



**HAL**  
open science

# Combining HoloLens and Leap-Motion for Free Hand-Based 3D Interaction in MR Environments

Fakhreddine Ababsa, Junhui He, Jean-Rémy Chardonnet

► **To cite this version:**

Fakhreddine Ababsa, Junhui He, Jean-Rémy Chardonnet. Combining HoloLens and Leap-Motion for Free Hand-Based 3D Interaction in MR Environments. 7th International Conference on Augmented Reality, Virtual Reality, and Computer Graphics, Sep 2020, Lecce, Italy. pp.315-327, 10.1007/978-3-030-58465-8\_24 . hal-03001870

**HAL Id: hal-03001870**

**<https://hal.science/hal-03001870>**

Submitted on 12 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Combining HoloLens and Leap-Motion for Free Hand-Based 3D Interaction in MR Environments

Fakhreddine Ababsa<sup>1</sup>(✉), Junhui He<sup>2</sup>(✉), and Jean-Remy Chardonnet<sup>1</sup>(✉)

<sup>1</sup> Arts et Metiers Institute of Technology, LISPEN, HESAM University,  
Chalon-sur-Saône, France

{Fakhreddine.Ababsa, Jean-Remy.Chardonnet}@ensam.eu

<sup>2</sup> ENSEA, Cergy, France

junhui.he@ensea.fr

**Abstract.** The ability to interact with virtual objects using gestures would allow users to improve their experience in Mixed Reality (MR) environments, especially when they use AR headsets. Today, MR head-mounted displays like the HoloLens integrate hand gesture based interaction allowing users to take actions in MR environments. However, the proposed interactions remain limited. In this paper, we propose to combine a Leap Motion Controller (LMC) with a HoloLens in order to improve gesture interaction with virtual objects. Two main issues are presented: an interactive calibration procedure for the coupled HoloLens-LMC device and an intuitive hand-based interaction approach using LMC data in the HoloLens environment. A set of first experiments was carried out to evaluate the accuracy and the usability of the proposed approach.

**Keywords:** Mixed reality · Calibration · Natural interaction

## 1 Introduction

Since a couple of years, both augmented reality (AR) hardware and applications became convincing and widespread in many fields, for example in medicine [1], education, and industry. Microsoft, Facebook and Apple have shown their interest in AR applications, believing in the viability of this technology. Following this trend, the HoloLens head-mounted device, which was released by Microsoft in 2016, is one of the leading MR devices. Features include drawing holograms in the user's field of view and enabling interaction with real-world environments. One of its worthwhile features is its own custom holographic processing unit (HPU), which allows complex calculations to be embedded. However, interaction techniques are limited. It proposes two hand gestures only: air tap and bloom. It cannot track precise information about the position of the hands or identify other hand gestures. On the other hand, the Leap Motion Controller (LMC) is a peripheral device dedicated to high accuracy hand tracking. Originally released in July 2013, its goal was to provide an alternative to the traditional mouse and keyboard by proposing free-hand interaction. Its size is quite small, making it possible to be used in

combination for example with head-mounted displays in virtual reality. To fill the shortcomings of the HoloLens in terms of interaction, we coupled the LMC to the HoloLens to provide hand tracking data to the HoloLens and create enriched gesture-based interaction in mixed reality. Note that other headsets used for Virtual reality offer already an integration with LMC, such as Oculus. In our case, we developed a communication tool between the two devices to enable data transfer, then we focused on the hand tracking part. As the HoloLens and the LMC have different coordinate systems, it is important to calibrate the coupled device in order to project the 3D points collected by the LMC on the coordinate system of the HoloLens. As long as the spatial configuration of the two devices does not change, the calibration process does not need to be repeated. In this paper, we propose a 3D point based calibration approach in order to achieve more complex and natural 3D interaction. In addition, to collect data, we propose a semi-automatic procedure involving the user without constraining his/her movements. Only the LMC and the HoloLens are used to detect the spatial position of the hand. The entire calibration process uses the user's fingers as a reference and does not require any other instruments. Therefore, the main contribution of this paper is to provide an interactive calibration procedure for the coupled HoloLens-LMC device. Our approach uses the virtual object rendering done by the HoloLens to collect 3D point coordinates from both the HoloLens and the LMC. Based on the calibration results, several demos using free hands to interact with virtual objects are presented. We carried out a set of first experiments to evaluate the accuracy and the usability of the proposed approach.

