Build Helper: Intuitive Interaction System for Assembly Tasks using Mixed Reality

XUE BINGYU
44181025-2

Master of Engineering

Supervisor: Prof. Jiro Tanaka
Graduate School of Information, Production and Systems
Waseda University

February 2019

Abstract

In modern days, "Assembling items" are tasks that people often encountered with. For mass production, assembly is indeed an essential criterion in factories for realizing ideas to creation; On another hand, for individual cases, Assembly is not a new word, whether it is for building some furniture, some toys, some electronics or even fixing some of your devices. There are multiple shreds of evidence showing the importance of assembly tasks in daily life. However, there are yet no perfect solutions in markets and previous researches.

This paper proposes an idea called "Build Helper" that we think might be one of the better solutions for daily assembly tasks. Different from any other existing works, the "Build Helper" tried to co-exist virtual contents with the real objects in a mixed reality environment, and use this virtual contents as more understandable and intuitive graphical instructions when compared with text instructions while maintaining functions such as instructions creation support, hands-free experience, intuitive instructions and good interactions at the same time. With the "Build Helper", we can perform something such as automatically generate the instructions based on components' data, show intuitive 3D instructions with virtual contents when the user operates the work hands-free and active troubleshooting powered by our recognition techniques.

In this research, we have invited some participants to test the usability and efficiency of our system. We got positive feedback through the preliminary user study.

Keywords: mixed reality, assembly instructions, interactive

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Jiro Tanaka, for the continued support of my study and research. He is an important person who leads me into the HCI field and helps me out with my research throughout all these years. During my times of study at Waseda University, he uses his knowledge, patience, motivation and enthusiasm guide me a lot. He makes me understand the importance of reading papers from conferences and absorb innovations from other researchers. He mentors me on how to express my research perspective in seminars and presentations. He leads me the entire process of developing idea pieces into a research topic. And he also gave me countless pieces of advice on how to improve my research and write this master thesis. It is my great honour to be a student of Prof. Jiro Tanaka and made something happened in the IPLAB.

Secondly, I would like to thank all the members of the IPLAB, especially my senior and coach Mr Dou Hao from helping me gather all my thoughts into ideas and my lab-mate Mr Hommaru Katsuya for generously providing me with his devices and support for my research. I wish to thank all the members of the IPLAB for all the encouraging support. You guys are not only my best partners in my research but also my best friends in my life.

Finally, I want to give my gratitude to my parents. Thank you for giving me the financial support and mental support throughout all my life. I love you and you will always be my firmest support.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Introduction

Assembling items are tasks that people often encounter in modern days. These kinds of work are not that easy. Sometimes, people need better instructions to understand the manual and construct correctly. Therefore, we think a better way of assembling items are required. "Build Helper" is an intuitive interaction system for assembly tasks using the mixed reality that we think might be one of the better solutions for daily assembly tasks.

In this research, we tried to use a mixed reality. With the help of MR, merging of real and virtual worlds to produce new environments and visualizations where physical and digital objects co-exist and interact in real-time becomes possible [1]. In our system, we convert traditional text instructions into more understandable and intuitive virtual objects. A virtual object can be a virtual component indicates where the user needs in this step, a virtual arrow mark shows where and how a user should use the screw, or a 3D virtual model indicates what will be like when finished. Therefore, it is much easier to understand and compare with text instructions.

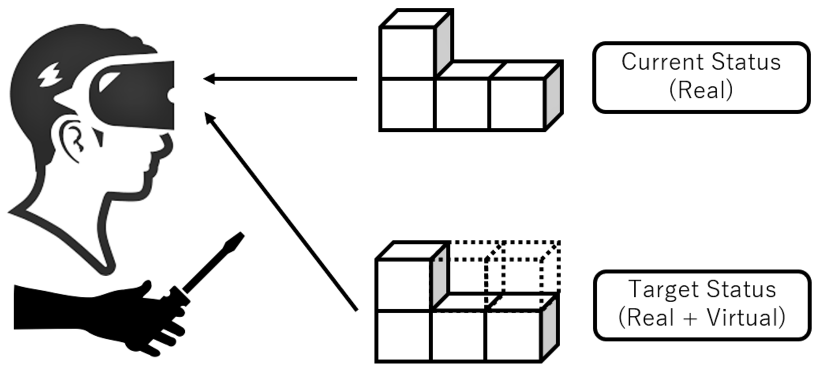


Fig. 1.1 MR and Assembly

Other than traditional manuals or instructions, our idea is to make a Build Helper which can provide interactions between the user and the system. Instead of just showing the steps, we think to be about to monitor the progress of the user and avoid mistakes is a very important point.

Different from any other solutions that exist, our proposal can achieve the following functions.

1. System can automatically generate the instructions;
2. System can provide intuitive instructions instead of texts;
3. System can interact with the user by object recognition and gesture recognition technique;
4. System can support a hands-free experience;
5. System can support some troubleshooting technique to prevent mistakes.

For the above reasons, the "Build Helper" can be considered as a relatively new and effective solution for daily life.

1.2 Organization of the thesis

The rest of the thesis is organized as follows: Chapter 2 introduces the background of the thesis and also analyzed how we figured out our standards and requirements. Chapter 3 is about the related works which show how we figured out the technology we want to use in our research. Chapter 4 will briefly talk about the research goal and also the approaches. Chapter 5 is the system design part, where the design concept and ideas will be introduced and the mechanism and algorithm design will also be told. Chapter 6 will be the system implementation part where the detailed environment and implementation will be talked. Chapter 7 will be about the evaluation, we will talk about the usability and efficiency of our system. The last part, Chapter 8, will be the conclusion and future work part, where we will conclude the previous content and talk about future possibilities.

Chapter 2

Background

2.1 Assembly in daily life

Assembling items or another term that more widely used in modern days, "DIY"s are tasks that people often encountered frequently. Whether it is for building a DIY PC, for fixing some of your electronic device or even more common work such as setting up some furniture. This kind of tasks is not that easy for average people. Sometimes, people need better instructions in order to construct correctly. Some articles claim that the current instructions can be confusing or hard to read [2]. Nearly 40 percent of people deal with an "imagination gap" when comes to the setup of a DIY furniture [3]. More and more facts are showing the importance of assembly tasks in daily life, but we yet not find out a perfect solution.

2.2 Criteria of better assembly method

When people talking about the "instruction" to assemble something. Usually, the first thing comes to mind is the traditional paper instructions.

In fact, there are too many drawbacks for the paper instruction method.

- The text or somethings a little bit figures are too hard to understand. Even if you have great ability to understand things, read through all the texts are time-consuming and things could be even worse if multiple paper instructions are required to read (e.g. DIY PC);

- Sometimes, user need to constantly check the paper instruction to make sure everything is on the right track. Inevitably, one of the user's hand will be occupied by this paper instruction. Of course, using only one hand is definitely harmful in terms of efficiency and user experience;
- The most important drawback for the paper instructions is it is only a "paper". Paper won't tall and point out the mistakes. For amateur users, be able to teach and correct them is surely helpful.

In summary, we recognize that the traditional method for instruct assembling items is not intuitive, not hands-free and definitely no helpful.

Chapter 3

Related Work

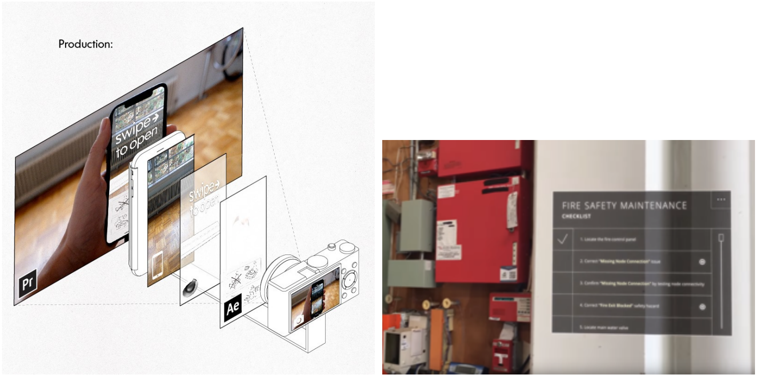
With the help of information technology, the human can mass produce products with incredible speed and quality, which is the famous "Third Industrial Revolution".

There are a lot of works that tried to combine these two together[4], whether for Virtualizing the workflow[5], optimising the work[6] or even collaborate among designers[7]. Indeed, information technology and assembling suit well in the last decades.

Augmented Reality is relatively new in this decade. The main idea for AR technology is to provide more intuitive context information to users rather than traditional text or papers. In recent years, AR technology and its device are widely introduced into the smartphone industry. Therefore, AR technology is something that a regular person can access every day.

In recent studies, researchers tried to use AR technology into varies of field[8][9][10], such as enhancing travelling experiences[11] or make richer interaction on everyday surfaces[12].

There are also several examples of using AR technology to solving some assembling tasks practically.



(a) Assemble AR (b) AR ASSIST

Fig. 3.1 practical use cases

"Assemble AR" is a smartphone application that offers a better assembling experience with IKEA furniture. [13] By using AR technology, This work introduced an idea of digitizing a paper instruction into AR instruction. It using a smartphone to display the AR contents side by side with the real task, showing AR contents of picture or illustration, target user could receive a better understanding when compared with a traditional paper manual. However, this method is not a complete solution. Because the instruction is not intuitive enough and the instructions can only be "played" step by step, which is not good enough, due to the target user are amateur who cannot distinguish whether a step is been finished correctly. Furthermore, this application requires the user to hold a smartphone, which is not so efficient, because a lot of tasks require two hands to operate.

A company called eon reality also came up with an application called "AR ASSIST". [14] Which allows work to wear a Microsoft HoloLens during the work. HoloLens will show contents such as a checklist or labels by the holographic display. Target worker can perform actions based on the description of the holographic contents. However, this system relied on a group of professionals works in the control centre, and the control centre gives instructions or orders manually, which is not that suitable for our use case.

Chapter 4

Research Goal and Approach

4.1 Goal

The fundamental goal of this research is to build **an intuitive interaction system for assembly tasks**.

A lot of previous approaches showed in the last section are indeed great improvement when compare with the simple paper instruction method. However, these approaches are still far away by both usefulness and intuitiveness.

After analyzing why assembly tasks are not that easy, certain criteria for a better assembly method is summarized. Therefore, we think our goal is to accomplish a system that meets the following requirements.

1. Intuitive compare with plain text;

2. Hands-free;

3. Interactive with the user;

4. Troubleshooting

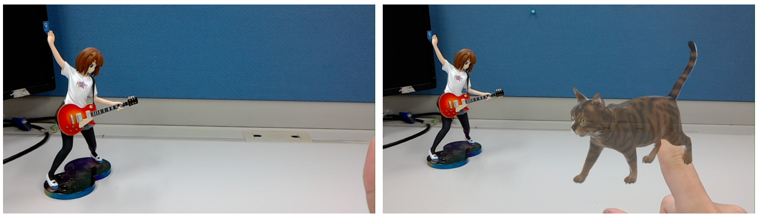
4.2 Use Case

Assembling task can be performed for producing the product in a factory by workers or DIY by amateur users. In this research, we will focus on helping those non-professional users to assemble the item in daily life, such as DIY PCs, build Legos or assemble furniture. Compare with professional workers, these amateur users are more rely on current paper instructions for assembling.

Different from the fact that we are targeting amateur users in this system, we do not want this system to have any specific target tasks. In fact, the framework of the system should be suitable for any kind of assembling tasks and not limited to any specific jobs. Thus, a "Universal Design" of the framework is required.

4.3 Approach

This research concentrated on using mixed reality HMDs to provide intuitive interaction for assembly tasks.



(a) without MR (b) with MR

Fig. 4.1 View of Mixed Reality

As you may notice the difference between figure 4.1 and figure 4.2. Mixed reality HMDs are able to enable the user the ability to see both real objects (the figure) and virtual objects (the cat) at the same time. Users are required to manipulate with the real objects and perform the task that intuitively displayed with virtual objects. Some of the virtual objects are 3D models that identical to a real component that user need and some of the other virtual objects are symbols or indicators which help the user to perform certain actions.

