https://doi.org/10.1016/j.cag.2018.04.004.

Preprint Version. Manuscript submitted to and accepted by Computers & Graphics

Mid-air Interaction with Optical Tracking for 3D Modeling

Abstract

Compared to common 2D interaction done with mouse and other 2D tracking devices, 3D hand tracking with low-cost optical cameras can provide more degrees of freedom, as well as natural gestures, when shape modeling is done in virtual spaces. However, though quite precise, the optical tracking devices cannot avoid problems intrinsic to hand interaction, such as hand tremor and jump release, and they also introduce an additional problem of hand occlusion. We investigate how to minimize the negative impact of these problems, and eventually propose to use hands in a way similar to how it is done when playing the theremin – an electronic musical instrument controlled without physical contact by hands of the performer. We suggest that the dominant hand controls manipulation and deformation of objects while the non-dominant hand controls grasping, releasing and precision of interaction. Based on this method, we describe a generic set of reliable and precise interaction gestures for various manipulation and deformation tasks. We then prove with the user study that for the tasks involving 3D manipulations and deformations, hand interaction is faster than common 2D interaction done with mouse.

1. Introduction

We use our hands in many ways for various modeling and assembling tasks. For example, different ways of grasping are developed for taking objects with different sizes and shapes. To deform an object, we may poke, press or smooth it by one or several fingers or any other parts of a hand. Two hands are used for twisting, bending or squeezing, where motions of wrist, shoulder, or the whole arm are involved. Moreover, the second hand can be used either in a symmetric way, e.g., to hold large objects together with the first hand, or for auxiliary control, e.g., to constrain motion of the object. The developed hand interaction ways allow us to deal with the most basic modeling tasks. However, when working on the precision tasks, two intrinsic problems of hand interaction exist: hand tremor and jump release. Hands tend to tremor when there is no physical support. Jump release causes the object to slightly displace when the fingers are being removed from it. These problems seriously affect the precision of object placement, especially for small components.

Historically, 3D modeling with computer uses 2D tracking devices, such as mouse and touchpad, which are based on tracking positions of one or a few points on a plane. As a result, while minimizing the negative impact of tremor and jump release in the modeling process, the plethora of natural hand gestures was replaced with only a few basic 3D manipulations which are often rather unnatural and even inefficient when implemented using successive 2D tracking operations. For example, an arbitrary 3D rotation is split into several subsequent rotations around principal axes.

Progress in computer vision allows mid-air hand motion to be tracked by affordable optical sensors, such as Leap Motion controller, uSens Fingo and Creative Senz3D. This way of interaction has boosted many applications in VR games and entertainment, however, very little was done in using mid-air interaction in precise shape modeling and virtual assembling [1]. In this paper, we investigate whether interaction in 3D modeling can be improved by optical hand tracking so that 3D tasks can be performed as natural as in real life and as efficient as using common 2D-tracking based interaction while the problems created by hand tremor, jump release, and hand occlusions are minimized. We survey the related works in Section 2 and test a few common optical sensors in

Section 3. We then illustrate in Section 4 what happens when common mid-air interaction ways are implemented for precise shape modeling and propose a new way of interaction in Section 5. The proposed method is validated by the user study described in Section 6. Discussion and conclusions are given in Section 7 and 8.

2. Shape modeling with hands – from real-life to digital

2.1. Limitations of natural hand motion

Before considering mid-air interaction for solving modeling and assembling tasks, it is important to understand the precision limits of hand motion in real-life interaction tasks.

In various studies, e.g., in [2], it was observed that precise in-hand manipulation has the best performance only when the sizes of objects range from 48 mm to 59 mm. Due to the constraints imposed on finger joint motion, in-hand rotation capabilities are also quite limited, depending on the direction of rotation axis [3, 4]. Interaction becomes extremely difficult when the size of an object is smaller than 5 mm, e.g., it may become challenging to screw a small nut onto a bolt or to thread a needle. Although various precision tools (such as a tweezers, needle threaders, etc.) can be used, hand tremor still adds to the difficulty of operations in such scale. The tremor amplitude varies between $0.4 \pm 0.2mm$ for young and $1.1 \pm 0.6mm$ for older people [5, 6]. Hand tremor is also closely associated with jump release which may happen when fingers are not simultaneously removed from the surface of the object while it is being placed. A little momentum created by the delayed finger motion may slightly displace the object. Furthermore, these problems may be further affected by arm fatigue which is resulted from long-time operations.

Therefore, although hand tracking devices are quite precise and they are able to capture sub-millimetre displacements of the fingers, direct transfer of our real-life hand interaction to virtual environments may not be always feasible for achieving high efficiency of interactive shape modeling. On the other hand, like when using tools and magnifying glasses in real life, the intrinsic problems of hand manipulations can be potentially reduced by using constraints and varying precision and coordinate mapping in virtual modeling environments. Keeping the limitations of hand motion in mind, we explore the ways of virtual interaction that can minimize or totally avoid negative effects of these problems.

2.2. Interaction using 2D tracking devices

Common 2D tracking devices can be efficiently used for 2D tasks (e.g., sketching and manipulation on a plane) and some 3D tasks that can be mapped into 2D (e.g., 3D object selection with the ray casting method). A common optical mouse has an accuracy of 800 to 1600 dpi (dots per inch), e.g., [7, 8], which corresponds to 31.8 to $15.9\mu m$. There are, however, tests showing that the actual accuracy is in fact lower than that. For example, a mouse sensor ADNS6010 with stated $12.7\mu m$ (2000 dpi) was tested in [9]. The position error when moving 1024 μm was 144 μm for static measuring and the maximum error was 187 μm for dynamic measuring. The obtained accuracy still allows a standard mouse to be used precisely for the purpose of shape modeling. However, when it comes to 3D operations such 2D tracking devices become inefficient due to limited motion space and degrees of freedom. Special interaction techniques were developed for such tasks to be performed with 2D tracking devices.

3D widget (also called 3D gizmo or manipulator) is a common method used by most shape modeling tools, e.g., Autodesk Maya. Manipulation of an object is performed by interacting with translation, rotation and scale widgets embedded into it. Some special widgets were also designed for deformations, such as tapering, twisting, bending and their combination [10]. The widgets interactively control the parameters, including range, extent and direction of deformation. They can be efficiently and precisely used with 2DOF operations, but become inefficient for arbitrary operations of more degrees of freedom. For example, aligning two objects can be divided into a sequence of translations and rotations. Aside from widgets, 3D tasks have also been implicitly mapped to 2D space using various intuitive techniques. Virtual trackball, Shoemake's Arcball and Two-arc Valuator methods allow 2DOF of 3D rotation to be controlled simultaneously by a mouse [11]. A sliding method was proposed for 3D translation in [12], where the object being moved was sliding along another one. Although for some specific cases interaction with 2D tracking devices can be designed in a more efficient way than 3D tracking [13], the insufficiency of DOFs always makes it not as convenient as direct manipulation in 3D space.

Interaction with 2D tracking devices can be potentially improved if we use dexterous motion of hand and fingers mimicking the way how we use them in real life. However, it should also be understood that continuous holding hands in the mid-air without physical support will cause arm fatigue [14]. In order to further explore whether existing interaction can be improved with 3D hand motion, we will survey the commonly available hand tracking devices, as well as related virtual interaction techniques. In order to achieve the desired accuracy of modeling, we expect from the tracking device a precision comparable with the accuracy of a common optical mouse.

2.3. Technologies of 3D hand motion tracking

Glove-based and vision-based systems are commonly used to track motions of hand and fingers. Other tracking methods, such as multi-fingered exoskeleton haptic devices, acoustic and light sensors, and electrical signals in muscles [15], are either too expensive, not accurate enough, or still in their early stages of development.

The major component of a glove-based tracking system is a cloth glove with special bend sensors sewn inside to measure the bending angles of the finger joints. Tracking of global hand motion requires additional trackers to be put on hand or wrist. Absolute positions of fingertips cannot be directly tracked, but can only be calculated according to the global hand position and bending extent of the fingers. The global hand motion can be tracked quite precisely, e.g., a professional optical tracking system OptiTrack Prime 41 has a precision of up to 0.2 mm [16]. Therefore, the precision of glove-based systems when tracking the global hand motion is comparable to that of the mouse. However, it becomes less accurate when tracking the tip of a bent finger, the accuracy of which is affected by several problems including nonlinearity, cross-coupling and noise. Therefore, when tracking separation distance of a pinching gesture, which was performed by connecting thumb tip and index fingertip, the actual separation distance when using the professional glove-based system CyberGlove was measured as 7.53 cm from raw data, and 5.25 cm after linear calibration [17].

