International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2023)
J.-M. Normand, M. Sugimoto, and V. Sundstedt (Editors)

Comparison of Touchless Interaction With One and Multiple Optical Sensors

T. Novacek[1], R. Kondac[1] and M. Jirina[1]

[1]Faculty of Information Technology, Czech Technical University in Prague †

Abstract

In this research, we compared the precision and ease of use of hand-tracking with one and three optical sensors. We created two test scenes that simulated real-life scenarios, one focusing on smoothness and intuitiveness and the other one focusing on precision and tracking range.

We conducted tests with 25 participants and measured the precision and effectiveness of their work with basic user interface elements and 3D objects.

This research showed that using multiple optical sensors for hand-tracking greatly improves the precision of the tracking, widens the tracking range and provides more smooth interaction. On average, 80% of the users preferred using three sensors for the interaction because it allowed more users to finish the tasks (74% on average) in a shorter time (15% on average) and with more precise results (43% on average) compared to same tasks done with just one sensor.

CCS Concepts

• ***Human-centered computing*** → ***Usability testing; Gestural input; Haptic devices;***

1. Introduction

With the rise of virtual and extended reality, controllers like keyboards and mice are becoming obsolete. We need to be able not only to stand but also move, jump and run and still be able to control the virtual environment.

A myriad of ways allows just that, but some of them are better than others. Handheld controllers allow precise tracking but lack the life-like approach to controlling the virtual environment. A more natural way is provided by hand-tracking by optical sensors, digitising the motion of the user's hands and fingers and transferring it to the virtual world, making it as easy to control as the real world.

In our previous research [NMJ21], later extended in [NJ21], we presented a set of algorithms that fuse data from multiple optical hand-tracking sensors. By using multiple tracking data sources, we could provide precise tracking of hands and fingers without the need for the user to hold or wear anything. This approach allowed tracking users' hands from multiple angles to provide more data about the position and rotation of their hands and fingers.

This research extends our previous work by conducting a series of tests with users and comparing the usage of one sensor and three sensors in the same scenarios.

We use the Ultraleap Stereo IR 170 sensor [Ult22b] as a provider for the hand-tracking data; however, the concepts for fusing data from multiple optical sensors are generic and can be applied to any optical sensor.

We had four hypotheses that we wanted to confirm:

- **H1: Three sensors will provide smoother and more precise interaction.**
- **H2: With three sensors, it will be easier to work with the elements that are far from the centre of the scene.**
- **H3: With three sensors, testers will be able to finish the tasks faster.**
- **H4: Users will prefer hand-tracking when working with objects in 3D space, but they will prefer a keyboard and mouse for working with UI elements.**

The paper is organized as follows: the previous work on hand-tracking with optical sensors is described in section 2; the Ultraleap sensor and our approach to hand-tracking with multiple optical sensors are presented in section 3; the conducted experiments are described in 4; the discussion can be found in section 5; the conclusion and future work are given in section 6.

2. Previous work

Virtual and extended reality uses various ways for the user to control the virtual environment with his own hands. The user can hold some physical controller with buttons and touch pads, wear gloves

† The MultiLeap library was created in collaboration with Vrgineers.

DOI: 10.2312/egve.20231317

on his hands that track his movements, or his movements can be tracked by some external sensors [NJ22]. The most common external tracking is in the form of optical sensors that provide the most natural and life-like interaction that allows the user to use his hands freely without the discomfort of holding or wearing any kind of controller.

However, optical hand-tracking has two significant disadvantages – possible occlusion of tracked hand and short tracking range [NJ22]. With the occlusion, hands and fingers can be hidden behind some object or the hand itself, thus making it untrackable by the sensor. The second problem is that the sensor's tracking range usually does not exceed one meter. Both disadvantages can be overcome by using multiple sensors to track the hands from multiple angles, either to make the tracking more precise by having more tracking data or by tracking different areas completely, thus extending the tracked space.

To combine the tracking data from multiple sensors, we first need to synchronize them in time and space. The time synchronization is pretty straightforward – simply by utilizing the timestamps of the tracking data provided by the sensor. The synchronization in space is more complex. Since most of the sensors do not use absolute but relative coordinate systems, their tracking data streams must be synchronised in space to use in the same application.

First, the relative coordinate systems must be converted to absolute so that the hand-tracking data can represent the same object. This means that the sensor has to know its position and rotation to some predetermined point in space to translate its tracking data from relative data to absolute. The positions and rotations of the sensors to each other are sometimes known beforehand. Thus, it can be hardcoded in the application. However, this means that there can be only one setup of the sensors, which cannot be changed. However, space synchronization can also be done automatically – by determining the position of every sensor to each other and then computing the correct rotation and translation accordingly. We call this automatic process **space calibration** for clarity.

Several research institutes already used multiple Leap Motion Controllers to track the users' hands, mostly with simple static setups, which served as a prototype and focused on either more precise tracking or enlarging the tracking space.

Placid et al. [PAC*21] used two Leap Motion Controllers to create Virtual Glove – a virtual hand-tracking system to track the hand from multiple angles. They use two virtual machines on the same computer to get the data from the two sensors. The position of the sensors is hardcoded in the system, which makes any movement of the sensors impossible. Also, their research is limited to the use of just two sensors. The fusion of the tracking data is done by taking the data about the tracked hand from the sensor that sees the part of the hand more clearly.

