Developing a Kinect based Holoportation System

Soumya Chiday

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
Chiday, Soumya, "Developing a Kinect based Holoportation System" (2018). Browse all Theses and Dissertations. 2231.
https://corescholar.libraries.wright.edu/etd_all/2231

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.
DEVELOPING A KINECT BASED HOLOPORTATION SYSTEM

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

by

SOUMYA CHIDAY
B.Tech., Jawaharlal Nehru Technological University, India, 2016

2018
Wright State University
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Soumya Chiday ENTITLED Developing A Kinect Based Holoportation System BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

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
ABSTRACT


Holographic communication and distributed collaboration offer great potential for empathic computing to help remove the cost, distance, language and expertise barriers in many social and economic activities. Recent advances in AR-enhanced communication as evident by Microsoft Holoportation technology demonstrate the progress toward fully immersive collaborations distributed and remotely. Current holoportation system requires the use of extensive camera-arrays and powerful server system due to the computation demand and sensory needs to capture and reconstruct the subject of interests. Thus, they suffer in mobility and applicability in real-world scenarios.

In this thesis, we present an ultra-portable holoportation system design that requires only modest hardware supports that include a Kinect v2 sensor and PC/Laptop in addition to the HoloLens devices. Using Unity 3D engine on Universal Windows Platform and HoloToolKit sharing service, we have implemented the software for holoportating a 3D holograms of a person from a remote location to a local HoloLens device over the Internet.
This low-cost and ultra-portable holoportation system could be potentially used in various industries for skill training, real-time assistance from subject expert.
# TABLE OF CONTENTS

1. INTRODUCTION ........................................................................................................... 1  
   1.1. Telepresence ........................................................................................................... 1  
   1.2. Augmented Reality ................................................................................................. 1  
   1.3. Potential Applications in Different Fields .............................................................. 2  

2. MICROSOFT HOLOLENS ................................................................................................ 4  
   2.1. What is HoloLens? .................................................................................................. 4  
   2.2. Specifications .......................................................................................................... 5  
      2.2.1. Cameras and Sensors: ..................................................................................... 6  
      2.2.2. Voice Commands: ......................................................................................... 7  
      2.2.3. Hand Gestures: ............................................................................................ 7  
      2.2.4. Spatial Mapping: .......................................................................................... 8  
   2.3. Limitations ............................................................................................................... 9  
   2.4. HoloLens and Kinect Collaboration ........................................................................ 9  

3. MICROSOFT KINECT ..................................................................................................... 11  
   3.1. What is Kinect? ...................................................................................................... 11  
   3.2. Kinect SDK ............................................................................................................ 12  
   3.3. Cameras and Data Streams ..................................................................................... 12  
      3.3.1. IR Projector and Depth Camera: ..................................................................... 12  
      3.3.2. RGB Camera: .............................................................................................. 13  
      3.3.3 Skeletal Data: .................................................................................................. 14  
      3.3.4. Construct 3D holographic video: ................................................................. 14  

4. HOLOPORTATION ........................................................................................................... 15  
   4.1. What is Holoportation? ......................................................................................... 16  
      4.1.1. Cave Camera Set-Up: .................................................................................... 17  
   4.2. Mobile Holoportation ............................................................................................. 18  
   4.3. Limitations ............................................................................................................ 19  

5. IMPLEMENTATION OF KINECT-BASED HOLOPORTATION PROTOTYPE SYSTEM .... 20  
   5.1. Unity 3D Engine .................................................................................................... 20
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1. An expert helping customer remotely</td>
<td>3</td>
</tr>
<tr>
<td>Figure 2.1. Components of HoloLens</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.2. Bloom Gesture give this Home Page on HoloLens</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.3 Spatial Mapping of Living Room</td>
<td>9</td>
</tr>
<tr>
<td>Figure 3.1. Components of Kinect</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3.2. IR Emitter Projects Dots in Living Room to generate depth data</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4.1. Empathic Computing</td>
<td>16</td>
</tr>
<tr>
<td>Figure 4.2. Cave Camera Set-Up</td>
<td>17</td>
</tr>
<tr>
<td>Figure 4.3. A person wearing HoloLens talking to his daughter’s hologram</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4.4. Manipulatable recorded holographic videos</td>
<td>18</td>
</tr>
<tr>
<td>Figure 4.5. Person in Car (Server End) and Holoported person (Client End)</td>
<td>19</td>
</tr>
<tr>
<td>Figure 5.1. Prototype Architecture</td>
<td>20</td>
</tr>
<tr>
<td>Figure 5.2. Transform components of a game object</td>
<td>22</td>
</tr>
<tr>
<td>Figure 5.3. Two devices connected on HoloToolKit Sharing Service</td>
<td>23</td>
</tr>
</tbody>
</table>
Figure 5.4. Image diffused as material on depth mesh .................................... 26

