Gesture-based User Interface Design for Static 3D Content Manipulation Using Leap Motion Controller

Naveed Ahmed

Department of Computer Science
University of Sharjah
Sharjah, United Arab Emirates
Email: nahmed@sharjah.ac.ae

***Abstract*—We present a new hand gesture-based user interface to efficiently manipulate static 3D content using the Leap Motion Controller. Our method uses intuitive gestures utilizing only the movement of the thumb and the index finger coupled with the hand rotation. These gestures not only allow the user to rotate the 3D content around the three axes but also scale the 3D content. We implement these gestures using the Leap Motion Controller and present the implementation details. We perform a comprehensive user study to demonstrate the effectiveness of our user interface in terms of both usability and user experience. The user study shows that a gesture-based user interface can be employed as a viable mechanism to manipulate 3D content and can be used in a number of applications.**

Keywords–Gestures; Leap Motion; Gesture-based User Interface; User Interface Design.

I. Introduction

The rapid change in technology has brought a number of ways to interact with software applications on a wide range of devices. Devices like Kinect, PS Move, Wii Remote brought motion controls to the gaming audience. Among these devices, Kinect [1] allows a complete passive motion capture solution without any direct physical interaction with the input hardware. Since, it can capture the motion of the person at real-time [2], Kinect has been widely employed to provide a gesture-based user interface to control a number of desktop applications [1]. It can detect high level arms and body motion, but it is not possible for it to capture fine grain hand movements up to the level of individual fingers.

With the advent of touch-based mobile devices, users are more comfortable with the use of hand gestures, especially finger-based multi touch interfaces, to control the applications. An interface that allows similar interactions on the desktop environment would be more efficient and user friendly. Any hand gesture control can also be easily employed in virtual and augmented reality applications. For a user, interacting with any application using a gesture controlled system is more natural, because it follows familiar user interface paradigm and unifies the user interaction across different platforms.

Incorporating gestures for user interaction first requires the detection of gestures. There has been a number of methods proposed to detect gestures using a number of devices. A number of methods have employed Hidden Markov Models to detect gestures [3] [4]. Other methods are proposed using AdaBoost [5], multi-layer perception [6], principal component analysis [7], Histogram of Oriented Gradients (HOG) features [8], and depth data [9]. A survey of 3D hand gesture recognition is presented by Cheng et al. [10].

Following the gesture recognition, there are a number of studies performed to evaluate the interfaces in terms of their usability and user experience. Farhadi-Niaki et al. [11] presented a usability study for arms and hand gestures to be used in common desktop tasks. Bragdon et al. [12] presented touch and air gestures for supporting developer meetings. A usability study of gestures in the virtual reality environment was presented by Cabral et al. [13]. Villaroman et al. [14] presented gesture-based user interface design using Kinect. Their work shows that even with the limitations of Kinect in terms of fine-grain gesture input, it works efficiently for common tasks. Bhuiyan et al. [15] presented gesture-based controls for common day-to-day tasks and studied their effectiveness in a number of real-world scenarios. Ebert et al. [16] show the limitations of a gesture-based interaction for manipulating CT scan data. Liao et al. [17] presented a gesture-based command system for interactive paper, and show that gesture-based interfaces are well suited for new technologies. Wachs et al. [18] presented a number of vision-based hand gesture applications.

Recently, Leap Motion [19] introduced a new way to interact with the desktop systems using hand and finger-based gestures. Leap Motion primarily targets hands recognition, and can track all 10 fingers with up to 1/100th millimeter accuracy [19]. The Leap Motion has a 150° wide, and 135° deep field of view (FOV). Within this FOV it can track within 8 cubic feet of 3D space. An example of the controller can be seen in Figure 1. In contrast to Kinect, which provides full body color, depth and pose data, the Leap Motion Controller provides specific set of parameters related to hands and fingers. Leap Motion SDK supports a limited number of gestures, e.g., Pinch, swipe, and tap [19]. Recognition of general purpose gestures from the Leap Motion data is an active research problem. Potter et al. [20] presented a study on viability of using Leap Motion for the sign language recognition. Guerrero-Rincon et al. [21] presented a gesture control system using Leap Motion to control a robot. Marin et al. [22] presented a generic framework to detect gestures using both Kinect and Leap Motion. In contrast, there are not many user studies that not only implement customize gestures but also perform a user study to demonstrate the effectiveness of usability and user experience of a hand gesture-based interface using the Leap Motion Controller.