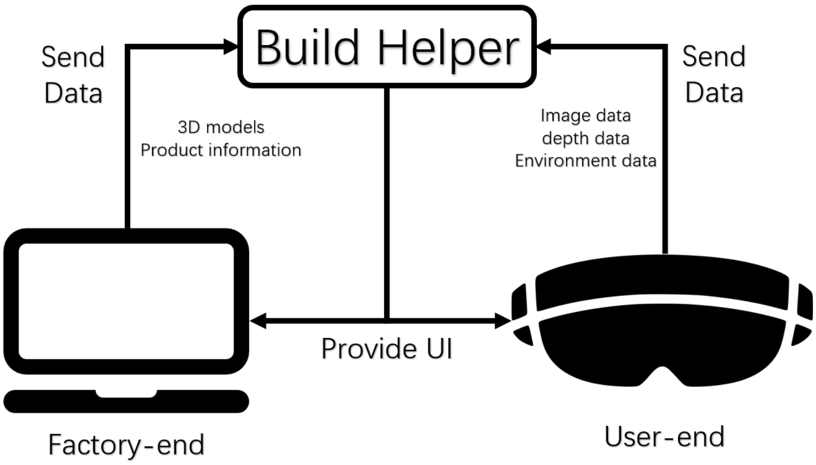
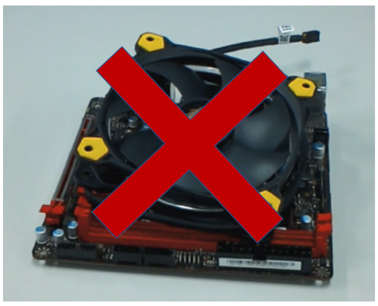
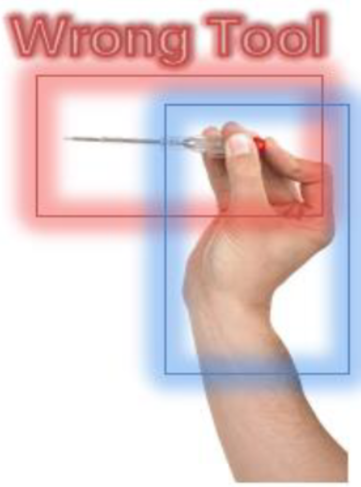


Fig. 4.2 Two End-users Structure

This system does require 3d models that pair with real components one by one. In order to solve this problem, we also introduced another user end that target makers or factories. By applying them simple user interface, they can upload all the component details digitally, which include a 3D model and detail information for all ports and sockets. Compare with traditional ways of doing the same job, which writes text instruction and draw picture instruction, we believe our simple user interface is consuming lesser time and money [15].



(a) Mistake Occurred



(b) Mistake on Proceeding

Fig. 4.3 Mistake Handling

A big challenge for the user when using the paper instruction is to maintain correctness. When some mistakes been made, it is very hard for amateur users to figure out where is the mistake. This system can monitor a user's movement and trigger some error message while a mistake has been made. Furthermore, by detect user's hand, the system can even avoid mistakes from happening by interrupting wrong actions.

In summary, we use several different technologies to meet our requirements. We use Holographic display to achieve more intuitive experience; head-mounted display to free our hands; hand gesture recognition to achieve some interactions and we achieved troubleshooting with the combination of object recognition and hand gesture recognition.

Our users are basically following the holographic instructions and perform the same action while maintaining hands-free. The user can perform hand action to interact with our system and our system also runs actively for monitoring if there is any mistake needs to correct.

4.4 Novelty

The novelty of this research mainly reflects in these aspects:

1. We proposed combining Mixed Reality, Object Recognition and Gesture Sensing together into an HMD unit for daily assembly tasks;

2. We proposed a "Universal Designed" framework in order to transform information of all components into a virtual assembly instruction;

3. Some prior works are using recognition and AR for accurate action. However, we proposed to focus on providing a more understandable and more user-friendly solution for amateur users.

Chapter 5

System Design

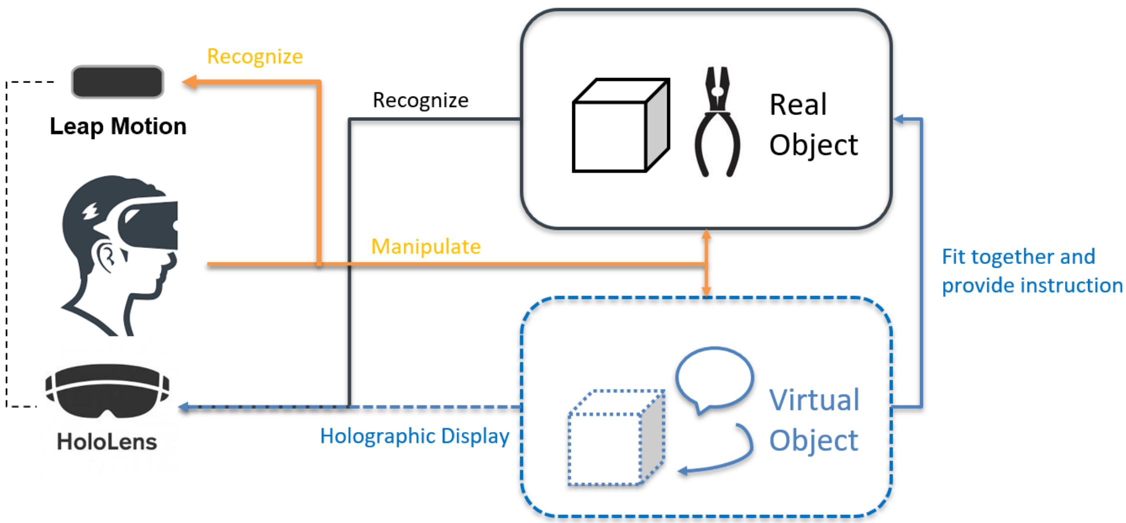


Fig. 5.1 System Design Overview

In this chapter, we will introduce our system design and each important pieces of our approach. Figure 5.1 shows the overall structure of our system.

The Mixed Reality HMD offers the user a combining view of the real object and virtual object. In our system, the real object can be considered as components or tools and the virtual objects may contain 3D models of component and indicators such as arrows or tool-tips. We fit these objects together and provide the instructions for assembly tasks. Our user may manipulate with both real objects for assembling or virtual objects for seeking additional information. The recognition device is used to detect and track real objects and human hands.

In order to achieve this figure, our system requires a lot of information to make a start. We made a factory end for accessing the information of the components we need. which includes the 3D model of the product, ID for all the slots and sockets within each component and also some other detail information.

With this information, our system can understand "what we have" and "what we are going to build". Then we used our original algorithm to analyze information of each component and produce a structured and visualized result. In this algorithm, we go through all the components. Based on their slot and socket ID, we tried to pair them up and create a link if they can be connected together. The system can visualize the assembly process because we have the 3D models of all the components and we have the coordinates of each socket.

After that, our system use objects recognition and detection to understand the real components and its coordinates related to the camera, augment these real components with holographic virtual objects and indicators to show how the user can perform the action in order to assemble it.

Following subsections are details information about key parts of this research.

5.1 Support two kinds of users

The main goal of this paper is to provide an intuitive system that helps the amateur user to assemble things. However, in order to accomplish this gold, we need help from the factory side.

In the traditional way of assembling things, such as an item of furniture from IKEA. The maker, IKEA design the paper instructions as well as the product. On other word, users understand how to build this furniture because they have the paper instruction, which contains a lot of assembling information from the maker. Therefore, our system also needs information from the maker side.

Compare with the traditional paper-based instruction, which need carefully designed text and picture contents, our system has to be simple and easy to use.

The system in this paper only needs maker to provide basic raw data. In modern days, makers are using machines and pipeline to manufacture their product. Which means they can output raw information such as dimensional data or serial IDs easily.

In our system, we only need the maker/factory side to upload this raw information as an end-user. After that, we user algorithm compute these data into instructions. That is why we think our system supports maker/factory and customer/user as our two end-users.

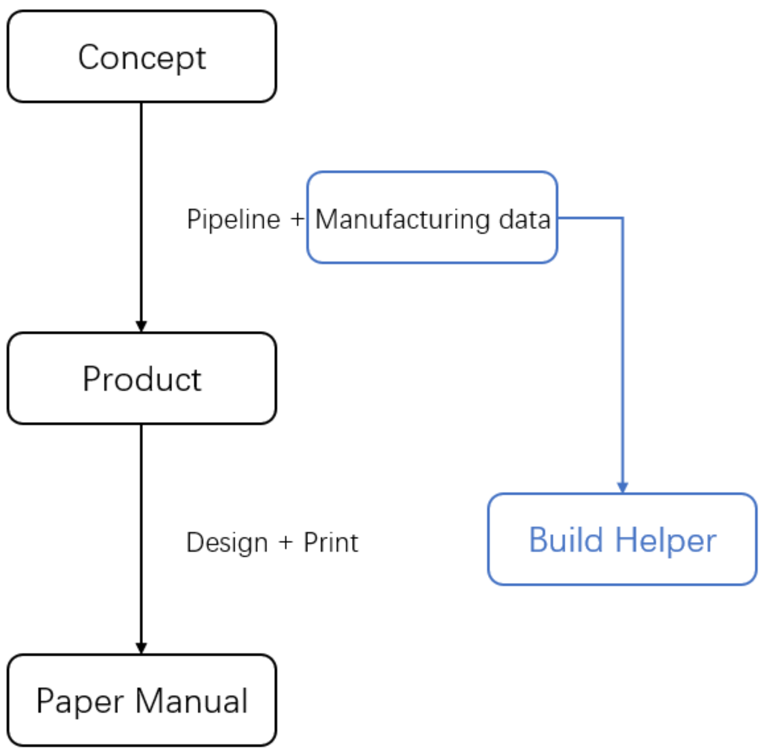


Fig. 5.2 Convenience of the factories user-end

As you can find out from the above figure. our system provides more convenience experience for the maker/factory side. In the traditional way, makers use pipeline and manufacturing data to make a product from concepts. However, they have to hire additional staffs to design the paper manually. Because it is usually paper and needs to print out, it might be costly. On other hands, our system can directly compute the manufacturing data into contents of the Build Helper.

The manufacturing data we need is fairly simple. The factory users who are using our system are requested to use our 2D user interface in order to upload the product information.

Firstly, they are asked to upload the 3D models (.fbx file) of their 3D product and the package of it, Then they can assign values such as "name", "description", etc. to the model. By drawing 2D rectangles which shows on following the figure, the user can assign the position value for a slot/socket and specify the function of it.

This side of our system will generate three files for each product. One is two 3d model data files (.fbx file), the last one is a sheet for storing all the details (.CSV file). Ideally, these files will be stored on the database of the Maker/Factory companies. The Consumer side using our system can access and download these files via internet connection.

The factory side user will provide the system with these information. After that, the system will use a certain algorithm to generate instructions automatically.

5.2 Automatic generation of instruction

In our system, we designed an algorithm that computes all the raw data from the maker/factory side, and transfer these data into intuitive instructions. If we are looking at the paper instruction. we can understand the paper instruction is a set of steps that need to perform in a particular sequence. on another word, the paper instruction is the combination of sequences and each single steps. Inside each step, users are asked to put two things together. A step is to perform a single action to certain components. A single action can be a break to two parts, which are "where", stands for "where should I insert" and "how", stands for "how can I do that (need tools or not)". In summary, what the system need to be generated is a sequence of step, that include needed components and how these component linked together.

To make it easier to understand the work here, we need to break it to details and analyze the problem first.

5.2.1 Analyzing the problem

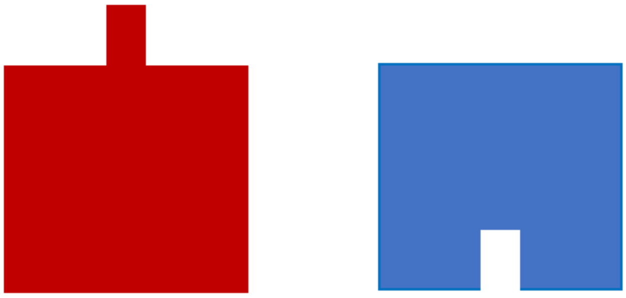


Fig. 5.3 Assembling case 1

Figure 5.3 is showing the most simple case of an assemble action. The user needs to assemble the red piece and the blue piece into one piece. The core problems of this case are "what are the pieces needed" and "how to put in together". The answers are"the red piece and the blue piece" and "put the blue piece at the top of the red piece where the socket can fit". However, the real situation will be much more complicated.

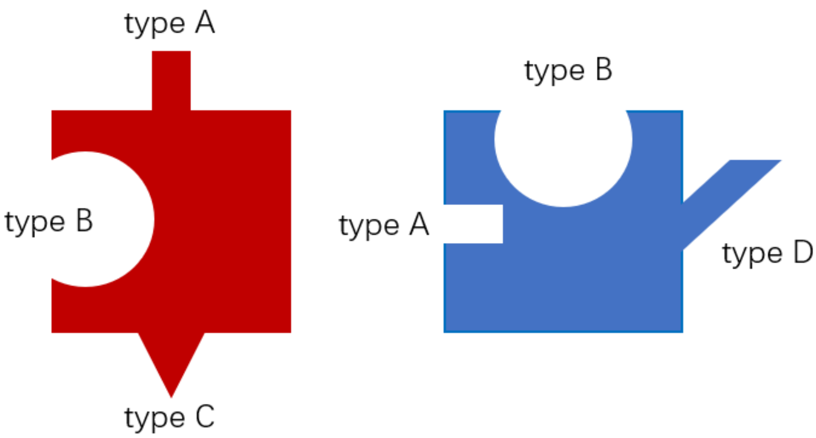


Fig. 5.4 Assembling case 2

Figure 5.4 is also showing two pieces that need to be assembled. But, this time the blue piece and the red piece have multiple sockets and plugs. In this case, we can understand how to fit these two pieces by the physical constraint of the plugs/sockets. However, the system cannot make this kind of judgement by just looking at it. Therefore, assigning IDs into these plugs/sockets is essential. The system can identify the plug and the socket will fit by matching them with the same ID.