Vision-based systems track hand motion with cameras and analyse the hand postures from the captured images by using computer vision algorithms. Absolute positions of palm and fingertips can be located by using optical techniques. For example, Leap Motion controller tracks hand motion in 3D space using two monochromatic infrared cameras. The accuracy of tracking the tip of a pen was within 0.2 mm when the pen was static, and 2.5 mm when the pen was moving [18]. Another test [19] showed that the accuracy of tracking of a static fingertip was less than 0.01 mm in a 30 cm area above the device, while the overall accuracy was below 0.5 mm in its entire sensory area. Therefore, vision-based systems show a better performance than glove-based systems when tracking the absolute position of a single fingertip. However, the precision decreases significantly when it comes to tracking fingertips which are located close to each other. According to [20], the error of Leap Motion controller was 0.878 cm when the distance between the thumb tip and index fingertip was 1 cm. Only when the separation distance was larger than 5 cm, the error was smaller than 0.04 cm This result was also confirmed in [19], which reported that the device failed to track the constant tip distance (21.36 mm, *sd* = 0.023 mm) of two sticks – the tracking was so unstable that the deviation was not limited to 0.8 cm.

In summary, both glove-based systems and vision-based tracking have their own pros and cons. Glove-based systems have a stable performance of tracking hand motion, but the tracked fingertips may shift when the fingers are bent. Vision-based systems can track absolute positions quite precisely, but their performance is inconsistent and unpredictable due to wrong estimation of the ambiguous images of occluded hands. Both of these systems can have a precision comparable to the mouse when tracking the position of a single point on hand. However, we believe that using vision-based tracking is a more promising and suitable for the purpose of designing interaction for shape modeling. On one hand, optical sensors are more convenient and affordable rather than glove-based systems for shape modeling on a daily base. On the other hand, vision-based tracking has received much research attention, which resulted in improvements on the tracking quality in recent years. Thus, works [21, 22] already show that the previously mentioned major problems of optical tracking have been solved to some extent. Optical tracking is becoming more popular than tracking with glove-based systems. Therefore, we have selected vision-based technology for hand tracking to conduct the research in this paper.

2.4. Natural hand interaction

With reference to whether collision with virtual hands (driven by real hands) is involved, natural interaction techniques can be classified into two types: collision- and gesture-based methods.

Collision-based methods mimic natural interaction by implementing collisions between virtual hands and objects. So-called, “colliders” are assigned to virtual hands and objects, and the interaction is computed either purely algorithmically [23] or using a physics engine. Real-life non-prehensile interaction ways, such as pushing and hitting with a hand, can be realistically simulated by a physics engine, while additional algorithms [24, 25, 26, 27] need to be implemented for grasping in order to define the frictions and prevent finger penetration. Aside from rigid interaction, some methods were also proposed to define interaction with deformable objects, such as a piece of clay [28], freeform surface [29], and elastic objects [30]. Collision-based interaction is thus intrinsically natural – real-life interaction ways can be directly used in the system without much explanation to the users.

Generally, collision-based methods are computationally expensive to be implemented in complex virtual environments. For example, in Unity 3D system a mesh collider can only be generated on a convex mesh which has less than 256 triangles. In contrast, gesture-based methods do not require collisions to be calculated, which thus makes them to be more efficiently used, as well as they are not limited by the size of the mesh. Gesture-based methods control different interaction with a set of pre-defined gestures, which can be recognized directly from the images captured by cameras or computed from the articulated hand model. Such methods can be natural when gestures used in real-life are implemented for virtual interaction. For example, the pointing gesture [31, 32] combined with ray casting method can be used for virtual object selection. Mimicking how we take objects with thumb and fingers, the pinching gesture was implemented for grabbing objects, followed by translation and rotation by moving hands or wrist rotation [33]. However, the methods can only allow some simplified interaction ways to be implemented, which makes it less realistic than using the collision-based methods. For example, natural interaction ways, such as claying with fingers, cannot be realistically simulated with such methods.

Both collision-based and gesture-based methods are affected by the problems of hand tremor and jump release, which become serious when small-scaled hand motion is involved. For collision-based interaction, it is difficult to precisely release the object being grasped when the tracked fingers sink into virtual objects. These problems were solved by several ways including 1) provision of additional visual feedback with a hope to use a light grasp [34], 2) optimization of the algorithm for object releasing [35], and 3) design of additional algorithms for enhancing the stability, e.g., to dynamically change the precision according to the hand motion speed [36]. In addition, occlusion of fingers, when a hand is used for holding objects, can be estimated as proposed in [37]. However, with one optical sensor the occlusion problem cannot be solved in principal. Both collision-based and gesture-based interactions have their own advantages and intrinsic problems when they are used for implementing efficient and natural interaction. However, there is no apparent evidence which method is more suitable to be used in shape modeling interaction to achieve the goal of making virtual interaction as natural as in real life and as efficient as common 2D-tracking based interaction. Therefore, both methods should be tested in order to have a better understanding of their limits. In the following two sections, we select an optical hand tracking device and implement natural interaction algorithms with it.

3. Optical sensors selection and test

3.1. Optical sensors comparison

There are many kinds of commercial optical sensors available in the market. In order to implement real-life interaction, we consider such sensors that can track small-scale hand and finger motion with high precision. Available sensors that can satisfy our requirements are Creative Senz3D, uSens Fingo and Leap Motion controller. They can track both hand and finger motions with their official SDKs. Besides them, Microsoft Kinect for Xbox One also has the potential tracking capability to track finger motion [38]. However, its official SDK mostly concentrates on tracking the whole body motion, while a hand is simplified to three points only. Therefore, we only tested Creative Senz3D, uSens Fingo and Leap Motion controller.

All of the three sensors can track small-scale hand motion in the area above or in front of the device. Creative Senz3D is designed to be placed on the monitor and track hand motion in front of it. uSens Fingo is to be used in combination with a head mounted display (HMD), i.e., to track hands in front of it. Leap Motion controller can either track hand motion above it, or in front of it when it is attached to HMD. The initial test showed that the hand tracking quality with Leap Motion controller was significantly better than that of Creative Senz3D. Seriously affected by occlusions, tracking of fingers and palm motion by Creative Senz3D was very discontinuous, which was prohibitive for fine modeling tasks. Both Leap Motion controller and uSens Fingo have a good quality of tracking global hand motion, as well as finger motion when they are not occluded. Leap Motion controller has a better performance in estimating the finger occlusion.

Leap Motion controller was then selected as the tracking device for this research. This is also because Leap Motion controller provides C++ SDK for Windows, while uSens Fingo only provides Unity SDK, which limits its possible application. We will still keep the tracking capabilities of the other two devices in mind, and consider to further propagate to these sensors what we will design for Leap Motion controller.

3.2. Testing Leap Motion controller

In the initial test, Leap Motion controller was placed on a desk to track hand motion above it. It was observed that the effective range of the device, where hand and finger motions can be reliably tracked, was around 40 cm in width and 40 cm in depth. The reliable vertical tracking area starts from 10 cm above the device up to the height of more than the maximum reach of hands when sitting in front of the desk. One hand can slightly overlap the other without loss of tracking. Palm position and orientation could be more reliably tracked than finger motion, and therefore to be preferred for tracking as input for 3D modeling. Fingers motion can only be tracked when they are not occluded. However, even when the fingers are facing the device, the separation distance between the thumb tip and the fingertips can not be tracked when it is below 3 cm, which accords with the tests referenced in Section 2.3.

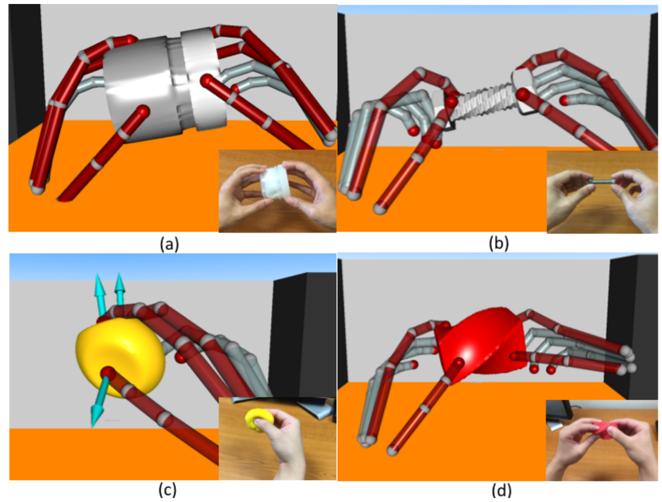

Figure 1: Real-life and virtual tasks in the experiment of collision-based interaction. (a) Putting a cap onto a container. (b) Screwing a nut onto a bolt. (c) Making elastic deformation with one hand and (d) twisting deformation done by two hands.

