Usability Study of Leap Motion Controller

Alberto Scicali[1] and Hans-Peter Bischof[1 2]
[1] *Computer Science Department, Rochester Institute for Technology, 102 Lomb Memorial Dr., Rochester, NY 14623, USA*
[2] *Center for Computational Relativity and Gravitation, Rochester Institute for Technology, 102 Lomb Memorial Dr., Rochester, NY 14623, USA*
{afs2842, hpb}@cs.rit.edu

Keywords: Usability Study, 3-Dimensional Input Devices, Unity

Abstract: The qwerty keyboard and a mouse are still the main input devices used today. The last upgraded mouse was introduced by Apple in 2009; the Magic Mouse. The Magic Mouse is a multi touch mouse and can be used to recognize simple gestures. Computer games and visualization environments are typically played out in a 3D world projected on a 2D screen, or a 3D environment. A 2D mouse is not very useful in a 3D environment. On July 2013 Leap Motion released a device which can be used as a 3 dimensional mouse. This paper presents the usability experiments and first results.

1 INTRODUCTION

A long time go a keyboard was the only mechanism to communicate with computer programs. Operations we have no problem to perform today, such as Cut and Paste operations, took a while learn if you have been using an editor like vi(William Joy, 1977). Programming tools, like the vi, was written for developers by developers and a perfect tool for creating code, but a terrible tool for any non-developer.

The introduction of graphical computer interfaces changed the way in which we interact with computers completely, especially after the mouse came along. The usage of a computer became so much easier for most users after the release of Microsofts mouse compatible version of Word. The next milestone came in 1984 when Apple released Macintosh 128 with a update version of the Lisa mouse (Pang, 2002).

The world of interaction with a computer has not changed for the next 25 years or so, because there was apparently no need for it. The qwerty keyboard layout, introduced to the market 1874, is the most common keyboard up till today, despite the fact the reason for this layout, jamming keys, does not exist anymore. Tetris could have been played with a keyboard, but games in a 3D world required something more sophisticated, such as console game controllers. However, moving freely in 3D space was a challenge, even when using a game controller.

The Kinect(Zhang, 2012) was another game changer. The Kinect allowed interaction with a game without a game controller, or to better describe it, the human body and voice are used as the game controller. The Kinect uses a motion sensing device and voice recognition system to interact with games.

The next device, which became available using sensing technology was the LeapMotion Device (Weichert et al., 2013). The LeapMotion is an inexpensive optical sensor capable of recognizing position of finger tips, palm vectors up to 0.2mm accuracy. This device allows to study gesture recognition algorithms in a detailed way.

This paper discusses scenarios in which these motion sensing devices outperform a keyboard, or a mouse, and also discusses their limitations.

2 PROBLEM DESCRIPTION

The inputs coming from a mouse or a keyboard is discrete and do not need any or very limited interpretations. A mouse down event is a discrete event at a given position on the screen and carries depending on the environment information about the position of mouse pointer, time of the click, etc. A double, or a triple click is in event over time, and will only count as double/triple click if the two events happen in a defined time window. A mouse can move on a limited 2D space and the basic functionality has not changed for a long time. Up until Apples Magic Mouse(Loyola, 2010) opened the window to 2D gestures a bit; swipe between pages or full screen applications and a double tap which gives access to mission control.

A mouse is a appropriate input device, if a correlation between between the action and reaction can be created. Tetris(Burgiel, 1997), the 2D tile matching puzzle game, can easily played with a keyboard. Rotating of a tile, move to the left, right, and down are the only needed operations to play this game, and these operation can easily be mapped to keys. A keyboard or a mouse is a less then optimal input device, if you want to play table tennis in a 3D world, because many parameters must be controlled at the same time. The position of the racket, orientation, speed and direction must be controlled at the same time. It is simply to difficult to map these analog signal to discrete events like a click, and movement of a mouse on a 2D surface. On the other side, a hand can be easily moved in 3D space to control a racket in the desired way.

The Leap Motion(Weichert et al., 2013) device is motion sensing device for hands. It senses objects near by like the hands.

This paper discusses scenarios in which a motion sensing devices outperform a keyboard, or a mouse, and also discusses their limitations.

3 LEAP MOTION DEVICE

The LeapMotion device is a very small and inexpensive usb device. It tracks the position of objects in a space roughly the size of the top half of beach ball. The precision of what the sensors is up to 0.01 mm. This means the sensor can identify 700,000,000 points in its view area, which is extremely precise.