In this work, we present a new user interface to manipulate

 ISBN: 978-1-61208-538-8

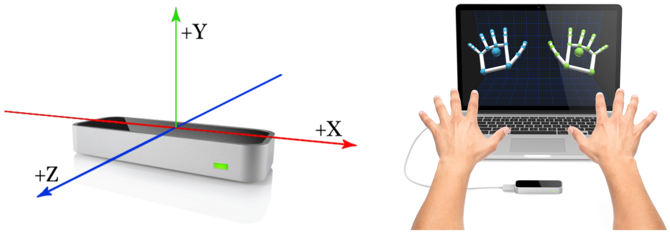

Figure 1. (left) Leap Motion Controller with the overlay of 3D right-handed coordinate system. (right) Tracking results are depicted on the computer screen if the hands are placed above the controller within the tracking distance.

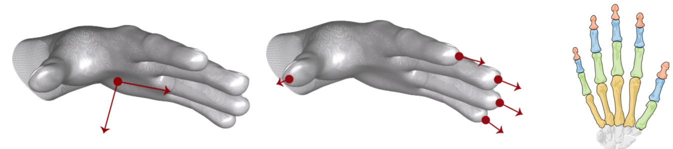

Figure 2. (left) Leap Motion Controller hand tracking. The palm normal and palm direction vectors are shown. (middle) Finger tracking with the direction vector of each finger is shown. (right) All the bones that are tracked by the Leap Motion Controller.

static 3D content using the Leap Motion Controller. Manipulation of 3D content is one of the widely performed user interaction in the computer graphics applications. A number of software, e.g., 3dstudio, Maya, CAD, Blender etc., employ the manipulation of 3D content. In general, a mouse-based interface is used for the user interaction. We implement four gestures, only using the thumb and the index finger to rotate and scale the 3D content. Three gestures are used to rotate the 3D content along the x, y and z axes, while the final gesture is used to uniformly scale the 3D content. We detail the implementation of the gestures, and then a comprehensive user study that evaluates the usability and user experience of the gesture-based user interface.

In the following section, we will first present the gesture-based user interface design and implementation (Sect. II), followed by the user study (Sect. III). Afterward, we will discuss the results (Sect. IV), followed by the conclusions (Sect. V).

II. User Interface Design & Implementation

Our gesture-based user interface is implemented using the Leap Motion Controller. As mentioned in the previous section, the Leap Motion Controller primarily focuses on the capturing of hands and fingers. The Leap Motion Controller uses optical sensors and infrared light, and the coordinate system is directed such that the y-axis is pointing upward when the device is placed horizontally on a surface. It employs a right-handed 3D coordinate system. It has a FOV of 150° wide, and 135° deep. The effective range of the Leap Motion Controller is from 25 to 600 millimeters or 1 inch to 2 feet above the device. The Leap Motion Controller coordinate system, along with the tracking result can be seen in Figure 1.

The Leap Motion Controller tracks hands and fingers in its field of view. The hand tracking provides the identity of the hand, along with its position, palm normal, direction, the arm to which the hand is attached and the list of fingers associated with the hand. A visualization of hand tracking can be seen in Figure 2(left). In addition, the Leap Motion Controller tracks individual fingers and tracks the position and direction of each finger. The visualization of the finger tracking can be see in Figure 2(middle). We make extensive use of these tracking parameters in our user interface design. If required, the Leap Motion Controller can return the position and orientation of each anatomical finger bone. Even though we do not use this data for our user interface, but it can be incorporated for more complex gestures. A visualization of the anatomical finger bones tracked by the Leap Motion Controller can be seen in Figure 2(right).