Kiselev et al. [KKC19] connected three Leap Motions Controllers over a local network to achieve more precise gesture detection. The setup can be seen in figure 1. Tracking the same plane with devices #2 and #3 so close to each other makes the value of the third device debatable. They also do not fuse the tracking data, so no space or time synchronization of the sensors is used. Instead, they created their own dataset that contains the tracking data from all three sensors at once and the detection of the gestures is done by simply inputting three sets of data. This also means the dataset must be recreated for a different setup.

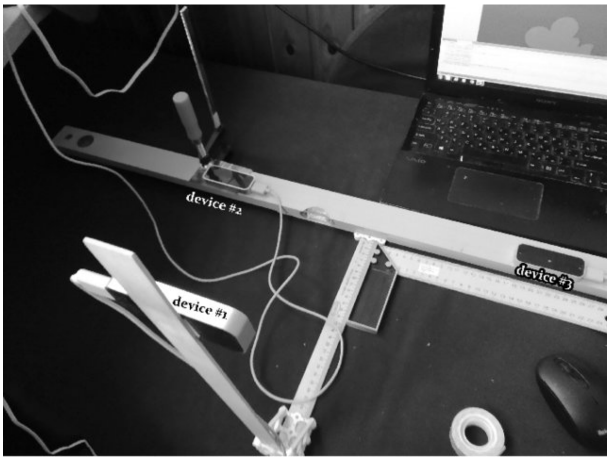


Figure 1: *The setup for the Multi Leap Motion system from Moscow Institute. [KKC19]*

Yu Wang et al. presented a hand-tracking system with multiple Leap Motion Controllers strapped on a headset [WWJ*21], as seen in figure 2. Their goal was to fix the short tracking range of the sensors for the VR environment. In this setup, they used five sensors whose fields of view partially overlapped so they could overcome incorrect chirality often reported by the sensor when the hand reaches the sides of its field of view. They achieved a 34% horizontal and 37% vertical enlargement of the tracking area compared to only one sensor.

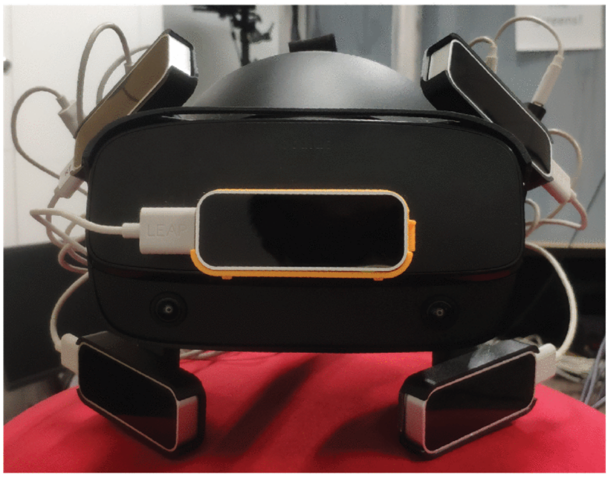


Figure 2: *The setup for the Multi Leap Motion system strapped on an HMD. [WWJ*21]*

A method based on the Least Square Fitting algorithm was used to calibrate multiple LMCs, which is based on the idea that every sensor shares its field of view with at least one other sensor. The front Leap Motion Controller was set as the reference sensor, the centre of the palm is used as the main traceable point, and the fusion of the other hand joints is related to the palm-joint tracking states.

A drawback is that this system again uses a static placement of the sensors and precomputed values for the translation and rotation of the tracking data. Another drawback is the placement of the sensors because the additional weight on the headset adds discomfort for the user and the need to have each sensor connected to separate computers due to the need to process the data on a separate physical machine.

Fok et al. [FGCT15] used two Leap Motion Controllers to detect American Sign Language. They used data fusing technology previously used for detecting car environment [AK22]. They used Kabsch algorithm-based calibration [Kab76] for space synchronization and covariance intersection [ARM01] for data fusion. The details of the synchronization, confidence computation and the number of computers were not presented.

Teleoperation system [HZW*17] created by researchers from Tsinghua University used five Leap Motion Controllers to widen the tracking area. Again, the transformation from relative to absolute hand-tracking data is done statically because the positions of the sensors to each other are known. They use a weighted average of the hand-tracking data to achieve the fused hand-tracking data.

All of these research projects use a static number of sensors, and their position cannot be changed. Re-configuring and recomputing the positions of the sensors is needed when a new sensor is added, or any of them is moved. Also, an additional physical or virtual machine is required if a sensor is added. Even though Leap Motion Controllers are cheap, the need for horizontal or vertical scaling of the computers that need to be done to meet the needs of the sensors makes these multi-sensor projects financially demanding.

3. MultiLeap system for hand-tracking

In our previous research [NMJ21], later extended in [NJ21], we presented a set of algorithms to fuse data from multiple optical hand-tracking sensors. We named the outcome of the research MultiLeap library because we used multiple Ultraleap optical sensors to obtain the tracking data. However, the algorithms used in the library are universal and do not depend on any specific tracking sensor. Our approach fixes the short tracking range and occlusion of the hand and fingers and generally provides more precise hand tracking.

