

# Operating Virtual Panels with Hand Gestures in Immersive VR Games

## Experiences with the Leap Motion Controller

Yin Zhang and Oscar Meruvia-Pastor<sup>(✉)</sup>

Department of Computer Science, Memorial University of Newfoundland,  
St Johns, NL A1B 3X7, Canada

oscar@mun.ca

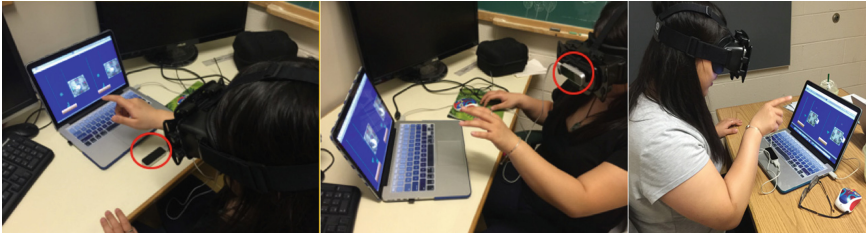
<http://www.cs.mun.ca/~omeruvia/research>

**Abstract.** Portable depth-sensing cameras allow users to control interfaces using hand gestures at a short range from the camera. These technologies are being combined with virtual reality (VR) headsets to produce immersive VR experiences that respond more naturally to user actions. In this research, we explore gesture-based interaction in immersive VR games by using the Unity game engine, the LeapMotion sensor, a laptop, a smartphone, and the Freefly VR headset. By avoiding Android deployment, this novel setup allowed for fast prototyping and testing of different ideas for immersive VR interaction, at an affordable cost. We implemented a system that allows users to play a game in a virtual world and compared placements of the leap motion sensor on the desk and on the headset. In this experimental setup, users interacted with a numeric dial panel and then played a Tetris game inside the VR environment by pressing the buttons of a virtual panel. The results suggest that, although the tracking quality of the Leap Motion sensor was rather limited when used in the head-mounted setup for pointing and selection tasks, its performance was much better in the desk-mounted setup, providing a novel platform for research and rapid application development.

## 1 Introduction

Virtual Reality (VR) gaming platforms have become quite popular, with VR headsets such as the Oculus Rift [22], HTC Vive [15], PlayStation VR [25] and Samsung Gear VR [23] being released recently. In spite of their novelty, these devices are not part of the first wave of attempts to bring VR into the consumer market. Back in 1995, the Virtual Boy [8], released by Nintendo, stands out as one of the exemplars of the first wave of commercial virtual reality devices, which did not manage to establish a permanent market for the technology. Some of the reasons for the failure of earlier waves to establish permanent markets include a lack of usability, reduced user adoption, and the high cost of earlier VR technologies.

The most recent rising of commercially available Virtual Reality owes credit in part to Palmer Luckey [7] who founded Oculus back in 2012.



**Fig. 1.** Experimental setup for operation of virtual panels through hand gestures.

His Oculus VR headsets set off a new wave of interest in Virtual Reality which continues until today. In the meantime, Apple [29], Samsung and HTC helped popularize the modern smartphone with high-resolution screens, accurate motion (and other) sensors and compact form factors that made them an appropriate fit to the technology required to make convincing VR.

As Google Cardboard [10] first illustrated in 2014 and later Samsung with the Gear VR in 2015, a low-cost, affordable and more casual form of VR using Smartphones is possible. By sliding a smartphone into a head-mounted display, users can get access into relatively simple virtual worlds. In addition to this, there was another significant development in the hardware that contributed to a more complete and affordable VR experience: portable, short-range depth sensing cameras [2]. Portable and short range depth sensors allow computing devices such as PCs, laptops and smartphones to capture its immediate environment as well as help identify a user's hands and/or face in the proximity of the devices [13, 21, 24, 28].

In this work, we use the Leap Motion as the depth sensing camera and as a gesture recognition device. As shown in Fig. 1, the Leap Motion controller is originally designed to be placed on a physical desktop, facing upward, or it can also be mounted on a VR headset for first-person interaction [1]. The device uses three LED emitters to illuminate the surrounding space with IR light, which is reflected back from the nearby objects and captured by two IR cameras [18]. The device's software installed on the connected PC will analyze the captured images in real-time, determine the position of objects, and perform the recognition of users' hands and fingers. Its suitability for a variety of tasks has been the focus of recent research [1, 3, 12, 27].