We implement four gestures to manipulate static 3D content using the Leap Motion Controller. The most general manipulation for the static 3D content is rotation and scaling. The rotation takes place along the x, y and z axes needing three gestures, and since we only implement the uniform scaling, it requires only one more gesture. Before describing the actual gesture recognition, we would like to formally define the hand and finger parameters that are used to implement the gestures:

- P_i is the 3 space position of each fingertip. The index i is from 0 to 4, identifying each finger starting from thumb (0) to the baby finger (4).
- D_i is the 3 space direction vector of each fingertip. The index i is from 0 to 4, identifying each finger starting from thumb (0) to the baby finger (4).
- N is the 3 space palm normal vector.

The first step before detecting any gesture is to define and identify a trigger. The trigger does not mean that a gesture will definitely occur but increases its likelihood. In our implementation, we define the trigger as the open hand with a spatial difference between the thumb and the index finger. We name the trigger as the "Neutral Pose", and all of gestures start form the Neutral Pose. An example of the Neutral Pose can be seen in top left image of the Figure 3, 4, 5, or 6. In principal, the Neutral Pose only depends on the thumb and the index finger, but for the gesture to work, it does not matter if other fingers are open or closed.

To detect the Neutral Pose, our system continuously monitors P_0, P_1, D_0, and D_1. If the distance between P_0 and P_1 is between 4 to 6 cm and D_0 and D_1 do not change over 30 frames, then it classifies the current pose as the Neutral Pose. Current values of P_0, P_1, D_0, D_1, and N are stored, and the system now actively searches for one of the four gestures, under the assumptions of each gesture. Henceforth the stored values for the Neutral Pose will be referred as P_0^{np}, P_1^{np}, D_0^{np}, D_1^{np}, and N^{np}. If there is a significant change in any of the four parameters then the Neutral Pose is classified as lost and the system again waits till the Neutral Pose is identified. Below we formally define the three rotation gestures and one scale gesture.

A. Rotation

3D rotation is characterized by the three rotations pitch, yaw and roll, which are the names given to rotations around x, y and z axes respectively. Once the Neutral Pose is detected, our implementation system actively searches for one of the three rotations or the scaling. Our rotation gesture is extremely

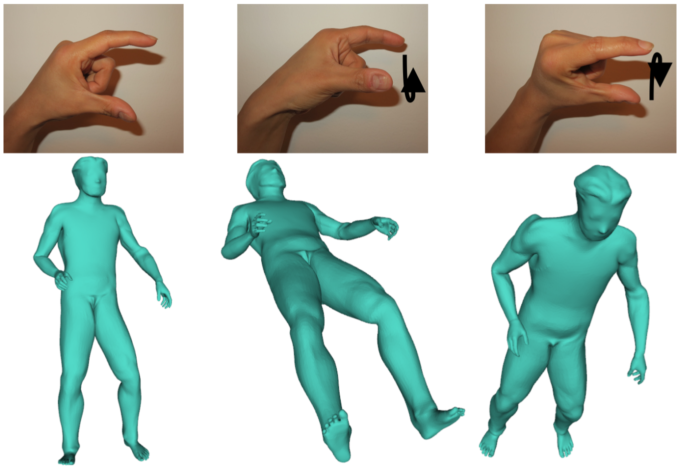

Figure 3. (left, top) The neutral pose is shown. (left, bottom) The 3D model in an arbitrary pose can be seen. Middle and right images show the rotation gestures with the corresponding rotated 3D model along the x axis.

intuitive as it is defined by the actual 3D rotation of the hand in the real world. Our system detects that if the values for P_0, P_1, D_0, and D_1 are similar to the Neutral Pose but the value of N is changing then one of the rotation gesture is being performed. Formally, the system actively monitors for $N.N^{np} > \sigma$. Where $N.N^{np}$ is the cosine of the angle between the two vectors. σ is the threshold that controls the sensitivity of the rotation gesture. In our case, it is equal to 0.349066, which implies that a rotation gesture is detected if the palm is rotated along one of the axes by more than 20° under the condition that the values of P_0, P_1, D_0, and D_1 are similar to P_1^{np}, D_0^{np}, and D_1^{np} respectively.

