Augment the Multi-Modal Interaction Capabilities of HoloLens Through Networked Sensory Devices

Subhadra Tummalapally

Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/2230

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
AUGMENT THE MULTI-MODAL
INTERACTION CAPABILITIES OF HOLOLENS
THROUGH NETWORKED SENSORY DEVICES

A Thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science

by

SUBHADRA TUMMALAPALLY
B.Tech, GITAM University, 2016

2018
Wright State University
WRIGHT STATE UNIVERSITY
GRADUATE SCHOOL

December 13th, 2018


_____________________________________
Yong Pei, Ph.D.
Thesis Director

_____________________________________
Mateen M. Rizki, Ph.D.
Chair, Department of Computer Science

Committee on Final Examination:

_____________________________________
Yong Pei, Ph.D.

_____________________________________
Mateen M. Rizki, Ph.D. (Co-Advisor)

_____________________________________
Paul Bender, Ph.D.

_____________________________________
Barry Milligan, Ph.D.
Interim Dean of the Graduate School

Augmented Reality (AR) places virtual contents along with real objects to enhance user perception in the physical environment. It is a technology that has shown great potential in many industries such as medicine, education and transportation. AR devices, such as Microsoft HoloLens, come with multi-modal interaction capabilities, such as voice, gesture and gaze, to support intuitive and reliable human-computer interactions. However, HoloLens gesture support is limited to simple gestures like tap, bloom and hold, it stops short to support meaningful interaction through full hand gesture and motion tracking, which may be critical in field operations of various industry practices in noisy environments, such as maintenance, repair, surgical operation and training.

This thesis research intends to augment the HoloLens with fully-fledged hand gesture and motion tracking capabilities by interconnecting it with a Leap motion controller through the Internet, resulting in a functional prototype AR device capable of hand gesture/motion controlled operation. This prototype also enables the use of AR as a training device for skills that require hand operations, such as in surgery, repair, maintenance and etc. In this thesis, we present our system design and prototype
implementation and testing, and also present a use case to illustrate its usage and value in real world working environments.

KEY WORDS: Augmented Reality, HoloLens, Leap Motion Controller, Gestures, Integration, Real-time applications.
## CONTENTS

1. INTRODUCTION .................................................................................................................. 1  
2. HOLOLENS .......................................................................................................................... 5  
   2.1 HARDWARE SPECIFICATIONS: ......................................................................................... 6  
   2.2 HUMAN-HOLOLENS INTERACTION BUILDING BLOCKS: ............................................. 8  
      2.2.1 GAZE: ....................................................................................................................... 8  
      2.2.2 GESTURE: ............................................................................................................... 9  
      2.2.3 Voice: ...................................................................................................................... 10  
   2.3 HOLOGRAM BUILDING BLOCKS: ..................................................................................... 10  
      2.3.1 Spatial sound: ......................................................................................................... 10  
      2.3.2 Spatial anchors: ....................................................................................................... 11  
      2.3.3 Spatial mapping: ..................................................................................................... 11  
3. LEAP MOTION CONTROLLER ............................................................................................. 13  
   3.1 Motion Tracking Data: ..................................................................................................... 13  
   3.2 Use Leap for Computer Interactions: .............................................................................. 16  
4. DEVELOPMENT OF A NEW PROTOTYPE AR DEVICE .................................................... 18  
   4.1 UNITY 3D: .................................................................................................................... 19  
   4.2 Holo-Toolkit Sharing Service: ....................................................................................... 20  
   4.3 Streaming Leap Tracking Data Into HoloLens Using Sharing Service: ....................... 23  
      4.3.1 Building for HoloLens: ........................................................................................... 27  
   4.4 Holographic Remoting ................................................................................................... 28  
   4.5 PROTOTYPE DEVELOPMENT: ...................................................................................... 30  
5. CONCLUSION ....................................................................................................................... 34  
REFERENCES ......................................................................................................................... 36
LIST OF FIGURES