## 2 Related Works

Free hand-based 3D interaction is a topic well explored for many years. Lyons [9] proposed camera-based gesture inputs for three dimensional navigation through a virtual reality environment. In [2] the authors showed that the physical interaction between virtual and real objects improves user experience and thus enhances the feeling of the presence of virtual contents in the real world. Ens et al. [5] proposed a mixed-scale gesture interaction technique, which interleaves micro-gestures with larger gestures for computer interaction. Their idea is to create a design space for applying micro-gestures within a greater gestural lexicon in order to adapt the interaction to the task performed by the user. Several input modes are often combined in order to overcome the problems related to the variability of gesture interaction. For instance, using a Leap Motion controller (LMC) allows accurate recognition of natural gestures. Thus, Khademi et al. [7] suggested free-hand interaction to rehabilitate patients with stroke; they modified the Fruit Ninja game to use the LMC hand tracking data, enabling patients with arm and hand weakness to practice rehabilitation. Blaha and Gupta built a virtual reality game displayed on an Oculus Rift head-mounted display and coupled with an LMC to help people with amblyopia restore vision in their amblyopic eye [3].

Most of the developed applications using an LMC are proposed in virtual reality, while MR environments suffer from a lack of applications using an LMC, because of equipment limitations. Furthermore, gesture recognition is almost never used in applications involving the HoloLens. Garon et al. [6] identified the lack of high quality depth data as greatly restricting the HoloLens's potential as a research tool. They mounted a

separate depth camera on top of the HoloLens, connected it to a stick computer, which was then used to stream the depth data wirelessly to the HoloLens. Several frameworks propose combining a HoloLens with a Kinect to enable multi-person collaboration on 3D objects in an MR environment. The HoloLens 1 are used so that all participants can view the same 3D objects in real time, while the Kinects are used to expand the available gesture interactions with custom gestures. But still, current research on the HoloLens usually adds a depth sensor on it to get more information. In fact, the LMC can also get data about depth, but it can provide more personalized gestures in mixed reality. In this frame, Köhler proposed a combination of the HoloLens and the LMC [8] which is very close to what we present here. However, the way he manages the data is different from our approach. He used the data collected by the LMC and analyzed the photos taken by the HoloLens to transform the 3D coordinates of the hand into 2D coordinates in the HoloLens screen using Perspective-n-Point, a well-known problem to estimate the distance between a set of  $n$  3D points and their projection on a 2D plane [4]. His work provides the possibility to create new gestures. However, he only gets 2D coordinates of the hand and loses 3D information, therefore 3D interaction cannot be achieved.