Once the rotation gesture is identified, we need to determine the axis of rotation. We determine the axis of rotation along one of the three axes using the following algorithm:

- Let N_{xy} and N_{xy}^{np} be the projection of N and N^{np} on the default xy plane at z=0.
- Let N_{xz} and N_{xz}^{np} be the projection of N and N^{np} on the default xz plane at y=0.
- Let N_{yz} and N_{yz}^{np} be the projection of N and N^{np} on the default yz plane at x=0.
- Compare $N_{xy}.N_{xy}^{np}$, $N_{xz}.N_{xz}^{np}$ and $N_{yz}.N_{yz}^{np}$, which is the dot product of the vectors projected on each plane respectively.
- If $N_{xy}.N_{xy}^{np}$ is maximum then the axis of rotation is along the z axis.
- If $N_{xz}.N_{xz}^{np}$ is maximum then the axis of rotation is along the y axis.
- If N_{yz} and N_{yz}^{np} is maximum then the axis of rotation is along the x axis.

Once the axis of rotation is determined, the corresponding dot product is used to work out the angle of rotation and the 3D model is rotated around that axis based on the angle of the rotation. An example of 3D rotations can be seen in Figure 3, 4, and 5.

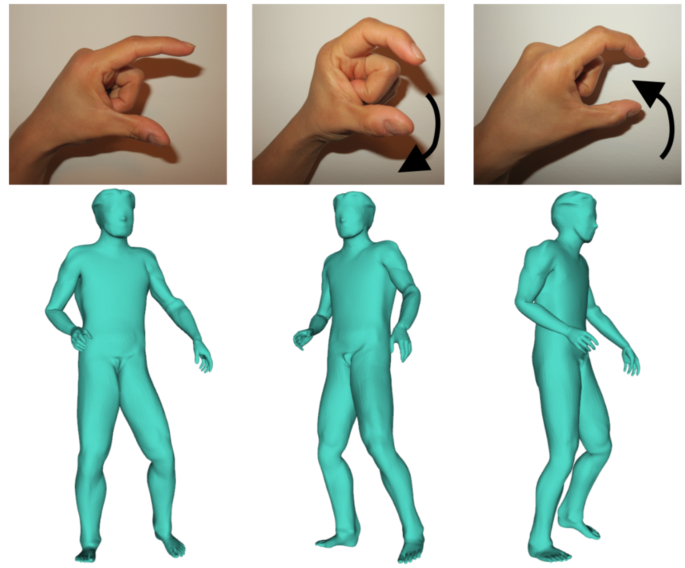

Figure 4. (left, top) The neutral pose is shown. (left, bottom) The 3D model in an arbitrary pose can be seen. Middle and right images show the rotation gestures with the corresponding rotated 3D model along the y axis.

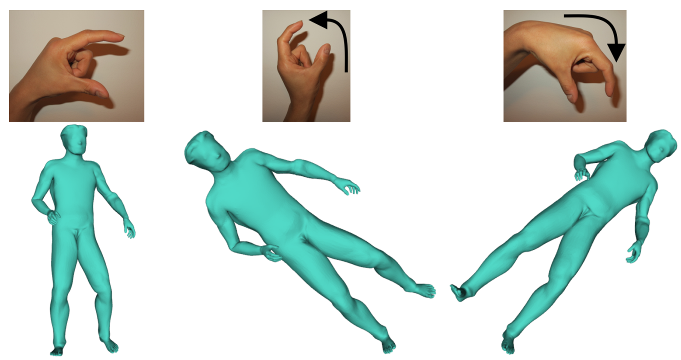

Figure 5. (left, top) The neutral pose is shown. (left, bottom) The 3D model in an arbitrary pose can be seen. Middle and right images show the rotation gestures with the corresponding rotated 3D model along the z axis.