Figure 1.1 Showing difference between VR and AR [1] 2
Figure 1.2 Mixed Reality Technology [2] 3
Figure 2.1 HoloLens [3] 6
Figure 2.2 Interaction with hologram using Air Tap gesture [3] 9
Figure 2.3 Spatial Mapping [3] 12
Figure 3.1 Coordinate System of Leap Motion Controller [5] 13
Figure 3.2 leap motion fingers and bone structure [5] 15
Figure 3.3 showing virtual hand alignment on human hands in Visualizer 17
Figure 4.1 Sharing hologram between two users using sharing service [4] 21
Figure 4.2 Sharing Service in UNITY 22
Figure 4.3 Use Case diagram to show establishment of network connection [6] 24
Figure 4.4 tracking physical hand moments and performing actions 26
Figure 4.5 sending data from server to client 27
Figure 4.6 Holographic remoting in unity 29
Figure 4.7 Holographic remoting in HoloLens [7] 30
Figure 4.8 Leap hand reaching car wheel 32
Figure 4.9 Leap hands trying to Remove car Wheel 32
Figure 4.10 Leap Hands removes car wheel 33
1. INTRODUCTION

We are living in a world where technology growing day by day. Imagine if we have technology through which we can see, feel and hear what others can’t. In other words what if we have technology to get computational graphical elements or objects into our real world? If we get such technologies soldiers can find where enemies are located, surgeon can guide juniors in proper way to do surgeries and mechanics can repair automobiles following instructions provided by subject expert.

Augmented Reality (AR) [6] is such technology through which we can feel, hear and see virtual objects in real world by placing them along with real world objects. Here makes it different from what we call as virtual reality. When we are using virtual reality technology it takes user into imaginary virtual world, we do not see any real-world objects, nor we have contact with real world objects until we come out form it. While using Augmented Reality we are placing virtual objects along with real objects. By doing this we are not losing connection with our real world. Example: Pokémon go game. Microsoft took a step ahead in this technology and created Augmented reality glasses called HoloLens.
In Figure 1.1, we can see difference between virtual reality and augmented reality. The person with virtual reality glasses can’t see anything in the real world or interact with real world object. While in augmented reality we are placing object in real-world we can see and interact with both virtual and real-world objects at the same time.

HoloLens has unique features which are developed towards the Augmented reality. Using HoloLens, we can create holograms and place these virtual holograms in real world without using any markers. Microsoft called this device as Mixed Reality Device because its combining physical reality with digital reality. With great infrared sensor capabilities and see-through lenses, it creates spatial mapping using which HoloLens can embed and align holograms to the real world. This device has its own operating system so that we do not tether this with any device.

HoloLens takes inputs like Gaze, Gesture and voice commands as input to make interaction with these holograms. Gaze means simply gazing at application we can select it then by hand gestures or voice commands we can perform actions on holographic applications. We have detailed discussion about HoloLens capabilities, software and hardware functionalities in later chapter of the thesis.
However, these interaction methods available in HoloLens have many limitations. HoloLens only provides limited gesture identification and support, and creating customized gestures is not provided so far. Using voice input we cannot interact with holograms all the time especially when we are working in noisy environments. For instance, when senior expert surgeon is trying to explain about a surgery to their junior surgeons, using voice commands alone may not be able to conveniently demonstrate the safe operation procedure continuously and consistently, which is essential to communicate clearly to the trainees.

In this thesis we come up with solutions to address this problem by integrating networked position/gesture/motion tracking sensory systems, such as leap motion controller and/or Kinect, with HoloLens that potentially support more capable gesture/motion based human-AR device interactions. For instance, Leap motion controller is hand tracking sensor device which take human upper arm, hand and fingers as inputs and produce high-quality tracking of write/hand joints. When we place our hands top of this device its stereo infrared imaging sensors that scan the human hand in very high video sampling frames rate (>100 frame per second) and send these data to leap machine learning engine to produce 3D motion information of the hand. So, by this integration, our goal is to sending this hand motion
tracking results in real time to HoloLens, using which we can perform better actions with holographic applications.

In the coming chapters of this thesis, we will look deep into HoloLens and leap motion software and hardware controls and their features. Then we show how we are integrating this to devices, followed by discussions on some real-world applications with these augmented capabilities.
2. HOLOLENS

HoloLens is a Holographic Computer built into headset by Microsoft. It leads a new era towards Augmented Reality. Through HoloLens we can create and visualize the virtual holograms clearly and pin them to real world. It differs from the virtual glasses available in market which takes us to virtual world. With HoloLens virtual holograms are pinned to a user’s real world that literally augments the user’s view. Without losing connection with real world we can visualize and feel the virtual holograms within the augmented reality. That’s the reason Microsoft named it as Mixed reality [10] headset.