### 3 Communication Between the HoloLens and the LMC

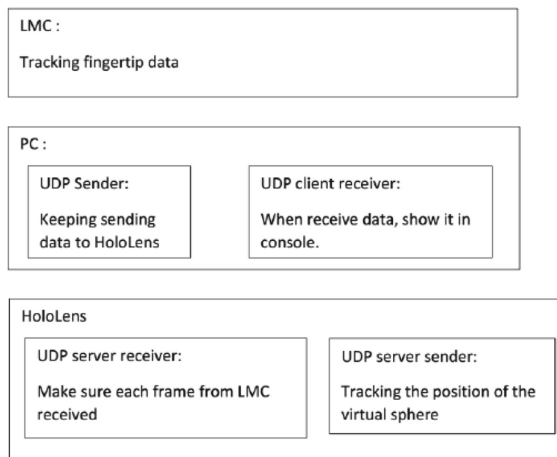
The current version of the LMC needs to be physically connected to a computer using a USB cable to provide enough power to infrared LEDs and to enable fast enough data transfer. For the HoloLens, the only physical ports are a 3.5 mm audio jack and a Micro-USB 2.0 port. Unfortunately, only use the Micro-USB port can currently to install and debug applications, charge, and access the Device Portal. Therefore, the built-in processing unit of the HoloLens cannot be used directly with the LMC and a separate computer is needed to allow the LMC to stream data wirelessly to the HoloLens. As shown in Fig. 1, the LMC was attached on top of the HoloLens using some sticky paste. The LMC was not fixed vertically but with an angle of  $45^\circ$  to enable gesture tracking within the field of view of the HoloLens and to track the hands at the chin or chest level. The inclination of the LMC allows thus better covering this area.



**Fig. 1.** The LMC is attached on the top of the HoloLens and connected to a computer

To exchange data between the LMC and the HoloLens, the UDP protocol was chosen due to its lightness and high speed. Figure 2 shows the architecture of the developed communication tool, with an external computer as a medium to transfer and process data

between the LMC and the HoloLens. In practice, HoloLens works as a server and the PC as a client. Initially, the HoloLens IP address is determined and then used by the client to send its first message to the server. The server in turn retrieves the client's IP address, allowing client-server communication. In our implementation, the LMC SDK is used to capture the fingertips' position at each frame. As long as a new frame is generated, it will be sent to the server (HoloLens). On the client side, the messages are received/sent in different threads. This allows the client to listen to the channel while sending a message. Similarly, on the server side a unity event is invoked to reply to the message sent by the client. This architecture allows synchronized and real-time communication between the LMC and the HoloLens.



**Fig. 2.** Communication architecture between the HoloLens and the LMC

## 4 Calibration Procedure

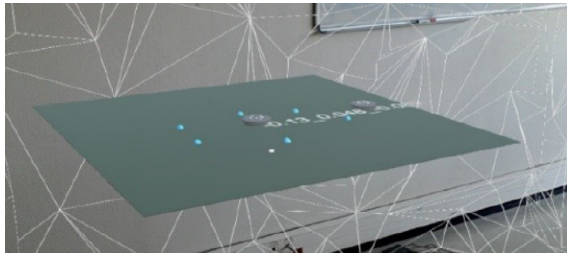
The main idea of the calibration procedure is to get the coordinates of the same points in different coordinate systems. After careful consideration, we choose fingertips of one hand as the detecting points because fingertips are easy to access from the LMC. The second hand is used to click on the hololens controller once the alignment between the finger and the marker point is completed. This triggers the recording of data. Indeed, fingers can move flexibly and be detected easily by the LMC. The entire calibration process is summarized as follows and will be detailed in the next subsections:

1. Detect the position of a fingertip and get the coordinates from the LMC.
2. Attach a virtual plane to a real plane (either a desk or a wall).
3. Coincide the fingertip with marked points on the plane.
4. The HoloLens records the coordinates of the marked points and the LMC records the fingertip simultaneously. Send the two coordinates to the computer.

5. After collecting a set of points, analyze the data and compute the rigid transformation.
6. Use the computed transformation to change the coordinates of the LMC points in the HoloLens coordinate frame and display in the HoloLens. Compare the error between the virtual and the real points.

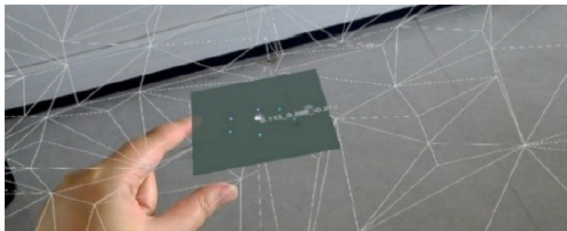
#### 4.1 3D Points Collection

As shown in Fig. 3, to collect 3D points, we developed a simple application on Unity3D where we display a calibration plane with six small balls embedded (what we called above the marked points). The data collected from the LMC is displayed in the middle of the plane.