Figure 5.5. Colored 3D hologram at the Server End ........................................... 27

Figure 5.6. Mesh created from the Depth data on Server (Right) and Client (Left) connected by HoloToolKit Sharing Service ................................................................. 28

Figure 5.7. Colored 3D Hologram at the Client End ............................................. 28

Figure 5.8. Use Holographic Remoting to debug Hologram view before Deploying as a HoloLens App ................................................................. 29

Figure 5.9. Hologram in client HoloLens .............................................................. 31

Figure 5.10. Stick Man defining human joints ....................................................... 32

Figure 5.11. Stickman aligning with depth mesh ................................................. 33

Figure 5.12. Random Avatar aligning with depth mesh ....................................... 33

Figure 5.13. 3D scanned model of a person using 3D scan app (all view) ............ 34
1. INTRODUCTION

1.1. Telepresence

Telecommunication methods keep on striving to achieve more and more realistic ways of telepresence [9]. Imagine: attending meetings as a virtual person, consulting your doctor without any physical appointment, playing with your kid at home some thousands of miles away while travelling, or keeping records of events for future reviews. These new approaches are more efficient, and potentially more productive, thus, increase customer satisfaction with reduced cost. Applications are developed or under development to support distributed collaboration to achieve this kind of presence in every field.

1.2. Augmented Reality

AR is also called as Mixed Reality. Unlike Virtual Reality, it enhances the visualization of the real world by adding virtual objects to the real world. These virtual objects like images, video, audio, text and more are system generated on real objects or wherever the user wants to place. One can observe some floating windows, virtual person on couch, a virtual football goal post in living room, etc., from their view when wearing a head mounted device or using some of the latest smartphone or tablets [10]. Instant information results can be displayed as it often can recognize the environment using such devices. AR can transform the present communication system between people remotely and with real time updates.
1.3. Potential Applications in Different Fields

Smart Vehicles: AR is now ready to strike in the cars as virtual application viewing tool like navigation, speed, gas level and all what we can see now in meters.

Shopping: Products in catalogues can be viewed online and can be known how exactly it looks like whether it is a dress or an object in your room or any spare part. Real time manuals can also be included for easy-to-access user experience.

Education: Viewing orientation of any complicated physical object can be made easier by being able to see through the internal parts and illustrate its working dynamics and from different angles and details.

Medicine: Doctors can constantly stay in touch with the trainees, nurses and/or patients, as well as performing remote consultation, surgery assistance and fresher training.

Industry: Help mitigate the subject expertise shortage and productivity issues in, e.g., maintenance, repair, manufacturing, or transportation, by making subject expert easily accessible to technicians in the fields.

Gaming: One of the leading driving forces behind AR right now is the rapid emergence of AR-based video games which have gain tremendous popularity among the players. Game environment changes with the user's actual environment upon the interaction with the present real objects could substantially enhance the experience and engagement. For example, virtual world and characters can be built into real environments through Environmental FX.
Architecture: As both design and communication tool, it provides more efficient workflow for the developer with instant and intuitive 3D visual effect and the ability to show the same to their customers.

*Figure 1. An expert helping customer remotely*

In short, these technologies will show great potential to transform every aspect of our daily life if implemented wisely.
2. MICROSOFT HOLOLENS

Holographic headsets belonging to augmented technology may potentially become the next major computing platform after PC, laptop and smartphone/tablet. With distributed apps that wirelessly connected to the cloud infrastructure, it shows an extremely promising future in fields related to gaming, designing, architecture, education, medicine and so on. One of the pioneering AR devices is Microsoft HoloLens, which is a complete computer by itself. With many of the AR capabilities to support meaningful application, it stands out as the most capable AR platform to date.

2.1. What is HoloLens?

Simply put, HoloLens is a head-mounted wearable computer. It has an Operating System called Windows Mixed Reality (formerly Windows Holographic) which is a mixed reality platform introduced as part of the Microsoft Universal Windows Platform (UWP). This accommodative design choice makes any app that is developed for UWP be able to run properly in HoloLens. Users are provided with enhanced perception of the actual physical environment with extra holographic effects that can be seen and heard while wearing a HoloLens. Unlike experiencing a totally different virtual world like VR, HoloLens can strive to integrate the virtual and actual world seamlessly together by gaining sufficient understanding of the environment, and enabling intuitive association between virtual and real objects, and supporting human-friendly interactions.
2.2. Specifications