3.1. Ultraleap sensor

Ultraleap Stereo IR 170 sensor by Ultraleap [Ult22b], the successor of the Leap Motion Controller [Ult22a], is one of the most successful hand-tracking sensors on the market. To track the hand, the sensor uses infrared light to illuminate it and then tracks the reflection of the infrared waves by two built-in cameras. The images from the sensor are processed by undisclosed (proprietary) classical vision techniques, which resolve the position of hands and fingers.

The hand-tracking data are then provided via API as frames – structured information about 27 distinct elements of the hand, such as the position and rotation of the palm, fingers and bones. The information about each finger is further divided into individual units according to bones and joints, as displayed in figure 3.

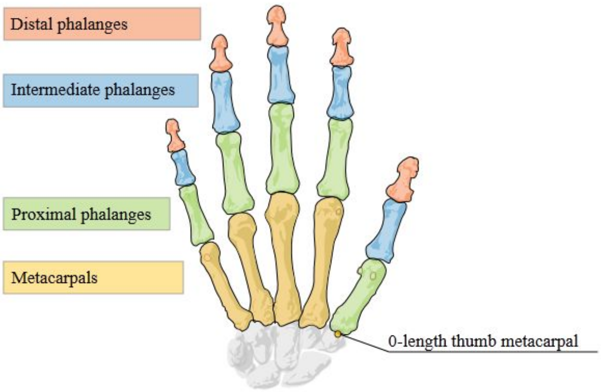


Figure 3: *Ultraleap hand parts.*

If part of the hand is occluded or out of the sensor's field of view, the tracking software estimates the tracking data for the missing part according to the visible part. The tracking software also provides interpolation and extrapolation of the tracking data to provide smoother tracking.

3.2. Fusing data from multiple sensors

In the last years, we presented several articles in which we proposed and tested a way to use multiple optical sensors. When precision, field of view and ease of use are concerned, our approach outmatched the results of hand-tracking with only one sensor. Our first article mostly focused on finding a way to fuse the hand-tracking data from multiple optical sensors [NMJ21]. In the second article, we extended this fusing algorithm and made the algorithm for synchronising the sensors easier and faster to use [NJ21]. In the third, yet unpublished article, we compared two approaches for synchronizing the hand-tracking data in space and improved the computation of hand-tracking data confidence, which we also proved superior to the computation provided by Ultraleap. Our approach allows us to extend the tracked space and make the tracking more precise by adding more sensors.

Our hand-tracking approach with multiple optical sensors will now be briefly described. For further information, please see our previous research, [NMJ21] and [NJ21].

Our research focused on three main algorithms – the **space calibration algorithm** [NMJ21], which is used for synchronizing the hand-tracking data in space so it can be fused, **hand-tracking data confidence algorithm** [NJ21], which determines the sensor that tracks the hand the best at a given time, and **hand-tracking data fusion algorithm** [NMJ21] that processes the tracking data from all the connected sensors and fuses them into one output.

The **space calibration algorithm**, presented in [NMJ21], and later extended in [NJ21], synchronizes the data streams from all

Proceedings published by Eurographics - The European Association for Computer Graphics.

hand-tracking sensors. The goal is to transform (translate and rotate) the tracking data from all sensors so they are in absolute coordinates, not relative.

Initially, one sensor, called the pivot, is marked as calibrated. Then, for every time t, the calibration algorithm stores **space calibration samples** – the hand-tracking data from all of the sensors that see the hand, along with the corresponding hand-tracking data from the pivot. When a predefined number of hand-tracking data for every sensor is stored, the correct translation and rotation are computed.

The rationale of this algorithm is very simple – the hand-tracking data from all of the sensors for time t for hand h are the same, just in a coordinate system relative to the corresponding sensor. Since the hand-tracking data for time t for sensor s_1 is just an affine transformation of the hand-tracking data from sensor s_2, we can then compute the translation and rotation of the tracking data by estimating the transformation by point-cloud matching of data representing the hand reported by the sensors.

Our second algorithm for precise hand-tracking with multiple optical sensors is the **hand-tracking data confidence** algorithm [NJ21]. It determines which sensor tracks the hand better at a given time.

This confidence value depends on both the angle of the hand to the sensor and the distance between them. In general, the sensor that sees the palm from the bottom can determine more information about the hand and thus has a higher hand-tracking data confidence than the sensor that sees the hand from the side (and probably sees the hand and fingers at least partly occluded). A sensor that is too close or too far from the hand will have a lower hand-tracking data confidence value than a sensor that is at the optimal distance from the hand (for example, 20 centimetres for the Ultraleap sensor). The distance is determined by each specific sensor type and should be provided by the manufacturer. The hand-tracking data confidence value is recomputed in real-time with every hand-tracking data sent by the sensor, so it always corresponds to the current hand pose.

The third algorithm we presented in our research was the **hand-tracking data fusion** algorithm [NMJ21]. It builds up on top of the previous two algorithms – it takes the tracking data from the synchronized sensors and their hand-tracking data confidence to compute a weighted average of the tracking data from all of the sensors. Since all the sensors provide some information about the hand, but some sensors have more precise data for a given time, we can determine which sensor affects the resulting tracking data the most. Because the hand-tracking data confidence changes with every frame, the weights used for the average computation must also change dynamically for every frame.

Combining these three approaches allows us to provide precise hand tracking by tracking the hand from multiple angles with multiple sensors.