**Fig. 3.** Calibration plane

The whole plane is attached with a “Taptoplace” function. The HoloLens provides spatial mapping allowing to understand the environment. Users can use the cursor to tap on the plane to move it according to the surroundings (see Fig. 4). The virtual plane must be placed according to the real world, so that it can help users accurately determining the position of their fingertip.



**Fig. 4.** “Taptoplace” according to the spatial mapping

According to the principle of camera projection, the HoloLens will project three-dimensional objects on a two-dimensional screen in front of the user’s eyes. Hence, some points with different 3D coordinates will have the same 2D coordinates on the screen.

We need to use a real object because in this way we can compute the exact spatial coordinate by querying the depth information given by the coordinate in the Z-axis

direction. Two buttons are displayed on the plane. The button on the right is a “change button”: clicking on it will change the rotation of the calibration plane from horizontal to vertical and vice-versa.

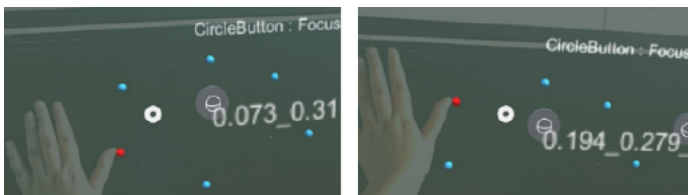
The aim of this button is to make the plane compatible with both vertical and horizontal surfaces in the real world.

Once the user fixes the virtual plane according to the real world as shown in Fig. 5 (in this case we attached it to a computer screen), the HoloLens creates a spatial anchor in the world coordinate frame. Therefore, the user can get the relative position between the HoloLens camera and the virtual plane in real time. The button on the middle is a “send button”. Clicking on it will allow the HoloLens to send data for calibration. After the first tap, one marked point (blue ball) will turn red.



**Fig. 5.** Changing the rotation of the plane and attaching it to a physical surface (Color figure online)

The user positions one fingertip, for example the thumb, on this point (see Fig. 6). If he/she clicks on the send button, both the coordinates of the ball from the HoloLens and the coordinates of the fingertip from the LMC will be sent to the computer. Another ball on the plane will then become red. This operation is done for each point. To help clicking on the buttons, a cursor always appears in the middle of the field of view and can be moved by moving the head. Clicks are performed using the HoloLens built-in clicker, not to affect point collection by the LMC.



**Fig. 6.** 3D points acquisition from HoloLens and LMC (Color figure online)

## 4.2 Calibration Approach

Once data collection is complete, we get two points clouds in two different frames (see Fig. 7). The first one is collected by the HoloLens, we name it the “reference point cloud”, while the second one is collected by the LMC, we name it the “source point cloud”. The relationship between these two point clouds is given by:

$$\begin{pmatrix} H_x \\ H_y \\ H_z \end{pmatrix} = R \cdot \begin{pmatrix} L_x \\ L_y \\ L_z \end{pmatrix} + t \quad (1)$$

The transformation matrix is composed of two parts: a Translation vector ( $t$ ) and a Rotation matrix ( $R$ ). The problem is to find the transformation matrix between both point clouds, knowing that the point clouds should be aligned. Several algorithms have been proposed in past work. In our case, we compared two algorithms. We will not focus on specific mathematical calculation, we will only briefly introduce their principle here. Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either. Subsequent paragraphs, however, are indented.

