# Fusion of Leap Motion and Kinect Sensors for Improved Field of View and Accuracy for VR Applications

**Stanford EE267, Virtual Reality, Course Report, Instructors: Gordon Wetzstein and Robert Konrad**

Adam Craig* & Sreenath Krishnan †

## Abstract

Kinect and Leap Motion sensors are commonly used for tracking applications today. However, both sensors have failure modes in particular tracking scenarios. The Kinect sensor is very sensitive to the orientation of the user with respect to the sensor and is susceptible to occlusions. The Leap Motion senor has a limited range and field of view, but is far more accurate than the Kinect. In this paper, these two sensor are combined to utilize the range and field of view of the Kinect sensor with the accuracy of the Leap Motion. A virtual reality (VR) game is used to demonstrate this tracking system and evaluated via user feedback. Based on the feedback from users it can be seen that users perceive the combined system to be more seamless to interact with as they do not need to worry much about the physical space of hand interaction.

**Keywords:** virtual reality, hand tracking, Kinect, Leap Motion

## 1 Introduction

In recent years there has been an explosion of virtual reality (VR) technologies and applications. The growth of the cell phone market has produced cheaper displays and more accurate sensors that have enabled virtual reality concepts to become feasible and affordable. However, the immersion of virtual reality environments can always be improved. One way of improving the realism of virtual environments is to allow humans to manipulate the virtual space with motions in the physical space.

Thankfully, there has also been an increase in the ability to track human motion. IMUs, vision systems, infrared depth cameras, and laser range finders are all technologies that can allow for the tracking of motion in the physical space. Two sensors that use infrared depth measuring technologies are the Leap Motion and the Microsoft Kinect. However, both these sensors also have their own tracking faults.

The Kinect sensor has a range that can allow for full body tracking, however, with this large range comes increased errors. Also, many of the tracking algorithms require a particular orientation of the tracked user. Once the user starts to become perpendicular to the sensor, the tracking breaks down. The Leap Motion has a very limited range but has very accurate hand tracking. Also, since the Leap Motion is usually mounted onto a head mounted display, it has the ability to overcome the occlusions experienced by the Kinect sensor.

The goal of this paper is to develop an improved tracking strategy for VR applications. The range and field of view of the Kinect sensor will be paired with the accuracy of the Leap Motion to result in more realistic manipulation of the virtual space. A VR demo will be prepared to allow for user feedback of this improved tracking strategy to gauge the potential of this technique while identifying potential shortcomings.

*sacraig@stanford.edu
†ksree@stanford.edu

## 2 Related Work

Kinect and Leap Motion technology has been used for a variety of applications. Kinect's primary use is for gaming, yet it has been used in a research capacity as well. A recent study [Frati and Prattichizzo 2011] used the hand tracking capabilities of the Kinect sensor to improve the stimulation provided by a worn haptic device. The Leap Motion has also been used in a research capacity as early as 2014. In [Bassily et al. 2014], a Leap Motion is used for the control of a 6 degree of freedom (DOF) robotic arm for the development of assisted devices.

Work as also been done to evaluate the tracking provided by these sensors technologies. [Obdrzalek et al. 2012] evaluated the Kinect's pose estimation compared to established tracking techniques. What they found was that under the right circumstances the Kinect sensor can have tracking errors of less than 5cm, however, there can be high variations of around 2cm between static frames. Once the angle of the user begins to approach 90 degrees to the Kinect sensor, the tacking algorithms break down and the Kinect can no longer accurately track the motions of a user.

The Leap Motions capabilities have also been evaluated. [Weichert et al. 2013] reported that the Leap Motion had a standard deviation of below 0.2mm for static tracking with a slightly larger 1.2mm standard deviation for tracking of dynamic objects. This is order of magnitudes below the tracking errors produced from the Kinect sensor. However, a similar paper by [Guna et al. 2014] produced similar results, but also noted that the accuracy of the of sensor degrades when obstacles move beyond a 250mm distance from the sensor. This is well below the roughly 4m tracking range provided by the Kinect sensor.

Projects have been conducted that aimed to combine and pair these sensors to overcome some of the drawbacks outlined in previous studies. [Asteriadis et al. 2013] used multiple Kinect sensors to overcome the occlusion and lose of accuracy due to changes in orientation of the object being tracked. However, no published research project was able to be identified that pairs a Kinect sensor with a Leap Motion device.

