Penerbit Akademia Baru

Journal of Advanced Research in Computing and Applications

Journal homepage: www.akademiabaru.com/arca.html
ISSN: 2462-1927

3D Object Manipulation Using Speech and Hand Gesture

T S Jou[1], M S H Salam[1,*] and A F Ahmad[1]

[1] Department of Aeronatical, Automotive and Offfshore Engineering, Fakulti Kejuruteraan Mekanikal, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia
[2] Department of Mechanical Engineering, Faculty of Engineering, Kano University of Science and Technology, Wudil, Nigeria

ARTICLE INFO

Article history:
Received 9 January 2023
Received in revised form 1 March 2023
Accepted 12 May 2023
Available online 10 June 2017

Keywords:
Augmented Reality, 3D object manipulation, gesture interaction, natural language processing

ABSTRACT

Natural user interface (NUI) is an interface that enables users to interact with the digital world in the same way they interact with the physical world through sensory input such as touch, speech, and gesture. The combination of multiple modalities for NUI has become the trend in user interface recently. There is significant progress in advancing speech and hand recognition technology, which makes both become effective input modalities in HCI. However, there are limitations exist that degrade the performance which includes the complexity of vocabulary and unnatural hand gestures to instruct the machine. Therefore, this project aims to develop an application with natural gesture and speech input for 3D object manipulation. Three phases have been carried out, first is data collection and analysis, second is application structure design, and the third is the implementation of speech and gesture in 3D object manipulation. This application is developed by using Leap Motion Controller for hand gesture tracking, and Microsoft Azure Speech Cognitive Service and Microsoft Azure Language Understanding Intelligence Service for natural language speech recognition. The evaluation has been performed based on the accuracy of the command recognition and usability and user acceptance. The results show that the approaches developed in this project able to make good recognition of the speech command and gesture interaction while user experience testing shows high level of satisfaction in the application.

1. Introduction

User interface (UI) is the visual part of computing devices where human-computer interaction (HCI) takes place. It is the environment that allows the users to communicate with computing devices or a machine. Traditionally, UI can be divided into the phase of Command-Line Interfaces (CLIs), followed by the phase of Graphical User Interface (GUI) [1]. Even though GUI has been successfully dominated for a long time, it is still not considered as the natural interface for users. Besides, as the devices have evolved in terms of physical size, a new way of more natural and intuitive interaction will be needed [2]. Thus, NUI has been introduced to pave the interaction way with computers more naturally and intuitively than ever before.

* *Corresponding author*
E-mail address: sah@utm.my (M S H Salam)

NUI refers to an interface that enables users to interact with the digital world in the same way users interact with the physical world, through sensory input such as touch, speech, and gesture [3]. The reason is due to its attractiveness in emulation of human daily "real-world" gestures that perfectly match the expectation on the way technology should work, for example, speaking with the device [4]. NUI stressing about the way of user interaction and the user experience. It utilized regular human behaviour and experience in making new technologies that are more intuitive and easier to learn due to the shorter learning curve [5]. Therefore, NUI can minimize the cognition overhear and allow the user to fully integrate into the device.

Research has suggested that a combination of modalities, which are speech and gesture, is preferred as it is more natural and reliable than speech or gesture alone [6]. Among all those modalities, gestures occur very frequently in human natural communication. Hansberger and his team have agreed that the performance for tasks like manipulating a 3D object, drawing, verbal task, has highly improved with an interface that supports multiple modalities [7].

However, research from Baig and his team had shown that there are still limitations exist that degrade the performance of speech-gesture recognition in 3D object manipulation [8]. The limitation includes the complexity of vocabulary used to instruct the machine. Besides, achieving high accuracy on gesture recognition causes some developers to overlook user preferences and design some gestures that are not naturally used in daily life.

Therefore, this work is another attempt to develop an application that supports hand gestures and speech input for 3D object manipulation. The motivation is to provide an intuitive environment that supports natural interaction between users and the digital world in 3D object manipulation. The remaining document is organized as follows: Section 2 is the background study of this work. Section 3 discusses the methodology of the application. In Section 4, the implementation of the application is discussed. Evaluation and results are presented in Section 5 and section 6 concludes the paper.

2. Methodology

An interface that mimics human-to-human interactions could reduce the learning time users required on the technology or application [9]. Among natural input modalities, the combination of speech input and hand gestures, which are abundant in everyday life, has been extensively used in recent applications. Gestures are beneficial for manipulating a virtual object directly, whereas speech input is useful for abstract tasks [8].

In everyday life, humans unwittingly produce hand gestures during communication with each other. Research has proved that humans even apply gestures when speaking on a phone or when there is no audience around [10]. The reason humans rely on gestures is that it is able to help in deciphering meaning. Thus, hand gestures have become one of the core fields to focus on in human-computer interaction for delivery commands and information.

There are a few hand gesture recognition technologies, which include glove-based, haptics, and sensor-based motion capture [8]. The first commercially home user available data gloves, Nintendo Power Glove, was introduced in 1989. Even though data gloves provide more accuracy in the interaction process compared to the normal static mouse and keyboard, research shows that it makes the user feel uncomfortable as it is cumbersome with a load of cables and no longer feels natural as in normal life. On the other hand, haptics is the technology of transmitting and understanding information through the sense of touch, without controllers or wearables. AIREAL, is a haptic technology that enables users to interact with the 3D object, experience free-air textures, and receive haptic feedback on the performed gestures [11]. Lastly, sensor-based gesture-recognition technology is used raw data from the sensor and manipulates the data to generate

position, orientation, and gesture type [12]. For instant, Microsoft Kinect, and Leap Motion Controller (LMC) are popular gesture recognition devices that implement this technique.