4. Implementation of natural collision-based and gesture-based interaction

Both collision-based and gesture-based interaction should be tested with an emphasis paid to the natural way of interaction, i.e., we attempted to exactly mimic in the virtual modeling environment the ways we use hands in the real world.

As a test-bed, we designed and implemented four typical hand interaction tasks for collision-based interaction, which were /1/ putting a cap onto a container, /2/ screwing a nut onto a bolt, /3/ making elastic deformation with one hand, and /4/ twisting deformation done by two hand, as shown in Figure 1. We attempted to exactly mimic in the virtual modeling environment the ways we use hands in the real world. Hand motion was restricted within the best tracking area, as tested in 3.2, with a width and depth of 40 cm and a height from 10 cm to 40 cm. 1:1 coordinate mapping of the tracking area to the virtual modeling space was used. The sizes of virtual objects were such that they could be grasped with the distance of 3 cm between the thumb and the fingers, as measured in Section 3.2.

To make a lower benchmark, we used our own implementation of the collision detection rather than a physics

Table 1: Time spent on real-life and virtual interaction of each task

	Task 1		Task 2		Task 3		Task 4	
	Real	Virtual	Real	Virtual	Real	Virtual	Real	Virtual
Average time (sec.)	3.8	9.1	4.4	17.2	3.5	6.6	3.2	7.6
Standard deviation	1.2	2.1	1.4	4.5	1.4	2.6	0.9	3.9

engine. In the implementation, five point "colliders" were put on the five fingertips. The shapes were defined using implicit functions, which allowed for precise collision computation between virtual fingers and objects. In addition, correction algorithms were also devised to visually prevent finger penetration into objects. Visual feedback in forms of changing finger colors and displaying arrows indicating force exerted was also provided in order to compensate the loss of force feedback. Following the idea proposed in [39], the precision of interaction had to be enhanced when the hand motion speed is low. This had to allow the virtual objects to be precisely placed while minimizing the problems of hand tremor and jump release. We then proceeded to the user study to compare the virtual interaction with the real-life interaction.

In the user study, 16 participants (8 male, 8 female) were invited for the study. Average age was 28 (sd = 3.12). 5 of them had an experience with mid-air interaction. The mid-air interaction devices which they used before were Leap Motion controller, Microsoft Kinect, Wii and HTC VIVE. They were required to perform the real-life tasks followed by the corresponding virtual interaction. The time they spent on each task was recorded.

T-test was performed to compare the time of virtual and real-life interaction in each task. However, with reference to Table 1, the results showed that the virtual interaction was significantly slower than the real-life interaction ($p < 0.01$ for all the tasks). The biggest time difference was observed in Task 2, where quite similar hand movements were involved as in Task 1, but the sizes of virtual objects were smaller. We tried several ways to improve the efficiency of virtual interaction, including using stereo display with Oculus Rift and making virtual grasping easier. However, the improvement we could achieve was very incremental. Several intrinsic problems summarized below prohibited us to further continue with the collision-based interaction.

- Missing of touch feedback (which was expected in collision-based interaction) made it difficult to transfer real-life interaction ways efficiently to virtual interaction. It took more time to reach and touch virtual objects, which became even more difficult when the sizes of objects were small. Screwing the nut with fingertips became clumsy in virtual interaction. Indeed, it was observed by Garbaya and Zaldivar-Colado [40] that adding force feedbacks with the multi-fingered exoskeleton haptic device made virtual tasks to be done more efficiently.

- The quality of collision-based interaction highly depends on the accuracy of the tracking technology. With the selected optical sensor, virtual finger motion could not be controlled precisely due to the occlusion problem and limited accuracy of tracking some hand postures.

- Although the virtual finger penetration was corrected visually, the actual unconstrained finger penetration caused difficulty in releasing objects, which was even more serious when a finger sank deeply into the object. This case prevented the problem of jump release from being completely solved algorithmically.

Next, we proceeded to the gesture-based interaction. The idea was to recognise many gestures with only a few algorithms [41, 42] while the precision was controlled dynamically during releasing the objects. We also avoided as much as possible to display the virtual hand but rather displayed objects moved by it. We designed a set of real-life interaction tasks involving manipulations of virtual and real objects. Paying attention to efficiency, we requested the users in the study to perform the same simple operations on virtual and real objects and compared the time spent on the tasks (refer to Figure 2).

In contrast to the collision-based interaction, the participants could complete the tasks within the allocated time, however the designed interaction was still difficult to be

Figure 2: The tasks of gesture-based interaction.

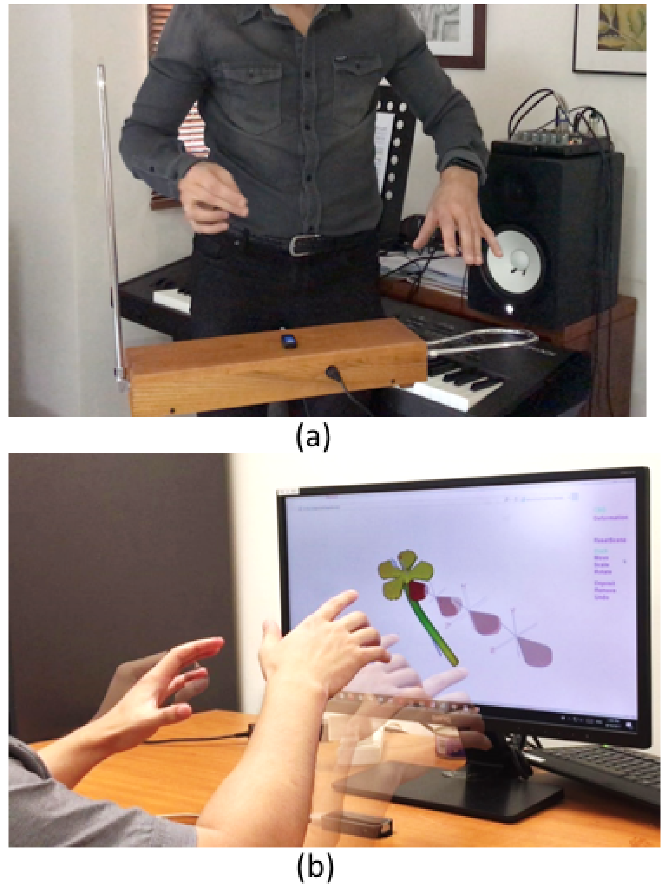

Figure 3: (a) Playing the theremin (Moog Etherwave Standard). (b) Theremin-style gestures proposed for 3D modeling in mid-air.

performed precisely, mostly due to the occlusion problem happening when two hands were close to each other, as well as when self-occlusions happened during various twisting gestures performed by one hand. Despite that some of the participants gradually learned how to adjust their hand postures and positions to minimize occlusion during interaction, we still could not achieve the desired perfection of interaction.

According to the results, we concluded that the gesture-based interaction was more promising than collision-based methods to achieve higher efficiency with optical tracking. Several problems prevented collision-based interaction to be used as efficient as real-life interaction. In comparison, gesture-based interaction saved the time used for colliding with virtual objects. The performance did not drop down significantly when dealing with small objects because collision computation was avoided. The occlusion then remained to be the main problem, and the main challenge became to find a way how to eliminate it.

5. Making shapes like playing music

5.1. Principles of Interaction

With the goal to eliminate the negative effect of the occlusion problem, we propose to use hands in a way similar to how it is done when playing the theremin – an electronic musical instrument controlled without physical contact by hands of the performer [43]. Theremin consists of two metal antennas that sense the relative positions of the thereminist's hands. The pitch of notes is controlled by one hand and the volume is controlled by the other hand (Figure 3(a)).

The hands never make any occlusion and the precision of the instrument can be tuned for each individual player by adjusting the range of the notes and volumes. We then hypothesized that if fine music can be played by moving hands in the mid-air without any physical support, we then should be able to make fine shapes in the virtual space by moving hands in the mid-air as well. Therefore, we decided to consider such kind of bimanual interaction where grasping and moving are separated between two hands (Figure 3(b)), specifically:

- The hand tracking space is divided into two unequal and non-intersecting zones: one bigger zone for tracking the dominant hand and another smaller zone for tracking the non-dominant hand.
- The dominant hand is only tracked for changing position and orientation. This can be the tracking position of the whole hand or individual fingers.
- The non-dominant hand is tracked for triggering two events – grasping and releasing – as well as for continuous changing of the precision of operation. Depending on the capability of the tracking device, it can be done either by moving the hand up and down, as the thereminist is doing to change the volume, or by closing/opening fingers as in the grasping gesture.
- Whenever it is possible, we do not display virtual hands but rather objects which are being manipulated by them.