## 3 Method

### 3.1 Overview

The goal of this project is to pair the Kinect and Leap Motion sensors to improve tracking for virtual reality applications. The application for this project will be an Angry Birds demo that requires the velocity and launch time of the users right hand as input. The important hardware used for this project will be a Leap Motion, Kinect sensor, and a Head Mounted Display (HMD) (with an IMU attached). The Unity game engine will be used to interface these devices and to render the virtual environment.

### 3.2 Leap Motion

The first sensor used for this project is the Leap Motion v2. The Leap Motion is a sensor that can track hand positions using 2 cameras and 3 IR LEDs [COLGAN 2014]. An image of this sensor can be seen in Figure 1.

**Figure 1:** *Image of Leap Motion Sensor*

Because of the wide angle cameras used, the field of view for the sensor is fairly good for capturing local movement. The field of view of the sensor is 120 deep and 150 wide. With a roughly 2 foot square located 2 feet above the sensor. A visualization of this field of view can be seen in Figure 2.

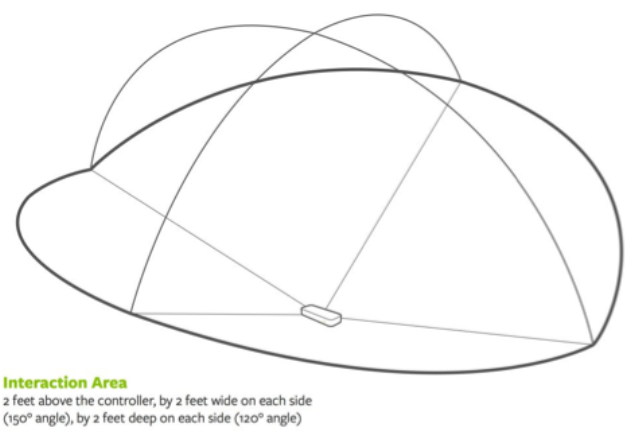


**Figure 2:** *Visualization of Field of View of Leap Motion*

This leap motion will be mounted to the head mounted display looking forward. As noted in previous work, the accuracy of this sensor is better than that of a Kinect, with 1.2mm standard deviation during dynamic tracking. However, this accuracy drops significantly at about half of the full 2 foot range.

The tracking software that comes with the Leap Motion is not only able to track the positions of the hand but also provides the normal vector to the palm, the position of each joint of the hand, and the recognition of unique gestures, such as pinching, swiping, and punching.

### 3.3 Kinect

The second sensor used for this project is the Kinect v2 for Windows. This sensor also utilizes infrared technology to track the distance to objects in its given frame. The Kinect sensor uses a three separate cameras (two depth cameras and a color camera) [MACCORMICK 2011]. An exploded view of this sensor can be seen in Figure 3.

**Figure 3:** *Exploded view of Kinect Sensor*

The field of view of this sensor is much less than that of the Leap Motion, however, it has a significant range increase. The Kinect has a horizontal field of view of 57 and a horizontal field of view of 43. This range is approximately 0.7m to 6m. This means that it has the ability to track much more of the physical user than the Leap Motion. However, as noted in Related Work, the Kinect sensor is much nosier than the Leap Motion, with non-negligible variations even with static tracking. A visualization of this field of view can be seen in Figure 4.

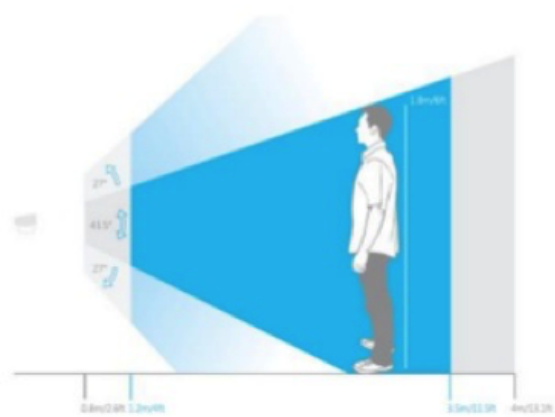

**Figure 4:** *Visualization of Field of View of Kinect Sensor*

The Kinect sensor will be placed facing the user, with the user perpendicular to the Kinect. This will result in the best tracking of the user. Therefore, the two sensors will be facing each other, allowing for the users hand to be tracked in the physical space, from two orientations.

### 3.4 Head Mounted Display

The head mounted display for this project will be a collection of hardware put together for the EE267 virtual reality class. The housing for this head mounted display is the View Master VR Start Pack seen in Figure 5.