4. Experiments

In this research, we conducted a series of experiments with users to confirm the effectiveness of our approach for hand-tracking with multiple optical sensors. We created two test scenes in Unity game engine [Uni23] that focused on interaction with virtual objects. We let the users work with them in two tests each – in one, only one optical sensor was used, and in the second, we used three. The goal was to compare the precision and intuitiveness of the controller system when single and multiple sensors are used. To mitigate the bias of users getting to know the test during the experiment and thus having a better result in the second run, no matter the number of sensors, some users did the experiments with three sensors first and one sensor later.

The sensor's placement can be seen in figure 4. In the tests where only one sensor was used, only the bottom sensor was plugged in. The left and right sensors are 70 centimetres apart, so their fields of view overlap as little as possible, so they would not be redundant.

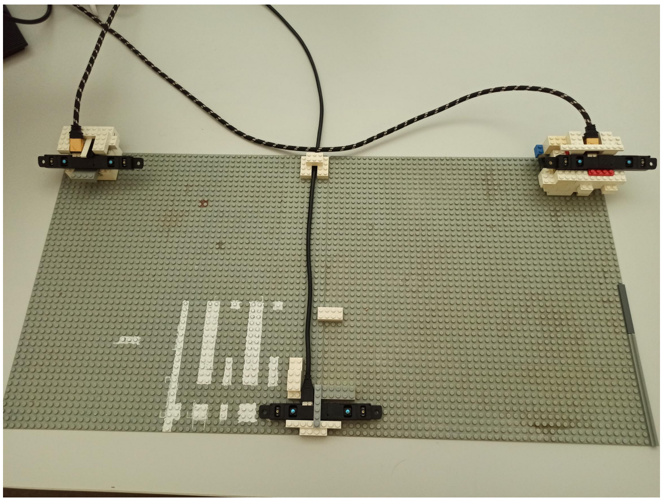


Figure 4: *Sensor placement used during the experiment.*

The experiment was not conducted in virtual reality because our goal was not to test the VR experience, just hand-tracking. We felt that overwhelming the users with additional stimuli provided by the virtual reality could make them lose focus on the interaction itself. We also did not want the users to suffer from the additional discomfort of wearing a VR headset, which can get heavy and tiring. Thus, the scene was displayed on a classic LCD display with Full HD resolution next to the setup with the hand-tracking sensors. This also allowed us not to focus the research on the concept of *presence* [Flo05] or to deal with *motion sickness*, which is very common in VR setups and similar experiments.

For the users to feel comfortable, they were allowed to either sit or stand or even switch between these two options as they needed.

4.1. Scenes

We created two test scenes in Unity, and each focused on a different kind of user-scene interaction. One scene, the **Cubes scene**, where the users were supposed to interact with 3D objects, and the **UI scene**, consisted of user interface elements like buttons and sliders.

The elements in the scenes were placed so they could be reached even when only one sensor was used, even though some elements were on the bounds of the sensor's field of view.

4.1.1. Cubes scene

In the first scene, the user was supposed to build a tower from cubes. The scene was initially empty, containing just one button that creates a new cube when pressed. The user then could stack the cubes on top of each other, trying to build the highest possible tower in four minutes. The current time was logged every time the user achieved the new highest tower (e.g. when the user first built a tower tree cubes high).

The goal of this scene was to test how the users can interact with 3D objects, in contrast to the previous scene with only 2D elements.

The initial *Cubes scene* can be seen in figure 5.

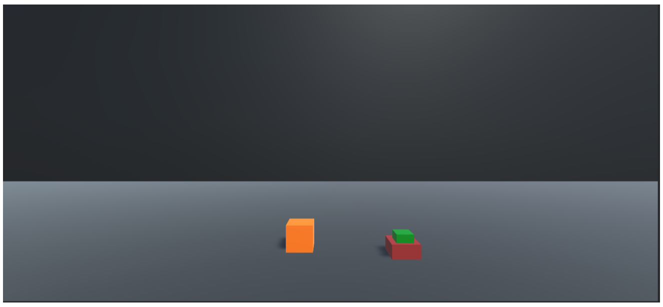


Figure 5: *The Cubes scene used in the experiment.*

4.1.2. UI scene

The goal of this scene was to confirm the usability of our approach to basic user interaction elements. Even though virtual reality provides novel ways to interact with the virtual world, simple menus and basic means of input are still necessary.

This scene was based on one of the example scenes provided by Ultraleap as part of their Unity plugin [Ult23]. This scene contained nine buttons, a horizontal slider and a 2D slider (where the slider button moves both horizontally and vertically).

First, we added digit labels to the buttons, thus turning them into a telephone keypad. Pressing the keypad buttons makes the pressed numbers appear above the keypad. We also added a delete button that removes the last entered digit. This part of the scene mimics the dialling of a phone number.

We also added a number label to the horizontal slider, which changes according to the slider's value, with a minimum value of 0 and a maximum value of 100, as often used for volume settings. The user's goal was to set a specific volume level via this slider.

The 2D slider was used to test precision and ease of interaction. Lines in the shape of the letter *N* were displayed on the slider as guidelines that the user was supposed to follow with the slider button. When the user finished the movement (got the slider button from one end of the symbol to the other), the precision of the movement and the time the movement took were displayed to the user. The user could reset and try the test again up to three times. The best time and precision of these three attempts were saved. The precision was computed as an Euclidean 2D distance between the centre of the slider button and the line itself.