B. Scaling

Since we only implement the uniform scaling, we need only one gesture to detect if the 3D content is being scaled up or down. The scaling gesture is defined by the fixed direction N with respect to N^{np}, but with the changing distance between P_0 and P_1 with respect to the distance between P_0^{np} and P_1^{np}. The following algorithm is used to detect the scaling gesture:

- Let $D^{np} = \sqrt{(P_0^{np} - P_1^{np})^2}$, which is the distance between the thumb and the index finger at the Neutral Pose.
- Let $D = \sqrt{(P_0 - P_1)^2}$, which is the distance between the thumb and the index finger after the detection of the Neutral Pose.
- if $D - D^{np} > \rho$ then the 3D content is being scaled up.
- if $D - D^{np} < -\rho$ then the 3D content is being scaled down.

The choice of parameter ρ controls the sensitivity of the scaling gesture. In our case, we set it to 1 cm, so that if the

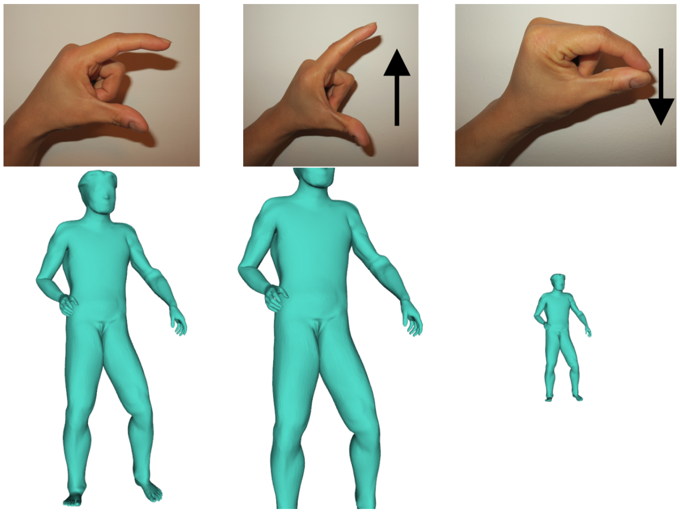

Figure 6. (left, top) The neutral pose is shown. (left, bottom) The 3D model in an arbitrary pose can be seen. Middle and right images show the scaling gestures with the corresponding uniformly scaled 3D model.

distance between the thumb and the index finger is increased by 1 cm then it is scaled up. Similarly, if it is decreased more than 1 cm then it is scaled down. An example of the scaling gesture can be seen in Figure 6. The above described gestures are used to manipulate the 3D content using both rotation and scaling. In the following section, we will describe the user study that validates the effectiveness of both the usability and the user experience of these gestures.

III. User Study

In order to evaluate the gesture-based user interface, we performed a user study over 20 participants (10 male, and 10 female). All the users were already familiar with manipulating 3D content using a mouse-based interface and their age was from 18 to 30. None of the users had any experience of manipulating 3D content using a gesture-based interface. At the beginning of the user study, they are asked to make note of the current scale of the 3D model. They performed the following steps in the user study:

- Rotate left along the x axis.
- Rotate down along the y axis.
- Scale up a little.
- Rotate right along the z axis.
- Scale down to half the size.
- Rotate right along the x axis.
- Rotate and scale the 3D model such that goes back to the original scale and facing towards the user.

After the completion of the user study, the users were asked the following questions:

1) Have you ever used a gesture-based interface before? (Yes/No).
2) Rate your satisfaction with the rotation interface on the scale of 1 to 9 with 1 being very simple, and 9 being very difficult.
3) Rate your satisfaction with the scaling interface on the scale of 1 to 9 with 1 being very simple, and 9 being very difficult.
4) Compared to a mouse-based interface how would you rate the effectiveness of the gesture-based interface in getting the tasks done? 1 very effective, 2 effective, 3 same, 4 worse, and 5 far worse.
5) Compared to a mouse-based interface how would you rate the learning curve of the gesture-based interface? 1 very easy, 2 easy, 3 same, 4 difficult, and 5 very difficult.
6) Compared to a mouse-based interface how would you rate the memorability of the gesture-based interface? 1 better, 2 good, 3 same, 4 bad, and 5 worse.
7) How would you define the experience of using the gesture-based interface in one word? e.g., exciting, fun, boring etc.
8) Given the choice of the mouse or gesture-based user interface, which one would you recommend?