**Figure 5:** *Image of the View Master VR Starter Kit*

Inside, the display is 6 inch LCD with a 1080p resolution. The Topfoison display comes with an HDMI driver board. In order to track orientation, an InvenSense MPU-9255 IMU is mounted to the front of the View Master headset. In order to interface the IMU and send orientation data to a computer, an Arduino Metro Mini is used. All of these items are placed together to make an affordable virtual reality headset.

Because this headset does not have a strap, the user is required to hold the HMD to their face. Therefore, only one hand is able to act as input to the virtual reality application. An image of a user holding the headset with the IMU and Leap Motion attached can be seen in Figure 6.

**Figure 6:** *Headset being worn by a user with Leap Motion and IMU attached*

### 3.5 Unity Game Engine

In order to interface with the sensors, drive the HMD, and render the virtual environment the Unity game engine was used. A single scene was rendered for the demo using the Unity Cardboard plug-in.

The Kinect Unity Pro Package based on the Kinect SDK 2.0 was used for the integration between the Kinect and Unity. This enabled access to Kinects BodyManager class which allows access to multiple skeletons being tracked in a single frame and the joints of these skeletons. For the purpose of this project, the HandRight joint type will be used for tracking.

Similarly, the Leap Motion interfaces with Unity via a Unity Core Assets library provided by Leap Motion developers. This enables the tracking of both the right and left hand as well gesture recognition through simple function calls.

The sensor data is tracked per frame via a script attached to the main Cardboard object in the unity engine. This script then determines the trigger event and launch velocity of projectiles in the virtual reality application. The physics package of unity then determines the outcome of the projectile and collisions with obstacles in the scene.

### 3.6 Tracking Algorithm

In order to improve the tracking given by an individual sensor, the data of these two sensors will be fused. For every frame, the velocity from the previous frame will be stored in a buffer. Once a trigger event is detected (notifying that the object is to be launched), the data in the buffers will be used to produce the velocity of the object at launch.

There is one trigger event that is tracked by both sensors. If either the Leap Motion or the Kinect, detects that the users hand is open, it will trigger a launch event. Having two separate triggers for each sensor allows for the ability to launch objects even if one sensor is not able to track the users hand. This improves the field of view of the developed system and can mitigate the effect of occlusions.

In order to merge the velocity buffers, a weighted average of the two sensors will be used. When both sensors are able to track the data, the confidence level of the Leap Motion and Kinect sensors will be used to determine the weighting of the two sensors. If one sensor is not able to track the hand for that frame, it will have no effect on the velocity. Essentially, its weighting for the weighted average will be 0. Since the Leap Motion is the more accurate for tracking at the distances expected for normal usage, it will have a higher weighting (as long as the confidence level of the measurement remains high).

### 3.7 Virtual Reality Application

To test the improved tracking stated in section 3.6, a virtual reality demo that allows the user to launch objects at a scene is developed. This application takes the form of an Angry Birds game. This game requires the user to hit or knock over a certain number of objects: spheres that represent pigs. These pigs are placed on a wooden structure that can also be knocked over with the birds launched by the user.

The developed game has 10 pigs arranged around 4 wooden structures. An image of this scene can be found in Figure 7.

This scene takes places in a 6 sided room that forces all the birds launched to remain in the scene and creates the potential for objects to bounce off walls and collide with other obstacles. An image of the bird launched by the user, can be seen in Figure 8.

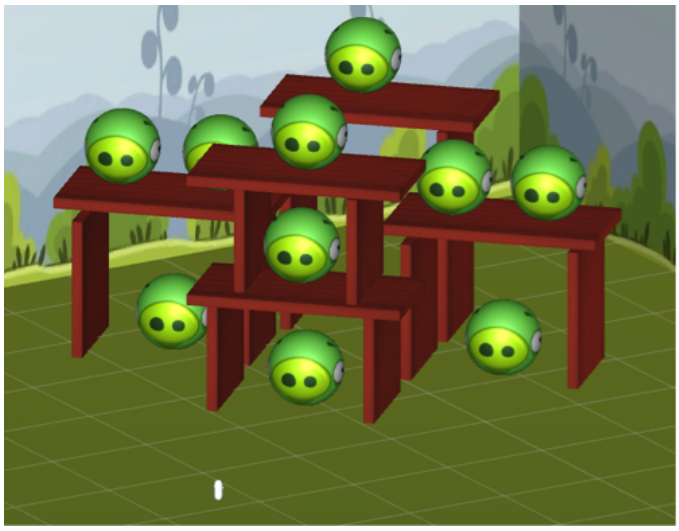

**Figure 7:** *Image of Scene for Virtual Reality Application*

**Figure 8:** *Image of "Bird" launched by User*

There is no limit on the number of objects launched by the user, but there is a 1.5 second delay between throws. After all pigs disappear, the demo will be restarted for the next user.