HoloLens supports three types of human-friendly interactions namely gaze, gesture and voice, in addition to USB-connected mouse or keyboard. By using any type of these interactions, a user can interact with the augmented reality presented through HoloLens. There are hand gestures like air tap to open application, bloom to start windows. We can also use voice commands like ‘close’ to close any application, and we can give customized voice commands to hologram to perform the required operations. Similarly, we can use gaze interaction to interact with holograms like we can move hologram from one place to another by simply gazing at them. So, by using any of this interaction types we can easily interact with holograms. Based on user comfort level we can customize these interaction types while building hologram based applications.
2.1 HARDWARE SPECIFICATIONS:
HoloLens is a complete holographic computer built on Windows 10 OS frame. HoloLens use see-through high definition lenses, also called wave guide lenses, to project holograms into user’s audiovisual view of real world, it has spatial sound capability through which we can hear from the holograms even when they are behind us. It also uses advanced sensors to capture the information about the physical environment and project and align holograms into it. Capable of operating untethered, HoloLens has its own high-resolution Central Processing Unit and Graphics Processing Unit. It has semi-transparent holographic glasses which works as screen for projecting holograms in to real world. HoloLens has to process Tera bytes per second of data captured from various built-in sensors through Artificial intelligence, such that we are able to see and interact with Holograms. Thus, in addition to
the CPU and GPU, HoloLens developed its own processor called Holographic Processing Unit (HPU) with TPM 2.0 support to enable such operation more battery power- and computation-efficient.

For object orientation and head tracking it has 1 Inertial Measurement Unit (IMU) with the combination of accelerometers, gyroscope and magnetometers. HoloLens has 1 2megapixel camera to take photos and record the video. It has 4 environmental studying cameras two on each side, which provides us basis for head tracking capabilities. It has one depth camera which serves us two main roles. One is hand tracking and another one is using depth camera we are able to do surface reconstruction through which we are able to place holograms in physical world. It also has four microphones and one ambient light sensor through which we are capable of doing noise compression. Along with these HoloLens has stereo type speakers exactly on top of ears which provide sound for holograms through which we are feeling the reality in holograms. For connection purpose HoloLens has Wi-Fi and Bluetooth applications. While coming to battery if we charge it fully, we can use HoloLens for 2-3 hours on full charge. It stays 2-3 weeks on standby. All the functionalities work fine even if we plug it in for a charge.

For application developments, HoloLens supports Universal Windows Platform (UWP). So, any app, either 2D or 3D, built on UWP can run on a HoloLens.
2.2 HUMAN-HOLOLENS INTERACTION BUILDING BLOCKS:
As we mentioned above, we can interact with holograms using gaze, gesture, and voice.
We will see now how they actually works.

2.2.1 GAZE:
Gaze [13] is the primary way to interact with holograms. While wearing the HoloLens, we can gaze at an object which we want to interact with. Although it looks like using the user’s eyes to interact with objects, HoloLens actually uses the user’s head orientations to gaze at object to interact not the user’s eyes. For visual cues, Gaze uses cursor to indicate its direction in the view. As user’s head moves, the cursor follows their point of attention. Usually the cursor is non-directional as it only indicates at which object user is currently gazing. Consider this gaze vector as laser pointer between two eyes, as we are looking around when the object intersects with this ray vector, HoloLens associates the cursor with that object. Cursor also provide depth and dimensional information of the object that it is intersecting. By intersecting ray vector with spatial mapping of the physical environment, we can also place the holograms in real world. And, we can conveniently differentiate on which user is gazing either on virtual holograms or real-world objects using this spatial mapping mesh. Once we target the object using gaze, we can perform other operations using hand gestures or voice commands along with gaze input. For example, you can gaze at a hologram and air tap using hand gesture to complete a selection operation, so that hologram will then follow your gaze. Alternatively, you can give some voice commands along with gaze to interact with a hologram.
2.2.2 GESTURE:
This is like a mouse click on our desktops. Once we target the object using gaze, we can start performing operations on the target object using hand gestures[13]. Once we place our hand in the field of view that means we are ready to perform actions on targeted object. The two core gestures of HoloLens are Air tap and bloom. Air tap is almost the same as mouse click, i.e., by using air tap, we make a selection. Bloom is specifically used to start the main menu in HoloLens. Using bloom is almost like going back to home screen by clicking windows button in keyboard. By using hand bloom gesture in front of HoloLens we can open home screen. By performing another bloom gesture while at the home screen, we then close the home screen. HoloLens also supports limited navigation gestures and manipulation gestures for rotation of object and resizing of object. It should be noted that HoloLens only recognize the ready state that means pointing index finger up or pressed state means holding index finger down. Other hand movements are not recognized by HoloLens. Furthermore, to perform the actions successfully on target, the gestures should be in a range of gesture frames.