On average, it took users less than a minute to perform the tasks, and they were immediately moved to the questionnaire after the completion of the tasks. On average it took 5 minutes for the users to complete the tasks and the user study. The results of the study are discussed in the next section.

IV. Results and Discussions

Our user study has seven closed questions (1 to 6, and 8), and one open question (7). Questions 1 to 6 focus on the usability, while the last two questions evaluate the user experience. Even though our usability questions have a scale of five or 9, we have consolidated the results into three classes for the ease of presentation. The results are presented as the percentages of the actual participants. The following table highlights the usability aspect of the gesture-based control:

TABLE I. USABILITY EVALUATION RESULTS BASED ON THE USER STUDY.

Question	Positive	Neutral	Negative
1	0%	0%	100%
2	70%	20%	10%
3	80%	20%	0%
4	70%	10%	20%
5	90%	10%	0%
6	95%	5%	0%

Similar to the usability analysis, to facilitate the presentation, we have classified the user experience analysis into two categories. For the question #7, we have classified the words e.g., fun, exciting, satisfactory etc., as positive, and the words e.g., boring, difficult etc., as negatives. Question #8 has only two possible answers, so there is no need for further classification. The following table shows the results of the user experience survey:

TABLE II. USER EXPERIENCE EVALUATION RESULTS BASED ON THE USER STUDY.

Question	Positive	Negative
7	95%	5%
8	90%	10%

As can be seen from the user study that the gesture-based interface is highly favored by the users both in terms of usability and the user experience. For the user experience, many of the users expressed that our interface is innovative and intuitive. It is evident from the results that the gesture-based interface is similarly effective to manipulate the 3D content compared to a mouse-based interface. Users not only managed to rotate and scale the 3D content easily using the gestures, but they are also able to complete the task efficiently without taking too much time. In addition, also in terms of learning the gesture-based interface proves to be easier to learn because users find it easy to pick up intuitively. Similarly, memorability of our interface is much better because once used they are easy to recall without any cues or prompts.

Our user study is subject to a couple of limitations. We do not specifically evaluate the ergonomics and fatigue factor associated with the gesture-based interface. Even though we did not receive any negative feedback in terms of user experience related to ergonomics or fatigue, these are important factors that must be evaluated using a long-term study. We will consider evaluating them in the future work. Similarly, we do not do a quantitative comparison of the time taken to complete various tasks using the mouse and the gesture-based interface. We plan to incorporate it in the future user study.

Despite the limitations, we show that it is possible to implement a single handed gesture-based interface to manipulate static 3D content using the Leap Motion Controller. Our user study demonstrates the effectiveness of the usability and user experience of such interface.

V. CONCLUSIONS

We presented a new hand gesture-based interface to effectively manipulate static 3D content using the Leap Motion Controller. We detailed the implementation of the four gestures. Our gesture recognition method first recognizes the Neutral Pose and then actively searches for either the rotation or the scaling gesture. The rotation gestures are recognized by using the rotation of the palm normal vector to find out the axis and angle of rotation. The scaling gesture is recognized by the changing distance between the thumb and the index finger. We performed a comprehensive user study to show the effectiveness of the usability and user experience of the gesture-based interface. The user study shows that gesture-based interface is easy to use, simple to learn and has a higher level of memorability. It resulted in a very positive user experience and most of the users recommended the interface over a mouse-based interface. The user study shows that our method is viable for manipulating static 3D content and can be employed in a number of applications. In future, we would like to add more gestures to add translation, in addition to rotation and scaling. We would also like to add gestures using both hands to control multiple 3D objects at the same time. In addition, we would also like to explore a gesture controlled system for controlling animated 3D content.

REFERENCES

[1] MICROSOFT, "Kinect for microsoft windows. http://www.kinectforwindows.org/," November 2010, Last Access: Jan, 2017.