The resulting *UI scene* can be seen in figure 6.

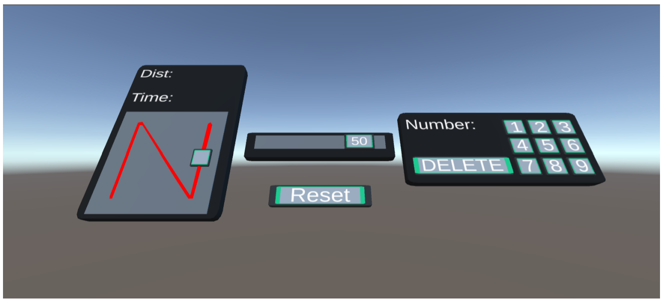


Figure 6: *The UI scene used in the experiment.*

4.2. Preparation for the tests

The users were asked to fill out three questionnaires during the experiment. First, before the test started, a questionnaire focused on their previous experience with virtual reality and various controllers for VR. The second and third questionnaires were the same; the users filled them out after concluding the *Cubes* and *UI* tests, respectively. All three questionnaires were a combination of single- and multiple-choice questions and open questions.

These questionnaires focused on the user experience – if the user could interact with the world seamlessly, without lag or jitter, or if the interaction felt natural. They also were asked to compare their experience with hand-tracking to interacting with the scene with a keyboard and mouse.

The instructions for the test were handed out to the testers beforehand so they could read it in advance and they could return during the test when needed. It took about 30 minutes to complete both tests, and the testers were allowed to take a break between them. First, the *Cubes* test was conducted, then the *UI* test.

The users also signed a form stating that they agreed with the experiment.

4.3. Information about the testers

Twenty-five people participated in the experiments – 14 men, 10 women, and one who did not want to disclose. The average age was 25.24, with a deviation of 4.15612. The testers were people who volunteered after a faculty-wide announcement about the planned experiment.

A majority (56%) of testers had tried VR 1–3 times before, 16% tried it 4–5 times, 20% more than five times and 8% had no previous experience with VR. 84% of testers never tried touchless interaction for VR.

4.4. Results

After each test, the users were asked to fill out a questionnaire that focused on the comparison of their feelings during the test related to the controllers.

Also, the times to complete the tasks and the precision were compared when one and three sensors were used.

Proceedings published by Eurographics - The European Association for Computer Graphics.

4.4.1. Cubes

This test was very well accepted by the users. According to the feedback received, it was “interesting” and “fun”.

However, most of the users complained about short tracking range when working with just one sensor, which resulted in difficulties in reaching for cubes that were on the edge of the scene. The users also often struggled with the hand disappearing completely when it was occluded or too far away from the sensor. Because of that, a lot of users were not able to build more than three cubes on top of each other with one sensor.

When three sensors were used, more users were able to build three or more cubes on top of each other, as can be seen in table 1. The height of four cubes was achieved by 13 users, which is an improvement of 160% from 5 users who were able to do the same with just one sensor.

In the table, *Improvement* denotes the difference between the corresponding values in the tests with one and three sensors. The higher the improvement, the better the results with three sensors. The user who was able to build 5 cubes on top of each other with one sensor was not the same as the one who was able to build 5 cubes on top of each other with three sensors.

	One sensor	Three sensors	Improvement
Two cubes	25	25	0%
Three cubes	17	23	39%
Four cubes	5	13	160%
Five cubes	1	1	0%

Table 1: *The tower’s heights in the Cubes task*

In the feedback for the test with three sensors, users stated that the movement was smoother and more precise, and the tracking range greatly improved. This can also be seen in the times of the tests (table 2), where stacking two and three cubes on top of each other with three sensors took half the time as compared to one sensor. Stacking four and five cubes with three sensors was achieved in a longer time on average because even not-so-dexterous users were able to achieve it, but not so fast, which resulted in a higher average time. However, achieving the goal in a long time is still better than not achieving it at all.

In the table, *improvement* denotes the difference between the corresponding values in the tests with one and three sensors. The higher the improvement, the better the results with three sensors, *mean* is the mean difference between times of tests with one and three sensors, *stdev* is a standard deviation, *DOF* denotes degrees of freedom. You can also find values connected to a paired t-test conducted, which we used to compare the means of the results of one- and three-sensor setups. t-value and p-value are computed with a critical value corresponding with the DOF for a 95% confidence interval. For two and three cubes, we can reject the null hypothesis of no difference and say with a high degree of confidence that the true difference in means is not equal to zero. For four and five cubes, we cannot reject the null hypothesis since we don’t have enough samples for a paired t-test.

From histograms in figure 7, it can be seen that in most cases, three sensors helped more users stack more cubes on top of each other in less time. Since only one user managed to stack five cubes with one sensor and one with three sensors, no relevant information can be extracted.

84% of the users stated that they preferred to work with three sensors, 4% said that they preferred one sensor, and 12% could not decide.

However, the users often said that they would improve the physics of the scene, for example, by adding more weight to the cubes so they are not tripped over so easily.

4.4.2. UI scene

This test was finished only by 20 out of 25 users when one sensor was used. For the rest, the sensor could not track their hand properly on the sides of its field of view – three users were not able to press the buttons to dial the phone number, one user was not able to finish the 2D slider, and one user was not able to finish both of these tasks.