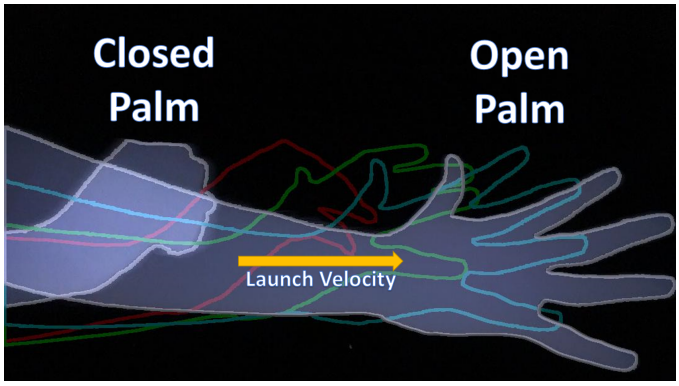


**Figure 9:** *Launch trigger*

The trigger gesture used in this game is shown in Figure 9. In order to launch an angry bird, the user starts with a closed palm and moves the palm while opening it. When the sensor detect that the palm is open, it find the launch velocity based on the user's palm velocity and a launch is performed in the direction the user is looking at

## 4 Results

Due to the inability to evaluate the tracking algorithm with a known velocity or use an established tracking system to verify the results of the tracking algorithm, user feedback and testing will be used to evaluate the application developed for this project.

A questionnaire was developed that asks users to rate how well they perceived the tracking and how easy it was to control the objects being launched by the user. Additional comments were also asked for to determine points of friction with the demonstration. Some of the users were also tested with only one of the sensors active.

Users found it more realistic working with the combined system. They often complained about lack of control when launching from the field of view of the Kinect alone, showing the noisier nature of Kinect sensor. After getting used to the game, users naturally tend to prefer Kinect-only physical space for fast throws and Leap workspace for slower and precise throws. However some of the users felt a discomfort with the throwing gesture and found that launching in the direction you are looking at to be slightly unnatural.

## 5 Analysis and Conclusion

A VR Hand tracking system that integrates Kinect and Leap motion sensors is developed and discussed here. The integrated system performs better than individual systems. Users can interact using their hands in a wider physical space due to Kinect's superior range at lower precision and for finer control, the user can use the limited physical space of the Leap motion.

## Acknowledgments

To Robert Konrad for providing hardware and a starting point for Unity development and Gordon Wetzstein for instruction of virtual reality concepts during EE267 lectures.

## References

ASTERIADIS, S., CHATZITOFIS, A., ZARPALAS, D., ALEXIADIS, D. S., AND DARAS, P. 2013. Estimating human motion from multiple kinect sensors. In *Proceedings of the 6th International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, ACM, 3.

BASSILY, D., GEORGOULAS, C., GUETTLER, J., LINNER, T., AND BOCK, T. 2014. Intuitive and adaptive robotic arm manipulation using the leap motion controller. In *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*, VDE, 1–7.

COLGAN, A., 2014. How does the leap motion controller work? (http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/).

FRATI, V., AND PRATTICHIZZO, D. 2011. Using kinect for hand tracking and rendering in wearable haptics. In *World Haptics Conference (WHC), 2011 IEEE*, IEEE, 317–321.

GUNA, J., JAKUS, G., POGAČNIK, M., TOMAŽIČ, S., AND SODNIK, J. 2014. An analysis of the precision and reliability of

the leap motion sensor and its suitability for static and dynamic tracking. *Sensors 14*, 2, 3702–3720.

MACCORMICK, J., 2011. How does the kinect work? (http://users.dickinson.edu/ jmac/selected-talks/kinect.pdf).

OBDRZALEK, S., KURILLO, G., OFLI, F., BAJCSY, R., SETO, E., JIMISON, H., AND PAVEL, M. 2012. Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. In *Engineering in medicine and biology society (EMBC), 2012 annual international conference of the IEEE*, IEEE, 1188–1193.

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.