Figure 2.2 Interacting with hologram using Air tap Hand Gesture [13]
2.2.3 Voice:
This is the third form of giving command to interact with the holograms. Instead of using hand gestures we also give voice commands to perform operations on holograms. While we gaze at an object of interest, we can give voice input to perform necessary selection operation. Instead of air tap to select application using hand gesture we can simply say ‘select’. We can also customize our voice commands and holograms can reply to our commands. We can simply say ‘hey Cortana’ and then give voice input for what you want to search for. Voice input is best and easy way to interact with holograms in any noise-free environments.

2.3 HOLOGRAM BUILDING BLOCKS:
In this section, we will look at the different audiovisual features of holograms.

2.3.1 Spatial sound:
Generally, holograms are combination of light and sound. We can view the holograms when it is in our line of sight. When we are unable to visualize object, we can find them using sound. HoloLens use aural component of mixed reality by simulating 3D sound with direction, environment and distance. By using these spatial sounds developers can adjust sound in 3-dimensional space around the user, which gives more realistic experience with holograms. Although we can only visualize holograms in user’s point of gaze, the spatial sound can come in from all directions. If you are close to the hologram, the sound volume increases. If you move away from the hologram, its sound decreases. Spatial sound in HoloLens uses customized Head Related Transfer Functions (HRTF) which simulates the sound and reflects their directions and distance to the ear. All the sounds which are coming from holograms are mostly spatialized. In HoloLens we can use 10-12 different types of spatial sounds [3].
2.3.2 Spatial anchors:
Spatial anchors [3] are one of the best features available in HoloLens. Each anchor is associated with one coordinate system. To pin holograms to real world we use spatial anchors. Once we give anchor to any hologram it will be fixed to that place until user removes it from there. Using this spatial anchor, we can also pin hologram in multiple places.

2.3.3 Spatial mapping:
Spatial mapping is one of the core feature in HoloLens to place holograms in real world. With this spatial mapping [3] it creates virtual layer over physical in order to accommodate virtual holograms in the physical world. To do that first it scans user’s environment and creates 3D virtual mesh inside HoloLens using which we are going to place holograms in user surface.

The top picture in Figure 2.3 illustrates the creation of virtual mesh by doing mapping with user surface. After completion of mapping it creates virtual mesh surface and place holograms on top this mesh which is show in the bottom picture (placing objects on cushion and table).
What we discussed till now are the selected features of HoloLens. There are some disadvantages too to give input to holograms when using gaze, gesture and voice. The hand gestures for HoloLens are very limited as we can’t use full hand motions. It only recognizes tapping and bloom. And in noisy places we can’t use voice to give input to holograms. These limitations are results of limited battery life and computation capacity of current HoloLens. To overcome limited hand gestures in this thesis we are augmenting HoloLens with leap motion sensor which takes hand and finger moments as input. By paring this leap hand tracking device to a HoloLens we can use full hand functionality in HoloLens to support human-HoloLens interaction. It will help overcome the problems of limited hand gestures, particularly, when voice may not functional properly, e.g., in any noisy environments.
3. LEAP MOTION CONTROLLER

Leap motion controller is a hand motion tracking device which takes upper arm, hand and finger movements as inputs and produces comprehensive list of joint/tip position information in real time through its machine learning engine[2]. It is a small USB device which consists of 2 infrared cameras and 3 LED’s. It can capture hands approximately from 1 inch to 2 feet from the device within a field of view of 150 degrees. Hand tracking works better when leap controller has clear and high-contrast view of objects [2].

![Right handed coordinate system of leap motion controller](image)

Figure 3.1 Right handed coordinate system of leap motion controller [1]

3.1 Motion Tracking Data:

Leap motion tracking is achieved through its machine learning engine, which has structure models of human hand. When we connect a controller to a computer using USB
cable and place our hands in its field of view, these sensors and infrared camera capture stereo videos of our physical hand. It then compares the collected stereo video data (our physical hand) with internally stored human hand model and produces 3D tracking data. Leap will output the tracking data in the form of FRAMES. These frames consist of 3D position data of identified hand, figure, joints. Estimate is made at time even when particular view is not available. Leap cameras can produce 200 frames per second, but the processing speed may be limited due to computer capabilities [1]. The leap motion controller follows right handed coordinate system as shown in Figure 3.1.