The first method is the Umeyama algorithm [10]. We denote the reference points as  $\{H_i\}$  and the source points as  $\{L_i\}$ ,  $i = 1, 2, \dots, n$ . The problem is to find the similarity transformation parameters ( $R$  and  $t$ ) giving the minimum value of the mean squared error  $\varepsilon^2(R, t)$  of these two point clouds:

$$\varepsilon^2(R, t) = \frac{1}{n} \cdot \sum_{i=1}^n \|H_i - (R \cdot L_i + t)\|^2 \quad (2)$$

However, we soon discovered that this algorithm causes large errors. In fact, with this method, the result is obtained through many mathematical transformations due to the coordinate matrices  $\{H_i\}$  and  $\{L_i\}$ . This method requires that the point’s coordinates be very accurate. It only works when the two coordinate groups can be perfectly matched. Since the coordinates are collected manually, errors obviously occur that must be taken into account in the transformation calculation. Another choice to align two point clouds

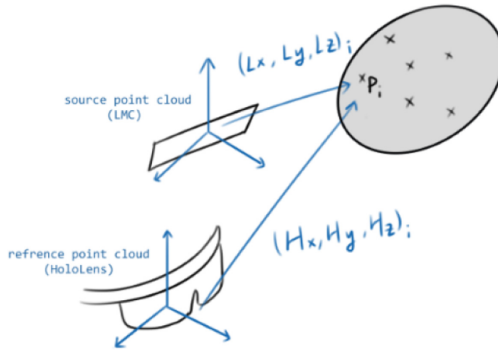


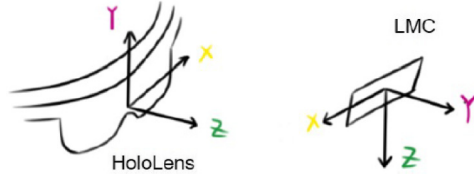
Fig. 7. The points cloud defined in the HoloLens end LMC reference frames



is the Iterative Closest Point (ICP) algorithm [12]. The principle of this algorithm is to find the closest points of the source point cloud in the reference point cloud. The whole algorithm is performed by multiple iterations, in which the points with larger errors are excluded. The algorithm steps are:

1. Find the closest points of the source point cloud (collected by the LMC), match them in the reference cloud (collected by the HoloLens).
2. Use a root mean square metric minimization technique to estimate a transform matrix, which will best align each source point to its match, found in the previous step. This step may also involve rejecting outliers prior to alignment.
3. Transform the source points using the obtained transformation matrix and repeat the steps until a defined threshold is reached.

This algorithm has some limitations. First, in order to match the source cloud to the reference cloud, we should know approximately the position of the reference cloud. Otherwise, finding the nearest point will be impacted. It means that before applying the algorithm, the source cloud should be transformed as close as possible to the reference cloud. Because of the arrangement of the two reference frames as shown in Fig. 8, we have to preprocess the source point coordinates ( $L_i$ ) so that they are closer to the reference coordinates ( $H_i$ ). The solution is to apply a rotation matrix:



**Fig. 8.** HoloLens and LMC coordinate reference systems

$$r = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (3)$$

After preprocessing, we get  $L'_i$ :

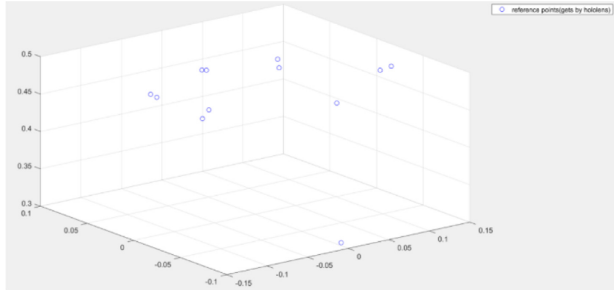
$$L'_i = r \cdot L_i \quad (4)$$

$L'_i$  can be considered then as the new source point cloud, it only needs to be rotated with a relatively small angle to be aligned with the reference point cloud. The new transformation matrix ( $R', t$ ) can be found using the ICP algorithm, where:

$$H_i = R' \cdot L'_i + t = R' \cdot r \cdot L_i \quad (5)$$

The second limitation of ICP is a practical one. Indeed, when the number of 3D points increases, the probability of errors becomes greater. The best solution is then to limit

the number of points and ensure that most of them are on different planes with different depths according to the Z axis. Figure 9 gives an example of twelve reference points, which are located approximately in the same plane.