In the proposed way of interaction, both hands are still doing natural grasping and motion. Using separate motion spaces, they will not occlude each other, as when the theremin is played. If the hands will be facing the sensor, the finger motion can be reliably tracked. There will be no jump release either because the release event, triggered by non-dominant hand, will not affect the object motion controlled by the dominant hand. Moreover, an ability to continuously change the precision of operation with another hand will allow us to solve the problem of hand jitter since the precision can be increased gradually before releasing the object.

Based on our experiments with natural gestures described in Section 4, we also believe we will achieve higher efficiency if we avoid displaying virtual hands, since observation of the motion of the virtual objects/instruments controlled by the hands may be more essential for the user than the ability to see the simulated hand. However, whenever needed, a simplified not obstructing virtual representations of the fingertips or palms still can be shown.

5.2. Technical details of interaction design using Leap Motion controller

The specifics of Leap Motion controller, compared to other hand tracking devices considered in Section 3, is in its ability to precisely track individual fingers. Therefore, motions of fingers from the non-dominant hand were tracked to recognise the grasping/releasing gesture. To minimize finger occlusion, the palm of non-dominant hand should face the device while performing the grasping and releasing motion, which is also ergonomically comfortable for the user. Specifically, the palm should face down when Leap Motion controller is used, while should face the user when uSens Fingo is attached to the HMD. Alternatively, the user can opt for moving the non-dominant hand up and down instead, which can be used for tracking by the devices that cannot track finger motion (e.g., by Creative Senz3D).

The palm motion of dominant hand is tracked for most of the operations, while fingers are only tracked when they can improve existing interaction and on the condition that they are not occluded. The user can hold the hand in any convenient way (with all the fingers extended or with the fingers bent), since it is the hand position and orientation which will be tracked by the controller.

In Figure 4, the two tracking areas set for Leap Motion controller are illustrated without showing the depth dimension. The same best tracking area is used, as tested in 3.2, with a width of 40 cm and a height from 10 cm to 40 cm above the device, as well as the minimum separation distance of 3 cm between thumb and fingers for the non-dominant hand. In this hand motion range, elbows can be rested on the surface of the desk so that arm fatigue can be minimized during interaction. However, it is essential that the previously fixed 1:1 coordinate mapping becomes variable when the precision of operation is being continuously changed by the non-dominant hand. It allows us to filter the hand tremor and jump release displacements as well as to increase the overall precision of modeling.

Since there are no any visible boundaries in the physical hand tracking space, we provide visual feedback to keep the hands within their tracking areas. Thus, if the dominant hand moves outside its tracking area, the respective 3D cursor or object controlled by it will stop following the hand until it returns back to the tracking area. For the non-dominant hand, there is also a visual indicator of the degree of grasping which also works only when the hand is located within its tracking area.

The extent of grasping is measured by the height of the non-dominant hand or distance between the thumb tip and the index fingertip, which is denoted as D. Suppose the maximum and minimum thresholds of D are D_{max} and D_{min}. When the object is firmly grasped by the non-dominant hand ($D \leqslant D_{min}$), the control/display (C/D) ratio r is set as the minimum value r_1 so that 3D displacement of the dominant hand maps to the entire visible shape modeling area, i.e. motion from corner to corner in the tracking space will result in the cursor/object motion from corner to corner in the modeling space. Also, rotation of the wrist will map to the maximum angle of rotation of the object in the modeling space. When the non-dominant hand is open ($D \geqslant D_{max}$), the virtual object will no longer be controlled by hand motion so that $r = \infty$. To minimize the negative effect of hand tremor, when releasing the object (by slowly opening or moving up the non-dominant hand $D \rightarrow D_{max}, D > D_{min}$) the C/D ratio $r = r_1 + (D - D_{min}) * (r_2 - r_1)/(D_{max} - D_{min})$ is linearly increasing towards the maximum value r_2 so that bigger displacements of the hand result in smaller displacements of the cursor/object, and bigger angles of the wrist rotation will result in smaller rotation angles in the modeling space.

5.3. Design of the Actual Gestures and Interaction

The proposed bimanual interaction way allows any actual operations to be implemented. We considered designing an interaction interface with basic functions including menu selection, object manipulation and deformation, as well as scene manipulation. In addition, we wanted to allow the actual interaction techniques to be used by all kinds of optical tracking devices with different levels of tracking capability. The minimum requirement is to be able to track two hands as two 3D moving points, just as in the case of Creative Sens3D. Therefore, alternative techniques are also provided for such cases. The proposed interaction interface description follows.

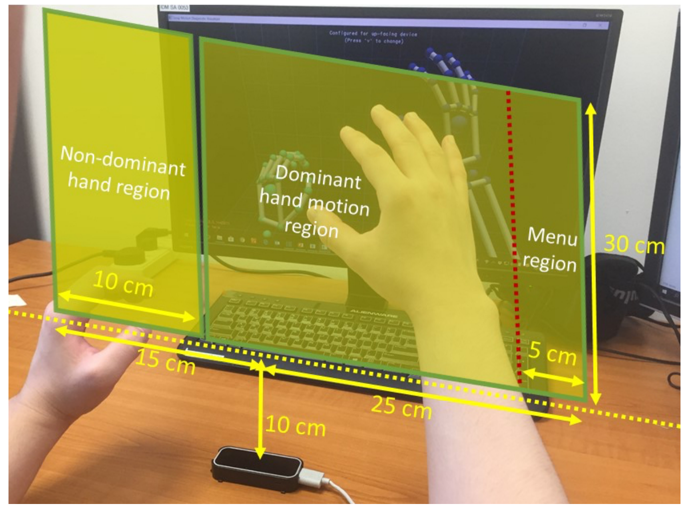

Figure 4: The interaction space of two hands.

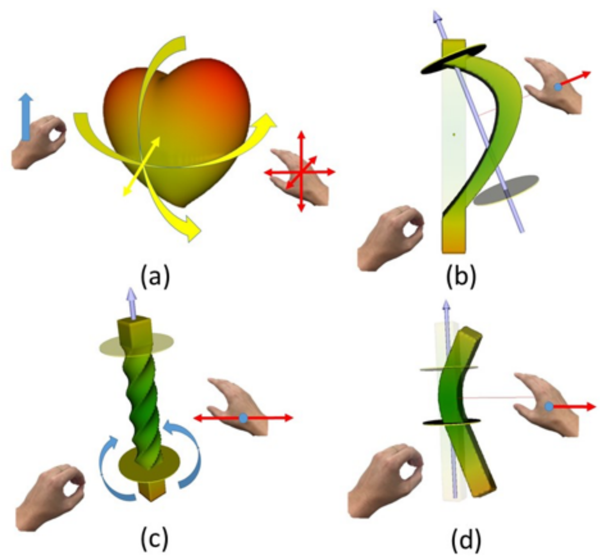

Figure 5: Interaction techniques for (a) Rotation (b) Tilting (c) Twisting, and (d) Bending.

Menu selection. The menu items are browsed by moving the dominant hand. The position of the dominant hand is displayed as a 3D cursor. When it moves in vicinity of the menu, 2D tracking is triggered so that the cursor is always placed over the menu items regardless of occasional moves of the hand in the third dimension. The precision of browsing is increasing when the grasping is being triggered by the non-dominant hand by closing or moving it down. This interaction method both minimizes the influence of hand tremor and prevents from accidental selection.

Object manipulation. In all object manipulations, the object has to be first selected/grabbed by the non-dominant hand. This is done depending on the device used, either by closing the hand or by moving it closer to the device. When being slowly released, the precision of the object placement is gradually increasing.

- *Translation.* The selected object is following the dominant hand motion while it is being grabbed by the non-dominant hand. Usually the palm position is tracked as input. However, it can also be switched to tracking index fingertip position when the user is pointing with the finger.

- *Rotation.* The first way of rotation directly maps the wrist rotation to the object rotation. Since the wrist rotation is limited and constrained, the second way of rotation can be used which is based on rotation of the visual aid – a bar connecting the centre of the object with the 3D cursor moved by the hand. Tracking of fingertips can also be used as input since the finger motion is more dexterous than the wrist motion. However, this way can only be used for axis-constrained rotation with some specific directions when the finger occlusions are minimized.

- *6-DOF placing.* The dominant hand controls translation and rotation (constrained by the wrist rotation) simultaneously with 6-DOF motion in the space.

