summer 2013, more as a prototype than a final product. Therefore until now only a few applications have been developed with optimizations for this type of user interaction and very few scientific works have been published. The device has the dimensions of a USB stick (80 mm length and 13 mm width) and allows for precise and fluid tracking of multiple hands, fingers or small objects in free space.

There are very few details known about the Leap Motion Controller's inner structure and its basic operational properties [8]. One clear fact is that it uses infrared imaging for object tracking, containing two cameras and three infrared LEDs to improve lighting conditions. Figure 1 shows an image of the controller's hardware setup.

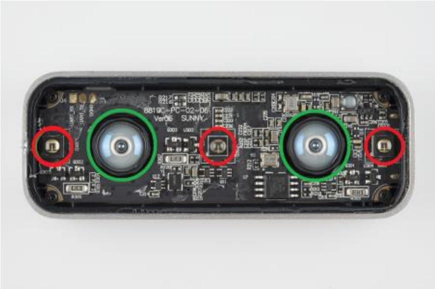

Figure 1. The 3 infra-red sensors are marked with red, while the 2 cameras are marked with green

According to the manufacturer, the sensors accuracy in position detection is about 0.01mm. However, recent research [6], [13] has shown that an accuracy of roughly 0.02mm can be obtained in realistic scenarios. In order to get objects positions from the stereovision images, all the calculations are performed on the host computer. This can result in a high machine load: about 20% processor usage for Intel Core I3, 2.4GHz laptop [12]. The gestures have an almost instant response time because the controller detects any movement with 300 frames per second rate. The data is transmitted through USB cable to the computer, which process and parses it into objects.

The developers of the Leap Motion Controller have released until now SDKs only for Windows and Macintosh platforms. Therefore a computer is needed to process all the data received from the controller and pass the results to the smartphone. An Android SDK, which is currently under development, would streamline the application development and would significantly improve the user experience.

Google Cardboard

As briefly mentioned before, Google Cardboard is a Google project to make an affordable, cardboard made, Virtual Reality headset that uses a smart phone as the screen. It is intended as a low cost system to encourage interest and development in VR. The reason why we chose to use this headset is because of its simplicity, being a do-it-yourself project, which means you can built your own cardboard using the plans from the official website. A Google Cardboard HMD is built using the following parts: a piece of cardboard cut into a precise shape, two 40 mm focal distance lenses and a ring-magnet combo for triggering actions. The lenses are set back from where the phone is placed at a precise distance so that they can each observe only half of the smart phone screen.

How does Google Cardboard work?

In [7] it is explained that: “Most human beings use what is known as binocular vision to perceive depth and see the world in 3D. The binocular vision system relies on the fact that we have two eyes, which are approximately 3 inches apart. This separation causes each eye to see the world from a slightly different perspective. The brain fuses these two views together. It understands the differences and uses them to calculate distance creating our sense of depth and ability to gauge distance.”

The assembly works by presenting two slightly different images, one to each eye. The application that runs on the phone is splitting the screen in two and is displaying the same view twice, but with a slight translation on one of the sides. Because each of the users’ eyes is looking at a slightly different image than the other, the 3D effect is created. Furthermore, as the rest of the view field is blocked by the cardboard, the environment visualized by the user becomes immersive.

The sense of immersion in this Virtual Reality is further improved through head tracking technology that uses phone’s sensors to detect any head movement and adjust the view accordingly. The accelerometer and gyroscope provide all the necessary data for the head tracking effect, so the user can freely look around and have that virtual reality experience.

In the initial Cardboard design, basic user interaction is provided using the magnet placed on the side of the cardboard, which acts as a click or a press. Using the magnetometer, the phone is able to sense magnetic changes caused by the magnet movement, and can transmit the input further to the application. However, this type of interaction is very limited and enables the user to only confirm or indicate a specific element. All the possible actions must be therefore described through two states controls (free state / selected).

Smartphone

Our pilot application has been tested on 5 different smartphones, namely: Samsung Galaxy Note 2, Sony XPeria Z1, HTC One M7, LG G2 and Samsung Galaxy Note 4. The application was developed to support versions of Android of 2.3 or greater.

Samsung Galaxy Note 2 has Mali-400MP GPU, which is pretty old piece of technology, compared to the other smartphones mentioned, which all have incorporated at least Adreno 320 GPU. These differences are visible in terms of performance, because the GPU is the responsible hardware for 3D processing.

Also, the screen resolution plays an important role in terms of image quality. Again, Galaxy Note 2, with a screen resolution of 720 x 1280, rendered poor images compared to the other smartphones.