The motivation of this research is to explore VR gesture control with a low-cost combination of equipment, including a smartphone, an infrared controller and a VR headset, to address implementation challenges and propose solutions to overcome the major barriers to being able to control, with hand motions and gestures, a simple VR game. In this research, we tested our software with four users, who tested the system under the 4 conditions described below (2 hardware setups X 2 applications) and provided unstructured feedback on their experience. Through the implementation of this project, we show how a low cost solution to develop VR applications using a laptop, Unity, a VR Headset, an Android phone

and the Leap Motion controller can be used to address existing limitations and suggest solutions to usability issues in these platforms.

## 2 Implementation

### 2.1 Project Setup

This research is implemented in Unity 5.3 on OS X El Capitan system on a MacBook PRO. In addition, we have the Leap Motion SDK\_Mac\_2.3.1, Splashtop Streamer [26], as well as Leap Motion Core Asset 2.3\_1 [19] package installed in the Unity engine. Meanwhile, the Samsung S5 phone is equipped with the Splashtop Streamer client and is inserted in the Freefly VR [9] headset.

The Leap Motion SDK connects the Leap Motion controller to the computer, allowing the captured data from the Leap motion to be delivered to the analyzing software on the computer in real-time (see Fig. 2). The Leap Motion Core Asset package, which is imported into the Unity engine, will function as the connector between the Unity engine and the Leap Motion analyzing software, which translates the input data from the analyzing software as a control signal and generate the corresponding hand model and gestures. The Splashtop Streamer mirrors the PC screen onto any mobile devices with the Splashtop client installed and in the same network. This novel setup allows us to avoid Android depolyment, as the program executes directly from the laptop to the HMD. This makes development much faster than the process of deploying an executable for the HMD (an Android phone) each time the source code changes. A prototype in development can be tested using the actual HMD rather than on a software android simulator running on the laptop.

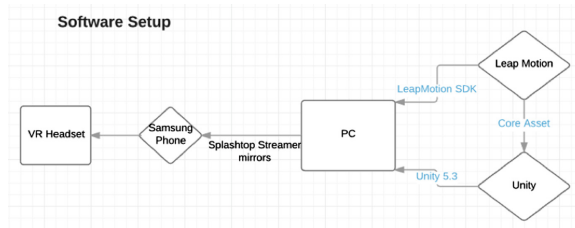


Fig. 2. Software setup flow chart.

As for the Hardware part, the Leap Motion sensor is always connected to the PC through the USB port, whereas the Samsung phone is wirelessly connected, mirroring the PC screen. The Samsung Phone is slid into the Free Fly VR headset and calibrated for horizontal and vertical axis with the help of the Free Fly VR SDK (see Fig. 3). In this research, we tested two conditions with regards to the location of the Leap motion controller, either horizontally placed on the desk with the device looking up (referred to as “desk-mounted”) or mounted on top of the VR headset facing away from the user’s head (or “head-mounted”).



**Fig. 3.** Hardware setup.

## 2.2 Applications

**Dial Panels.** To explore the experience of point touch accuracy using the Leap Motion gesture controller, we simulated the dial panel with 9 number buttons and two symbol buttons: # and \* (see Fig. 4). The button's color will change in response to users' actions upon detection of a collision with the toggle buttons of the panel. A box collider component placed on each button allows buttons to react to a virtual collision with the hand model generated by the Leap Motion SDK. Multiple canvas layers on each button work under the button color changing script, allowing it to change colour when it is picked by the hand model to provide a clear visual feedback.

Shadow casting has been previously found useful to localize objects in virtual environments [6, 11, 16]. In our system, a shadow of the controlling hand is cast along with the hand model on a panel plane beneath the buttons layer, to provide a sense of the distance between the hand and the panel through visual feedback.

During the implementation process, we found that the hand model generated sometimes did not accurately reflect the real world's hand pose or position. In addition, sometime multiple fingers would trigger simultaneous collisions on the panel. This was especially confusing for the user, as most users typically intended to push a button with just one of their fingers. For this reason, we modified the collision detection to be limited to the index finger, so only one finger can trigger the collision detection action.

**Games.** We also implemented the well-known Tetris game [20], which is displayed within the VR environment (see Fig. 8). The game is controlled by the user through a control panel which is placed besides the falling blocks. The user then moves his or her hand around and presses on the buttons that will control whether the falling block is displaced to the left or right, sped towards the bottom, or rotated around. For the implementation of the game control, we built a game control engine. The game control functions described above are triggered by detecting the collision between the buttons in the control panel and the hand model (see Fig. 5).