**Fig. 9.** Example of collected points

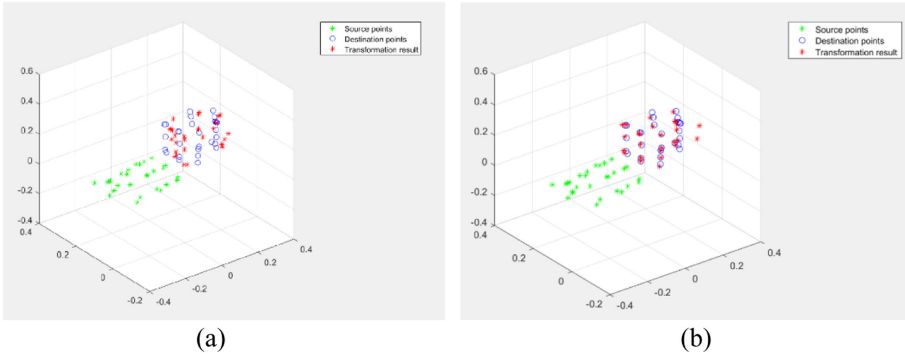
One point that is a little bit far from the others is displayed and is considered as wrong. In the current collection process, we can collect multiple points at a fixed distance, then change the distance between the HoloLens and the virtual object (just move the head closer to the virtual plane) to continue collecting points.

### 4.3 Calibration Results

First, we compared the results of our method with the naïve approach that uses raw LMC data without taking calibration into account. The finger positions provided by the LMC and projected into the HoloLens reference frame are far away and do not match the real hand. This baseline error measurement clearly shows the need of a space coordinates transformation and quantify on the initial mismatch magnitude. Figures (10-a) and (10-b) show the results obtained by the Umeyama and the ICP algorithms when processing the same data set. With Umeyama, we get a mean error of 0.0869 m while with ICP we get a mean error of 0.0189 m. It can be seen that because of some errors while collecting the data, there are two points that have a large deviation. ICP can limit their impact, while Umeyama is easily affected with a larger estimation error.

### 4.4 Error Analysis

After calibration, we used the obtained transformation matrix to mark the hand position in the HoloLens screen. We represent the fingertips of the index finger and the thumb by two red balls. As shown in Fig. 11, there are misalignment errors, especially when the hand is not parallel to the vertical plane. However, we have improved previous existing works like [8] where the translation error reported by the author is roughly a 4.5 cm along the z-axis; while our approach gives an error of about 1.8 cm. In our case the main factor affecting the registration accuracy is human behavior. All the points are collected manually and using the user's fingertips as reference. During this process, deviations are unavoidable. The spatial mapping provided by the HoloLens also presents deviation



**Fig. 10.** Calibration results obtained with (a) Umeyama and (b) ICP algorithms

compared to the real environment, especially when the interaction distance is close. This error cannot be ignored. The accuracy limits of the LMC can be another reason. According to past work, the LMC cannot reach a theoretical accuracy of 0.01 mm in real conditions, but a high precision (an overall average accuracy of 0.7 mm) for gesture-based interaction is still possible [11].



**Fig. 11.** Registration errors (Color figure online)

## 5 Free-Hand Interaction

To demonstrate the feasibility of natural interactions in MR environment using LMC data, we implemented intuitive hand-based interaction techniques in the HoloLens environment. These interactions are not proposed for given scenarios, but are chosen to explore simple interaction metaphors allowing virtual object manipulation. The aim is to quickly set up a user evaluation to test the proposed method's ease-of-use. We thus used the 3D coordinates of the fingertips and the center point of the left hand to define three manipulation techniques:

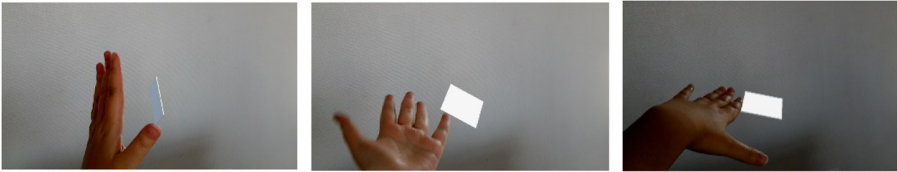
*Select by touch:* When the index finger collides with the virtual cube in the HoloLens coordinates frame, the cube is selected, and its color changes to green (see Fig. 12).



**Fig. 12.** Select and move a virtual cube with free hand (Color figure online)

*Translate:* To move the selected object, the thumb and index finger of the hand must touch each other. To drop the object it simply needs to separate the two fingers.

*Rotate:* The selected virtual object is rotated according to the 3D rotation of the left hand (Fig. 13). For that, we use the coordinates of the five detected fingertips and the center of the hand to estimate the plane corresponding to the palm of the hand. Finally, the normal to this plane is computed, which defines the 3D rotation applied to the virtual object when the hand rotates.

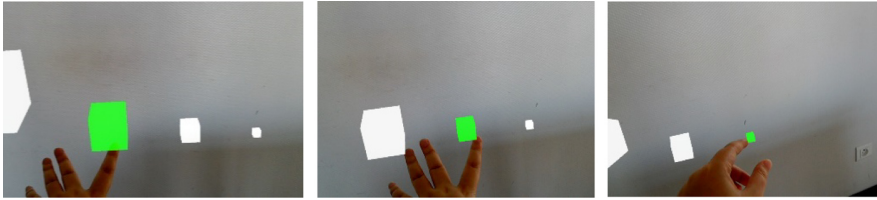


**Fig. 13.** Rotate a plane according to the rotation of the hand

In order to evaluate the usability of our free-hand interaction approach, we carried out different pre-experiences. We defined the interaction space above a 120 cm × 80 cm table. The user sat in front of the table wearing a HoloLens. An LMC mounted on the front of the HoloLens, facing down, allows the detection of the user's hands in his vision field.

We showed the user four virtual cubes in the AR scene with different sizes and depths (Fig. 14). The considered sizes are, respectively: 0.1 m, 0.05 m, 0.025 m and 0.01 m. We asked the user to select the different cubes in turn using his index finger, move them around and place them on the table. We found that the interaction with the three largest cubes is done correctly. However, when trying to touch the smallest one, we face difficulties, which is quite normal; as the positioning of the cube in the HoloLens mixed reality environment has a theoretical error in the range between 0.01 m and 0.02 m.

The second experiment consisted in selecting a cube and rotating it in all directions using the Rotate interaction metaphor. We found that the user could rotate the virtual object placed on his hand in a natural and intuitive way. This is explained by the fact that the user performs a physical gesture in accordance with the direct manipulation done on the virtual object. Indeed, the mental load of users is reduced when their gestures reflect real-world metaphors that allow them to use the most intuitive gestures they desire.



**Fig. 14.** The usability evaluation

Finally, a qualitative evaluation was carried out. Its purpose was to compare the ease and comfort of use between the proposed approaches and the baseline scheme (HoloLens integrated interaction). We have found that overall the interactions that we propose are less tiring and require less effort and concentration, they can be repeated several times over a long period without generating handling errors or hand fatigue. This is not the case with the baseline scheme.