For the section "Connection between two end-users", we understand that we have the serial ID information and the dimensional information of all the plugs/sockets in each component thanks to the help of our maker/factory side user. Nevertheless, we need some kind of form to store these data.

Color	Type	Plug or Socket	Where
Red	type A	Plug	top
Red	type B	Socket	left
Red	type C	Plug	bottom
Blue	type A	Socket	left
Blue	type B	Socket	top
Blue	type D	Plug	right

Table 5.1 Assembling case 2

Table 5.1 shows the data we need to solve the problem in figure 5.4. The system can check the IDs from the red piece and the blue piece. Once the ID "type A" is matched with "plug" and "socket" checked, we can identify this match and assemble these two pieces by the dimensional data. In this case, it is the "top" of the red piece and the "left" of the blue piece can be attached together.

On the other hand, the real situation might be much more complicated.

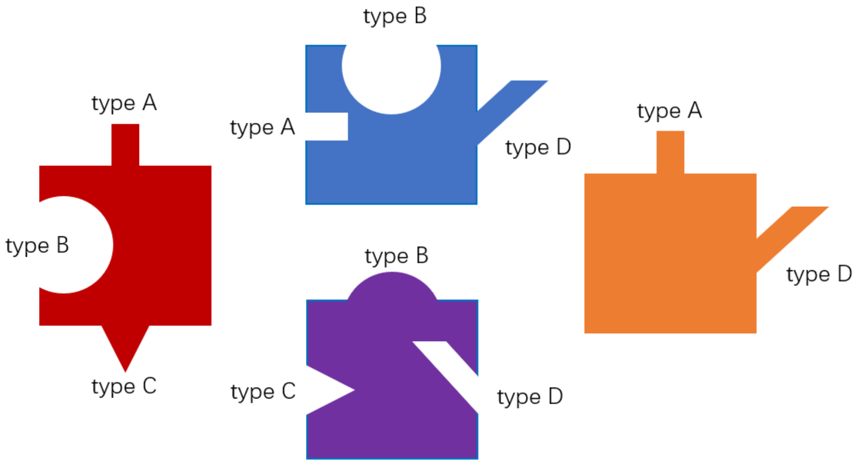


Fig. 5.5 Assembling case 3

In figure 5.5, there are multiple pieces with multiple different kinds of plugs/sockets need to be assembled. Also, these plugs/sockets are not in a one-to-one relationship, which even increases the level of complexity. In a situation like this, simply try it one by one is very inefficient. Therefore, the system needs to simplify the data first.

5.2.2 Structuralize the component data

After the system gathered all the information from each component for the real case, a table(.csv) will be made to store this information. each row will contain the detail information for one single component. For example, a CPU many contains "ID", "Name", "Socket ID" and the coordinate of the socket. The system will put all of the components together as the table shows below.

ID	Name	SV	SS	SID	x	y	z
90001	Motherboard	-3	1	-AM4	-0.274	0	-0.052
0	null	null	2	-DDR4	-0.075	0	-0.252
0	null	null	null	-DDR4	0.057	0	-0.252
0	null	null	1	-PCIEx16	-0.069	0	-0.008
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

Table 5.2 Component table(raw)

- SV: Socket Volume, how many kind of socket;
- SS: Socket Size, how many same sockets for this type;
- SID: Socket ID, important ID to determine whether a link can be created;
- x,y,z: Coordinate data for that socket.

In this table, the negative value (except in coordinate) shows that is a socket needs to plug and the positive value (except in coordinate) shows that is a plug. "0" (except in coordinate) and "null" shows data of the last component is not yet finished. Therefore, there serve the structure and have no actual quantity meanings.

For example, "0" in table 5.2 shows the information of the "Motherboard" is not ended yet, and the "null" in the column "SS" shows the third row is the extension of the second row. In this case, it means the product "Motherboard" have 3 kinds of plugs/sockets (SV value = 3), there are all sockets (SV is negative). For the second kind of plugs/sockets, there are two of them (SID = 2) and they are DDR4 sockets (SID = -DDR4).

Let's look at table 5.2 row by row.

In the first row, 90001 is a motherboard. "-3" shows it has 3 different kind of socket that needs to be inserted with other components. "1" shows there is only one socket for this type, which is "-AM4". Again, minus shows this is a socket that needs to be plug-ed in something. The last three values show the coordinate of the "-AM4" socket.

In the second row, the first value "0" shows this is the continuation of the motherboard. The "socket size" value "2" means it got two "-DDR4" type of socket.

However, not all of these plugs/sockets are required to assemble. For example, we have two DDR4 sockets but only one DDR4 device, also we have to PCIEx16 device for the PCIEx16 socket in motherboard.

When our algorithm computes this table. Firstly, it will check the non-zero entries of the "ID" column, and understand there are totally four components. Then, the system will try to pair the "SID" column. For example, "-AM4" will be paired with "+AM4". The "SS"(Socket Size) will be decreased if there is no pair for any socket; if the "SS" is already "1", then the "SV" of the component will be decreased by 1. In this process, the row that its "SID" got no pairs will be eliminated.

ID	**Name**	**SV**	**SS**	**SID**	**x**	**y**	**z**
90001	Motherboard	-3	1	-AM4	-0.274	0	-0.052
0	null	null	2	-DDR4	-0.075	0	-0.252
0	null	null	null	-DDR4	0.057	0	-0.252
0	null	null	1	-PCIEx16	-0.069	0	-0.008
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

Table 5.3 Component table (matching)

In this case, the system will compute row by row with the "SID" value. The row 1 pair with the row 5, the row 2 pair with the row 6, then row 3 and the row 4 have no pairs.

Once the system finds row 3 has no pair, it will try to decrease the value of the "SS" entry in the same row. However, here is a "null" value. so the system will change the value of "SS" from "null" to "0" and look up to the previous row, decrease it by one to "1".

The same thing also happens in row 4. The system will find "PCIEx16" have no pair, then decrease SS by one to "0". Because the "SS" is already decreased from "1" to "0", the "SV" of the component will be also decreased by 1.

Following table shows what happened after the action. The Colored entries mean that the values have been changed.

ID	Name	SV	SS	SID	x	y	z
90001	Motherboard	-2	1	-AM4	-0.274	0	-0.052
0	null	null	1	-DDR4	-0.075	0	-0.252
0	null	null	0	-DDR4	0.057	0	-0.252
0	null	null	0	-PCIEx16	-0.069	0	-0.008
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

Table 5.4 Component table (decreased)

After this process, the system will eliminate the rows with zero values in "SS" entry. The table below shows what it would be after the rearrange.

ID	Name	SV	SS	SID	x	y	z
90001	Motherboard	-2	1	-AM4	-0.274	0	-0.052
0	null	null	1	-DDR4	-0.075	0	-0.252
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

Table 5.5 Component table (computed)

Once this rearrangement is completed, the system eliminates the extra information and only kept the sockets that need to perform some action. The system can now create holographic 3D models based on this table.

5.2.3 Link connection

After our system gets the pairs based on "SID"s, it will seek the link connection information online.

SID	Group	Name	olPriority	olRequirements
AM4	PC	CPU-Motherboard	0	null
DDR4	PC	Memory-Motherboard	1	null

Table 5.6 Link connection table

The system will look up the information online to get the priority to overwrite and requirements for action. For example. In the above table, "olPriority = 0" means the top priority and "olRequirements = 0 means no need of tools". "Structuralize the component data" and "Link connection" are important steps to figure out required components and needed actions. Once these two steps are done, the system can proceed to "Net-map Strategy" in order to compute the sequence of the assembly.

5.2.4 Data visualization and Net-map Strategy

In the last part, we simplified the component data and we figured out to connect plug/socket by their socket ID (SID). After that, we managed to record the connection by socket ID (SID) and name it. The next task is to visualize these data.

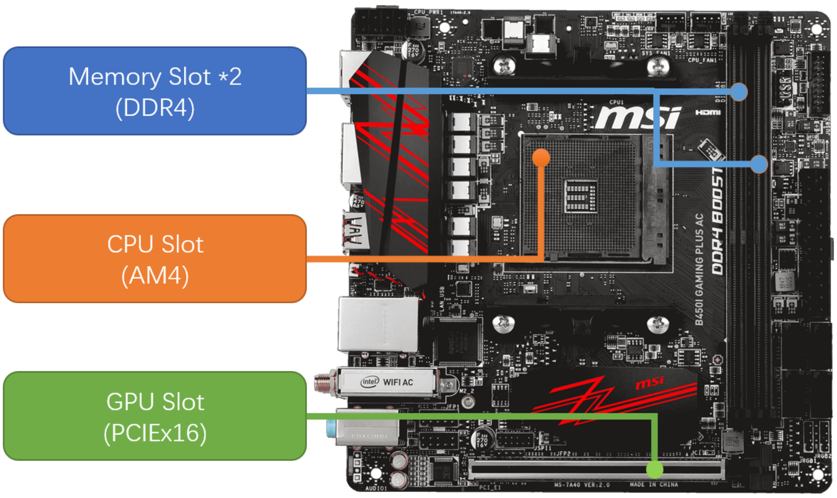


Fig. 5.6 Net-map1

From this figure, we can understand this motherboard has three kinds of the slot. Memory got two slots. Then, we can try to simply this figure.

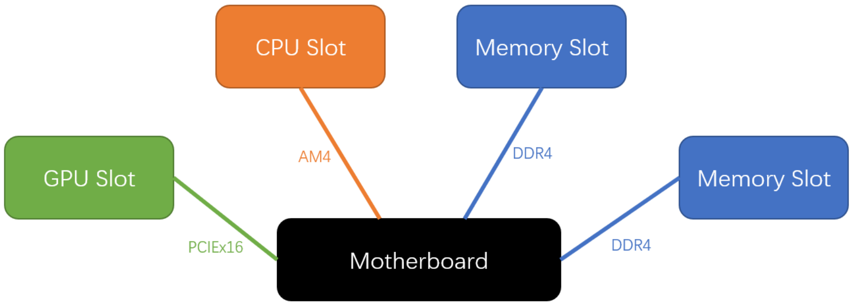


Fig. 5.7 Net-map$_2$

In this figure. Squares(nodes) represent as components and the lines represent as the "Link" between two components. Actually, this figure is the visualized version of our component table and link table.

In table 5.5, the system scans the "SID" entries and find two pairs. Then the system will display these components with the 3D models and locate them by the dimensional information in table 5.5. Then, the system will check the "AM4" and "DDR4" in the "Link connection table".

These will be transferred into the image in figure 5.7. The squares are based on "ID" and the lines are based on "SID" from table 5.5.

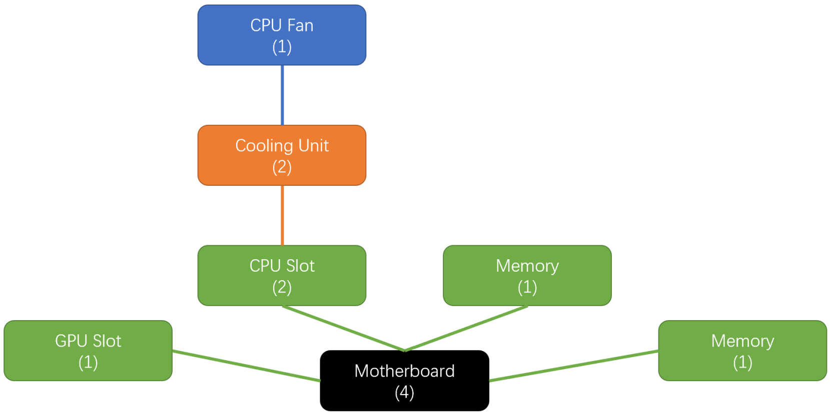


Fig. 5.8 Net-map3

After we add more components into this figure, we can find a natural layer type of relationship appeared. The motherboard connects with four components and becomes the most inside layer. This kind of "Net-map" can be displayed by the 3D models of real components and have the potential to give the user a relatively better understanding of how things can be assembled.

Fig. 5.9 Net-map using real components

In our system, we generate a hierarchy of components by this "Net-map" with the dimensional information. In figure 5.9, lines with different color such as red, green and yellow demonstrate the "Link connection" between two components.

Usually, we think the component with most connections should be assembled first because after the components around it finished, it will be very hard to access to that inside component physically. Therefore, the default generated sequence is Black -> Green -> Orange -> Blue for figure 5.8.

5.2.5 Fixed sequence of the instructions

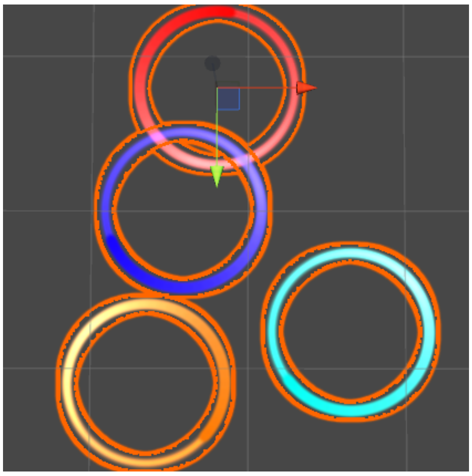
In the designing phase of this system, we constructed an algorithm based on the Net-map. Basically, the node which connects with the most other nodes will be the first to compute. It was working well, but we decided to set this as an alternative method.