With three sensors, all users were able to finish all the tasks because they could operate in a wider space thanks to the combined tracking range. Three sensors were also more precise with both the 2D slider and the test in general, so they could finish it in a shorter time, as can be seen in table 3.

In the table, *Precision* denotes the precision of the 2D slider test (the distance between the centre of the slider button and the desired path). *Improvement* denotes the difference between the corresponding values in the tests with one and three sensors. The higher the improvement, the better the results with three sensors. You can also find values connected to a paired t-test conducted, which we used to compare the means of the results of one- and three-sensor setups. t-value and p-value are computed with a critical value corresponding with the DOF for a 95% confidence interval. For the overall time, time for the 2D slider and the precision for the 2D slider, we can reject the null hypothesis of no difference and say with a high degree of confidence that the true difference in means is not equal to zero. N/A means that the value is not applicable to the number of finished users.

From the histogram in figure 8, it can be seen that when the users worked with three sensors, it was again much easier for them to finish the task because they achieved it in a shorter time.

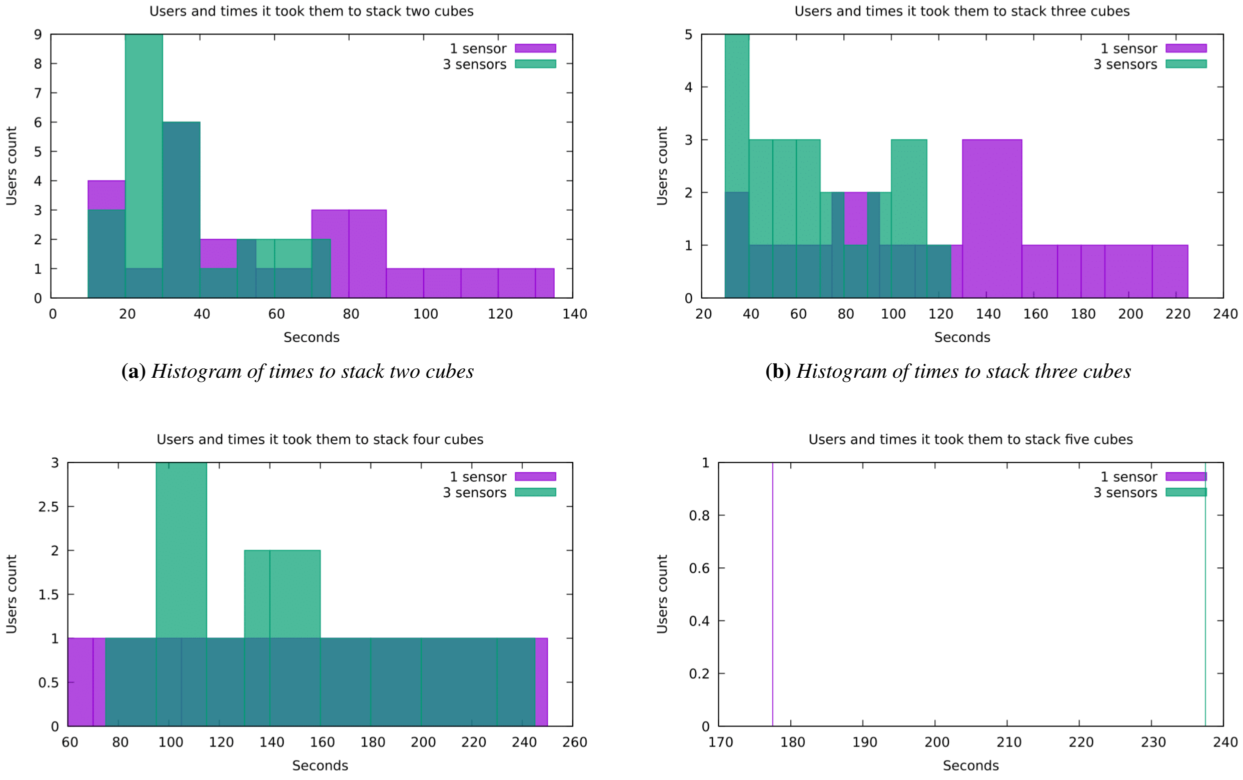
76% of the users stated that they preferred to work with three sensors, 8% said that they preferred one sensor, and 16% could not decide.

5. Discussion

The results show that using multiple optical sensors can improve the user’s work both in terms of time and precision. The only deviation is the *Cubes* test, where the fact that more people were able to finish the test had a negative effect on the average time to finish it. However, this is logical because with one sensor, only the most skilful users were able to build a tower out of four cubes, but even the less skilful were able to do so with multiple sensors. Unfortunately, the cubes were too high, and when four of them were stacked on top of each other, the user’s hand was too high from

	One sensor	Three sensors	Improvement	Mean	stdev	t-value	DOF	p-value
Two cubes	0:59	0:34	42%	25.40	36.10	3.52	24	0.002
Three cubes	1:57	1:05	44%	33.78	61.74	2.19	17	0.005
Four cubes	2:04	2:25	-17%	-5.5	50.63	0.22	3	N/A
Five cubes	2:59	3:59	-34%	-30	295.57	0.10	1	N/A

Table 2: *Times to complete the Cubes task*



(a) *Histogram of times to stack two cubes*

(b) *Histogram of times to stack three cubes*

(c) *Histogram of times to stack four cubes*

(d) *Histogram of times to stack five cubes*

Figure 7: *Histograms of users and times it took them to stack two (a), three (b), four (c) and five (d) cubes*