HoloLens has CPU and GPU like a laptop. It also has a special kind of processing unit called Holographic Processing Unit (HPU 1.0) with Intel 32-bit architecture which it needs to handle smooth streaming of terabytes of data which is achieved by silicon technology, sensor technology and optical engineering by reducing the power of processing to a great extent. This highly scaled silicon tech makes the device work on continuous streaming of Terabytes of data without freezing. To support any UWP applications, HoloLens run Windows Mixed Reality, a part of Windows 10 Operating System with human understanding features like voice, gaze and hand gestures as different kinds of input methods [5]. It supports the Visual Studio and Unity 3D platform for app development. It has 2GB RAM, 64GB storage. Wireless networks connections are available with Wi-Fi 802.11ac, Bluetooth 4.1 LE, Micro USB 2.0 [2].
To present 3D audiovisual effects, Microsoft HoloLens 1.0 has two see-through holographic lenses with 16:9 screen aspect ratio light engines. Holographic density achieves greater than 2.5k points per radian for high definition visual fidelity. It makes use of various sensors and artificial intelligence engines to understand the surrounding just like a human, e.g., one Inertial Measurement Unit with accelerometer, gyroscope and magnetometer, four environment understanding cameras, a mixed reality depth camera with 120*120-degree field of view, four mics, an ambient light sensor help in understanding. Cameras can capture pictures with 2MP and HD videos with a 2.4MP photographic camera. It also has 3D audio speakers with spatial sound that simulates the perceived location and distance of sounds to support fully immersive 3D audiovisual effects.

2.2.1. Cameras and Sensors:

HoloLens has four visible light cameras, a 2.4 MP video recording camera and sensors too like accelerometers and gyroscopes working together for environment understanding. These sensors detect the geometry of surroundings of the user. There is a depth camera with light sources capable of spatial mapping (SLAM) and sematic understanding. There is a laser module at the bottom which recognizes the gestures by tracking hand movements and event triggers. Inertia Measurement Unit (IMU) is used to center the graphical display using head tracking. Photons jouncing within the device, enter the lens and create the digital images. They are not RGB pixels, instead the displays in HoloLens are a color sequential display, which flash color channels of red-green-blue-green at 60Hz (individual color fields are shown at 240Hz). This is achieved through two small projectors at the bridge of the nose projects distributed light rays to waveguide lens
in between and makes the image formation at the back of our eye that appeared as a real object in our field of views. The holograms can be placed as if they are located at a particular location in the real-world using anchors, and the view of the user will change accordingly while he/she look or moves around [7]. Audio element can be pinned to a specific object to enable sound as well. It has two small speakers on headset right above the ears. Real world sounds were also audible. Finally, the user’s perspective can also be shared by others by broadcasting or recording the video through HoloLens.

2.2.2. Voice Commands:

HoloLens has mics to capture and/or record the sound/voice from environment. Through automatic speech recognition, voice command can be used to interact with applications and holograms, just like hand gestures. We can customize the voice commands based on the application’s specific needs [5].

2.2.3. Hand Gestures:

Gestures are the input methods like mouse clicks which work in concurrence with the gaze of the user. An object can be targeted with the help of gaze followed by any selection with gesture. In default, bloom for home page, Air tap to select, tap and hold to move, rotate, zoom in and out. However, it should be noted HoloLens itself doesn’t have a fully-fledged hand motion tracking capability for more comprehensive gesture based human interactions. There are successful efforts to augment HoloLens with hand tacking devices such as Leap motion controller for enhanced multi-modal interaction capabilities [5].
2.2.4. Spatial Mapping:

Built-in cameras scan the real-world surroundings continuously to create a virtual layout in which real and virtual items can co-exist and associate with each other naturally. Virtual objects take room in the real world, thus surfaces identified are used by the app to place holograms accordingly. This mapping is called Spatial Mapping. Virtual mapped surface is constructed on basis of continuous reflecting sensor data from real surfaces as shown in Figure 2.3. It refreshes with a rate of 5 fps. However, limitations exist with the current HoloLens spatial mapping capabilities. As shown in Figure 2.3, the mapping of two different objects with views that overlaps with other side to side without semantics could result in continuous map, and only reveals the geometry of the room, not individually identify objects. Moreover, transparent and reflective surfaces could impose a challenge and give false/missed identification in the virtual map.
2.3. Limitations

There are several noticeable limitations of HoloLens to support distributed collaborations. It starts with its limited Human-HoloLens interaction capabilities in primitive gesture and voice support. It cannot track bodies and objects by itself. The field of view impose additional limitation on how we can capture and share our close-range body activities to others. For example, when an expert surgeon to demonstrate an operation to trainees from his/her specific angle of views, the object may go beyond of the field of view and get chopped off when we try to get closer look.