Using leap motion controller, we can track the human hand in real time. Instead of storing the captured hands as picture leap produce a broad spectrum of hand data such as hand position, arm position, joint position as well as joint angles [1]. Its API provides measurements for distance in millimeters, time in microseconds, angle in radians and speed in millimeter/second. Frames are the root for leap data model. Using the frame data, we are able to reconstruct virtual hands to closely reflect the actual hand’s size, position and motions. Each frame is identified by its own frame id along with hands and finger data. Hands are represented using Hand class in leap. Leap motion controller gives precise data when object is clear and high-contrast view. Hand provides information like position and identity. Leap compares hand sensor data with internal human hand. When the hand is not completely visible it takes data from internal hand and compares with previously available observations along with currently available hand data to calculate each individual finger.
Each figure is differentiated by its type (thumb, index, middle, ring and pinky).

Each human finger except thumb has four types of bones like metacarpals, proximal phalanges, intermediate phalanges, distal phalanges. Human thumb has only 3 types of bones so leap treats thumb metacarpals as zero metacarpals for thumb. So, using this frame data, we can reconstruct virtual hands in visualizer [1].

![Diagram of human hand bones](image)

**Figure 3.2** Leap motion fingers and bone type structure [1]

Leap motion controller also takes motion factor changes of hand. Motion factors refer to the changes in the position of an object, such as scale, rotation and translation. By using the motion factors per frame or per hand we can create commands for different interaction tasks accordingly. For example, to change the position of an object, we can use finger motions by measuring the changes in the scaling factors of corresponding fingers in successive frames. Leap has provided gesture class and subclasses for developers to
identify and define hand movements as gesture-based command to perform corresponding operations.

3.2 Use Leap for Computer Interactions:

Now we will look into what we have to do to use leap motion controller to support touch free or noncontact interaction with computing devices. First, we have to download leap motion SDK from leap motion website based on our operating system either windows or mac, then run the setup program. After completion of setup we can see horizontal black box with white boarder in scroll bar. Now if we connect leap to system this black box turns to green that means your leap is in active state. After making connection if it is in red then your leap sensor is not working.

When we downloaded SDK, we will get application called Visualizer where we can see how leap is capturing stereo videos of our hand and create virtual hand there. By downloading Air Space Store, we can get different commercially available applications of leap which shows how Leap could be used in different applications.

Now, we can start developing different applications using these virtual hands, particularly for virtual and augmented reality applications, using UNITY 3D, e.g., as illustrated in Figure 3.3.
Unfortunately, Leap motion controller can’t be directly attached to a HoloLens device. It is not due to the USB connector, instead it is due primarily to the limited computation power and battery capacity of the HoloLens device for sufficiently supporting the Leap machine learning engine at such high frame rate. Thus, Leap has not released a driver for HoloLens.

In this thesis, we will take a networked sensory system approach to interconnect a leap controller with HoloLens through the Internet to augment HoloLens motion tracking and human interaction capabilities.
4. DEVELOPMENT OF A NEW PROTOTYPE AR DEVICE

As stated in earlier chapters, HoloLens has limited gestures tracking capabilities and creating customized gestures for holographic applications remains as a challenge for application developers. In this thesis we are going to present the development of a new prototype AR device that integrates Leap’s fully-fledged motion tracking capabilities with HoloLens, by which we can use the entire hand motions for human-AR interactions. Specifically, by interconnecting and integrating leap motion and its full hand tracking capability with HoloLens, we can enable users for more intuitive and maneuverable multi-modal interactions with the Augmented Reality (AR).

At the first glance, the integration of the Leap with HoloLens should be as simple as plugging Leap controller with HoloLens using USBs. Unfortunately, Leap motion controller can’t be directly attached to a HoloLens device. Although HoloLens has mini USB-2.0 for charging, it won’t take any input devices. This is due primarily to the limitations imposed by the computation power and battery capacity of the HoloLens device. Simply put, HoloLens may not have sufficient resource for supporting the Leap machine learning engine at such high frame rate. Thus, Leap has not released a driver for HoloLens.