[2] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in ICCV, 2011, pp. 415–422.

[3] S. Marcel, O. Bernier, J. Viallet, and D. Collobert, "Hand gesture recognition using input-output hidden markov models," in 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000), 26-30 March 2000, Grenoble, France, 2000, pp. 456–461.

[4] F. Chen, C. Fu, and C. Huang, "Hand gesture recognition using a real-time tracking method and hidden markov models," Image Vision Comput., vol. 21, no. 8, 2003, pp. 745–758.

[5] Y. Liu and P. Zhang, "Vision-based human-computer system using hand gestures," in Computational Intelligence and Security, 2009. CIS '09. International Conference on, vol. 2, Dec 2009, pp. 529–532.

[6] C. Yu, X. Wang, H. Huang, J. Shen, and K. Wu, "Vision-based hand gesture recognition using combinational features," in Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2010 Sixth International Conference on, Oct 2010, pp. 543–546.

[7] J. L. Raheja, R. Shyam, U. Kumar, and P. B. Prasad, "Real-time robotic hand control using hand gestures," in Proceedings of the 2010 Second International Conference on Machine Learning and Computing, ser. ICMLC '10, 2010, pp. 12–16.

[8] Y. Y. Pang, N. A. Ismail, and P. L. S. Gilbert, "A real time vision-based hand gesture interaction," in 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation, May 2010, pp. 237–242.

[9] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, Dec 2014, pp. 2368–2377.

[10] H. Cheng, L. Yang, and Z. Liu, "Survey on 3d hand gesture recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol. 26, no. 9, Sept 2016, pp. 1659–1673.

[11] F. Farhadi-Niaki, S. A. Etemad, and A. Arya, "Design and usability analysis of gesture-based control for common desktop tasks," in Proceedings of the 15th International Conference on Human-Computer Interaction: Interaction Modalities and Techniques - Volume Part IV, ser. HCI'13, 2013, pp. 215–224.

[12] A. Bragdon, R. DeLine, K. Hinckley, and M. R. Morris, "Code space: Touch + air gesture hybrid interactions for supporting developer meetings," in Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ser. ITS '11, 2011, pp. 212–221.

[13] M. C. Cabral, C. H. Morimoto, and M. K. Zuffo, "On the usability of gesture interfaces in virtual reality environments," in Proceedings of the 2005 Latin American Conference on Human-computer Interaction, ser. CLIHC '05, 2005, pp. 100–108.

[14] N. Villaroman, D. Rowe, and B. Swan, "Teaching natural user interaction using openni and the microsoft kinect sensor," in Proceedings of the 2011 Conference on Information Technology Education, ser. SIGITE '11, 2011, pp. 227–232.

[15] M. Bhuiyan and R. Picking, "A gesture controlled user interface for inclusive design and evaluative study of its usability." JSEA, vol. 4, no. 9, 2011, pp. 513–521.

[16] L. C. Ebert, G. Hatch, G. Ampanozi, M. J. Thali, and S. Ross, "You cant touch this: Touch-free navigation through radiological images," Surgical Innovation, vol. 19, no. 3, 2012, pp. 301–307.

[17] C. Liao, F. Guimbretière, and K. Hinckley, "Papiercraft: A command system for interactive paper," in Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology, ser. UIST '05. New York, NY, USA: ACM, 2005, pp. 241–244.

[18] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," Commun. ACM, vol. 54, no. 2, Feb. 2011, pp. 60–71.

[19] LeapMotion, "Leap motion controller. https://www.leapmotion.com/product/desktop," November 2012, Last Access: Jan, 2017.

[20] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, ser. OzCHI '13, 2013, pp. 175–178.

[21] C. Guerrero-Rincon, A. Uribe-Quevedo, H. Leon-Rodriguez, and J. O. Park, "Hand-based tracking animatronics interaction," in Robotics (ISR), 2013 44th International Symposium on, Oct 2013, pp. 1–3.

[22] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in 2014 IEEE International Conference on Image Processing (ICIP), Oct 2014, pp. 1565–1569.