Why is the screen resolution so important? Because the two lenses inside the Google Cardboard act as magnifiers, thus making the pixels visible for lower screen resolutions. Samsung Galaxy Note 4 produced outstanding results, with a resolution of 1440 x 2560 pixels with no visible pixel looking through lenses. But there is a trade-off between image quality, screen resolution and application performance: while you do get super quality images using quad-hd screens, the performance and responsiveness of the application is affected. The best overall results were obtained with the mid-range smartphones, with a resolution of 1080 x 1920 pixels, for which we recorder a good balance between image quality and performance.

CASE STUDY

As virtual reality and hand tracking are both emerging technologies, creating a natural user experience through their combination is not an easy process. In real life it comes natural to any human being to use the hands in order to interact with and control objects. The ability to use identical actions in virtual worlds, avoiding gestures that don’t make sense in real environments, would make the interaction as intuitively as possible.

The application that we developed tries to demonstrate this new way of interaction with a computer by taking advantage of the two affordable devices: Google Cardboard and Leap Motion Controller, in the context of a virtual tour of a zoo. The application also has educational purposes, providing an active experience for children who want to learn more about animals in an interesting and fun way. Three different ways of interaction with the environment have been implemented: using a Gamepad (Moga Controller), using Google Cardboard’s magnet and through the Leap Motion Controller.

In a VR experience, where your orientation is provided by the movement of your head, we found out that there is no place for "clickable" devices, such as the game controller, keyboard or a mouse. These devices are not capable of providing an immersive experience, as there is no genuineness in interaction with the scene. Because the orientation in the scene is controlled through the movement of users' head there is no other use for the previously mentioned devices than a "final" action over any virtual object. Google Cardboard's magnet makes sense only for interaction within a menu, whereas for any real-world object that has an equivalent in the virtual environment the Leap Motion Controller emulates perfectly. This is the main reason why we concentrated more on the interaction through this device.

The lack of an Android SDK not only limits the experience because the Leap Motion Controller cannot be placed on the head mounted display, but also makes a computer absolutely necessary in order to be able to use the data from the controller. Beside this hardware necessity, another software responsible for forwarding the data from the controller to the application is needed.

For generating the virtual world and manipulating its objects, the Unity IDE has been used along with its Game Engine, which provides a lot of useful built in features such as the physics engine. For head tracking capabilities we have made use of the Durovis Dive SDK [3]. This functionality could have been easily implemented using the raw data from the smartphone's gyroscope, or from its accelerometer, but the reason why this plugin was used is because it implements certain algorithms to reduce the motion sickness.

The hand tracking capabilities, on the other side, were implemented using the Leap Motion Controller SDK [10]. Trinus Gyre was used to glue together the data coming from application, controller and smartphone's sensors and project it on the phone's screen.

Figure 2 shows the menu scene that is opened when the application is launched. The image was taken in Unity, the upper part representing the perspective view of the scene, visible from the Editor. The interactive objects are the menu items, with the camera objects (there are two cameras, for left and right eye, even though from the image appears to be only one) providing orientation for them. Another object that can be seen is the leap motion controller, which obviously is placed in front of the cameras. The controller and the cameras are only visible in the Editor to give the developers a preview about their positions when the application will actually be run.

The bottom part of the image is what the user actually sees. Again, there are two perspectives (one for each eye) and with a closer look it can be seen that the images are slightly distorted providing some differences between them, so that the depth can be perceived. The interaction with the menu consists of two simple gestures that are combined into one, namely *KeyTap* and *Push*. These two gestures can be viewed in Figure 3.

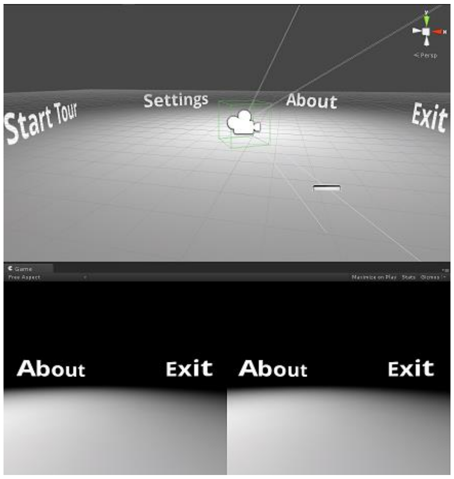

Figure 2. Menu Scene

A *KeyTap* is recognized when the tip of the index finger rotates down toward the palm and then springs back to approximately the original position, as if tapping. The tapping finger must pause briefly before beginning the tap. A *Push* gesture is recognized when your hand, while in the field of view of Leap Motion Controller, reaches the necessary downwards velocity (the hand needs to be parallel to the controller).