Actually, the sequence of assembling something is not something need to be generated. In fact, the sequence is a constant pre-set, because all the components are designed and made for assembled by a certain sequence. On other word, it meant to be assembled by a single sequence.

Therefore, the sequence of our system will be mainly based on the "olPriority" entries inside the link data. (From small value to large value)

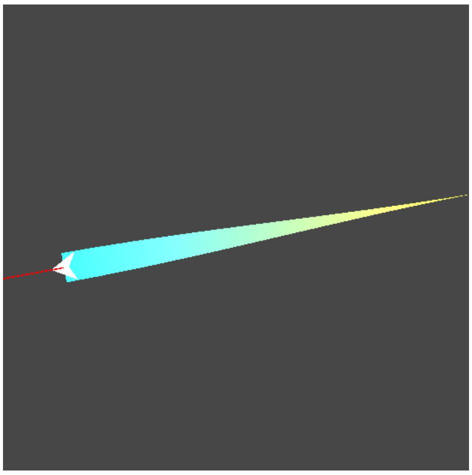
5.3 Holographic Display and Automatic Layout

Holographic Display usually is the combination of virtual components and zones.

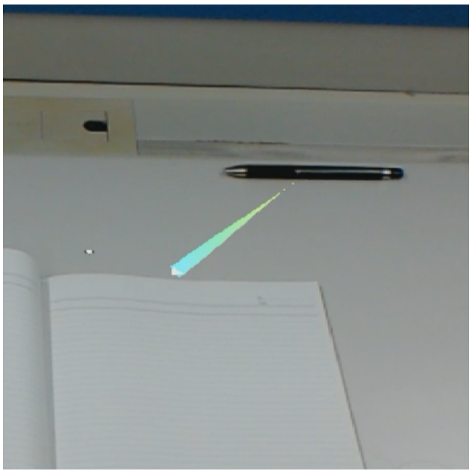


(a) Rings in Unity

(b) Rings



(c) Arrows in Unity



(d) Arrows

Fig. 5.10 Virtual Component (Indicators)

Virtual components contain 3D models and virtual indicators. Different virtual indicators suggest different activities that need to be made by our user. For example, the Rings are positional indicators to address where is the component, the Arrows are directional indicators to address which direction the user needs to move the component.

Zones contain Depot, Workspace and Board. Functions such as Object recognition, Object tracking or Collision detection are enabled differently in these three zones.

Depot

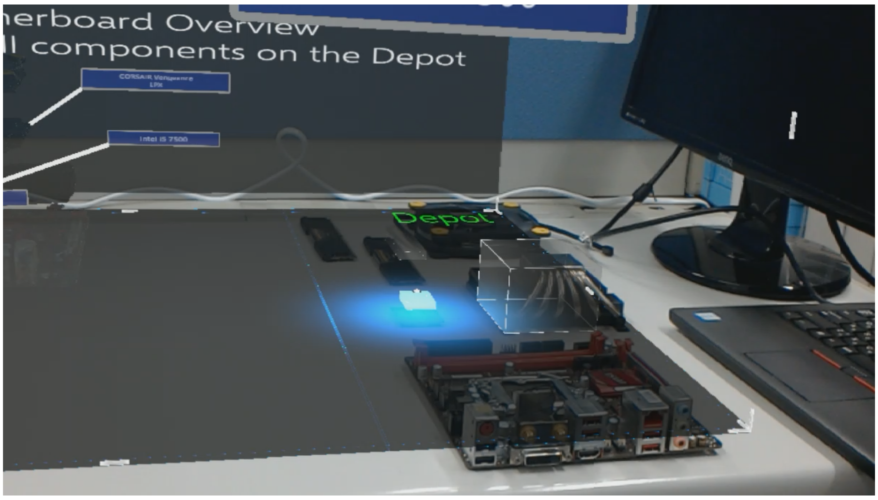


Fig. 5.11 Depot

Depot is where user can unpack all the component and place on. Object recognition function is enabled to recognize all the components. Once a component is recognized, an indicator will be shown with the component's name.

Board

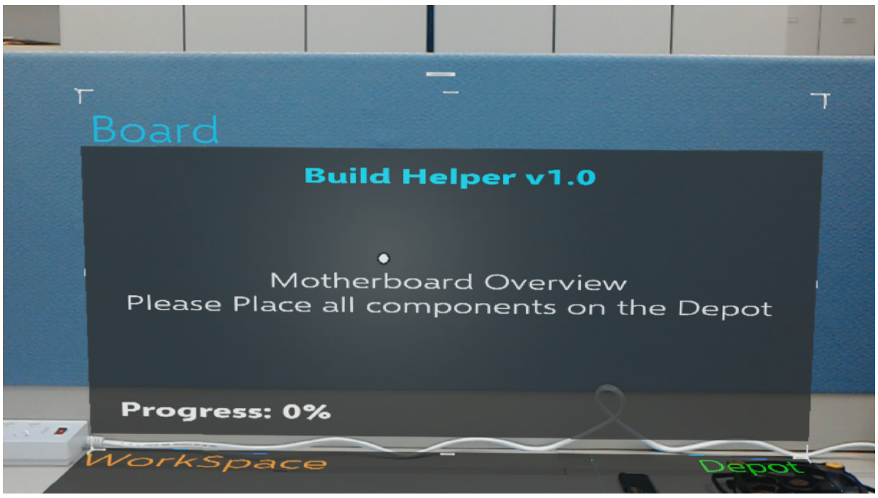


Fig. 5.12 Board

Board is a virtual object attached to a vertical surface. Additional information such as progress status and text instructions will be shown.

Workspace

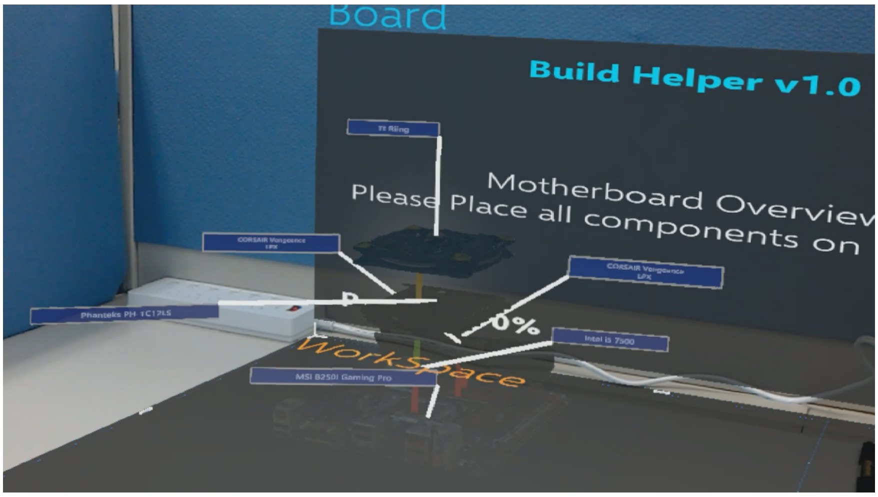


Fig. 5.13 Workspace

Workspace is the main zone for our system. Not only object recognition but also object tracking will be enabled. A certain virtual object will be shown to indicate how the user can perform the action and sensors will keep monitoring the change for preventing mistakes.

Reachable Domain

(a) RD in Unity (b) RD

Fig. 5.14 Reachable Domain

Our system is constantly sensing the user's hand gesture, movement and coordinates. A certain range of space that surrounds the user's hand is called reachable domain. If the user performs a grab gesture and certain real object is inside this domain. The system will treat it as the user is grabbing this object.

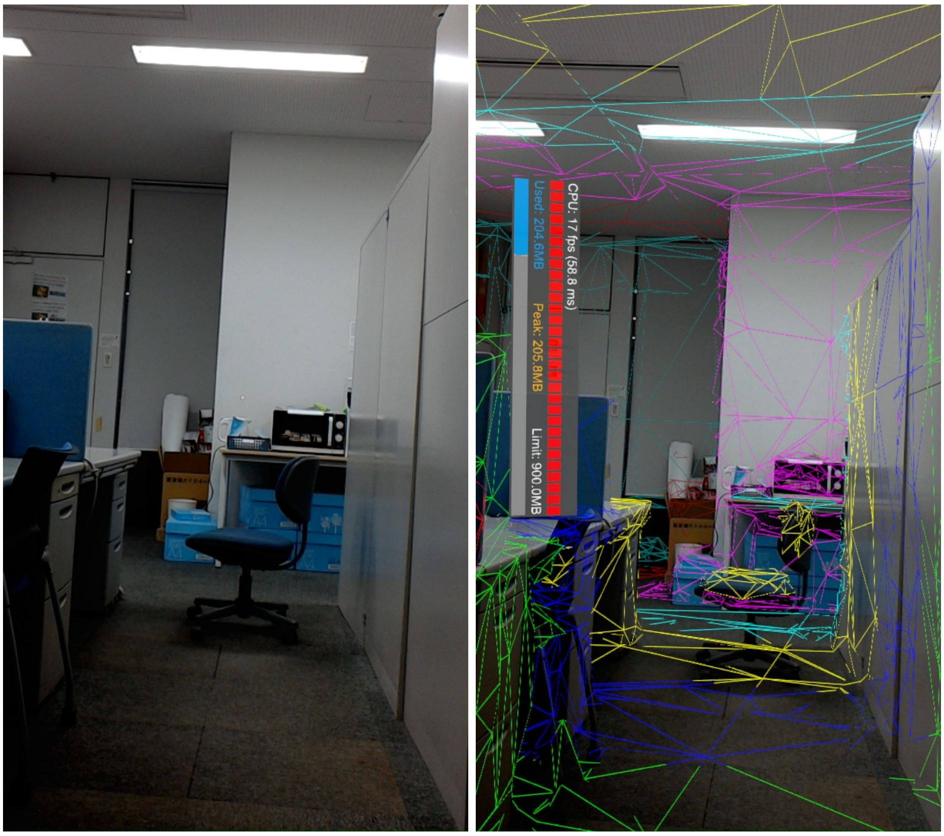
5.3.1 Automatic Layout

When people think about assembling something. the first image that comes out is usually a worker sits at a table and performing the work.

In order to achieve a user-friendly experience. We decided to put all the objects which including real objects and virtual objects on a table.

There is some previous research about fit the virtual contents with the shape of the space automatically. [16] Therefore, our system will automatically decide where to put all the contents and decide the size and layout.

Spatial Recognition



(a) Real Environment (b) Spatial Meshes in Real Environment

Fig. 5.15 Spatial Meshes

As the Figure shows, we use depth cameras to detect surfaces of the real world. By using the spatial understanding API of Mixed Reality Toolkit 2 (MRTK2). We can get the metadata of detected surfaces: Position, size, horizontal or vertical. These unclassified data are complex and cannot be utilized.

In a real environment, surfaces surrounded are usually very complex. To make the system understand the environment, we designed a simple understanding mechanism.

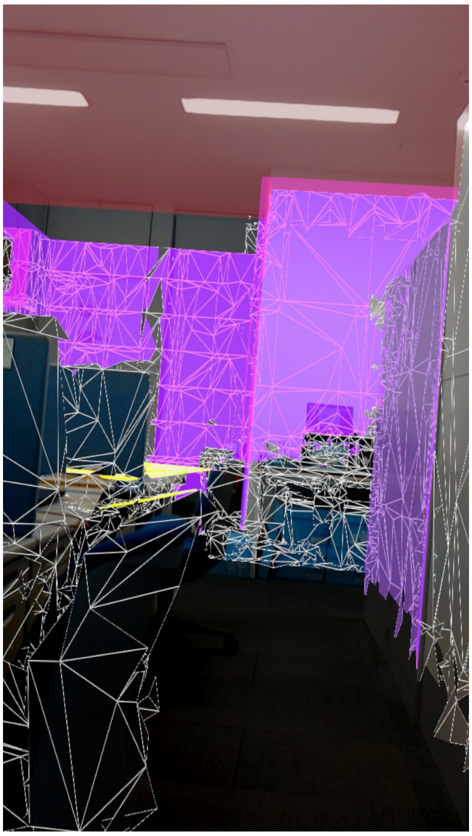
1. Ceiling: The highest horizontal surface in the space;

2. Floor: The Lowest horizontal surface in the space;

3. Wall: The Vertical surfaces in the space;

4. Platform: The horizontal surface that is not Ceiling or Floor.

(a) Spatial Meshes



(b) Understand the surroundings

Fig. 5.16 Surface Understanding

Through this spatial understanding mechanise, the system can find every Platform in the space. In order to place all the component to assemble, our system will use the largest Platform in the space. [17, 18]

Responsive Layout

After the Spatial Recognition, the system will try to place the Depot, Board and Workspace on the largest Platform. By accessing the data from the largest component for assemble, the system will understand if this Platform has enough space for the task. Once confirmed, our system will try to adjust the size of Depot and Workspace to place on the Platform. The Board will be attached on the nearest and visible wall near the Platform.

5.4 Object recognition

Object Recognition can be used to build rich and interactive experiences with 3D objects. These experiences could be augmenting a toy with 3D content to bring it to life, overlaying a user manual on top of a consumer electronics device or leading a new employee through an interactive training process for a workplace device.