Scene manipulation. This mode is triggered by the same grabbing/releasing gestures performed by the non-dominant hand when it is displaced to a higher position in its tracking area. The whole scene can be rotated by moving the dominant hand in the vertical plane, or dragged towards or away from the user by moving the hand away from and towards the screen, as shown in Figure 5(a).

Work-piece deformations.

- *Deformation Axis and Range Adjustment.* The deformation axis, as well as its range limits, can be moved, rotated or placed using the proposed object manipulation techniques as shown in Figure 5(b).

- *Twisting.* Twisting can be done by moving the dominant hand horizontally, as shown in Figure 5(c).

- *Tapering, bending and tilting.* After closing the non-dominant hand, a point initially placed on the deformation axis is translated with the motion of the dominant hand. For bending and tilting, the direction of deformation is defined by the direction of the bar, as shown in Figure 5(b) and (d).

5.4. Implementation

Aside from rigid manipulation and deformation of individual primitives, we also implemented the operation of combining them together. The tool object in the scene can be combined with the work-piece object using Boolean operations, which are accessible from the menu. The shapes are defined with FRep [44]. Specifically, a solid object is defined by inequality $f(x, y, z) \geq 0$. The points on, inside and outside the object result in 0, positive and negative values of the function, respectively. We used Virtual Reality Modeling Language (VRML) and JavaScript (VRML script) as our programming language, and Bitmanagement BS Contact as the viewer. This choice was motivated by the available function-based extension of VRML (FVRML/FX3D) [45] which permits geometric shapes and operations to be defined using analytic FRep functions straight in the VRML/X3D code together with the standard definitions of VRML and X3D. The hand tracking data was obtained from the Leap Motion SDK version Orion 3.2, and delivered to the platform using a plugin we developed and described in [46].

6. Evaluation of the results

Before undertaking the final user study we first tested the same cases as in our previous implementations, specifically, screwing a virtual nut onto a bolt as well as placing and deforming objects (See Figure 6). We observed that the tasks could be done more precisely and robustly than before since the occlusion problem was minimized with

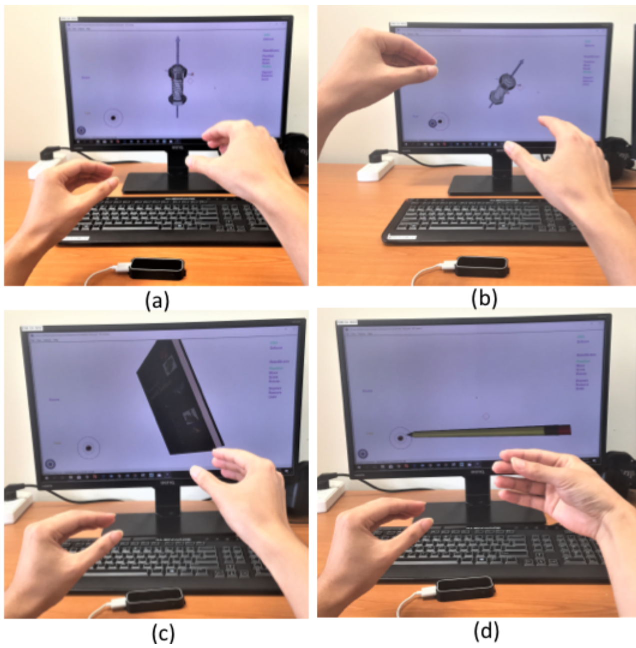

Figure 6: Implementations of the theremin-style interaction for natural interaction tasks studied in Section 4. (a) Placing the nut. (b) Rotating the bolt by rotating the scene. (c) and (d) are placing a pencil and a book.

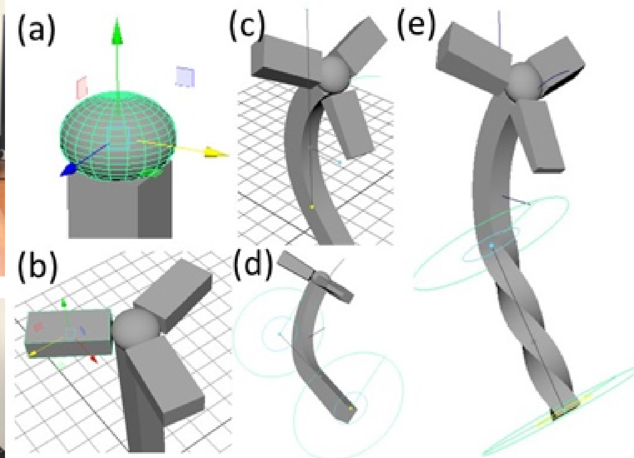

Figure 7: The shape modeling task contains five steps: (a) To place an ellipsoid on top of the block. (b) To assemble three petal blocks around the ellipsoid with 120° between any two neighbours. (c) To bend the upper middle part of the first block in a specific direction. (d) To align the deformation axis with the lower unbent part of the block and (e) to twist the unbent part of the block.

the new method. Having obtained these encouraging results, we then investigated whether the proposed mid-air interaction method could improve existing interaction of shape modeling with common 2D tracking devices.

The commercial modeling tool Autodesk Maya was used in this exercise. In Autodesk Maya, manipulation and deformation widgets were implemented for 3D tasks to be done with a mouse. As analysed in Section 2.2, the obstacle of using a mouse is that 3D operations have to be decomposed into a sequence of 2D operations. We were interested in whether such operations can be more efficiently done with the proposed interaction technique, while the intrinsic problems of hand motion and tracking could be minimized.

6.1. Task

We considered a common shape modeling case which required to /1/ Select items from a menu; /2/ Rotate the 3D scene with a work-piece object to any desired orientation; /3/ Select a tool-object and place it on, near or inside the work-piece object; /4/ Apply some CSG operations onto them (add and remove material); /5/ Perform deformations of the work-piece. The actual task (see Figure 7) contained the following five steps:

- In step (a), the ellipsoid had to be placed on the center of the top square face of the block while the bottom was touching the surface.
- In step (b), the three petal blocks had to be placed in the same horizontal plane with the ellipsoid while touching the surface. The blocks had to be individually manipulated so that the angles between each two neighbours were 120°.
- In step (c), bending had to be done at the upper half part of the first block. This required to 1) move up the bending widget, 2) adjust the bending range and drag out the bending bar to define the bending extent, and 3) to adjust the bending direction by rotation of the widget until the bar controlling bending penetrated through an edge of the block.
- In step (d), the twisting widget had to be aligned with the central axis of the lower unbent part of the body

block. This required to 1) align the widget with the edge of the lower unbent part of the block, and 2) to translate the widget to the center.

- In step (e), the bounding range had to be adjusted to exactly include the unbend part of the block before twisting the object.

The virtual model had to be viewed from different directions by rotating the scene in order to eliminate the misalignments. The size of the model displayed on the screen could also be adjusted by dragging the scene towards and away from the user. In Autodesk Maya, the scene could be manipulated by dragging the mouse while the "ALT" key is pressed. In the mid-air interaction, the virtual scene can be manipulated at any time when moving up the non-dominant hand, as illustrated in Section 5.3. In the mid-air interaction, grasping was performed by moving the fingers towards the thumb. The distance between the thumb tip and the index fingertip D was measured for activating events. The maximum and minimum thresholds D_{max} and D_{min} (which were illustrated in Section 5.2) could be adjusted for different hand sizes in the experiment, and they fell into the range between 3 cm and 7 cm. The maximum C/D ratio r_2 was set as 5 for translation and 2 for rotation in the experiment, while the minimum C/D ratio for both translation and rotation was set as $r_1 = 1$.

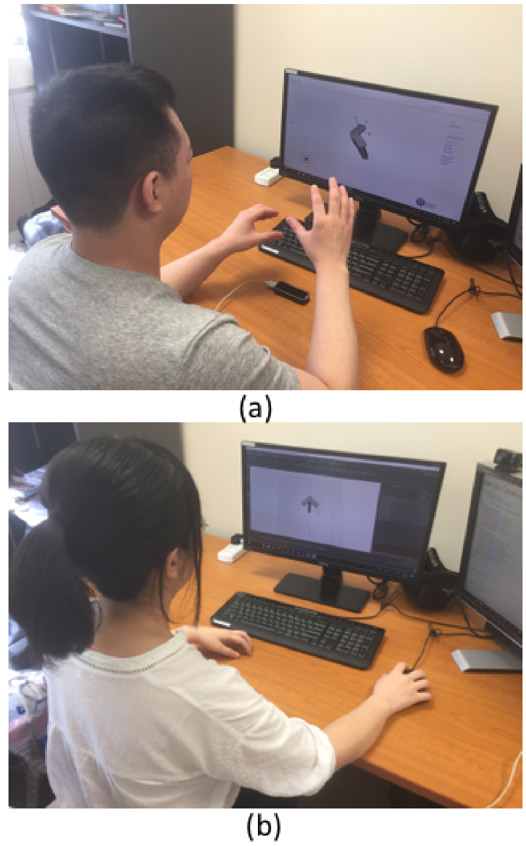

Figure 8: The setup for the user study for the (a) mid-air group and (b) Maya group.