Because the SDK lacks a production level implementation, the tracking has some errors now and then. Even a gesture as simple as KeyTap is not recognized all the time. To improve the interaction, we decided that the selection of an item is done using a combination of KeyTap and a Push gesture, this way increasing drastically the chances that the selection will be recognized. The Push gesture was configured so that the minimum length of the hand movement is fairly small, and thus the combination of the two gesture results in a single one, very similar to KeyTap. The selected item is the one that the user is looking at when the gesture is performed (there is a small dot that indicates a ray projected from the center of the screen).

When the user selects "Start Tour", the main scene is loaded. After the scene is loaded, the character controlled by the user is placed outside the zoo, where he/she is able to see the available interaction gestures. All the gestures available in this scene can be observed in Figure 3.

In our virtual environment the walking action is achieved using the **Point** gesture (Figure 3). When the user points with one finger above the controller, and as long as that gesture is performed, the character keeps moving in the direction that the user is looking at. To stop the character from moving, one either removes the hand from the Leap Motion's field of view, or simply closes the fist.

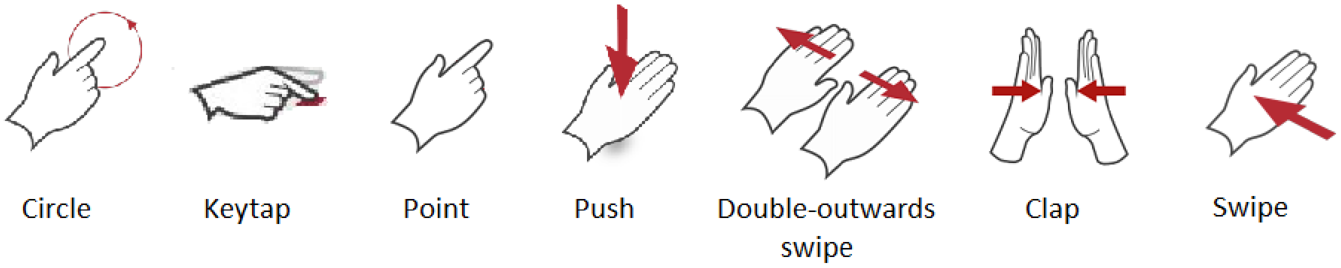

Figure 3. Interaction gestures available in the main scene

In order to enter the zoo, two actions need to be performed. The first one represents pressing a button, for which the system responds with an audio playing that gives information about the second action needed to be performed, that is opening the doors. This is accomplished using the *Double Outwards Swipe* gesture (Figure 3): the hands are placed initially above the controller, with the palms parallel to it and with their thumbs touching, and then perform a swipe with each hand in opposite directions. The character has to be placed in front of the doors for them to actually open, as it would make no sense to be able to open the doors from a far distance (see Figure 4).

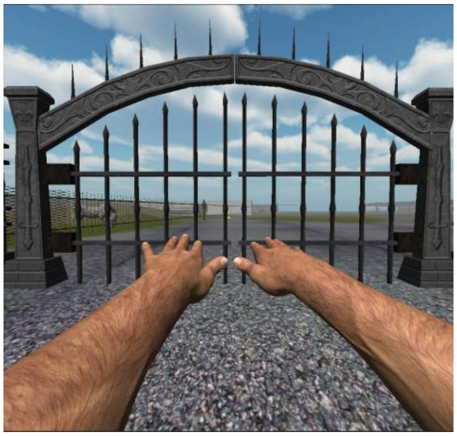

Figure 4. Single eye's view during the double outwards swipe gesture

The size of the virtual world is pretty big, because we used a 1:1 scale to make virtual hands and objects look as realistic and natural as possible. Because of this, the animals, which are able to move inside their cages, can get sometimes pretty far away from the fence. In order to “call” an animal to get a closer look of it, you need to look at it and perform a *Clap* gesture (Figure 3).

As the name of the gesture suggests, you have to clap your hands once and the animal will start walking towards you until it reaches the fence. From there, you can perform a *Circle* gesture (Figure 3) to get details about the animal. A screen appears in front of the camera with all the information, which you can dismiss by performing a *Swipe* gesture.

One of the biggest problems in VR is simulator sickness, which is caused by a conflict between different sensory inputs (ear, visual field and body position). Being able to control movement reduces the experience of motion sickness. The display responds to user’s movements all the time, even in menus. This is very important because, as in real world, you need to be able to change your orientation any time, regardless of the situation. Also, the virtual cameras rotate in a manner consistent with head and body movements.

Interaction challenges using Leap Motion in VR environments

Because the Leap Motion Controller is set on the table, it was virtually attached to the cameras so that the user is conscious about hands position with respect to the controller, so wherever the user is looking he/she is able to see his\her hands. If the controller would have been placed on the head mounted display, it would have made more sense to attach the controller to the “body” of the character, because our solution does not represent the way human body works.