Because our target users are amateurs. It is better to have an "eye" on them all the time. Therefore, we introduce the Object Recognition technology into our system, to make our system understand the user's hand and all the component that required to assemble.

Compare with traditional Image-based recognition, our system is mainly focused on 3D object recognition with Object Targets. Object Targets are a digital representation of the features and geometry of a physical object. They are distinct from image-based target types, such as Image Target, Multi Targets and Cylinder Targets that require the use of a planar source image.

Our system is using a camera to view the same image to our user, and try to compare all the object will the 3D Object target in our database. Once a match happened, the system will push all the attributes from the 3D Object target to this real object and recognize it.

5.4.1 Mixed recognition Strategy

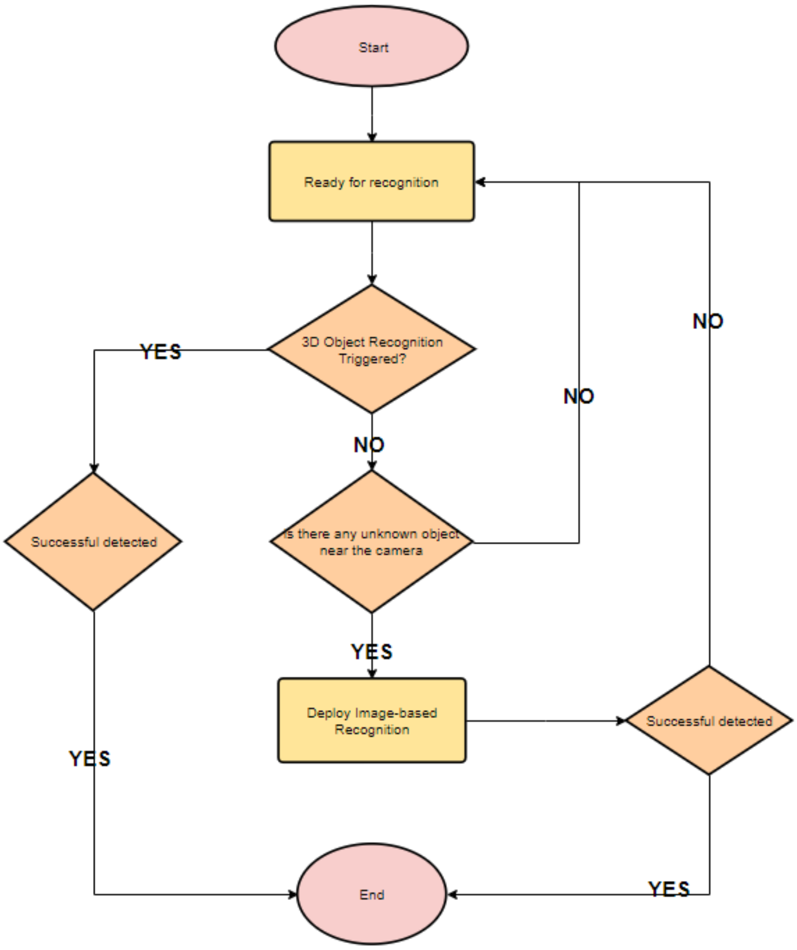


Fig. 5.17 Mixed recognition Strategy

Sometimes, the real object that needs to be recognized is just too small for our 3D Object Recognition mechanism. Therefore, we combined it will the traditional Image-based Recognition technology. When a user grabs an item and faces it to the camera with a short distance, this strategy will be triggered.

Our system will perform 3D Object Recognition and Image-based Recognition at the same time. The result of Image-based Recognition will get higher priority and overwrite the overall output.

5.4.2 Articulated Hands

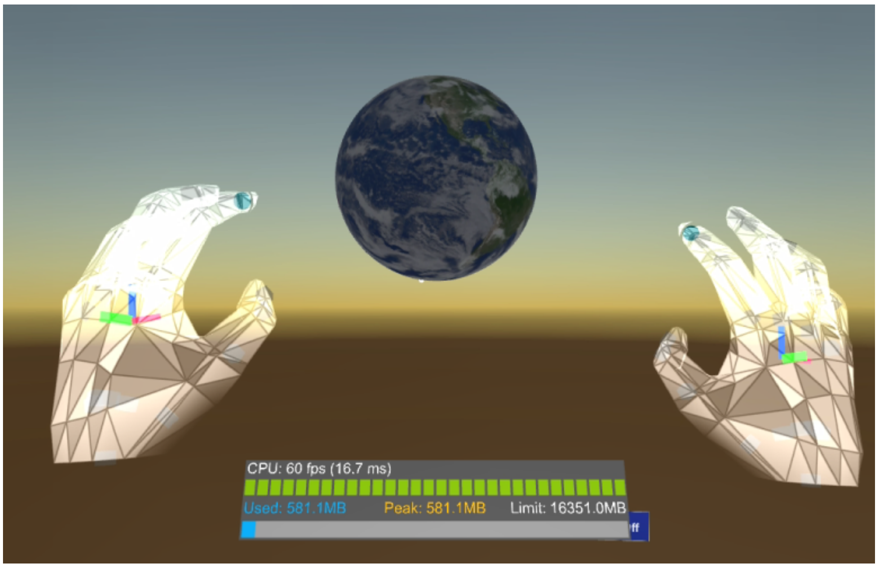


Fig. 5.18 Articulated Hands

We also introduce the Articulated Hands system in our system for better hand recognition.

Articulated Hands system simulates a fully articulated hand with joint position data. From these data, the system can understand what kind of gesture the user is performing, such as push, drag or grab. [19] With the understanding of the gesture and the positional data of the user's hand compared with nearest objects. The system can understand what the user what to do.

5.5 Interactions between user and System

One of our goals in this research is to make the system interactive. We think the Build Helper is a "guidance" system for assembly rather than an "instruction" system. Different from any other "instruction" system such as paper instruction, our system is not just throwing the "process" and "result" to the user. instead, We designed our system to be interactive. Thus, the system can communicate with the user while assembling, telling the user what is every component for. Additionally, our system can stop the user if any wrong action is performed.

5.5.1 Non-gesture design

Because our main focus for this system is the assemble the real object. Therefore, we think the user gesture to manipulate with the virtual object is unnecessary and troublesome. Using gesture for giving input to the system such as "go next step" is even worse because users need their hand to grab and hold real components. So normal gesture is indeed harming our goal to be Hands-free.

5.5.2 Criteria of a single assembly task

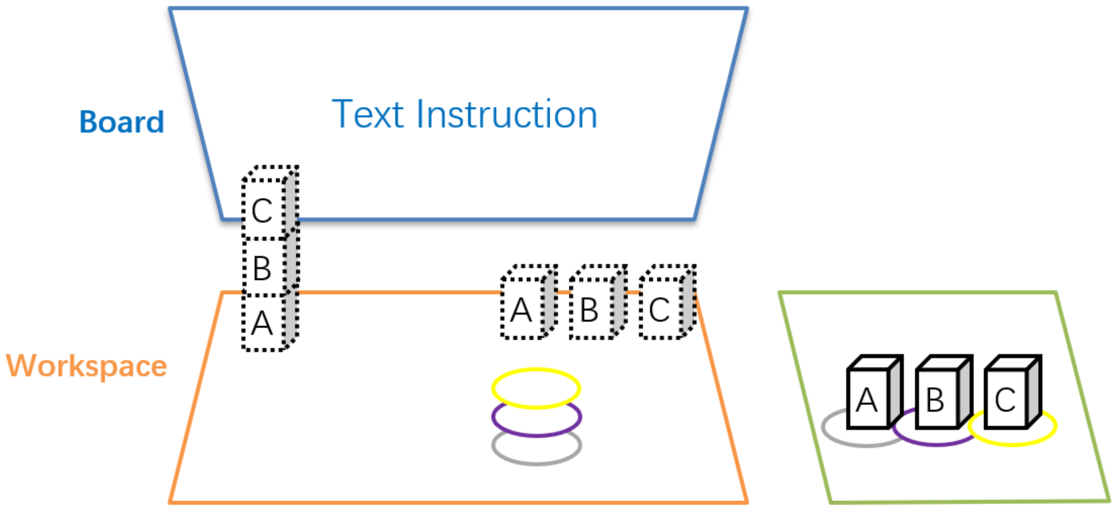


Fig. 5.19 what, where, how

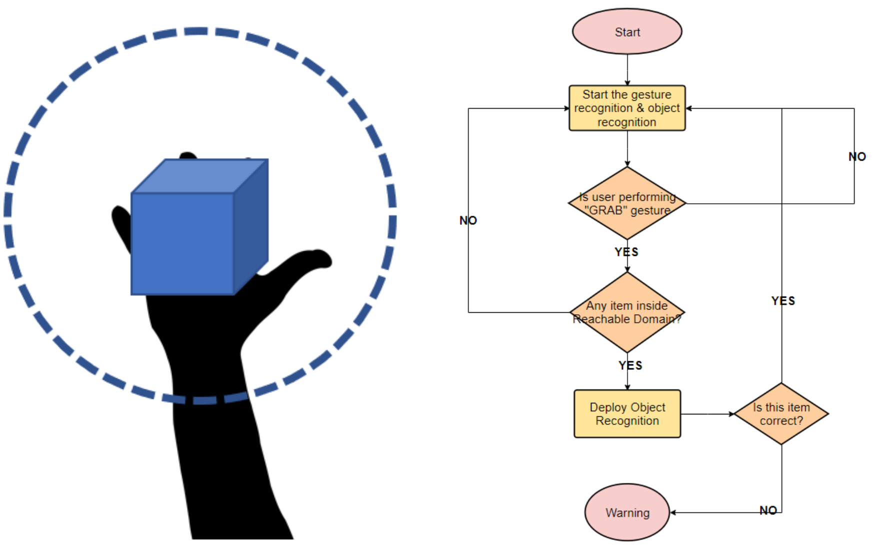
When we think of the Criteria of a single assemble task. Usually, we break up the task into these questions.

1. What is the component I need?
2. Where is that component?
3. How can I assemble that component?

In order to answer these questions, we designed 3 individual set of content across the Workspace and Depot. Here is our example of this design.

- In front of the user is the virtual components that show what kind of item is needed;
- In the right-hand side on the Depot zone. virtual rings are marking the position of the real component that needed for this task;
- In the left-hand side on the Workspace zone. virtual objects are stacked together to show how the user can make the action.

5.5.3 Dynamic troubleshooting



(a) Real Environment (b) Strategy of Dynamic Troubleshooting

Fig. 5.20 Dynamic trouble shooting

"Dynamic troubleshooting" is a technical term we used in our system that demonstrates the effectiveness of the monitoring and recognition function.

Our system using a mixed architecture of traditional image-based recognition technique and 3D object recognition in order to detect every frame that caught by our camera, especially when the user finished some actions. The system detects and distinguishes whether the user made a right action or not based on that caught frames. This process is the basis of our troubleshooting function.

However, this method is very limited, because the user has to perform any mistakes in order to trigger the system. In other word, this method can only tell the user there is a mistake when the mistake has already happened.

In order to prevent any mistakes from happening, we came up an idea of monitoring what is on user's hand, and by what is the item on user's hand to speculate whether it is right action or not. The task sounds simple but it is near impossible by only using Microsoft HoloLens.

We solved this problem by introducing an external device called Leap Motion into our system. The Leap Motion is a sensor that can detect the user's hand skeleton, and make a 3D skeleton model of it dynamically.

We tried to deal with this problem separately by dividing this into two tasks: "what kind of gesture is performing currently" and "What kind of object is near the user's hand". If our user is performing a grab gesture and there is an apple near the user's hand (inside "Reachable Domain"), the system will speculate our user is currently holding an apple.

5.5.4 Detail tray and Nature movement

Detail tray is a function that we came up with in order to boost the interactions between our user and the system while maintaining hands-free.

In the whole assembling process, our user is expected to use a lot of components that he/she is not familiar with. We think it is maybe very confusing or even dangerous When someone performs actions without knowledge. Thus, provide a function to give our user the information of each component is may be important. Our system use object recognition technology to detect and identify the components. After that, when using virtual labels to mark up all the names of each component. However, amateur users often feeling confused and struggled even with the names. If we attempt to display all the details information, the user's view will be very messy and a lot of things will be blocked by virtual labels because there are a lot of items in our user's view. In summary, we have to have an additional interaction method in order to call the detail information.

Not using the gesture is a very important strategy of our system. Because the main focus of using our system is to assemble items. Thus, using gesture may give more distraction to the user and the gesture detection system may be interfered with our object detect system badly. However, make the user interact with the system and create a rich interaction system is an essential thing for our system. Therefore, we designed a "Natural movement" input method for our system.

Fig. 5.21 Detail tray and Nature movement

The figure above shows our idea of combining our Detail tray and Nature movement of the user's hand.

Normally, when the user grabs something or he/she wants to perform some sort of actions, the palm is often facing down. On the other hand, when someone felt curious about something, he/she usually hold this item with the palm up and try to look at that object. It is a very natural gesture that happened in regular people. Therefore, we think if our user finds it is interesting for any components. Usually, our user will hold that component will the palm facing users hand or facing up.