In this thesis, we will present a networked sensory system approach to interconnect a leap controller with HoloLens through the Internet to augment HoloLens motion tracking
and human interaction capabilities. So, we will connect leap motion device to a PC, e.g., a pocket PC or tablet, from where we will send hand tracking data through Internet to the HoloLens. Moreover, as HoloLens is developed on Universal Windows Platform (UWP) and leap motion controller don’t have any software development kit for UWP we will use UNITY 3D platform which was supported by both devices for the system integration.

Specifically, we have two approaches to connect HoloLens and Leap motion together for sharing the motion tracking data: 1) using HoloToolkit Sharing (by creating server and client app). 2) using the Holographic Remoting utility tool available in Unity3D. In this thesis we integrated the prototype system using both approaches. In the rest of the thesis we will show how to successfully integrate the system and how the prototype AR system can be used in real time applications.

4.1 UNITY 3D:
Unity 3D is a video game engine platform developed by UNITY where we can develop 2D and 3D games and distribute them in different platforms by making a particular build to that platform accordingly. Before starting develop, a game using UNITY, we first download and install the proper version of UNITY based on our operating system properties which supports all our requirements and needs to develop the game. After completion of setup we can use UNITY in our system to develop 2D or 3D applications. We can download required software development kits (SDK) from unity asset store. Unity also supports scripting. Any script written in C# or JAVASCRIPT together with required Unity libraries can run on the Unity machine [9].
For our prototype system development, we will use C# scripting in Unity. We also download and install Leap Core SDK and HoloLens SDK which includes the Holo-Toolkit. These assets are imported into our Unity project.

4.2 Holo-Toolkit Sharing Service:
HoloLens has feature called network streaming through which we can see and record what the person wearing HoloLens is seeing just by entering IP address and password. It can also transfer and share holograms to multiple HoloLens. Spatial anchors are used by HoloLens to pin holograms to certain place in real world. So, if we are able to share these anchors together with information about holograms, we can see and exchange holograms information remotely using the anchor tags. To do that the user who wants to share a hologram creates anchor and stores it in anchor store form where, another user can get the anchor.

To share this anchor, we first have to establish a network connection. There are two convenient ways to handle the network connectivity in Unity. We can use Unity built-in networking function called UNET or the SHARING SERVICE solution provided by Microsoft. Through either way, we can establish network connectivity to exchange anchors and messages to support shared experience. This sharing service is built on top of open source multiplayer gaming network called RACKNET which has features like connection management, session management and broadcasting.

Simply put, sharing service creates a room to share the messages with multiple users. The user who want to share their information will send their anchor and/or the message to sharing service server. The users who wants to access this information joins the room and takes information from the sharing service. Sharing service self has functions for
socket and network connection. So, it acts as authority server, and the user who wants to share their information will give its IP address to the sharing service[10] server along with the information he wants to share. Then, other users can use this IP address to join the particular sharing room and access the corresponding content.

Figure 4.1 Sharing hologram between two users using sharing service [10]
To use the sharing service in our applications in Unity, you need to add the Microsoft mixed reality toolkit to the application. Then, drag the sharing prefab from this toolkit and add this to your game object which will be shared. Now, launch sharing service from toolkit icon on top of Unity and complete all required settings in your game project as shown in Figure 4.2. After launching the sharing service, enter the IP address of the content provider, i.e., the user who provide holograms to share, in sharing.exe window in the sharing prefab in game object. Now the users who want to access these game object in their game simply add sharing prefab into their game and enters this IP address there. So that they can join the network and can get the information for their game object. In general this is considered a distributed solution, as the server thread is running at the content provider, and there is no need for a centralized sharing services. This make the solution more scalable. The downside of this approach is the need to disseminate the server IP info to the potential participants who receives the shared data. Fortunately, for most distributed collaboration applications, this info is usually already available among the collaborators. Next, we look at the particular configurations to set up the server.

Figure 4.2 Sharing service in UNITY
In Figure 4.2, we can see black console with IP address. This is sharing service console which we launch form mixed reality toolkit. So, to make the connection, the IP address shown in the console will be entered at the server address column under sharing prefab (the highlighted box in the right panel). We always launch sharing service in server application which want to share information. By using server IP address, the people who want access contents, can now join the sharing Room.

### 4.3 Streaming Leap Tracking Data Into HoloLens Using Sharing Service:

This approach requires significant development efforts at prototype stage, but is very efficient once completed and when deployed. To begin this process of streaming, first we have to do these following steps.