While the software is constantly getting better at tracking potential hand poses, some will track better than others. In order to achieve the most reliable VR tracking pose, the users need to keep their fingers splayed and hands perpendicular to the Leap Motion’s field of view. Many development recommendations can be found in “Leap Motion VR Best Practices Guidelines” [11].

CONCLUSION

Virtual Reality environments allow the development of complex and very detailed learning scenarios that can be experienced by the users in a very similar way to the real world. Through this approach, harmful and dangerous situations or hard to create conditions can be thoroughly analyzed as many times as necessary, without endangering the user.

One of the main limitations in the spread of this technology in educational domain is related to the high costs usually involved in developing a VR system. Google Cardboard represents an affordable and very promising virtual reality experience for the masses. Although it is a “do it yourself” headset, the results obtained are very impressive and there are no changes required in the architecture of the mobile device. Any smart phone on the market, with decent resolution and graphical acceleration capabilities, can be transformed in a virtual display. Creation of a VR application is also an easy to do process if you're using an existing engine, such as Unity3D. The most important part for generating the depth sensation is the use of two cameras that present a slightly different image to each of the user's eyes. Another important step to achieve virtual reality is to enable the head tracking so that the user is immersed into the experience.

Until recently, most of the input devices were unfit to be used in immersive virtual worlds, feeling highly unnatural to the users. Leap Motion is a possible solution to this problem, but for now it feels more like a prototype. Even though the hardware setup seems very robust, it lacks a good software implementation for hands tracking. At the moment the device seems to be more intended for developers than for general public. The feedback provided by the users of our application leads to only one conclusion: for the interaction with the scene through Leap Motion, the functionality is only obvious after playing around with the device for some time.

In [1] Baudel and Beaudouin-Lafon explored the limitations of the gesture based interaction systems. One of the key findings was that gestural communication used more muscular activity than simple keyboard interaction, mouse interaction or speech. As a consequence, in order to avoid fatigue in users, the gestures must be chosen so they require minimal effort, be concise and fast to describe. Most of the times, the long use of this input approaches may induce fatigue in the user [2].

One other drawback we noticed is related to Leap Motion Controller and the inability to place the device directly on the head mounted display. As already mentioned, this limitation occurred because the only supported platforms for SDK are Windows and Mac. Due to this constraint the user experience is limited, he/she not being aware of his hands position relative to the controller. This also implies that a computer is required to receive the data from the leap motion controller and forward it to the application running on the smart phone.

REFERENCES

1. Baudel, T., Beaudouin-Lafon, M. Charade: Remote control of objects using free-hand gestures. *Communications of the ACM - Special issue on computer augmented environments: back to the real world*, 36 (7), 1993, pp. 28-35
2. Christova, P., Kossev, A., Kristev, I., Chichov, V.: Surface EMG recorded bybranched electrodes during sustained muscle activity. *J. Electromyogr Kinesiol* 9, 263–276 (1999)
3. Durovis Dive SDK : Available online: https://www.durovis.com/sdk.html
4. Google Cardboard. Available online : https://www.google.com/get/cardboard/ (accessed on 19/05/2015)
5. Google Maps. Available online : http://en.wikipedia.org/wiki/Google_Maps
6. Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., Sodnik, J. *An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking*. Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, Ljubljana 1000, Slovenia
7. Gurman. J.B. STEREO Home Page http://stereo.gsfc.nasa.gov/classroom/how.shtml
8. Jakus, G., Guna, J., Tomazic, S., Sodnik, J. Evaluation of Leap Motion Controller with a High Precision Optical Tracking System, *Human-Computer Interaction Advanced Interaction, Modalities, and Techniques*, 16th International Conference, HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014, Proceedings, Part II, pp.
9. Leap Motion App Store. Available online : https://apps.leapmotion.com/categories/vr-beta
10. Leap Motion Controller. Available online: https://www.leapmotion.com (accessed on 19/05/2015)
11. Leap Motion VR Best Practices Guidelines. Available online: https://developer.leapmotion.com/assets/Leap%20Motion%20VR%20Best%20Practices%20Guidelines.pdf
12. Leitão, P.M.O. Analysis and Evaluation of Gesture Recognition using Leap Motion, *Proceedings of the 10th Doctoral Symposium in Informatics Engineering - DSIE'15*, 2015, pp. 83 – 94
13. Weichert, F.; Bachmann, D.; Rudak, B.; Fisseler, D. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors 2013*, 13, pp. 6380–6393.
14. Winner, L. Do artifacts have politics? In *The Social Shaping of Technology* (2nd. ed.), Donald MacKenzie and Judy Wajcman (eds.). Open University Press, Buckingham, UK, 28-40.