2.4. HoloLens and Kinect Collaboration

HoloLens itself has no full-body capture and tracking technology, but it may take different kinds of data like body, hand and face from Kinect or Leap Motion Sensor. This thesis deals with the Holoportation Technology. Holoportation enables a person to be present remotely and in more than one place virtually at the same time. We believe continuous body tracking sensor like Kinect could be used to capture data and streams it to
distributed HoloLens devices running the holoportation application. As it refreshes at a high rate with the real-time data, it makes the remote virtual body move as if the person moves locally. This technology could take current communication services to another level.
3. MICROSOFT KINECT

Kinect is originally designed as an input device for pairing with Xbox gaming console by Microsoft. It enables the user to interact through an interface with the game console abolishing the game controllers, instead just by body movements and voice commands. We believe we can exploit the potential of Kinect for human body identification and tracking and with its depth sensor to help segment the body object in the video, such that we can construct an ultra-portable Holoportation prototype system in combination holographic modeling.

3.1. What is Kinect?

When plugged into a PC through a USB 3.0 port, Kinect can fetch raw data from its color and depth camera sensors with decent field of view. Then through its machine learning engine, it produces full-body tracking of 25 human body joints all under a latency of just 60 milliseconds. The newer version Kinect V2 made substantial improvement in detecting much smaller objects than the older Kinect V1. It has features like creating 3D images and can provide video surveillance.

Through its SDK, it also makes available all its raw data such as the color and depth images that can be exploited for different usages. In this thesis, we will use both the raw data and the results produced by its machine learning engine [11].
3.2. Kinect SDK

Microsoft released an SDK for Windows to allow application developments. We worked with the 2.0 version for Windows 10 OS to support Kinect V2 device and developed application scripts in C# in Visual Studio 2017 which also has debugger and app builder.

3.3. Cameras and Data Streams

3.3.1. IR Projector and Depth Camera:

Kinect has infrared projector which emits infrared light and scanning the space and the reflected signal is then read back by the Infrared Camera which is used to calculate the time of flight, i.e., how long it is taking to bounce back from an object, hence determining the corresponding depth with respect to the Kinect sensor. Kinect projects a specific pattern of dots that uses to optimize its measures [13].
If the pattern hits a flat surface, pattern remains the same even after receiving. If it is on curved surface, that pattern turns distorted, by analyzing the distortion Kinect can recognize which objects are closer or farther away as the depth.

Dot patterns reflected from the environment are read by the depth camera [11]. Unlike a webcam camera is composed of pixel defining RGB colors, here the depth camera uses pixels defining how far is it from the sensor and constructs a three-dimensional map of the surroundings. Such that, it becomes easier for the developer to separate object of interest with the background by simply removing the pixels farther from sensor than certain distance. IR depth data has spatial-temporal resolution of 512 * 424 at 30 fps and can track from 0.5 meters to 8 meters and provide a field of view of 70.6 * 60 degrees [12].

3.3.2. RGB Camera:

RGB streaming data acts like taking a two-dimensional color video. Each pixel has combination of Red, Green, Blue component values. Full HD spatial-temporal resolution of 1920 * 1080 at 30fps is provided with a field of view of 62 * 48.6 degrees. Color stream
is a sequence of image frames delivering at a rate of 12 to 31 fps depending on the resolution.

As we can see the difference in the resolutions, (RGB image has higher resolution than the depth image) it is important to correctly align the depth data with color data. Fortunately, the Coordinate Mapper function in SDK helps to align the respective depth and color points giving us decent 3D visualization.

3.3.3 Skeletal Data:

To track a person in the view is made easy by using Kinect because it has a pre-defined graphical model like skeleton in Kinect library. Human body joints are in Kinect class joint hierarchy with 25 elements [9]. These 25 joints are recognized on the object whenever Kinect detects a human body. Every joint has its orientation element along with the position in the space. It can detect and track up to six human bodies simultaneously. This is how multiple players can control the game through their individual body motion when playing Xbox game with Kinect together. In addition to joint tracking, Kinect can also recognize facial expressions to support features, such as face ID.

3.3.4. Construct 3D holographic video:

In this project, we plan to use Kinect and its sensors to help track continuous movements of a person in communication. These background pixels in the color video can be removed based on the depth data at the sender’s end and are then streamed to the receiver’s end which will be further processed to construct 3D holographic video of the person. Kinect also provides a great compatibility to UWP platform and HoloLens which help reduce our prototype development efforts substantially.
4. HOLOPORTATION

Human – Computer interaction is improving day by day through the continuous progress in implicit understanding between user and machine. Machines were built to respond to human actions through, e.g., gaze tracking, vocal instructions, face expression, finger movement, body posture and other natural measures using photosensors, eye tracker, infrared illuminators, heart rate sensors and many others. Natural collaboration has grown over years in distributed system designs due to the increase in network bandwidth, computing power, sensory capabilities and AR-based audiovisual presentation technologies. People can now share their experiences with others remotely better than ever. All these three features converging together can be used toward truly immersive Remote Collaborations by implementing Empathic Computing as shown in Figure 4.1. This basically makes a person to understand what another person can see, feel and hear. It can create and enhance the emotional experience and empathic connection during remote collaborations. The remote empathic computing experience can be improved by incorporating this idea with VR and AR [6].
4.1. What is Holoportation?