When our system detects the user is holding a component with the palms inside certain angle (object inside Reachable domain + object recognized as a component + palm in right angle). A Detail tray will be shown by one side of the user's palm.

The Detail tray is a surface that showing 2D information (text and picture) to the user, which is very similar to the "Board". Information such as detail description or how this component can be assembled will be shown.

Contents inside the Detail tray could be varied because different information is more desired in different cases. For example, when the user is trying to assemble a PC. Maybe what he/she wants to know is the detail description, the price of our options on the market. On the other hand, when the user is trying to assemble a Logo. Maybe the result is not as valuable as the previous case, because the important thing is to have fun in the process. Thus, The Detail tray could be showing information such as why these pieces are needed or how many same pieces left.

Furthermore, detect an angle or a movement when the user holding one of our target components have a lot of potentials to design more Nature movement input.

Chapter 6

System Implementation

6.1 Hardware Setup

We divided this system into two parts: the HMD Unit and server. To achieve this system, we need:

1. An HMD with Mixed Reality Display;
2. A set of sensors that understand the surrounding environment;
3. A camera that supports 3D Object Recognition;
4. A sensor that supports tracking Articulated Hands;
5. A computer as a server to store the component data.

Fig. 6.1 Required Hardware

Therefore, we choose Microsoft HoloLens. It is a see-through type head-mounted display with built-in depth camera set and a front camera enabled with 3D Object Recognition function. It blends cutting-edge optics and depth sensors to deliver 3D holograms pinned to the real world around the user.

Fig. 6.2 HMD Unit

To make the Articulated Hands tracking possible, we add a Leap Motion sensor to the HoloLens with 3D printed head mount frame. Because the Leap Motion itself cannot provide electricity and it does not have any wireless module, plus the micro-USB port of the HoloLens does not support OTG protocol. We connect the Leap Motion with a compute stick in order to send data to Hololens wirelessly (UDP protocol) [20] [21] [22].

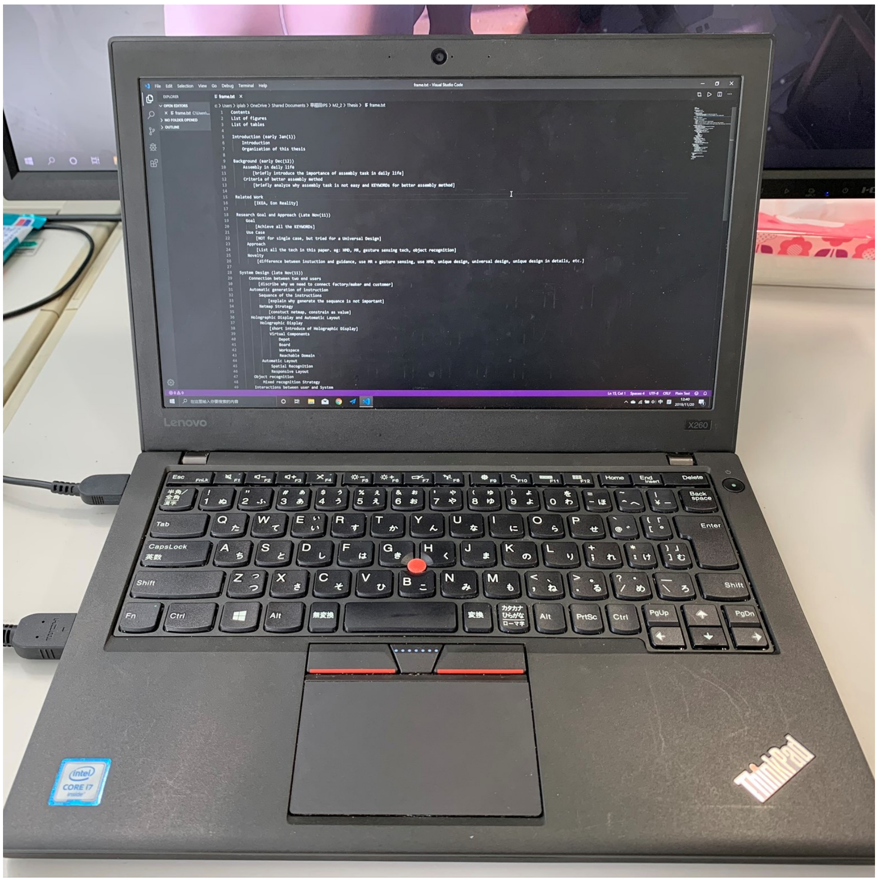


Fig. 6.3 Laptop

In order to store and access the component data (3D models and information), we use a laptop as a server. This laptop and HMD Unit are in the same network environment to increase access speed.

6.2 Software Environment

The Software Environment support is:

1. Mixed-Reality Toolkit v2.1.0 (MRTK2), it Supports environment data analysis and gestures input functionality of my system;
2. Vuforia Object Recognition SDK, it allows the user to detect and track intricate 3D objects;

3. Leap Motion SDK, it gives access to the raw tracking data from the Leap Motion service;

4. Microsoft speech-to-text API, it converts the user's voice commands into text.

To make a 3D environment and build it to HoloLens, we use Unity 2018 and Visual Studio 2019.

6.3 UDP Connection

Since Microsoft HoloLens does not support OTG protocol and a Leap Motion device contains no Bluetooth module. We decide to make a UDP connection between these two devices in order to make a connection.

The Leap Motion software installed on any computer with a Leap Motion controller provides raw tracking data through WebSocket. We made a WebSocket Server to broadcast JSON data of the Leap Motion and also developed a client on UWP platform that can run on HoloLens.

Each WebSocket package is containing tracking data that formatted in JSON, Which include:

```
"currentFrameRate": float
"id": float
"r": array of floats (Matrix)
"s": float
"t":  array of floats (vector)
"timestamp": integer
"devices": not implemented (always an empty array)

"gestures": array of Gesture objects
    (Attributes vary by gesture type)
    "center": array of floats (vector) -- circle only
    "direction": array of floats (vector) -- swipe, keyTap, screenTap only
    "duration": integer microseconds
    "handIds": array of integers
    "id": integer
    "normal": array of floats -- circle only
    "pointableIds": array
    "position": array of floats (vector) -- swipe, keyTap, screenTap only
    "progress": float -- circle, keyTap, screenTap only
    "radius": float -- circle only
    "speed": float -- swipe only
    "startPosition": array of float (vector) -- swipe only
    "state": string - one of "start", "update", "stop"
    "type": string - one of "circle", "swipe", "keyTap", "screenTap"

"hands": array of Hand objects
   "armBasis: the 3 basis vectors of the arm (array of vectors)
   "armWidth: float
   "confidence: float
   "direction": array of floats (vector)
   "elbow: array of floats (vector)
   "grabStrength: float
   "id": integer
   "palmNormal": array of floats (vector)
   "palmPosition": array of floats (vector)
   "palmVelocity": array of floats (vector)
   "pinchStrength: float
   "r": array of floats (Matrix)
   "s": float
```

Fig. 6.4 JSON Data$_1$

```
    "sphereCenter": array of floats (vector)
    "sphereRadius": float
    "stabilizedPalmPosition": array of floats (vector)
    "t": array of floats (vector)
    "timeVisible": float
    "type": string - one of "right", "left"
    "wrist: array of floats (vector)

"interactionBox": object
    "center": array of floats (vector)
    "size": array of floats  (vector)

"pointables": array of Pointable objects
    "bases": the 3 basis vectors for each bone, in index order, wrist to tip, (array of vectors).
    "btipPosition": the position of the tip of the distal phalanx as an array of 3 floats.
    "carpPosition": the position of the base of metacarpal bone as an array of 3 floats.
    "dipPosition:" the position of the base of the distal phalanx as an array of 3 floats.
    "direction": array of floats (vector)
    "extended": boolean (true or false)
    "handId": integer
    "id": integer
    "length": float
    "mcpPosition": a position vector as an array of 3 floating point numbers
    "pipPosition": a position vector as an array of 3 floating point numbers
    "stabilizedTipPosition": array of floats (vector)
    "timeVisible": float
    "tipPosition":  array of floats (vector)
    "tipVelocity":  array of floats (vector)
    "tool": boolean (true or false)
    "touchDistance": float
    "touchZone": string - one of "none", "hovering", "touching"
    "type": integer - 0 is thumb; 4 is pinky
    "width": float
```

Fig. 6.5 JSON Data_2

The motion factors, r, s, t, attached to Hand and Frame objects are snapshots of the motion occurring across frames. These factors must be combined with those of a previous frame to derive the relative motion.

- r – a 3x3 rotation matrix
- s – a scale factor
- t – a 3-element translation vector

For a 3D object, it has a rotation factor to indicates its rotation, a scale factor to indicate its scale and a translation factor to indicates its location. With these three motion factors, r, s and t. We can compute these three basic factors by following expressions.

- Rotation factor

$$rotation = r_{currentframe} * r^{-1}_{sinceframe}$$

- Scale factor

$$scalefactor = e^{s_{currentframe} \ - \ s_{sinceframe}}$$

- Translation factor

$$\overrightarrow{translation} = \vec{t}_{currentframe} \ - \ \vec{t}_{sinceframe}$$

We also developed an API for receiving and using these raw data and transfer it into MRTK2. Basically, Acting like we are using the upcoming HoloLens Gen2.

6.4 User Workflow

The workflow of each step is pretty straight forward.

1. The appearance of the packages will be displayed on the Board;
2. The user unpacks the required packages;
3. The user places the unpacked components on the depot;
4. The system recognizing the new item on the depot;
5. The system check these items in and label these items;
6. The system will ask the user to move one component to the Workspace first;
7. Indicators will be shown to tell the user how to assemble these two components together;
8. User can perform the action;
9. The system will check the result of this step, and proceed to the next one if everything is right.

6.5 User-end for the factories

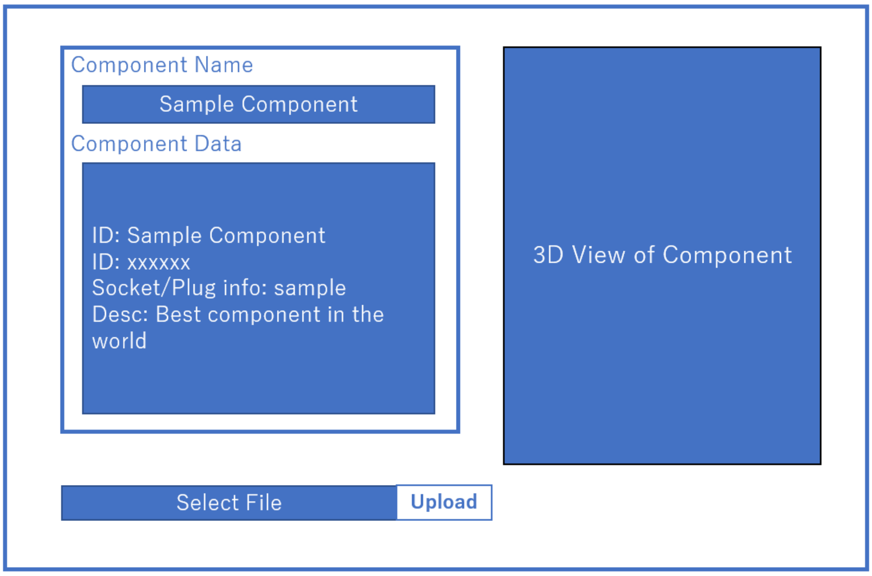


Fig. 6.6 Concept of Factory End UI

The fig. 6.6 is showing a concept of our factory end UI. We embrace the factory side user of our system provides the raw data with easy access. They need to upload a 3D model of the product(.fbx and .od) and a datasheet(.csv).

The 3D model is used to apply in the holographic display, which is an important part of our intuitive instruction display.

The datasheet usually contains data such as the name and description of the product. Additionally, It contains the information of all the sockets/plugs of this product.

For example, if our factory side user uploads a datasheet for a "Motherboard" product, the datasheet would have the information of sockets such as "CPU socket", "GPU socket", "Memory chip socket", etc.

For each socket/plug, we need serial ID and dimensional data in order to proceed with our algorithm. Here is another example to explain this.

The information of a CPU socket Name of this socket: Show what is this socket (e.g. "CPU socket") Serial ID: Show what kind of socket it is, for pair with suitable component (e.g. "AM4") Dimensional data of this socket: Show where is this socket relative to the motherboard (e.g. "-0.0057, -0.2679, 0.1293")

6.6 User-end for the consumers

The consumer is the main character for our system. By accessing the file uploaded by the factory side, our system can generate and offer an intuitive and user-friendly assembly experience. In order to achieve that, our system has to go through these steps:

1. Scan the package;
2. Automatic generate and adjust the working environment;
3. Automatic generate the assemble layout;
4. Dynamic instructions for each step.