The Leap Motion device uses three infrared camera to detect the reflection of light from users hands and arms. It also senses in on direction, therefore it can not see objects which are behind each other. The SDK V2 introduced a detailed skeleton model which allows a better prediction of the position of objects, like hidden fingers. The API supports access to the stream of frames or can be used to detect predefined gestures. We use both for levels for the implementation of the software.

4 EXPERIMENTS

4.1 Visualization

Visualization of data comes is the art of creating meaningful images, video, or audio out of a n-dimensional data set(Bischof and Dong, 2011). The

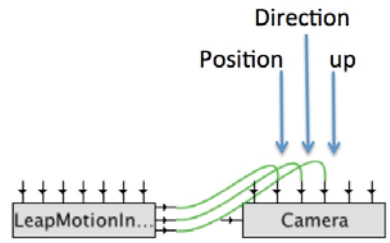

Figure 1: Leap Motion used as an Input Device

process includes typical a bit of experimenting with the attributes of the lights like positions, color, and directions etc. The position, direction, and up vector of the viewing point camera - and other parameters require typically some experimenting in order to create a meaning full visualization.

We used the Leap Motion device just to control the camera. Most visualization systems are using a 3D space, which means that camera, lights must be moved along the x, y, z axis. Defining the values of the coordinates with sliders, or defining the values in a text field typically does this. This is a cumbersome and not a very explorative undertaking. With the Leap Motion Device we can take the virtual camera and position it space and focus the camera at a particular position. We implemented this and used the left hands palm position and direction to arrange the camera in space. The up position was determined with the palm direction of the right hand. The Spiegel programming framework is easily extensible and we added just one component to include the Leap Motion device, see Figure 1. The LeapMotion component sends out new values along isa output channels if any of the value changes and the camera component reads the new value and applies the value to is state.

4.1.1 Observations

It was relatively easy to control the camera with hand, it got a bit more difficult to use two hands for the complete control of the camera. After a while it felt like hard work to keep both hands up and moving then them around.

The Leap Motion device sends information as soon as it detects anything, which at the beginning is more or less noise. We used a Finite State Machine(Brand and Zafiropulo, 1983) to control when the device starts and ends sending information.

A camera might be used inside a cube of length 1, or in a cube of much greater length, like 1000. We employed also a FSM to determinate a scalar value which mapped to our rather limited hand motion range.

The use of system was no far away from easy to use. To many things needed to be remembered and most people using the system have a very difficult time to move their hands independent form each other.

The Leap Motion device sends out up to 200 frames per second. The processing of 200 frames per second would be of little value, because the changes between each frame would be extremely minimal, because you can not move your hand in much1/200 of a second. In order to reduce the workload on the system we process only frames which are at least n seconds apart and n is variable. Experiments have shown that 1/10 second is a reasonable value.

4.2 Unity3D Games

The games created in the Unity3D game engine were intended to further exemplify the advantages of control that is offered with the Leap Motion device, especially when compared to a computer mouse. Each game was created with specific tasks to be completed that demonstrates the need for input devices that allows software to be controlled with 3-D controllers.

4.2.1 Interactive Globes

Two interactive globe applications were created in order to test varying methods of gesture controls with the Leap Motion device. One globe application, created with Java and jMonkey, is controlled with firm gesture controls that, once recognized, began traversing the globe; i.e. keeping your hands parallel and over 20cm would begin to move the camera away from the globe, effectively zooming out. The second globe application, created with JavaScript and the Google Earth plugin, implemented a complete one-to-one mapping between a users hand movements and how the camera moved around the globe; when a user moves their hand along the XYZ axis the camera will too, i.e decreasing the hand's Y axis will zoom the camera closer to the globe, while increasing the hand's Y axis will move the camera away from the globe. Then, movement in the X axis controls longitudinal panning and Z axis movement pans along the latitude.

4.2.2 Observations

Through experimenting with over one hundred anonymous users during Imagine RIT 2014, it was discovered that the one-to-one globe was much more intuitive for users to understand the controls to, with the instant visual feedback between the movement of their hand and the panning movements of the globe camera, users could easily relate what motions of their

Figure 2: GoogleEarth with LeapMotion integration

hands controlled the camera. While the first globe application, using the firm gestures, was difficult for users to understand the controls. When gesture controls are limited to a firm gesture, with no active feedback, it becomes less intuitive for users. The way in which the Leap Motions controls are utilized is a primary component in allowing the Leap Motion and other 3-Dimensional input devices in becoming more pertinent for users.