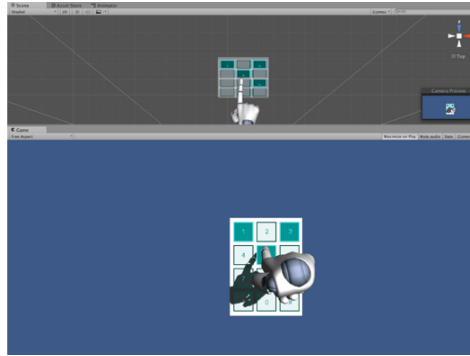


Fig. 4. Dial panel.

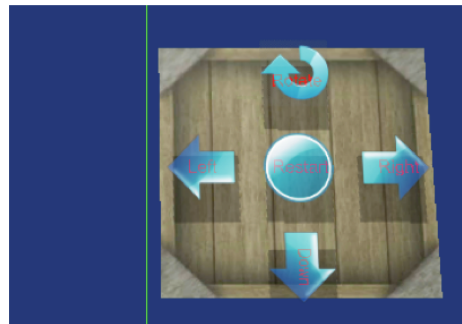


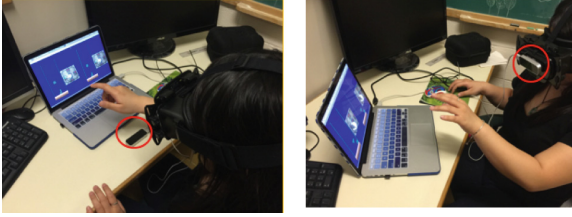
Fig. 5. Tetris control panel.

**Comparison Between Head-Mounted and Desk-Mounted Leap Motion Placement.** Both the implementation of the dial panel and the Tetris game are compared under head-mounted and desk-placed conditions. The leap motion is horizontally placed on the desk in the desk-placed condition, whereas it is mounted on to the top of the VR headset facing outside away from the user's head in the head-mounted situation (see Fig. 6). The complete setup with the four experimental conditions is shown in the accompanying video available at our website.

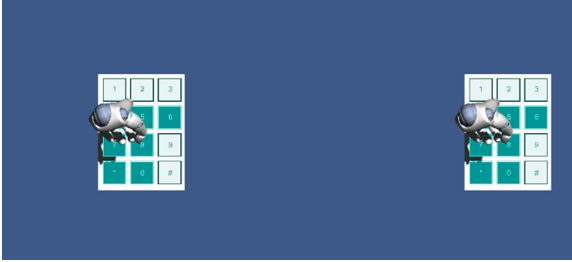
### 3 Results and Evaluation

When testing the implementation of the dial panel we obtained a stable result where users can predict the placement of the hand using the shadow effect and easily choose the button using the output information provided by the visual feedback (see Fig. 7).

As for the implementation of the Tetris game, the control panel can be difficult to operate during play. As a result of various evaluation sessions, we figured out several modifications on the gaming control panel regarding the size,



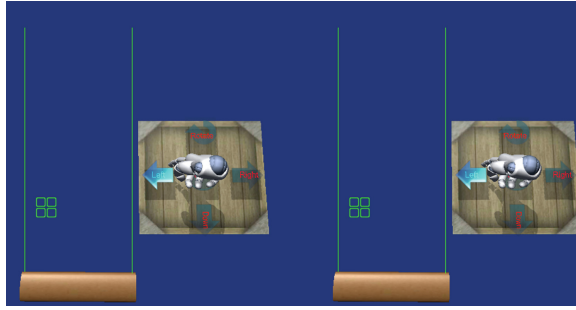
**Fig. 6.** Leap motion position comparison.



**Fig. 7.** Screen capture of dial panel simulation.

transparency, shadow effects and the visual feedback on collision, with the goal of improving the user experience. Eventually, the game's control panel became reachable and easier to control. However, the user's attention needed to correctly operate the panel was still causing some distraction when trying to control the blocks. Overall, the control panel did not feel as comfortable as using a regular keyboard, from the point of view of the gaming experience (see Fig. 8). This is in line with Barret and Krueger [4], who found that typing performance suffers when applying keystrokes on planar surfaces when compared to keystrokes on actual physical keys that displace upon typing. We hypothesize that the lack of haptic feedback on a planar virtual keyboard further reduces the user's performance.

With regards to the comparisons of the head-mounted versus desk-mounted conditions for the location of the Leap Motion controller, the head-mounted version more closely resembles the original setup of the Oculus Rift combined with the Leap Motion sensor, as used in most head-mounted setups [5, 14]. This eliminated the dependency between the user's hand and the placement of the sensor on the table or desk. However, we noticed that the Leap Motion controller did not behave as effectively as when placed on the desk and so the hand tracking was not as stable as in the desk-placed configuration. We hypothesize this is mainly because the user's hand orientation when pointing out was somewhat parallel to the line of sight of the Leap Motion controller, making it especially challenging for the internal Leap Motion algorithms to estimate the correct hand pose.