6.2. Equipment

We used a desktop computer and a Leap Motion controller. The desktop computer was Alienware Auroro R5 with the Intel Core i7-6700 Processor (4-Cores, 8MB Cache, Turbo Boost 2.0, up to 4.0GHz) and 32GB (4X8GB) DDR4 2133MHz SDRAM Memory. The Graphic card was NVIDIA GeForce GTX 1080 with 8GB GDDR5X. The monitor was 24 inches with a resolution of 1920×1200. The Leap Motion controller was connected to the desktop though a USB port. It was placed on the table facing up in front of the screen.

6.3. Procedure

24 participants (18 male, 6 female) were invited. The average age was 25.9 ($sd = 3.7$). All of them were students from the university or colleagues in the research lab. None of them had an experience with Leap Motion controller or Autodesk Maya. They were evenly divided into two groups: to use Autodesk Maya and to use mid-air interaction. The user study setups for the two groups are shown in Figure 8. The maximum and minimum thresholds of grasping D_{max} and D_{min} were calibrated for the mid-air group. Participants were required to open and close their fingers several times until they could comfortably reach the three ranges of grasping separated by the two thresholds. The objective and requirements were illustrated to the participants in the beginning. The participants were allowed to practice the task and then to perform it. The experimenter was sitting next to the participants to take videos and to give instructions. Finally, the participants were asked to fill in a questionnaire based on

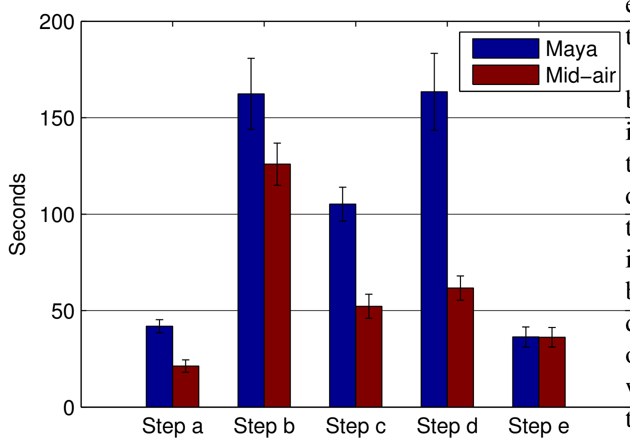

Figure 9: Average time spent on each step of the task.

a 5-point Likert scale, where 1 = strongly disagree and 5 = strong agree.

- Q1: It is easy to learn and acquire the interaction skills;
- Q2: It is easy to remember the interaction techniques;
- Q3: I feel comfortable during interaction;
- Q4: The system is easy to control;
- Q5: I can do the task as precisely as I want.

The user study took around 30 minutes.

6.4. Results

All the participants successfully completed the task, and 24 pieces of videos were collected and analysed. The results showed that the mid-air interaction (297.3 *sec*, *sd* = 66.7) was significantly faster than the interaction with Autodesk Maya (519 *sec*, *sd* = 149.8) for the modeling task ($p \approx 0$). By breaking down the time into individual steps, we found that for steps (a) - (d), mid-air interaction was significantly faster than that using Maya ($p \approx 0.05$ for (b) and $p \approx 0$ for the others), but there was no significant difference for step (e). The average time and standard error of each step when using mid-air interaction and Autodesk Maya can be seen in Figure 9.

Our initial proposition, explaining the time difference between the two kinds of interaction, was that mid-air interaction allowed for direct manipulations or deformations in 3D space instead of separating them into a sequence of 2D operations. In order to verify it, we counted the number of manipulations or deformation operations involved. We considered one operation as an action which begins with clicking the mouse button or closing the non-dominant hand, and ends with releasing the button or opening the non-dominant hand. The operations which we counted included manipulation of the scene and interacting with the scene objects. Menu selection was excluded from the counting, since they were inevitably involved but not relevant to the major interest of this user study. We found that the average number of operations when using mid-air interaction ($46, sd = 7.6$) was significantly less ($p \approx 0$) than that when using Autodesk Maya ($102.2, sd = 20.4$). The breakdown of the average number of operations in each step is shown in Figure 10. Furthermore, we found that there was a significant positive relationship between the interaction time and operations for both interactions using Autodesk Maya ($r = 0.69, F_{(1,N=10)} = 9.01, p < 0.05$) and mid-air interaction ($r = 0.60, F_{(1,N=10)} = 5.58, p < 0.05$). Finally, we divided the total time by the total number of operations. We then found that significantly lesser time was needed ($p < 0.05$) for performing one operation when using Autodesk Maya ($5.05, sd = 0.76$) than that using Mid-air interaction ($6.8, sd = 2.8$).

By observation of the user interaction, translation (step a), placing (step b), and alignment (step d) were indeed performed more efficiently with mid-air interaction, where decomposed sequence of 2D operations when using a mouse could be performed as integrated 3D translation or 6 DOF positioning. For the task of bending, the designed interaction technique allowed the bending position, extent and direction to be controlled simultaneously with 3 DOFs. In comparison, these operations had to be done with translation, rotation and bending widgets separately when using Autodesk Maya. It further complicated the task when the user had to switch between different widgets frequently in the process. Too many interactive arrows or lines were displayed, which confused the user and resulted in a higher rate of errors. However, the twist-

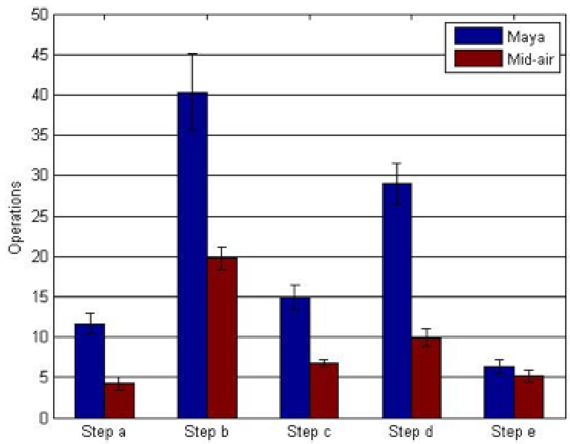

Figure 10: Average number of operations on each step.

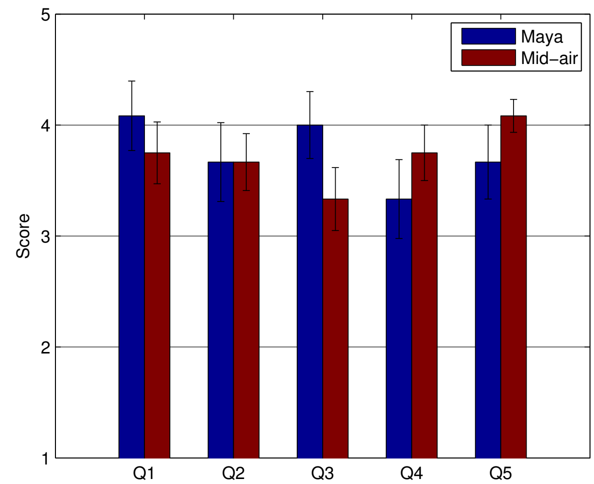

Figure 11: Results of the user questionnaire.

ing task only consisted of a few operations in both systems including adjusting the bounding range and rotating one end of the shape around its central axis. Therefore, similar number of operations and time were involved in both interaction paradigms, as it can be observed in Figure 10.

Next, we proceeded to participants' subjective evaluations of the interaction techniques. The results showed that there was no significant difference between mid-air interaction and using a mouse in aspects of learnability, mental load, comfort, controllability and precision, as shown in Figure 11. The proposed interaction technique was easy to learn (3.75, $sd = 0.97$) and remember (3.67, $sd = 0.89$). Most participants could acquire and remember the interaction technique easily, since it only involved natural ways of grasping and motion of hands. Compared to interaction using a mouse, mid-air interaction was still a new interaction way to the users which required some practice.