4.2.3 Paddle Game

A paddle and ball game was created early on using jMonkeyEngine and Java along side the Leap Motion API to create unique three dimensional experience. The task of the paddle ball game was to continuously bounce a tennis ball into the air with a racket. The virtual racket can be controlled through all axis, X, Y and Z. This allowed experimenters to reach into the 3-Dimensional world; interacting with a game in a way that is not possible with a typical 2-dimensional mouse. This experiment was an early example of the advantages that the Leap Motion device provided over a mouse.

4.2.4 Swatting Game

The first Unity3D game is the simplest, where a ball must be swatted into a goal using a 3-D hand model that is controlled and mapped to the user's hand. This game uses the X, Y and Z axis. A typical computer mouse only allows movement in two dimensions, by creating a game that allows a user to control objects in 3-dimensions. The user is capable of swatting the ball vertically, side to side and towards the far end of the designated arena. Without the 3-Dimensional capabilities of the Leap Motion, a mouse is incapable of swatting the ball into the goal without extra controls being added. The Leap Motion allows full 3-dimensional input controls without only one device, this allows for a more intuitive control and a lower learning curve for the controls.

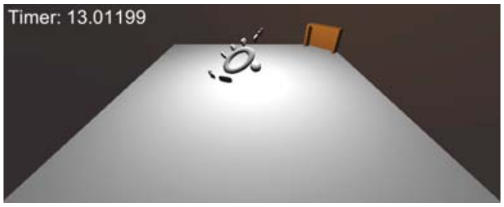

Figure 3: Swatting Unity3D Game

Figure 4: Labyrinth Unity3D Game

4.2.5 Observations

Without adding additional input controls to the game, a mouse isnt capable of effectively traversing the 3-D space. Using the LeapMotion, users were able to swiftly swat the ball into the goal. If the users were new to LeapMotion, they would take a little longer to adjust to the interaction between their hand and the controller. But the end result would always be with the user successfully completing the game.

4.2.6 Labyrinth Game

The Labyrinth game is a more difficult challenge, demanding users to navigate a sphere through a maze. The maze is directly connected to the hand, pitching and rolling as the user's hand does the same. This game tests the intuitive methods to control 3-Dimensional object by the rotation and pitch of a user's hand. As a hand rolls left or right and pitches forward and backwards, the maze does the same. Giving the user full control of the terrain and allowing gravity to pull the ball down the labyrinths corridors as the user tilts the floor. This game allows users to match the exact movements of their hand to the tilt-angle of the maze. A 2-Dimensional mouse would have to map out X and Y axis movement to the angular tilt, rather than directly using the user's hand. The absolute positions of the mouses Y and X axis controls the pitch and roll, respectively.

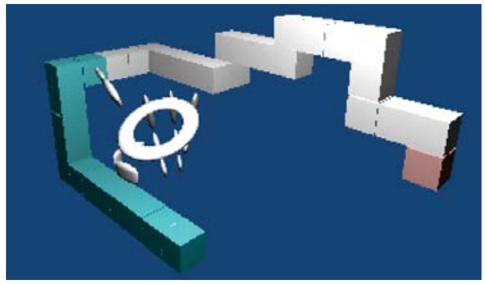

Figure 5: Trace Unity3D Game

4.2.7 Observations

This game was playable with both a mouse and the LeapMotion, however the when users used the LeapMotion the controls were much more intuitive; because of the direct 1-to-1 control of pitching and rolling their hand, the labyrinth mimicked their hand movement as expected. When users played the game with the mouse, it took more time to understand and effectively use the controls.

4.2.8 Trace Game

The final game, Trace, tasks the user to trace a 3-Dimensional path with a 3-Dimensional hand that acts as their own. This task forces the user to use all 3 dimensions to their fullest extent, as well as testing the accuracy of the Leap Motion device. A 2-D mouse is incapable of carrying out this task to completion, the mouse would have to accurately move a 3-Dimensional object along all 3 axis. To allow a 2-D computer mouse to complete the task, extra inputs would have to be added in order to allow the mouse to traverse the 3 axis; adding more inputs other than the mouse introduces more complications by forcing the user to use a combination of inputs, rather than just the mouse, or in the final case the LeapMotion. The Leap Motion device makes 3-Dimensional traversal of the path possible and accurate.