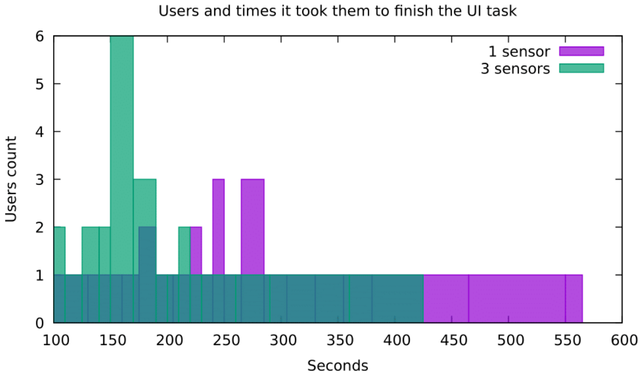


Figure 8: *Histogram of users and times it took them to finish the UI task.*

the sensor; it reached the upper tracking range of the sensor and the hand very often disappeared. Thus, it was almost impossible to stack five cubes on top of each other, effectively invalidating the results of the five-cube height tower test.

The users often complained about some physical aspects of the scenes, for example, the insufficient weight of the cubes, which made the tower easy to fall, or that it was hard to determine the depth of the scene (the distance of the hand from the UI elements). Some of them also complained about their arms hurting after a long time working with the hand-tracking controllers – an effect known as Gorilla Arms [HPM*17]. However, these complaints were the same for both one and three sensors since the problem is connected to the scenes and/or controllers for VR in general and cannot be fixed by adding more sensors.

Proceedings published by Eurographics - The European Association for Computer Graphics.

	One sensor	Three sensors	Improvement	Mean	stdev	t-value	DOF	p-value
# of finished	20	25	25%	N/A	N/A	N/A	N/A	N/A
Overall time	4:25	3:04	31%	86.6	123.68	3.5	19	$55x10^{-4}$
Time (2D slider)	11.57s	7.00s	40%	15.64	16.44	4.76	24	$77x10^{-6}$
Precision (2D slider)	36.72	21.09	43%	4.59	6.06	3.79	24	$90x10^{-5}$

Table 3: *Results of the UI task*

As far as the hypotheses are concerned, only three out of four were supported.

H1: Three sensors will provide smoother and more precise interaction was supported since users described the tracking with three sensors as smoother in both scenes. They were also able to build higher towers in shorter times with the *Cube* scene and shorter times and more precise control of the elements in *UI* scene.

H2: With three sensors, it will be easier to work with the elements that are far from the centre of the scene was also supported in both tests. In the *Cubes* scene, users were able to grab cubes that were pushed or fell away further from the centre of the scene. In the *UI* scene, using three sensors also resolved the cases where the hand would disappear before the dial was pressed, which previously made finishing this task impossible for some users.

H3: With three sensors, testers will be able to finish the tasks faster was fully supported in the *UI* test and partially confirmed in the *Cubes* test. When three sensors were used in the *UI* scene, the users were able to press the dial buttons faster and with fewer errors because the tracking was more precise and the tracked hand was not occluded that often. The 2D slider was also easier to operate, and the users were able to finish the task with a lower error rate.

In the *Cubes* test, stacking two and three cubes on top of each other was faster with three sensors but slower for four and five cubes. However, three sensors drastically improved how many users were able to stack four cubes on top of each other.

H4: Users will prefer hand-tracking when working with objects in 3D space, but they will prefer a keyboard and mouse for working with UI elements was confirmed only partially. In the *Cubes* test, 60% of the users stated that they would rather use a keyboard and mouse to control the scene (20% could not decide, 16% said it would be the same). In the *UI* test, 80% of the users stated that they would rather use a keyboard and mouse to control the scene (12% could not decide). This means that the users would prefer a keyboard and mouse for both scenes, which contradicts our hypothesis that they would prefer it just for the *UI* test.

Every complaint that the users had when working with one sensor was resolved by using multiple sensors. The rest of the complaints were connected to the test scenes in general (e.g. the lack of weight of the cubes or the hand-object collision detection). This was not addressed during the course of the experiment because, although it could improve the satisfaction of the users, it was not affecting the goal of the experiment as a whole.

Some problems, for example, the hand-object collision detection or grip sensitivity, are handled by the Ultraleap plugin for Unity and were out of the scope of our research.

6. Conclusion and future work

In this research, we compared the precision and ease of use of optical hand-tracking for virtual and augmented reality, more specifically when one and three optical sensors are used. For synchronizing the data from multiple optical sensors, we used the algorithms from the MultiLeap library presented in our previous research.

We created two test scenes that simulated real-life scenarios, one where the users worked with a simple user interface and one where the users were building a tower from cubes, both operated by hand-tracking. Twenty-five users participated in the tests.

This research showed that using multiple optical sensors for hand-tracking greatly improves the precision of the tracking, widens the tracking range and provides smoother interaction. On average, 80% of the users preferred using three sensors for the interaction because it allowed them to finish more tasks in a shorter time and with more precise results.

Using three sensors also helped more users to achieve desired goals, where, in some cases, there was a 160% improvement in the number of finished tasks.