Consider a speaker speaks to the audiences from a remote location but appear as if he/she is doing that face to face right in the same room [10]. A fully immersive telepresence could improve the interaction and sharing compared to today’s video conference. Family and friends can also get together every now and then without even travelling distance.

To achieve this immersive telepresence, Microsoft has introduced a new feature called holoportation where a 3D virtual person appears in real life, just like in animated movies. That hologram appears in full size, in all directions if we move around it. This hologram is captured using a cave camera system to get the precise 3D depth and color data. Although facial expression could be a challenge if users wear a head mounted device, it could become more miniaturized in the future.
4.1.1. Cave Camera Set-Up:

In Microsoft Holoportation project, a unique advanced capturing system is used to develop a precise 3D model which almost looks like a person. A series of depth reading cameras all around a space at different heights and angles are placed as shown in Figure 4.2, tracking movements, gestures, physical appearance of a person continuously. This is called a Cave Camera Set-Up. The streaming data captured by every depth camera are blended together by tools like Room Alive Kit resulting a well-built 3D model.

![Cave Camera Set-Up](image)

*Figure 4.2. Cave Camera Set-Up [1]*

It can assemble all the data gathered together and can create a high-quality 3D model. This 3D model is then projected by headset like HoloLens onto its lens. Continuous movement streaming is done by the sharing services available. The data is compressed to increase the resolution of the motion capture at the client end. User can scale the hologram as per their convenience. We can observe from the picture in Figure 4.3 that the hologram appears realistic.
3D video from the wearer perspective can be recorded along with the audio for future review purposes. Even recorded holograms can be individually scaled as shown in Figure 4.4 (left) due to the use of holograms. One can also inspect the scene again as a second person while the video runs virtually in front of you as shown in Figure 4.4 (right).

Figure 4.4. Manipulatable recorded holographic videos [1]

4.2. Mobile Holoportation

This is an advanced and challenging version of the regular holoportation technology by Microsoft. The motion capturing setup is fitted into a car for demo purpose [1]. This
mobile setup needs two depth cameras for 3D capture of the person in the back seat, 30 to 50 Mbps Wi-Fi or cellular data. More the cameras used, more clarity would be in the model. 97% of the bandwidth which is used regularly is compressed to 30 to 50 Mbps to achieve mobility. Background removal, light sources were other challenges. Client can view the person in car with a head mounted device as usual. Figure 4.5 shows the person in car waving and the virtual hologram of him sit in a chair at the client side.

![Figure 4.5. Person in Car (Server End) [1] (left) and Holoported person (Client End) [1] (right)](image)

However, the system still requires high computation power hardware to run smoothly.

### 4.3. Limitations

Multiple cameras were used in this system to get a sophisticated quality model and significant computing resource are required, which limit its usage in a truly mobile environment. We intend to look at the feasibility and performance of true mobile-friendly solution which will implement the same using a single Kinect with laptop at each end.
5. IMPLEMENTATION OF KINECT-BASED HOLOPORTATION PROTOTYPE SYSTEM

A system with Kinect connected at the server end, should make it possible for the clients to view the same scene what server is able to. This evolution in the communication field can be applied to any Augmented Reality applications. We designed this project to achieve the set-up like illustrated in Figure 5.1 architecture [15].

![Prototype Architecture](image)

*Figure 5.1. Prototype Architecture [15]*

5.1. Unity 3D Engine

We implemented this project using Unity 3D which is a platform to build 3D games. This Unity editor is used to develop games and applications with Mono Develop or Visual Studio 2017 as script editors. HoloLens version 2018.2.0f2 is installed. It comes with holographic emulator that run UWP apps on PC locally before deploying into HoloLens.
device for testing purpose. It should be noted that Hyper-V should be enabled in the Windows 10 settings for program features to support emulator feature.

We then import the HoloToolKit (for app developments related to HoloLens), Kinect View[16] (for app developments related to Kinect) and other custom tools, such as materials and shaders, and required assets for a new 3D project. Meta files are created for every asset folder, which has all the information on how things work together with that particular asset. These contain all the scripts, prefabs, components, materials used in the app development.

Next, we create the basic interactive game objects for the scene such as camera and directional light, followed by Source View, Data Sender and Sharing. We also have to set their appropriate positions in XYZ spatial coordinate system so that camera can see the Source View object in HoloLens.