## 6 Discussion

The HoloLens has its own process without using external devices, which provides a stable and fast AR rendering experience. Its spatial mapping is the basis for our calibration procedure. From the results we got, it can be concluded that the combination of the two devices can provide enhanced results in terms of free-hand interaction in MR environments. However, this study has also allowed us to identify some difficulties:

1. According to the official documentation on the HoloLens, the best distance interval to display AR objects in the HoloLens is between 1.25 m and 5 m. When a user wants to use his/her hand to interact, the operating area is always limited to 1.25 m.
2. The field of view of the HoloLens is not very large. It is hard to show many contents in a close range. In other words, the range of activity of the hand is very small. Moreover, the HoloLens screen consists in two separated screens which are located in front of the left and right eyes. When observing close-range objects, it can be difficult for users to combine the projections of both screens due to the change of the focal length of the eyes, which causes severe ghosting.
3. The performance of the HoloLens spatial mapping at close range is not very stable, misjudgments or instability occur frequently. Although the physical location of the LMC relative to the HoloLens is fixed, due to individual differences of user's 'eyes, calibration needs to be performed for each user.

## 7 Conclusion

In this study, we introduced free-hand interaction in mixed reality environments combining a HoloLens and an LMC. Two main issues have been solved. The first one concerns the real-time 3D points acquisition from the two devices. For that, we attached a virtual plane with several balls used as reference points to a real object. This idea allowed us

to acquire 3D points in the HoloLens reference frame and their corresponding points in the LMC reference frame. The second issue is calibration. We used the ICP algorithm to compute the transformation matrix between both frames. The obtained registration error is about 1 cm, which is quite acceptable to realize basic interaction. Finally, first experiments carried out proved the viability of the proposed free-hand interactions and their interest to improve the sensation of presence in MR environments.

Future work will include improvement of the accuracy of our solution by using additional depth sensors like a Kinect. Another interesting possibility would be to implement the whole 3D model of the hand in the HoloLens environment in order to develop more complex and natural free-hand interactions.

## References

1. Aruanno, B., Garzotto, F., Rodriguez, M.C.: Hololens-based mixed reality experiences for subjects with alzheimer's disease. In: Proceedings of the 12<sup>th</sup> Biannual Conference on Italian SIGCHI Chapter, CHItaly (2017), pp. 15:1–15:9, (2017)
2. Benko, B., Jota, R., Wilson, A.: Miragetable: freehand interaction on a projected augmented reality tabletop. In: Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems, pp. 199–208 (2012)
3. Blaha, J., Gupta, M.: Diplopia: a virtual reality game designed to help amblyopics. In: 2014 IEEE Virtual Reality (VR), pp. 163–164 (2014)
4. Carceroni, R.L., Brown, C.M.: Numerical methods for model-based pose recovery. Technical Report 659, University of Rochester, Computer Science Department (1997)
5. Ens, B., Quigley, A., Yeo, H.-S., Irani, P., Piumsomboon, T., Billinghamurst, M.: Counterpoint: exploring mixed-scale gesture interaction for AR applications. In: Proceedings of the 2018, CHI Conference on Human Factors in Computing Systems (2018)
6. Garon, M., Boulet, P., Doironz, J., Beaulieu, L., Lalonde, J.: Real-time high resolution 3d data on the hololens. In: 2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct), pp. 189–191 (2016)
7. Khademi, M., Mousavi Hondori, H., McKenzie, A., Dodakian, L., Lopes, C.V., Cramer, S.C.: Free-hand interaction with leap motion controller for stroke rehabilitation. In: Proceedings of the Extended Abstracts of the 32<sup>nd</sup> Annual ACM Conference on Human Factors in Computing Systems, CHI EA 2014, pp. 1663–1668 (2014)
8. Köhler, N.: Integration of a leap motion controller with the hololens to enable improved gesture interactions. Master's thesis, Aalto University (2017)
9. Lyons, D.M.: System and method for permitting three-dimensional navigation through a virtual reality environment using camera-based gesture inputs (2001)
10. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Patt. Anal. Mach. Intell.* **13**(4), 376–380 (1991)
11. Weichert, F., Bachmann, D., Rudak, B., Fisseler, D.: Analysis of the accuracy and robustness of the leap motion controller. *Sensors* **13**(5), 6380–6393 (2013)
12. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision* **13**(2), 119–152 (1994)