1. On any version of Windows 10, download and install Leap motion SDK from Leap Motion web site.

2. After completion of setup, open Leap visualizer which we mentioned in Chapter 3 and connect leap device to the PC and check in visualizer if hands are visible. Leap provides troubleshooting tips on its website to help you correctly install the SDK.

3. Next, download latest version of UNITY 3D which is supported by both devices from UNITY web page and run the setup.

4. Then, download software development kit (SDK) for HoloLens from its websites or from Git-hub which are supported by Unity version we have downloaded. HoloLens SDK is also called as Mixed Reality Toolkit. This is the basic process, which we have to do in both methods of interconnecting Leap and HoloLens.

5. Now, we will create two projects namely client and server in Unity. Leap is connected to a PC which would be our server. It will stream data to the client project which is deployed
in HoloLens. The data connection is established between the server and client projects using Sharing Service.

![Server Client Architecture](image)

**Figure 4.3 Server Client Architecture [8].**

As mentioned in Chapter 3, leap produces hand and finger motion data and stores them in frames which are the data being streamed from server to clients. To access these data at the server end (i.e., at the PC), we will add the Leap core asset into our Unity projects. The Leap core asset includes frame class together with different hand graphical models which we can use as virtual assets in our game applications.

We can then add leap motion controller prefab which is under prefab folder in this core assets into our project and import all the required scripts which we will need. Upon completion, when we connect leap device to the PC, we can control a virtual hand game object by moving our hands within the field of view of leap device. Leap SDK provide a rich set of hand models that help reduce the scripting required to make it operational.
Next, coming to the client end, we will first validate our solution use another PC as client before we move to a HoloLens. We will first add leap motion controller prefab and virtual hand models from SDK to the game object. Then major development effort is to establish the data connection. We are using Unity C# scripting to write the scripts that send hand and finger position and rotation data to client in the format which is supported by both Leap and HoloLens. These data can usually be in JSON format. For network connection, we will also use JSON serializer to encode the data to JSON string. Using custom messages class which is available as part of Sharing service, we send the JSON strings as message to client. At the client end, by using JSON deserializing parser, we decode this string and receive the motion data.

Finally, we can launch sharing service in server app and give the server IP address to client. Now, when we run the client project and we can control a virtual hand game object by moving our hands within the field of view of leap device at the client end as if a Leap device is directly connected to the client, here the HoloLens. At this point, we have successfully validate our client-server based networked sensory solution for supporting Leap-based interaction distributedly. Next, we will extend this approach to complete a prototype AR system with fully-fledged hand motion tracking capabilities for supporting enhanced multi-modal interaction.
Figure 4.4 Use leap to perform interaction in Unity
4.3.1 Building for HoloLens:
To deploy a project from Unity 3d to HoloLens, we need to make a build of the application for HoloLens. To make build for HoloLens, we have to download UWP metro support into the Unity project. With this download and any visual studio version above 2015 that supports the Unity version where we are creating the project, we can then add all the necessary functions which we need for HoloLens and Unity [9]. Now we can go to file and select UWP under player settings to do all necessary settings following the documentation provided by Unity and Microsoft and make a build for the project.

After successfully making the build, close the Unity project and open the project folder and open the visual studio solution which is under the build. Now in visual studio,
do necessary settings and choose remote device and run the application following Microsoft documentation. When we choose remote device, it asks for machine address. Turn on HoloLens and enter its IP address and continue if the debug was successfully done then we can see our app starting in HoloLens. Once it started there, we can see under sharing.exe saying it is connected then play the server project in PC by connecting leap. At this point, a prototype AR system with fully-fledged hand motion tracking capabilities for supporting enhanced multi-modal interaction is completed.

4.4 Holographic Remoting

Holographic remoting[4] is an alternative method to support the networked sensory connection between the PC and HoloLens. By adding holographic remoting to our PC and HoloLens app, we can visualize the same contents at the PC as what our application is presenting in a real HoloLens device or in emulator.