Scripts are needed to make the game objects interact with each other and respond to the input methods given by the player. The primary scripting APIs offered by Unity are in C#. Using scripts, you can set conditions for a particular event to occur. They can also control the graphical effects. Unity 3D also provides supports for C# debugging.

Finally, we will need Visual Studio to complete the application development by make the build for a target UWP device. In our project, we use VS 2017. Tools and extensions like Unity support and .NET backend must be enabled in VS 2017. File name should be the same as the class name. We can now complete the build that is ready for deployment.
5.2. HoloToolKit

HoloToolKit is a Microsoft APIs for application development with holographic displays. Camera and publish settings must be changed in Unity project to support this import. Clear flags must be set to solid color with a black background for HoloLens visibility. Spatial Mapping is enabled to map real world into virtual layout surface. UWP is selected for HoloLens as the target device. We also choose the following settings for supporting HoloToolKit:

- choose D3D type, so that it will build the application in three-dimension;
- enable ‘support to virtual reality’;
- select the C# scripts in solution;
- change the scripting backend to .NET.

To keep holograms associated with a fixed 3D location, HoloToolKit provides world anchors which will override the original transform components of a game object that determines the Position, Rotation, and Scale of each object in the scene as shown in Figure 5.2. It is attached to a game object which we want to fix it to a place in real world like a movie window to a wall, so it won’t move around. However, objects with world anchor can’t contain rigid body components with Dynamic physics [7].

![Figure 5.2. Transform components of a game object](image)

|
|---|---|---|
|Position| X 0 | Y -126 | Z 231 |
|Rotation| X -3.394 | Y -4.234 | Z -4.347 |
|Scale| X 4 | Y 4 | Z 2 |
5.3. HoloToolKit Sharing Service

Multiple clients can receive and share data that is compatible with HoloLens but restricted for certain actions. HoloToolKit provides a sharing service to connect to the remote client systems. It is basically a server executable that opens a sharing session with its local IP address.

Sharing service must be launched at the server end first and then we can run the Unity server scene. It creates a session to which any number of clients can be connected. This service should be given access to pass through the Firewall. Running a client scene should make it automatically connect to the session launched by the server. Clients configured with the server address and port number connects to the server and receives the streaming data in order to reconstruct a 3D hologram on the client side. Sharing has the ability to synchronize data across any client that is connected to current running session [8]. The remote client applications built for UWP using HoloToolKit sharing service can be deployed on HoloLens just like apps on our phones and PCs.

![Image of Sharing Service output]

*Figure 5.3. Two devices connected on HoloToolKit Sharing Service*
5.4. Components of Hologram

Holograms are constructed using mesh and texture. Mesh contains the structure of vertices and multiple triangle arrays for the object, while the texture provides the fine detail of the object surfaces.

5.4.1. Mesh:

Filter and renderer are needed to construct a mesh. Filter stores the mesh shape, while renderer takes the information from filter and adds customized game objects or materials to it. A mesh class has components like vertices, triangles, color, normals and tangents. Most fundamental ones are vertices; stored in vertices array of 3D vector of type Vector3. Triangles arranged in three-dimensional space gives a smoother solid texture to the object; triangles array is an integer array that stores the indices of the vertices within the vertices array, with which the triangle made up of, ordered clockwise. They were always taken as a group of three while arranging. Culling removes the internal parts which we cannot see for the sake of computation performance. Recalculating function of mesh makes the mesh react to the lighting effects automatically by Unity. Normal (vector perpendicular to mesh surface) is provided to each vertex to get balanced lightening. Normal and light direction angles are processed to get the reasonable shading effect [14].

5.4.2. Texture:

For every mesh triangle there exists a triangular texture printed image pinned to mesh, if we are working with 3D color objects. For this to happen, every vertex is accompanied by its corresponding unique 2D image UV coordinates [14]. UV is an array mapping the depth and color triangular regions.
5.5. Server Design

In Unity, we create a Data Manager which collects the depth data, color data, color texture directly from the Kinect sensor from Multi Source Frame Reader function.

Mesh is created according to the depth data. Vertices are created based on which triangles were built to give smoother texture to mesh. Normals are recalculated for shadow effect. Refresh has Coordinate mapper mapping the depth data and color space point (UV) before texture is applied. They must update for every frame it received to get a streaming 3D effect.

Kinect Color frame description and Depth frame description are used to describe the resolutions. Color Frame has a resolution of 1920 * 1080 and depth frame at 512 * 424. Color Texture is applied in the update function to refresh the texture for every frame captured. The respective color data of a depth data is an image segmented out from its background, overlaid on its depth mesh to construct as a 3D hologram. As the mesh changes every frame, the registered image also changes as per the depth showing a holographic video.