The results also showed that the proposed interaction technique was as controllable and precise as using a mouse (Q4 and Q5). By separating functions among two hands, the proposed interaction way effectively solved the problem of jump release and it also minimized occlusions. The ability of continuous changing precision of interaction allowed the participants to place objects precisely despite of hand tremor. By observation of the user interaction, their performance indeed improved significantly in comparison to our previous user studies presented in Section 4. The participants made fewer errors when they were preforming precision-controlled interaction in the mid-air. It was even more preferable than using a mouse for some 3D tasks. As reflected by some of the participants, they could not use a mouse precisely because it always failed to map the way how they wanted to interact in 3D mentally. In contrast, the mid-air interaction allowed them to work on the tasks in a more direct way, which was exactly how they planed mentally the interaction with their hands.

Finally, in aspects of comfort (Q3), a prolonged mid-air interaction caused arm fatigue, which made it less comfortable than using a mouse. However, some of the participants could also gradually learn to rest their arms on the desk, as it was actually anticipated, when they became familiar with the mid-air interaction.

Therefore, the obtained results proved our initial hypothesis that interaction in 3D modeling can be improved by using optical sensors. However, the naturalness had to be compromised to some extent to make sure hand motion can be reliably tracked so that the problems of hand tremor and jump release could be solved.

7. Discussion

It has to be admitted that separating natural grasping motion among two hands is a compromise due to the poor tracking quality of Leap Motion controller. However, it was reflected in the user subjective evaluation that the technique was not very difficult to learn since two hands were still performing natural motions. Functions were separated among two hands in accordance to the natural asymmetric way of using our hands in real life [47].

As for comparison with common interaction in shape modeling, the proposed interaction involved less number of operations compared to using the standard mouse-based interaction. However, on average it took a longer time to accomplish one operation than using a mouse, which made it not much advantageous when doing low DOF tasks. This was also confirmed by Burno et al. [48], who found that compared to mouse, gesture-based interaction was less efficient and with a higher error rate for 2D selection tasks. Therefore, according to these observations, mid-air interaction is more promising to be used for 3D modeling scenarios where arbitrary 3D interaction consists of the major part of interaction.

8. Conclusion

In this paper, we investigated whether 3D tasks could be performed as natural as in real life and as efficient as in common 2D-tracking based interaction with optical sensors. By comparison of the existing available optical sensors, Leap Motion controller was selected as the device for testing. In the experiments, we found that common ways of both collision- and gesture-based interaction could not allow us to achieve our goal, while gesture-based interaction was still more promising.

Inspired by the way of playing the theremin, we proposed a new gesture-based interaction technique which separated functions among two hands: the dominant hand was used for tracking position and orientation while the non-dominant hand was used to trigger events and to continuously control precision. Although from the first glance the interaction way is not as intuitive as common interaction, it can be better adapted to the tracking capability of the optical tracking device so that higher efficiency and precision can be achieved.

A user study was conducted to compare the proposed interaction technique with the common 2D-tracking based interaction. According to the subjective evaluation, no significant differences were observed between mid-air interaction and using a standard mouse in aspects of learnability, mental load, comfort, controllability and precision. The specific modeling task could even be more quickly accomplished with the propose interaction technique. The obtained results proved usefulness of existing optical sensors in precise shape modeling and virtual assembling tasks.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

References

[1] Cui, J, Kuijper, A, Sourin, A. Exploration of natural free-hand interaction for shape modeling using leap motion controller. In: 2016 International Conference on Cyberworlds (CW). 2016, p. 41–48. doi:10.1109/CW.2016.14.

[2] Bullock, IM, Feix, T, Dollar, AM. Human precision manipulation workspace: Effects of object size and number of fingers used. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2015, p. 5768–5772. doi:10.1109/EMBC.2015.7319703.

[3] Gloumakov, Y, Feix, T, Bullock, IM, Dollar, AM. Object stability during human precision fingertip manipulation. In: 2016 IEEE Haptics Symposium (HAPTICS). 2016, p. 84–91. doi:10.1109/HAPTICS.2016.7463160.

[4] Feix, T, Bullock, IM, Gloumakov, Y, Dollar, AM. Rotational ranges of human precision manipulation when grasping objects with two to five digits. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2015, p. 5785–5790. doi:10.1109/EMBC.2015.7319707.

[5] Sturman, MM, Vaillancourt, DE, Corcos, DM. Effects of aging on the regularity of physiological tremor. Journal of Neurophysiology 2005;93(6):3064–3074. URL: https://doi.org/10.1152/jn.01218.2004. doi:10.1152/jn.01218.2004; pMID: 15716367.

[6] Burkhard, PR, Langston, J, Tetrud, JW. Voluntarily simulated tremor in normal subjects. Neurophysiologie Clinique/Clinical Neurophysiology 2002;32(2):119 – 126. URL: http://www.sciencedirect.com/science/article/pii/S0987705302002964. doi:https://doi.org/10.1016/S0987-7053(02)00296-4.

[7] Genuine alienware kkmh5, modmuo usb wired scroll wheel laser black glossy gaming 3-button 1200 dpi mouse part numbers: Kkmh5, modmuo. https://www.amazon.com/Genuine-Alienware-KKMH5-MODMUO-3-Button/dp/B00C6DZH9S; 2018. [Online].

[8] Wireless mouse m235. https://www.logitech.com/en-sg/product/wireless-mouse-m235-2nd-gen#specification-tabular; 2018. [Online].

[9] Bachrat, M, Åalman, M. 2d position measurements with optical laser mouse sensor. https://www.scribd.com/document/184855267/2D-Position-Measurement-With-Optical-Laser-Mouse-Sensor; 2010. [Online].

[10] Conner, BD, Snibbe, SS, Herndon, KP, Robbins, DC, Zeleznik, RC, van Dam, A. Three-dimensional widgets. In: Proceedings of the 1992 Symposium on Interactive 3D Graphics. I3D '92; New York, NY, USA: ACM. ISBN 0-89791-467-8; 1992, p. 183–188. URL: http://doi.acm.org/10.1145/147156.147199. doi:10.1145/147156.147199.

[11] Zhao, YJ, Shuralyov, D, Stuerzlinger, W. Comparison of multiple 3d rotation methods. In: 2011 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems Proceedings. 2011, p. 1–5. doi:10.1109/VECIMS.2011.6053855.

[12] Oh, JY, Stuerzlinger, W. Moving objects with 2d input devices in cad systems and desktop virtual environments. In: Proceedings of Graphics Interface 2005. GI '05; School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada: Canadian Human-Computer Communications Society. ISBN 1-56881-265-5; 2005, p. 195–202. URL: http://dl.acm.org/citation.cfm?id=1089508.1089541.

[13] Teather, RJ, Stuerzlinger, W. Guidelines for 3d positioning techniques. In: Proceedings of the 2007 Conference on Future Play. Future Play '07; New York, NY, USA: ACM. ISBN 978-1-59593-943-2; 2007, p. 61–68. URL: http://doi.acm.org/10.1145/1328202.1328214. doi:10.1145/1328202.1328214.

[14] Lubos, P, Bruder, G, Ariza, O, Steinicke, F. Touching the sphere: Leveraging joint-centered kinespheres for spatial user interaction. In: Proceedings of the 2016 Symposium on Spatial User Interaction. SUI '16; New York, NY, USA: ACM. ISBN 978-1-4503-4068-7; 2016, p. 13–22. URL: http://doi.acm.org/10.1145/2983310.2985753. doi:10.1145/2983310.2985753.

[15] The myo armband. https://www.thalmic.com/; 2018. [Online].

[16] Aurand, AM, Dufour, JS, Marras, WS. Accuracy map of an optical motion capture system with 42 or 21 cameras in a large measurement volume. Journal of Biomechanics 2017;58(Supplement C):237 – 240. URL: http://www.sciencedirect.com/science/article/pii/S0021929017302580. doi:https://doi.org/10.1016/j.jbiomech.2017.05.006.

[17] Wang, Y, Neff, M. Data-driven glove calibration for hand motion capture. In: Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. SCA '13; New York, NY, USA: ACM. ISBN 978-1-4503-2132-7; 2013, p. 15–24. URL: http://doi.acm.org/10.1145/2485895.2485901. doi:10.1145/2485895.2485901.

[18] Weichert, F, Bachmann, D, Rudak, B, Fisseler, D. Analysis of the accuracy and robustness of the leap motion controller. Sensors 2013;13(5):6380–6393. URL: http://www.mdpi.com/1424-8220/13/5/6380. doi:10.3390/s130506380.

[19] Valentini, PP, Pezzuti, E. Accuracy in fingertip tracking using leap motion controller for interactive virtual applications. International Journal on Interactive Design and Manufacturing (IJIDeM) 2017;11(3):641–650. URL: https://doi.org/10.1007/s12008-016-0339-y. doi:10.1007/s12008-016-0339-y.