4.2.9 Observations

Users that played Tracer were able to successfully complete the task upon familiarizing themselves with the 3-D environment and the given 3-D Path. With the 1-to-1 mapping between the users hand and the rendered 3-D hand, users were able to smoothly control their hand movement with accuracy and trace the path. One noticeable negative was the inability for users to sense if they truly touched the path objective. When a block on the path is touched, the color changes to teal, however there is no limit on how far the hand/fingers can go through the block objects.

Two solutions to this issue would be to add more feedback, whether through auditory, tactile or more visuals, or to add collision properties between the 3-D hand and the blocks; this way the 3-D rendered hand model wont be able to move through the blocks.

5 RESULTS

With all experiments we have tried to create natural user interface to perform one task. We did not try to create a virtual drum set and ask a drummer if this could be used. we have been interested in the question if the Leap Motion Device could make it easier to perform a task in 3D space.

In all Unity3D game cases, the Leap Motion device was able to more achieve the game task. A computer is only able to be used fully in the labyrinth game, and that is with the necessity of remapping 2 axis to the tilt angle of the maze. The Leap Motion device becomes a more useful and familiar device for 3-Dimensional programs that need to access all 3 axis at a single time with a user's hand/s. There is simply no good way to control the movement of the mouse with the use of discrete events of a mouse or keyboard in an easier way.

The hundreds of users controlling the globe has shown that very simple gestures outperforming more powerful, but more complicated gestures. Most users could not perform to different movements with their hands in order to achieve a desired task.

The Spiegel experiment has shown that the Leap Motion is very useful in order to explore the visualization of a data set. It has also been shown that it is not as simple as find the palm position and direction and using it. FSM, start and end markers need to be used to be somewhat efficient. It is less precise than using exact coordinates, but precise enough to make it very useful.

6 CONCLUSION

Programs that are based in a 3-Dimensional environment are much more accessible with 3-D controllers such as the Leap Motion. A standard 2-Dimensional mouse will not be able to effectively manipulate and traverse the 3-Dimensional landscape without having to add extra inputs. The Leap Motion allows direct 1-to-1 mapping that gives users of varying skill the familiarity of using their own hand. However, these 3-Dimensional devices are only as useful and friendly to user as the user experience allows. If the controls to a program are created with hard-coded gestures that must be fulfilled within a certain degree, the learning curve to effectively using these devices increases. With the ability to access all 3-Dimensions, it is important to allow the controls to be intuitive to use and informative; giving feedback to the user so that they may quickly adapt their hand movements.

7 FUTURE WORK

Our experiments have strongly indicted that a 3 dimensional input device, like the Leap Motion Device, can be successfully used to control objects in a 3 or more dimensional space. It is not clear at this point which kind of gestures work well, or not so well. Future study will bring more clarity.

Additionally the previous work now leads to applying these techniques and uses to a real world, physical applications. The current goal is to map the Leap Motion device to a robotic arm so that it may be controlled with one-to-one hand movements. This could allow users to move a robotic arm with precision, over long distances.

We also will study how which aspects of a visualization environment can be successfully controlled by this kind of device. Under current consideration is lighting, paths of the viewpoints and lights, speed of the objects on their paths etc.

REFERENCES

Bischof, H.-P. and Dong, A. (2011). Directing a visualization ala kubrick. In *Proceedings of International Conference on Computer Graphics and Virtual Reality*. CSREA Press.

Brand, D. and Zafiropulo, P. (1983). On communicating finite-state machines. *J. ACM*, 30(2):323–342.

Burgiel, H. (1997). How to lose at tetris. *Mathematical Gazette*, 81:194–200.

Loyola, R. (2010). Apple's magic mouse offers multitouch features. Copyright - Copyright PC World Communications, Inc. Jan 2010; Document feature - Photographs; Last updated - 2012-07-19.

Pang, A. S.-K. (2002). The making of the mouse. *American Heritage of Invention and Technology*, 17(3):48–54.

Weichert, F., Bachmann, D., Rudak, B., and Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. *Sensors*, 13(5):6380–6393.

William Joy, M. H. L. (1977). An introduction to display editing with vian introduction to display editing with vi.

Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10.