**Fig. 8.** Screen capture of tetris game.

We found that the lack of haptic feedback in the prototype developed caused some sensory confusion. It was hard for users to figure out the depth of the panels containing the buttons to be clicked. Through the whole implementation of this research, we found out that achieving stable tracking and making sure the control panel was reachable and usable were the most challenging aspects of this work.

### 3.1 Motion Sensor Limitations

It has been reported that the Leap Motion controller does not perform in a way that can be reliably used as a professional tracking system [12] and that its performance as an input device for everyday generic computer pointing tasks is rather limited when compared with standard input devices [3]. In addition, we have found that the current version of the Leap Motion controller and the Leap Motion SDK provides at times unsatisfactory tracking results, in part due to the following technical limitations:

- The LeapMotion SDK doesn't produce an RGBD depth map, so it limits the general tracking of objects in 3D and the general usability of the controller when compared to other depth-sensing controllers that provide an RGBD image, such as Intel's RealSense [17] and Occipital's Structure sensor [21].
- Other items within the neighborhood of the hand (such as a piece of paper behind the hand) can cause the system to lose track of the hand altogether.
- The SDK's ability to correctly estimate the hand pose seem to vary, depending on where the controller is placed (desk- vs head-mounted). In particular, the current version of the Leap Motion SDK produces reduced tracking quality in the head-mounted configuration, especially when the user is making a fist, with hand poses where the fingers are pointing towards the depth-direction and parallel to the device's line of sight, and when the user has a single finger extended with the rest as a fist.

## 4 Insights on Interaction Improvement

After the implementation of the applications mentioned above, we came up with several suggestions for improving the interaction:

- Providing visual feedback by means of projecting a shadow of the hand model on the control panels made a positive contribution for users trying to assess how far to move their hand to press on the buttons of the panels. This was even more relevant the first time users were interacting with the system, as it was the time when they were forming a mental map of the relationship between their own hand movements and the representation of the hand model shown within the VR environment.
- A more natural choice for an application like the Tetris game would be the swipe gesture, where the blocks are displaced to the left or to the right by a physical waving of the hand within the VR scene.
- The lack of haptic feedback was disconcerting for most users, since it was easy for the hand controller to pass through the button and past the back of the virtual panel, making it hard to know when the button was clicked. For this reason, an additional controller, such as a joystick, or the hand-held button clicker device provided by the Freefly VR, might be better able to indicate the button click operation rather than using a pressing motion of the fingers to indicate a button click.
- Providing auditory feedback by having the users hear a sound when a button is clicked would help users to become certain that the button click operation was successfully registered.
- Current technologies can already bring users to experience an immersive VR experience at a low cost, but more reliable gesture controllers and better depth-sensing SDK's are still needed.

## 5 Conclusions

The built-in Leap Motion hand detection system presents some limitations in estimating the pose of a user's hand in the real world, but it still constitutes a workable device for rapid prototyping of real world gaming environment conditions, particularly when the controller is placed on the desk. At times, the lack of stability of the controller prevented the users from concentrating on playing the game itself. The low-cost smartphone used as a VR display, with the support from a simple and lightweight VR headset and a desk-placed depth sensor constitutes a convenient and affordable method for VR development for people who want to try the technology at early stages of development. Any game or similarly suitable type of application can be directly mirrored from the laptop to the proposed configuration, allowing for fast development of prototypes for research and evaluation of new ideas. We plan to expand this work with a user study where we compare the performance outcomes of people using the Leap Motion in the desk-mounted setup versus the head-mounted configuration across a variety of tasks within the VR environment.