6.6.1 Scan the package

Fig. 6.7 Scan the package

When the user is scanning the package, the system is trying to find the matching package throughout all of our cooperated maker's database. Once a pair of a match is founded, our system can recognize the product and download the data (.fbx file and .CSV file) of this product to our local folders. while each package is being scanned, our system will check if it is able to make a assemble process every time. if a minimum requirement is satisfied, our system will show the user it is possible to proceed to the next step. (e.g. the minimum requirement for a motherboard unit is a motherboard, a CPU and a memory)

6.6.2 Automatic generate and adjust the working environment

In this step, the system will ask the user to scan the surrounding environment first, this is a Spatial Mapping process. Our system using the depth sensors of HoloLens to create a 3D model for the surrounding environment. By analyzing the coordinates of surrounding meshes, our system can eliminate the surface that we do not need such as the floor, the ceiling and the walls.

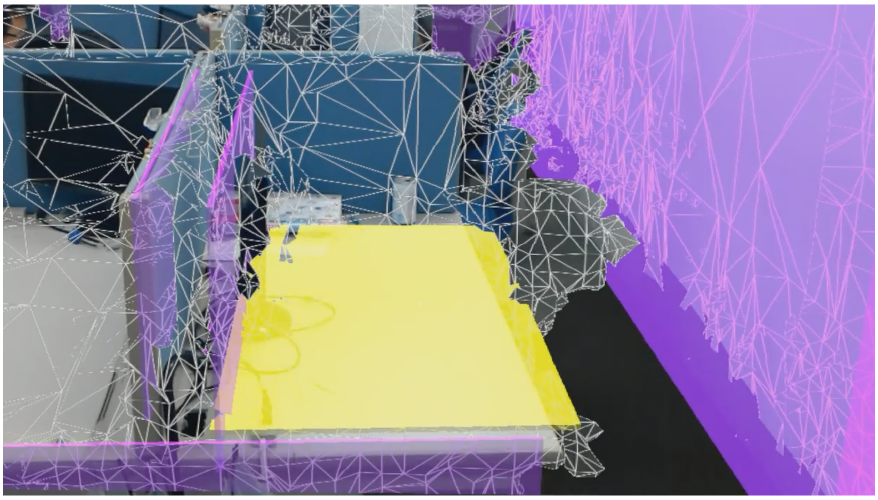


Fig. 6.8 Suitable platform

Our system will select a suitable surface that is big enough and also can be sat by aside (yellow platform in the figure). Then, the system will try to place the three main components, Board, Depot and Workspace. The strategy for adjusting these three components are listed below.

- The Workspace will adjust its size to four times of the largest object in all the process;
- The Depot will adjust its size to larger than the size that combined all the components in one step;
- The Board will be attached to the nearest visible wall towards the user;

- Our system can only proceed if all the requirements are satisfied.

6.6.3 Automatic generate the assemble layout

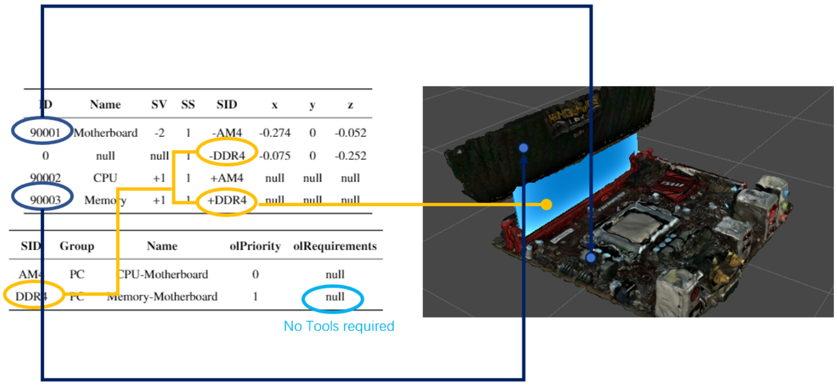


Fig. 6.9 How does the instructions generated

For example, if the user scanned the packages of a CPU, a Motherboard and a memory chip. Because the CPU and the Motherboard have same socket ID and also the Motherboard have another same ID socket with the memory chip. Two links will be generated. due to the higher priority, the CPU-Motherboard Link will be the first to proceed.

Our system will generate the 3D models for these two components and fit together (with some space) by the coordinate information of their sockets (inside .CSV file). This hierarchy will be placed on the far-left corner on the Workspace. When a new 3D model joins the hierarchy, the whole structure will adjust it’s coordinate to secure the visibility.

For a better understanding of this part, we will demonstrate the detail of making the graphical instruction from our computed table.

ID	Name	SV	SS	SID	x	y	z
90001	Motherboard	-2	1	-AM4	-0.274	0	-0.052
0	null	null	1	-DDR4	-0.075	0	-0.252
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

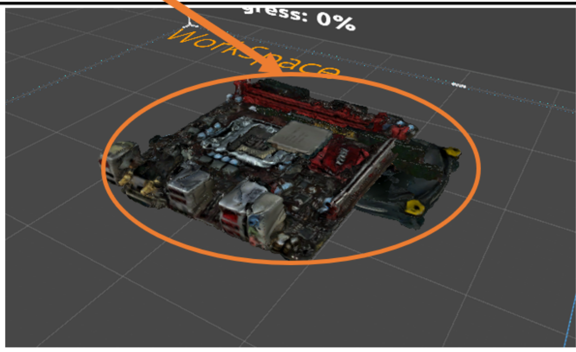


Fig. 6.10 Automatic generation: display the 3D model

Firstly, the system will look up to the "Component table". Our system will try to generate the 3D models of all components by reading their IDs. This process will be proceeded by the descending order of the datasheet. From figure 6.10, the system will access the 3D models by the IDs such as "90001", "90002" and "90003", then display it together without specifying coordinate.

ID	Name	SV	SS	SID	x	y	z
90001	Motherboard	-2	1	-AM4	-0.274	0	-0.052
0	null	null	1	-DDR4	-0.075	0	-0.252
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

Table 5.5 Component table (computed)

SID	Group	Name	olPriority	olRequirements
AM4	PC	CPU-Motherboard	$\boxed{0}$ Greater Priority	null
DDR4	PC	Memory-Motherboard	1	null

Table 5.6 Link connection table

Fig. 6.11 Automatic generation: figure out the sequence

The system will check the "SID" column for getting pairs. In this case, the system recognized "AM4" and "DDR4" connections. Then, it will check the priority of these two connections by seeking for the entries in the "Link connection table". For the figure, we can find out "AM4" connection have greater priority. Therefore, "Motherboard" and "Memory" will be executed first. Because the "Motherboard" has more connections than the "CPU", by the Net-map strategy, the system will execute the "Motherboard" first.

The compute sequence will be: "Motherboard" -> "CPU" -> "Memory".

Move to the origin

ID	Name	SV	SS	SID	x	y	z
90001	Motherboard	-2	1	-AM4	-0.274	0	-0.052
0	null	null	1	-DDR4	-0.075	0	-0.252
90002	CPU	+1	1	+AM4	null	null	null
90003	Memory	+1	1	+DDR4	null	null	null

Table 5.5 Component table (computed)

Fig. 6.12 Automatic generation: relocate the models

After the system sort out the sequence, it will try to relocate all the component by that sequence.

In this case, because the "Motherboard" has the highest priority, It will move to the origin point. Other components will move to a location based on the coordinate value of the "Motherboard". For example, the "CPU" can pair will the "Motherboard" with "AM4", then the 3D model of the "CPU" will relocate based on the coordinate of that AM4 socket. After this whole process, a hierarchy of components can be made.

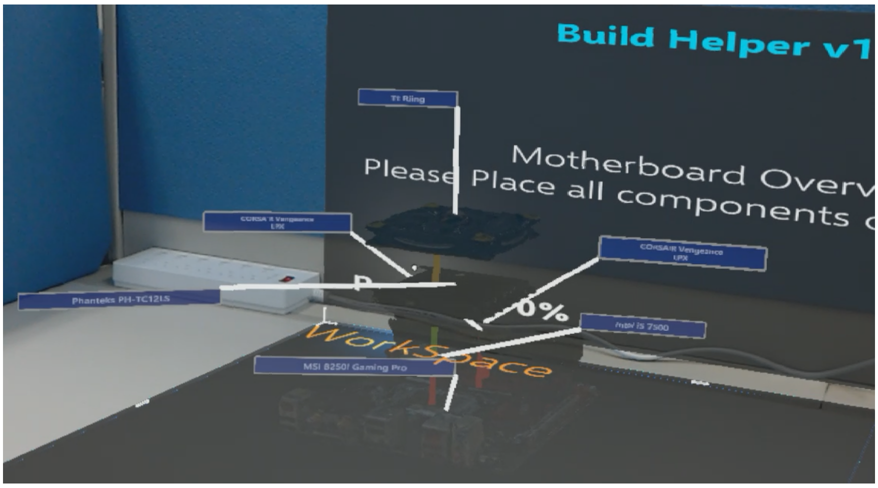


Fig. 6.13 Automatic generation: hierarchy of components

Although the hierarchy shows a good overview of how objects can be assembled, details for each step can be more important.

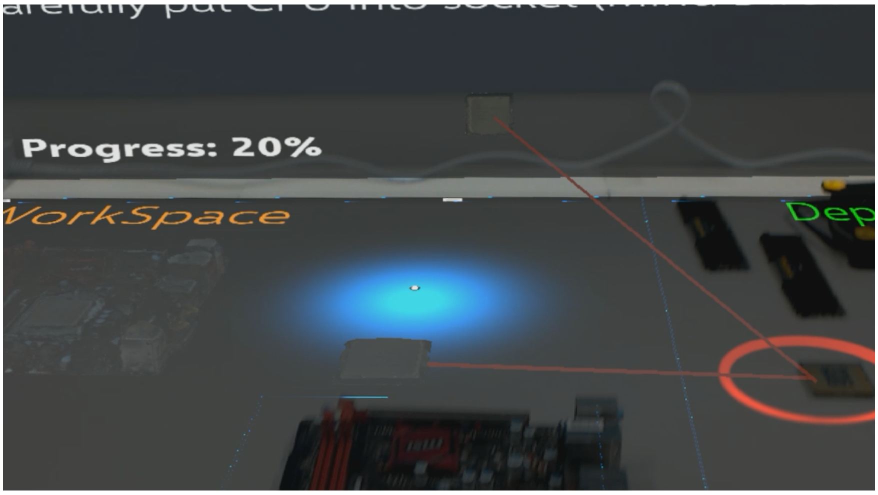


Fig. 6.14 How does the instructions displayed

In front of the user on the Workspace is showing the specific components (objects that connected with a Link) that are needed for a single step, that give our user clear instructions of what is required.

In figure 6.14. In front of the user is the item required, which is the CPU. It links to the actual CPU on Depot, then the connection to the virtual CPU indicator right above the actual motherboard. Which answers the question such as, "what is the component I needed", "where is the component I needed" and "where can I place it".

The remaining question is "how can I do it".

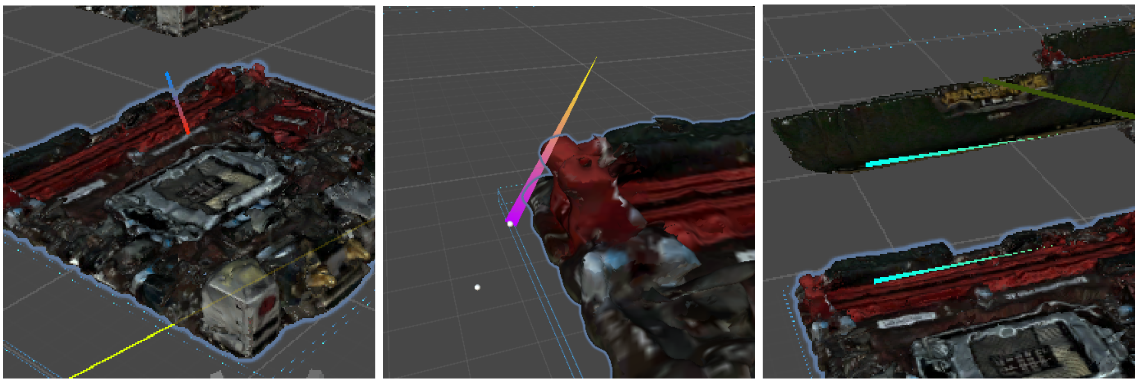


Fig. 6.15 Virtual signs

In our system, we mainly use the virtual sign to indicate how the user can perform the action. If a tool is required, the system will use the Board as the additional device to display text.

Chapter 7

Preliminary Evaluation

In this section, we introduce our preliminary user research and result in analysis. We asked our participants to accomplish several tasks with our system in order to verify whether our system can provide better experiences and better efficiency when compared with the traditional paper manual. Our discussion is based on the received results and received feedback from a questionnaire.

7.1 Participants

We invited in a total of 12 participants (6 females and 6 males), ranging from 20 to 27 years of age. All participants have basic computer skill and can be considered as an amateur user for assembling items. Each of our participants will experience both head-mounted display experience with our system and transitional methods, thus our control group is performed by the same person for each task.

7.2 Method

All participants are given a brief introduction of the system. Before each study, we introduced the basic operations of Microsoft HoloLens to the participants. After the participants became familiar with the device, we asked them to perform our task.

Our task is about asking our participants to use our system and perform several simple assembly tasks. Then perform similar tasks by following a paper manual. After that, the participants will be asked to fill in a questionnaire as shown in figure 7.1. The questionnaire has following questions and these questions use the 5-point Likert scale.