Our future work will include an algorithm that combines the hand-tracking sensors with positional sensors, like SteamVR tracking [Val20], which will create a possibility to have one (or more) of the hand-tracking sensors in motion, for example, attached to a head-mounted display.

References

[AK22] Aeberhard M., Kaempchen N.: High-Level Sensor Data Fusion Architecture for Vehicle Surround Environment Perception. *Proceedings of 8th International Workshop on Intelligent Transportation* (11 2022). 3

[ARM01] Arambel P., Rago C., Mehra R.: Covariance intersection algorithm for distributed spacecraft state estimation. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)* (2001), vol. 6, pp. 4398–4403 vol.6. doi:10.1109/ACC.2001.945670. 3

[FGCT15] Fok K.-Y., Ganganath N., Cheng C.-T., Tse C. K.: A Real-Time ASL Recognition System Using Leap Motion Sensors. In *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (2015), pp. 411–414. doi:10.1109/CyberC.2015.81. 3

[Flo05] Floridi L.: The Philosophy of Presence: From Epistemic Failure to Successful Observation. *Presence: Teleoperators and Virtual Environments 14*, 6 (12 2005), 656–667. URL: https://doi.org/10.1162/105474605775196553, arXiv:https://direct.mit.edu/pvar/article-pdf/14/6/656/1624373/105474605775196553.pdf, doi:10.1162/105474605775196553. 4

[HPM*17] Hansberger J., Peng C., Mathis S., Areyur Shanthakumar V., Meacham S., Cao L., Blakely V.: Dispelling the gorilla arm syndrome: The viability of prolonged gesture interactions. In *Virtual, Augmented and Mixed Reality* (07 2017), pp. 505–520. doi:10.1007/978-3-319-57987-0_41. 7

[HZW*17] Hu T., Zhu X., Wang X., Wang T., Junfeng L., Qian W.: Human stochastic closed-loop behavior for master-slave teleoperation using multi-leap-motion sensor. *Science China Technological Sciences 60* (01 2017). doi:10.1007/s11431-016-0434-x. 3

[Kab76] Kabsch W.: A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A 32*, 5 (Sep 1976), 922–923. URL: https://doi.org/10.1107/S0567739476001873, doi:10.1107/S0567739476001873. 3

[KKC19] Kiselev V., Khlamov M., Chuvilin K.: Hand Gesture Recognition with Multiple Leap Motion Devices. In *2019 24th Conference of Open Innovations Association (FRUCT)* (Moscow, Russia, Russia, 04 2019), IEEE, pp. 163–169. URL: https://ieeexplore.ieee.org/document/8711887, doi:10.23919/FRUCT.2019.8711887. 2

[NJ21] Novacek T., Jirina M.: Project Multileap: Making Multiple Hand Tracking Sensors to Act Like One. In *Proceedings of 2021 IEEE 4th International Conference on Artificial Intelligence and Virtual Reality (AIVR 2021)* (Taichung, Taiwan, 2021), AIVR 2021, IEEE, pp. 19–26. 1, 3, 4

[NJ22] Novacek T., Jirina M.: Overview of Controllers of User Interface for Virtual Reality. *PRESENCE: Virtual and Augmented Reality 29* (07 2022), 37–90. URL: https://doi.org/10.1162/pres_a_00356, arXiv:https://direct.mit.edu/pvar/article-pdf/doi/10.1162/pres_a_00356/2036227/pres_a_00356.pdf, doi:10.1162/pres_a_00356. 2

[NMJ21] Novacek T., Martin C., Jirina M.: Project Multileap: Fusing Data from Multiple Leap Motion Sensors. In *Proceedings of 2021 IEEE 7th International Conference on Virtual Reality (ICVR 2021)* (New York, NY, USA, 2021), ICVR 2021, IEEE, pp. 19–26. 1, 3, 4

[PAC*21] Placidi G., Avola D., Cinque L., Polsinelli M., Theodoridou E., Tavares J.: Data integration by two-sensors in a LEAP-based Virtual Glove for human-system interaction. *Multimedia Tools and Applications 80* (05 2021), 18263—-18277. URL: https://link.springer.com/content/pdf/10.1007/s11042-020-10296-8.pdf, doi:10.1007/s11042-020-10296-8. 2

[Ult22a] Ultraleap: Leap Motion Controller, Oct. 2022. Accessed 31 October 2022. URL: https://www.ultraleap.com/product/leap-motion-controller/. 3

[Ult22b] Ultraleap: Ultraleap Stereo IR 170, Oct. 2022. Accessed 31 October 2022. URL: https://www.ultraleap.com/product/stereo-ir-170/. 1, 3

[Ult23] Ultraleap: Ultraleap plugin for Unity, Mar. 2023. Accessed 19 Mar 2023. URL: https://developer.leapmotion.com/unity. 5

[Uni23] Unity: Unity, Mar. 2023. Accessed 19 Mar 2023. URL: https://unity.com/. 4

[Val20] Valve: SteamVR Tracking, Dec. 2020. Accessed 13 Dec 2020. URL: https://partner.steamgames.com/vrlicensing. 8

[WWJ*21] Wang Y., Wu Y., Jung S., Hoermann S., Yao S., Lindeman R.: Enlarging the Usable Hand Tracking Area by Using Multiple Leap Motion Controllers in VR. *IEEE Sensors Journal 21*, 16 (05 2021), 17947–17961. doi:10.1109/JSEN.2021.3082988. 2