Thus, the data to be streamed must include the arrays of Depth, Color and Color scape point (to map depth and color). These are enough for a client app to reconstruct the holograms. Unity registers the holograms, makes slight changes for every frame received by aligning the depth pixels with the color pixels. The segmented-out color image related to corresponding to the depth image is then diffused to the material of the hologram, i.e., it applies the material texture on top of the depth game object (i.e., mesh created from the depth data).
We used sharing stage, auto join session and custom messages scripts from HoloToolKit for sharing the data across the network. Three tags are created to be sent along with the message (the respective arrays) so that it will be easier for client machine to parse the data received. Three functions are implemented to send these three types of data.

In our experiment, we observed that the client end may experience a delay of 30 seconds at first. This is caused by transmitting data at the default frame rate of 60 frame per second which turn out to be overwhelming to our computers and networks. An adjustment is made to limit the transmitting rate at 1 frame per second. This help reduced the delay under 3 seconds. With more capable computers and network connections, we believe it will support higher frame rate and lower delay. Figure 5.5 illustrated the constructed hologram at the server end.
5.6. Client Design

The received message is stored in NetworkInMessage function of custom messages class. Empty mesh is created with the known resolution of depth and color data. The tag accompanied with data specifies to which array it belongs. Depth and color are mapped with respect to the color space point every time along with the calculated texture. Mesh looks like the Figure 5.6. The texture is diffused to the material of the mesh game object as in server and the final model looks like the Figure 5.7.

Figure 5.5. Colored 3D hologram at the Server End
Figure 5.6. Mesh created from the Depth data on Server (Right) and Client (Left) connected by HoloToolKit Sharing Service

Figure 5.7. Colored 3D Hologram at the Client End

5.7. Holographic Emulation

Holographic emulation is one more tool to see the 3D hologram as if in HoloLens. This is used for app testing in the process of development providing alternative tools for
debugging and displaying game scene on head mounted device without the actual deploying. It can be directly done within the Unity 3D editor once HoloToolKit is installed. Debugging and running app on the PC can then show the game in editor and Holographic emulation as prototype. Makes a developer’s job easier by saving time to make any changes before finalizing and deploying the application into holographic device.

Holographic Emulation also support a remote mode. When a HoloLens device is available locally or remotely, we can use the Holographic Remoting to connect to it and show the exact same scene as on the Unity editor running one a PC. The remote machine’s IP address is used to connect with HoloLens without build. This control window should be active throughout the running time. Overall operation seems as if it is deployed while spatial and head tracking are still active. Once started, Holographic remoting player on HoloLens shows IP address of itself while waiting for a PC to connect. Figure 5.8 illustrated the connection using Holographic Remoting.

![Image](image_url)

*Figure 5.8. Use Holographic Remoting to debug Hologram view before Deploying as a HoloLens App*
5.8. Building an UWP Application

5.8.1. Player Settings in Unity:

Configure the project with mixed reality settings. Apply project settings like XR, Direct3D, Windows UWP, sharing services, .NET backend scripting and spatial mapping layer. Virtual reality support should be added. Reality scene can be set by enabling the input manager prefab, default cursor prefab, mixed reality camera prefab instead of a normal camera game object, updating world space canvases. Universal Window Platform Capability is set to enable the services to work on HoloLens client like webcam, microphone, spatial perception, internet client server protocol, launching private network client server. Enabling network services turns this system to local holoportation prototype.

Right before building player settings are to be changed with the following configurations. Scripting runtime version to .NET 3.5 Equivalent, scripting backend to .NET, API compatibility level to .NET 4.x, accelerometer frequency to 60 Hz. Permit virtual reality in XR settings and check box the enable depth buffer sharing which ensures plane-based reprojection. Any extra capabilities can be added to the project in publishing settings. UWP recommend using IL2CPP package instead of .NET because later releases of Unity may not support .NET any more. After the settings and capability selections were done, required platform can be chosen to build an application in developer’s choice. We selected UWP to deploy into HoloLens [3].

5.8.2. Build Settings in VS2017:

A scene in unity 3D can be deployed into HoloLens by first building it in unity and debugging in Visual Studio 2017. UWP development, .NET desktop development, Game development with Unity extensions are added to VS2017 to make the application deploy
HoloLens. A sln format file is created after built successfully, run in VS2017. Set up the following: Target - Release, arm - x86, run on - remote machine.

5.8.3. Deploy:

Deploy and run the generated solution into Remote Machine through an IP address. If app is connected to the server, the same person appears to be in front of us. An app can be played on multiple client devices through sharing service [8] associated with the developing kit, thus helps remote people engaging with the server end person. These builds usually, compresses, adjust the frame rates to deal with delays (in server), sends a person to a remote place as a 3D model ready to interact regulating the network bandwidth.