[20] Hornsey, RL, Hibbard, PB. Evaluation of the accuracy of the leap motion controller for measurements of grip aperture. In: Proceedings of the 12th European Conference on Visual Media Production. CVMP '15; New York, NY, USA: ACM. ISBN 978-1-4503-3560-7; 2015, p. 13:1–13:1. URL: http://doi.acm.org/10.1145/2824840.2824855. doi:10.1145/2824840.2824855.

[21] Tkach, A, Pauly, M, Tagliasacchi, A. Sphere-meshes for real-time hand modeling and tracking. ACM Trans Graph 2016;35(6):222:1–222:11. URL: http://doi.acm.org/10.1145/2980179.2980226. doi:10.1145/2980179.2980226.

[22] Taylor, J, Bordeaux, L, Cashman, T, Corish, B, Keskin, C, Sharp, T, et al. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. ACM Trans Graph 2016;35(4):143:1–143:12. URL: http://doi.acm.org/10.1145/2897824.2925965. doi:10.1145/2897824.2925965.

[23] Boulic, R, Rezzonico, S, Thalmann, D. Multi-finger manipulation of virtual objects. In: In Proc. of the ACM Symposium on Virtual Reality Software and Technology (VRST '96. 1996, p. 67–74.

[24] Borst, CW, Indugula, AP. A spring model for whole-hand virtual grasping. Presence: Teleoper Virtual Environ 2006;15(1):47–61. URL: http://dx.doi.org/10.1162/pres.2006.15.1.47. doi:10.1162/pres.2006.15.1.47.

[25] Kumar, V, Todorov, E. Mujoco haptix: A virtual reality system for hand manipulation. In: 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids). 2015, p. 657–663. doi:10.1109/HUMANOIDS.2015.7363441.

[26] Talvas, A, Marchal, M, Duriez, C, Otaduy, MA. Aggregate constraints for virtual manipulation with soft fingers. IEEE Transactions on Visualization and Computer Graphics 2015;21(4):452–461. doi:10.1109/TVCG.2015.2391863.

[27] Jacobs, J, Froehlich, B. A soft hand model for physically-based manipulation of virtual objects. In: 2011 IEEE Virtual Reality Conference. 2011, p. 11–18. doi:10.1109/VR.2011.5759430.

[28] Sato, M, Savchenko, V, Ohbuchi, R. 3d freeform design: interactive shape deformations by the use of cyberglove. In: 2004 International Conference on Cyberworlds. 2004, p. 147–154. doi:10.1109/CW.2004.2.

[29] Fuge, M, Yumer, ME, Orbay, G, Kara, LB. Conceptual design and modification of freeform surfaces using dual shape representations in augmented reality environments. Comput Aided Des 2012;44(10):1020–1032. URL: http://dx.doi.org/10.1016/j.cad.2011.05.009. doi:10.1016/j.cad.2011.05.009.

[30] Shieh, MD, Yang, CC. esigning product forms using a virtual hand and deformable models. In: ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; vol. 3. 2006, p. 823–830. doi:10.1115/DETC2006-99171.

[31] Nickel, K, Stiefelhagen, R. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In: Proceedings of the 5th International Conference on Multimodal Interfaces. ICMI '03; New York, NY, USA: ACM. ISBN 1-58113-621-8; 2003, p. 140–146. URL: http://doi.acm.org/10.1145/958432.958460. doi:10.1145/958432.958460.

[32] Segen, J, Kumar, S. Gesture vr: Vision-based 3d hand interace for spatial interaction. In: Proceedings of the Sixth ACM International Conference on Multimedia. MULTIMEDIA '98; New York, NY, USA: ACM. ISBN 0-201-30990-4; 1998, p. 455–464. URL: http://doi.acm.org/10.1145/290747.290822. doi:10.1145/290747.290822.

[33] O'Hagan, R, Zelinsky, A. Visual gesture interfaces for virtual environments. In: Proceedings First Australasian User Interface Conference. AUIC 2000 (Cat. No.PR00515). 2000, p. 73–80. doi:10.1109/AUIC.2000.822069.

[34] Prachyabrued, M, Borst, CW. Design and evaluation of visual interpenetration cues in virtual grasping. IEEE Transactions on Visualization and Computer Graphics 2016;22(6):1718–1731. doi:10.1109/TVCG.2015.2456917.

[35] Prachyabrued, M, Borst, CW. Virtual grasp release method and evaluation. Int J Hum-Comput Stud 2012;70(11):828–848. URL: http://dx.doi.org/10.1016/j.ijhcs.2012.06.002. doi:10.1016/j.ijhcs.2012.06.002.

[36] Frees, S, Kessler, GD, Kay, E. Prism interaction for enhancing control in immersive virtual environments. ACM Trans Comput-Hum Interact 2007;14(1). URL: http://doi.acm.org/10.1145/1229855.1229857. doi:10.1145/1229855.1229857.

[37] Oikonomidis, I, Kyriazis, N, Argyros, AA. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In: 2011 International Conference on Computer Vision. 2011, p. 2088–2095. doi:10.1109/ICCV.2011.6126483.

[38] Sharp, T, Keskin, C, Robertson, D, Taylor, J, Shotton, J, Kim, D, et al. Accurate, robust, and flexible real-time hand tracking. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. CHI '15; New York, NY, USA: ACM. ISBN 978-1-4503-3145-6; 2015, p. 3633–3642. URL: http://doi.acm.org/10.1145/2702123.2702179. doi:10.1145/2702123.2702179.

[39] Frees, S, Kessler, GD, Kay, E. Prism interaction for enhancing control in immersive virtual environments. ACM Trans Comput-Hum Interact 2007;14(1). URL: http://doi.acm.org/10.1145/1229855.1229857. doi:10.1145/1229855.1229857.

[40] Garbaya, S, Zaldivar-Colado, U. The affect of contact force sensations on user performance in virtual assembly tasks. Virtual Reality 2007;11(4):287–299. URL: https://doi.org/10.1007/s10055-007-0075-5. doi:10.1007/s10055-007-0075-5.

[41] Cui, J, Kuijper, A, Fellner, DW, Sourin, A. Understanding people's mental models of mid-air interaction for virtual assembly and shape modeling. In: Proceedings of the 29th International Conference on Computer Animation and Social Agents. CASA '16; New York, NY, USA: ACM. ISBN 978-1-4503-4745-7; 2016, p. 139–146. URL: http://doi.acm.org/10.1145/2915926.2919330. doi:10.1145/2915926.2919330.

[42] Cui, J, Fellner, DW, Kuijper, A, Sourin, A. Mid-air gestures for virtual modeling with leap motion. In: Streitz, N, Markopoulos, P, editors. Distributed, Ambient and Pervasive Interactions. Cham: Springer International Publishing. ISBN 978-3-319-39862-4; 2016, p. 221–230.

[43] Cui, J, Sourin, A. Interactive shape modeling using leap motion controller. In: SIGGRAPH Asia 2017 Technical Briefs. SA '17; New York, NY, USA: ACM. ISBN 978-1-4503-5406-6; 2017, p. 3:1–3:4. URL: http://doi.acm.org/10.1145/3145749.3149437. doi:10.1145/3145749.3149437.

[44] Pasko, A, Adzhiev, V, Sourin, A, Savchenko, V. Function representation in geometric modeling: concepts, implementation and applications. The Visual Computer 1995;11(8):429–446.

URL: https://doi.org/10.1007/BF02464333. doi:10.1007/BF02464333.

[45] Lai, D, Sourin, A. Interactive visualization of mathematics in 3d web. In: 2012 International Conference on Cyberworlds. 2012, p. 122–129. doi:10.1109/CW.2012.24.

[46] Cui, J, Sourin, A. Feasibility study on free hand geometric modelling using leap motion in vrml/x3d. In: 2014 International Conference on Cyberworlds. 2014, p. 389–392. doi:10.1109/CW.2014.60.

[47] Guiard, Y. Asymmetric division of labor in human skilled bimanual action. Journal of Motor Behavior 1987;19(4):486–517. URL: https://doi.org/10.1080/00222895.1987.10735426. doi:10.1080/00222895.1987.10735426. arXiv:https://doi.org/10.1080/00222895.1987.10735426; pMID: 15136274.

[48] Burno, RA, Wu, B, Doherty, R, Colett, H, Elnaggar, R. Applying fitts law to gesture based computer interactions. Procedia Manufacturing 2015;3(Supplement C):4342 – 4349. URL: http://www.sciencedirect.com/science/article/pii/S2351978915004308. doi:https://doi.org/10.1016/j.promfg.2015.07.429; 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.