Download remoting player in HoloLens from app store. Remoting player is companion app through which we can visualize PC or UWP contents in HoloLens using HoloLens remoting. It only supports contents which are developed using holographic remoting.
Unity wraps the holographic remoting, libraries in a way so that we can use Holographic Remoting in UNITY 3D engine. Launch holographic remoting, into your game from windows scroll bar as shown in Figure 4.6. After completion of launching we can see holographic remoting application in our game as it is there in the right-side corner of the picture. Now under this Holographic application we will change emulation mode to remote from device and under the remote machine enter IP address of HoloLens which is shown in holographic remoting player as shown in Fig. 4.7. Now by hitting the connect button we can make connection with HoloLens. When connection was successfully established the connection status under holographic application will change to connected. After making connection, click the play button in Unity 3D and start the game. Now data will stream into HoloLens and we can see whatever the scene playing in Unity 3D also in HoloLens headset.
For our application, we add Leap assets and add one of the Leap provided hand model to our game object in Unity 3D and connect Leap sensor device to PC and hit play button so that we can see hand moments in HoloLens by making connection using holographic remoting.

![Holographic Remoting Player](image)

Figure 4.7 screen image of holographic remoting in HoloLens [4]

4.5 PROTOTYPE DEVELOPMENT:

Now we are successfully streamed Leap virtual hands into HoloLens by which we can use our entire hand moments as gestures in HoloLens. But when coming to real time applications, how are we going to use this hand gesture capabilities? Till what range are they useful? To answer these questions, we are developing one simple Augmented Reality app which emulate a typical procedure in car maintenance: disassembling and assembling a car wheel using virtual Leap hands controlled by your actual hands. We can make connection between HoloLens and Leap device using holographic remoting. This
application is simple but sufficient to illustrate the use of our proposed AR prototype system for efficient skill training, refresh and assessment to for students, technicians, surgeons, and other professions. It can be straightforwardly extended for many use cases. It can also give a very helpful hand for many who like to Do-it-Yourself.

To quickly put together the demo app, we directly use one 3D car model available from Unity asset store into our game and add Leap motion asset to that game scene. From leap core asset, we then add capsule hand model prefabs to game object. Leap motion handily facilitate us with a great feature called interaction manager which is a one of core behavior of leap motion controller. Interaction manager manages the interaction behavior of the game objects in scene. It provides necessary functions which we can manipulate and call them in game whenever we need them. It keeps a track record to keep track of objects and hands. Using this interaction manager, hand models of the leap motion device and adding some necessary scripting we can disassemble the wheel, rotate the car and assemble the wheel back to the car without touching the car with physical hands. Now using Holographic remoting, we can stream this to HoloLens and can demo steps to change a car wheel by visualizing things, which helps people to follow the safe operation procedures correctly and easily as shown in Figure 4.8 and 4.9.
Figure 4.8 Leap hands reaching the car wheel

Figure 4.9 Trying to Remove the car wheel using leap hand models.
From above figures 4.9 and 4.10 we can see that leap hands are reaching the car wheel. By taking physical hand moments leap hands removes the car wheel. We can see in figure 4.9 leap hand rotating car wheel in anti-clockwise direction to remove the wheel. In the above right corner circle showing wheel number (say number 3) and percentage (65%) rotation. when this percentage reach 100% the car wheel successfully comes out which we can see in figure 4.10.
5. CONCLUSION

HoloLens takes gaze, gesture and voice commands as input to interact with holograms. The main limitation of these input methods, particularly for supporting skills training and assessment when using hands, is that HoloLens doesn’t have a full-fledged hand gesture tracking capabilities, and voice commands is not intuitive and may fail when in a noisy environment. Because of these reasons we may not be able to interact with holograms properly.

To overcome these problems, we intend to develop a multi-modal interactive augmented reality prototype system. To rapidly conceptualize and prototype our AR system, we have used a distributed solution, where we use off-shelf devices such as leap motion controller together with its machine learning engine and HoloLens to produce AR system with full hand tracking capabilities. We have presented two different approaches to connect the Leap device to a HoloLens in prototyping: using HoloToolKit sharing service, or using holographic remoting player. We have successfully validated both approaches through our prototype design, development and testing. Furthermore, we developed an AR application which disassemble and assemble car wheels using Leap-controlled virtual hands, and demonstrate its superior interactive performance compared to using HoloLens by itself.

In conclusion, by integrating leap motion device with HoloLens in a networked solution, we have successfully built a novel augmented reality prototype system that is capable of using full hand gestures of human-AR interaction, and enable a broad range of
highly interactive AR applications, particularly for supporting skills training, refreshing and assessment when using hands, such as in maintenance, surgery, and etc. We believe, many meaningful real-world applications can be adapted, developed and deployed using our solution.
REFERENCES