5.9. Implementation Results

The application in the HoloLens can be run just like any mobile application by opening and parallelly running server scene in another PC would connect the HoloLens to it. It plays the client scene in the HoloLens, displays a hologram in the real time just like the shown in Figure 5.9 This shows our achievement in providing a fair quality of front view of the holograms in a single Kinect based Holoportation prototype system.

Figure 5.9. Hologram in client HoloLens
5.10. Rear End of Hologram

Due to the use of a single Kinect, front view of the hologram is captured and completed but leaving the back-end hollow due to lack of view. This doesn’t make the hologram complete if one examines the virtual person in HoloLens while roaming around. Hence, we came up with the idea to use the joint tracking data produced by Kinect to complete it with a customized graphical model that is based on the collected information about the person. Kinect detecting a human, can automatically generate a joint system which looks like the Figure 5.10 (used line renderers to look like a stick man structure) [9].

Aligning scripts are attached to the game objects which are meant to be aligned together. Transform position is adjusted to form a complete 3D figure by aligning the stick man [16] exactly with the partial body hologram.
The quaternion data of the avatar joints are aligned with the Kinect body joints [17]. The transforms are joined for both the depth and joint data views to acquire complete body with single camera system. In Figure 5.12, we can observe that the 3D model (on back) is aligned with the avatar (in front). If turns around the hologram they can view the back side when avatar is placed at back.
5.10.1. Scanned 3D Model for Rear End:

A customized avatar model can be created with the Microsoft tools available like 3D Builder and 3D scan instead of using imported avatar assets. The following model should be segmented out according to Kinect joint hierarchy into FBX format with Blender app. This scanned model can be imposed on to the stick man. The resultant model is used to cover the hollow rear end of the current working model.

![3D scanned model of a person using 3D scan app (all view)](image)

*Figure 5.13. 3D scanned model of a person using 3D scan app (all view)*

Figure 5.13 shows all angles of the 3D scanned model of a person. One can scan a person’s body with Kinect from all sides to get a precise model. This model can be broken with respect to the Kinect Joint Hierarchy. This complete body instead of a stickman on the back gives an impression of full 3D hologram.
6. CONCLUSION AND FUTURE WORK

A prototype single Kinect based Holoportation system is implemented instead of requiring a complicated and expensive cave camera setup. Kinect did a pretty good job by collecting decent amount of the object details. The reconstructed body at the client side is obtained with the similar quality as of server. We designed a portable and a steady model with Unity 3D application deployed into HoloLens. This system is experiencing significant delay while receiving the streamed data. This can be reduced by decreasing frame rate of the transmitted frames, as well as improved computer and networking capabilities. One limitation of the system is the back part of the present model is not complete as it cannot be seen by the Kinect. But it may not be a concern for most distributed interactions between people as we mostly engage in face to face interaction.

When the back view of the object is of interest, we have proposed to apply and align a 3D graphic model of a person, customized according to the joint data, with the depth data-based mesh model to complete the back end of the hologram to make it complete. Unlike using a cave system, a complete model can be gained with the help of only one camera with limited loss of fidelity. In the future, human internal parts can be included in the 3D human model for augmented medical demonstrations as well.
REFERENCES

[1] Microsoft, Holoportation, 2018;

http://www.imaginativeuniversal.com/blog/2015/01/26/top-21-hololens-ideas/

https://hololens.reality.news/how-to/hololens-dev-101-build-basic-hololens-app-minutes-0175021/

https://wwwpolygon.com/halloween/2014/10/30/7079943/kinect-ghost-hunting


https://docs.unity3d.com/560/Documentation/Manual/windowsholographic-anchors.html

[8] Microsoft/MixedReality213;
https://github.com/Microsoft/MixedReality213/tree/master/Assets/HoloToolkit/Sharing

https://blogs.windows.com/buildingapps/2017/04/18/building-telepresence-app-hololens-kinect/

https://program-ace.com/blog/top-6-uses-of-holographic-teleportation/

https://shiffman.net/p5/kinect/

[12] Ronald Smeenk, Kinect V1 and Kinect V2 fields of view compared, March 11, 2014;
https://smeenk.com/kinect-field-of-view-comparison/

[13] Robert Cong, Ryan Winters, How does the Xbox Kinect work;
https://www.jameco.com/jameco/workshop/howitworks/xboxkinect.html
[14] Unity Documentation, Unity User Manual (2018.2);
https://docs.unity3d.com/Manual/AnatomyofaMesh.html


[16] James Ashley, Unity 5 and Kinect 2 Integration, The Imaginative Universal, March 27, 2015;
http://www.imaginativeuniversal.com/blog/2015/03/27/unity-5-and-kinect-2-integration/

[17] Jason Odom, Add Full Body Motion Tracking to the HoloLens with a Kinect, 2017;