## References

1. Adhikarla, V.K., Sodnik, J., Szolgay, P., Jakus, G.: Exploring direct 3d interaction for full horizontal parallax light field displays using leap motion controller. *Sensors* **15**(4), 8642–8663 (2015)
2. Arthur, K.W.: Effects of field of view on performance with head-mounted displays. Ph.D. thesis, University of North Carolina at Chapel Hill (2000)
3. Bachmann, D., Weichert, F., Rinke, G.: Evaluation of the leap motion controller as a new contact-free pointing device. *Sensors* **15**(1), 214 (2015). <http://www.mdpi.com/1424-8220/15/1/214>
4. Barret, J., Krueger, H.: Performance effects of reduced proprioceptive feedback on touch typists and casual users in a typing task. *Behav. Inf. Technol.* **13**(6), 373–381 (1994). <http://dx.doi.org/10.1080/01449299408914618>
5. Beattie, N., Horan, B., McKenzie, S.: Taking the leap with the oculus HMD and CAD - plucking at thin air? In: Proceedings of The 1st International Design Technology Conference, DESTTECH2015, vol. 20, pp. 149–154. Elsevier (2015). <http://www.sciencedirect.com/science/article/pii/S2212017315002029>
6. Besacier, G., Tournet, J., Goyal, N., Cento, F., Scott, S.D.: Object and arm shadows: Visual feedback for cross device transfer. In: CHI 2014 Extended Abstracts on Human Factors in Computing Systems, pp. 463–466, CHI EA 2014. ACM Press, New York (2014) <http://doi.acm.org/10.1145/2559206.2574832>
7. Clark, T.: How Palmer Luckey created Oculus rift, November 2014. <http://www.smithsonianmag.com/innovation/how-palmer-luckey-created-oculus-rift-180953049/?no-ist/>
8. Edward, B.: Unraveling the enigma of Nintendo's virtual boy, 20 years later, August 2015. <http://www.fastcompany.com/3050016/unraveling-the-enigma-of-nintendos-virtual-boy-20-years-later/>
9. FreeflyVR: Freefly VR: how does it work, the complete guide (2014). <https://www.freeflyvr.com/freefly-vr-how-it-works/>
10. Google Inc.: Google cardboard Google VR (2016). <https://www.google.com/get/cardboard/>
11. Greene, E.: Augmenting visual feedback using sensory substitution. Master's thesis, University of Waterloo (2011). [https://uwspace.uwaterloo.ca/bitstream/handle/10012/6161/Greene\\_Eugene.pdf](https://uwspace.uwaterloo.ca/bitstream/handle/10012/6161/Greene_Eugene.pdf)
12. Guna, J., Jakus, G., Poganik, M., Tomai, S., Sodnik, J.: An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors* **14**(2), 3702 (2014). <http://www.mdpi.com/1424-8220/14/2/3702>
13. Higuchi, M., Komuro, T.: [Paper] robust finger tracking for gesture control of mobile devices using contour and interior information of a finger. *ITE Trans. Media Technol. Appl.* **1**(3), 226–236 (2013)
14. Hilfert, T., König, M.: Low-cost virtual reality environment for engineering and construction. *Vis. Eng.* **4**(1), 1–18 (2016). <http://dx.doi.org/10.1186/s40327-015-0031-5>
15. HTC Corporation: Vive - Home (2016). <https://www.htcvive.com/ca/>
16. Hu, H.H., Gooch, A.A., Thompson, W.B., Smits, B.E., Rieser, J.J., Shirley, P.: Visual cues for imminent object contact in realistic virtual environment. In: Proceedings of the Conference on Visualization 2000, pp. 179–185, VIS 2000. IEEE Computer Society Press, Los Alamitos (2000). <http://dl.acm.org/citation.cfm?id=375213.375238>

17. Intel Corp.: Intel RealSense Technology. <http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-shortrange.html> (2016)
18. Leap Motion Inc.: How Does the Leap Motion Controller Work? (2016). <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>
19. Leap Motion Inc: Leap motion developers (2016). <https://developer.leapmotion.com/unity/>
20. Noobtuts.com: Unity 2D Tetris Tutorial (2012). <http://noobtuts.com/unity/2d-tetris-game/>
21. Occipital, Inc.: Structure Sensor - 3D scanning, augmented reality, and more for mobile devices (2016). <http://structure.io/>
22. OculusVR: Oculus (2016). <https://www.oculus.com/en-us/>
23. Samsung Corporation: Samsung gear VR - the official Samsung galaxy site (2016). <http://www.samsung.com/global/galaxy/wearables/gear-vr/>
24. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., Izadi, S.: Accurate, robust, and flexible real-time hand tracking. In: Proceedings of CHI, the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM Press, April 2015. <http://research.microsoft.com/apps/pubs/default.aspx?id=238453>
25. Sony Interactive Entertainment: PlayStation VR (2016). <https://www.playstation.com/en-ca/explore/playstation-vr/>
26. Splashtop Inc.: Top-performing remote desktop and remote support (2016). <http://www.splashtop.com/>
27. Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the accuracy and robustness of the leap motion controller. *Sensors* **13**(5), 6380 (2013). <http://www.mdpi.com/1424-8220/13/5/6380>
28. Welch, G., Bishop, G.: Scaat: Incremental tracking with incomplete information. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 333–344. ACM Press/Addison-Wesley Publishing Co. (1997)
29. Whitwam, R.: How Steve Jobs killed the stylus and made smartphones usable, October 2011. <http://www.extremetech.com/computing/98923-how-steve-jobs-killed-the-stylus-and-made-smartphones-usable/>