Questionnaire

Name: **Gender:**

Date: **Age:**

Questions

The questions are based on 5-point scale.

Answer the following question by circling the most appropriate answer.

1. **"Hands-free" is very important during assemble.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

2. **"Intuitive instruction" is very important during assemble**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

3. **"Interactions between user and system" is very important during assemble**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

4. **"Trouble shooting" function is very important during assemble**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

5. **The system is easy to use.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

6. **The system is easier to understand than a paper manual.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

7. **The instructions of this system are more intuitive than a paper manual.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

8. **I feel that the system is useful in real life.**

   Strongly Disagree Disagree Neutral Agree Strongly Agree

Fig. 7.1 Questionnaire

7.3 Result

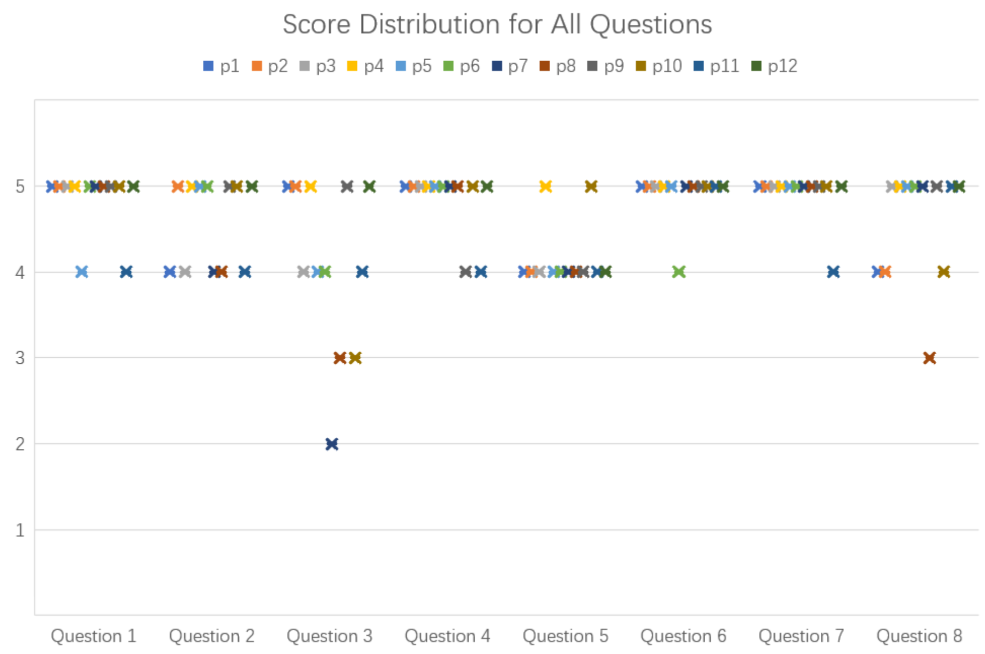


Fig. 7.2 Score Distribution of Questionnaire

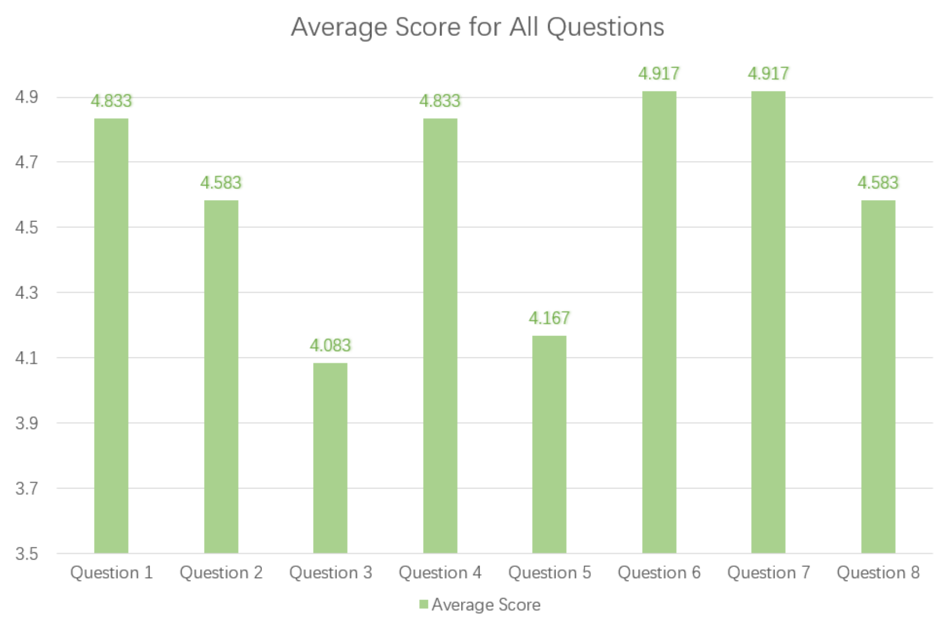


Fig. 7.3 Average Score of Questionnaire

Question 1, 2, 3 and 4 are used to test our basic standards of this research. In fact, we strongly valued these four standards as the most important criteria that need to be achieved.

The average score of question 1 is *4.833*, this proves that people do value our HMD setup and appreciated how they can freely move their hands. Question 2 is about how people value the "intuitive" factor when compared with simple text and pictures. Our average score (*4.583*) shows that normal user tends to favour more intuitive instructions. We have the average score (*4.083*) in question 3, which is not very as high as the previous two. We could understand that average users are more focus on their jobs rather than make interactions with our system. On another hand, we got one of the highest average score on question 5 (*4.833*), which means our user really need the "troubleshooting" function to avoid mistakes during the assembly process.

Question 5, 6, 7 and 8 are used to verify the usability and experience when using our system. The results of these questions are *4.167*, *4.917*, *4.917*, *4.583*. Based on these stats, we could determine that our system is easy to operate and much more understandable compared with a paper manual. Additionally, Our participants do think the system can be useful in real life.

In general, all participants valued our system higher than the traditional ways. This may signify that our system is designed to be reasonable and practical. It demonstrates that our system can potentially provide a new way of daily assembling, allowing users to get a better experience during the assembling process.

Overall, we got positive feedback through the preliminary user study.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

Due to the result of our preliminary evaluation, we think our system is qualified as a better solution for daily assembly tasks.

In general, we design a framework in order to build an intuitive interaction system for assembly tasks. Our system use mixed reality technology which displays the virtual contents that give virtual instructions by co-operating with real objects. We convert traditional a text manual into more intuitive and understandable 3D virtual instructions. Moreover, we designed our unique 3d instructions by showing "where", "what" and "how" for every single step throughout the entire workflow of the system.

Technically, we designed an algorithm that converts raw component data into instructions automatically. This algorithm can realize the relationship of all components by analyzing the socket/plug data provide of each component and generate the instructions for our system. Other than that, we created functions such as automatic layout, mixed object recognition, articulated hands support and active/passive troubleshooting.

For our factory user, we made a user end that offers them an easier way to upload product details; For our individual user, we built an intuitive interaction system for daily assembly tasks. Overall, we think we have successfully accomplished these following requirements.

1. Intuitive compare with plain text;

2. Hands-free;

3. Interactive with the user;

4. Troubleshooting.

8.2 Future Work

Although we have proposed a system that changes assembling experience completely. There are still some problems that need to be mentioned.

The hardware problem is always an issue throughout this research. Due to the extremely narrow view of HoloLens Gen 1, it is nearly impossible to make the user see the Board, Depot and Workspace at the same time when sits near. Besides, the viewing distance is also a big issue. The default and recommended viewing distance for HoloLens Gen 1 is 0.85m, while our system requires the user to see a lot of virtual objects around the hand distance. Holographic display that near needs extra forced configuration and the result display view is not comfortable, sometimes even causing dizziness.

The current object recognition method also has some problems in hardware. Because the Vuforia APIs we are using in this system relies on the central camera of the HoloLens, which is a 2D camera with not that good quality (on 2019 standards). The result is not that perfect even after our work for improving it. It will be greatly improved if we can find a way to borrow some of the environment sensing depth camera to enhance this job.

The connection between HoloLens Gen 1 and the Leap Motion sensor also causing some issues. Because the MicroUSB socket on HoloLens Gen 1 does not support OTG protocol. That means we cannot make a direct wire connection between HoloLens and Leap Motion.

Fortunately, the incoming HoloLens Gen 2 seems to be greatly improved to all of these problems. Therefore, using HoloLens Gen 2 to continuing work for this topic maybe produce better results.

In the software point of view, we think some cooperation between multiple HoloLens user for one assembly job might be worth taking a deeper dig. Cooperation and distribute the job to multiple users is always a problem for assembling. There is some research showing the potential for building an assembly helping system for multiple user's cases. [23]

Furthermore, there is some research showing the potential for doing cross-platform cooperation that including HoloLens. [24] Maybe using a device such as a smartphone to interact with the view of HoloLens user could be useful in terms of teaching and guiding manually.

References

[1] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. In *IEICE Transactions on Information and Systems*, pages 1321–1329. IEICE, 1994.

[2] Oliver Thompson. Why is ikea furniture so hard to assemble? https://www.transformingthenation.com.au/why-is-ikea-furniture-so-hard-to-assemble/, 2018.

[3] SPACE10. Assembling ikea's new ar app, without a manual. https://medium.com/space10/assembling-ikeas-new-ar-app-without-a-manual-c74c09d0488d, 2017.

[4] Ankit Gupta, Dieter Fox, Brian Curless, and Michael Cohen. Duplotrack: A realtime system for authoring and guiding duplo block assembly. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 2012.

[5] Shuxing Du, Yanfeng Liu, and Wenyan Chen. Research on geometric modeling virtual assembly technology based on component-level assembly feature. In *AIVR 2019 Proceedings of the 2019 3rd International Conference on Artificial Intelligence and Virtual Reality*, pages 17–20. ACM, 2019.

[6] Dunn Oliver, Jeff Renton, and Sarsfield Aidan. Lego batman: graphical breakdown editing - optimising assembly workflow. In *DigiPro '17 Proceedings of the ACM SIGGRAPH Digital Production Symposium*. ACM, 2017.

[7] Yanning Xu, Xiangxu Meng, Weiwei Liu, and Hui Xiang. A collaborative virtual environment for real time assembly design. In *VRCIA '06 Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 373–376. ACM, 2006.

[8] Giles Westerfield, Antonija Mitrovic, and Mark Billinghurst. Intelligent augmented reality training for assembly tasks. Springer Berlin Heidelberg, 2013.

[9] Steven Johnson, Madeleine Gibson, and Bilge Mutlu. Handheld or handsfree?: Remote collaboration via lightweight head-mounted displays and handheld devices. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work Social Computing*. ACM, 2015.

[10] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the conference on Human factors in computing systems*. ACM, 2003.

[11] Merja Haveri and Jouka Mattila. Enhancing traveling experiences with augmented reality. In *MindTrek '09 Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, pages 215–215. ACM, 2009.

[12] Robert Xiao, Chris Harrison, and Scott E.Hudson. Worldkit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *CHI EA '13 CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pages 2889–2890. ACM, 2013.

[13] adam Pickard. Assemblear. https://adampickard.com/AssembleAR, 2018.

[14] eon reality. Ar assist. https://eonreality.com/platform/ar-assist/, 2018.

[15] Daniela Rosner and Jonathan Bean. Learning from ikea hacking: i'm not one to decoupage a tabletop and call it a day. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 419–422. ACM, 2009.

[16] Hao Dou. Spatial recognition based mixed-reality shop system providing responsive store layout. Master's thesis, Waseda University, Japan, 2019.

[17] Mikko Kuhna, Ida-Maria Kivelä, and Pirkko Oittinen. Semi-automated magazine layout using content-based image features. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 379–388. ACM, 2012.

[18] Luca Chittaro and Roberto Ranon. Dynamic generation of personalized vrml content: a general approach and its application to 3d e-commerce. In *Proceedings of the seventh international conference on 3D Web technology*, pages 145–154. ACM, 2002.

[19] Microsoft Research. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. 35(4), 2016.

[20] Mathieu Garon, Pierre-Olivier Boulet, Jean-Philippe Doironz, and Luc Beaulieu. Real-time high resolution 3d data on the hololens. In *IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2016.

[21] Juin-Ling Tseng. Development of a low-cost 3d interactive vr system using sbs 3d display, vr headset and finger posture motion tracking. In *INTERNATIONAL JOURNAL OF ADVANCED STUDIES IN COMPUTER SCIENCE AND ENGINEERING*. IJASCSE, 2016.

[22] Kenneth R Moser, Sujan Anreddy, and J. Edward Swan. Calibration and interaction in optical see-through augmented reality using leap motion. In *IEEE Virtual Reality (VR)*. IEEE, 2016.

[23] C Ailie Fraser, Tovi Grossman, and George Fitzmaurice. Webuild: Automatically distributing assembly tasks among collocated workers to improve coordination. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1817–1830. ACM, 2017.

[24] Shihui Xu. Augmented reality with shared augmented object and interaction between hmd and non-hmd users. Master's thesis, Waseda University, Japan, 2019.