Speech is the primary and most original mode for humans to communicate with each other. Thus, speech is one of the natural modes that has been widely used as user input for the past few decades. The capability of identifying speech-language and converting it into respective written text in respective natural language is known as speech recognition [13]. Speech recognition has progressed immensely from only being able to recognize a few sets of sound to automatic speech recognition systems.

Nowadays, speech recognition has become one of the cutting-edge technologies. It allows the machine to understand a user's instruction or command by converting the human natural language into readable text through speech recognition. Speech recognition could be utilized in a range of advanced applications that might include machine learning, optimization, and so on. However, to achieve those advanced functions, the ability of the machine in understanding and converting the actual meaning of the input accurately is the foundation. Figure 1 shows the overview of the speech recognition process. First, the input speech signal will be analysed by Digital Speech Processing, and the features will be extracted. Then, matching of input with language model in the learning environment will be carried out, and the best fit will be returned as result output, which is in text format [13].

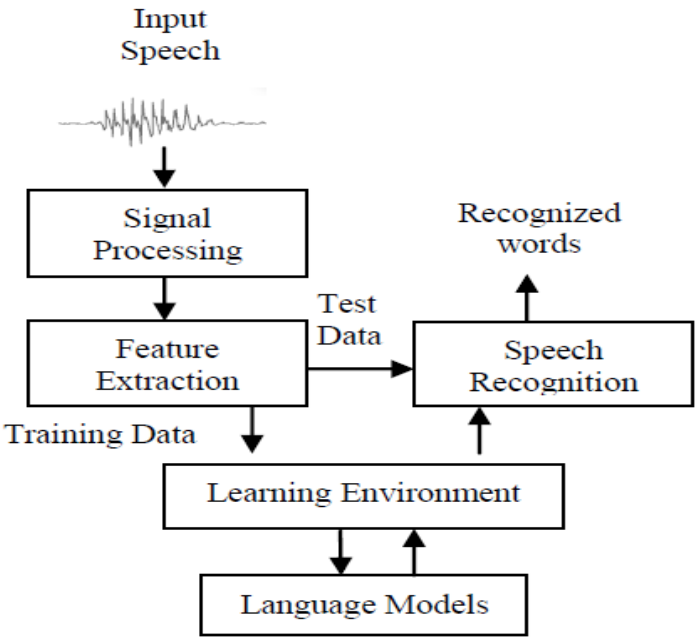


Fig. 1. Overview of Speech Recognition Process [12]

A number of research studies have been conducted to develop systems that interact using human behaviour and common language. The following are two examples related to the work. In 2018, Yongda and his team proposed a focus on human-robot interaction by using speech and gesture integration methods, as shown in Figure 2. This system utilizes Microsoft Speech SDK for speech recognition where the natural human language is transferred to execution instruction while Leap Motion Controller for gestures recognition [14].

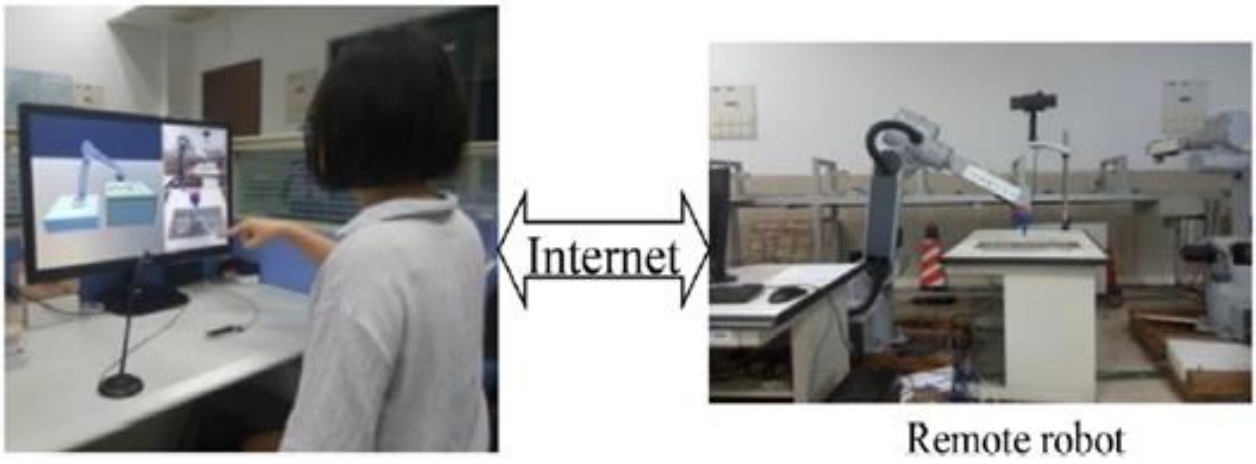


Fig. 2. UTM-LST delta wing VFE-2 profiles

Another example is "Paint that Object Yellow". It is an object manipulation system with multiple modalities interaction in a virtual reality environment as shown in Figure 3 [15]. It applied a typical VR visualization method with the use of head-tracked HMD and speech-gestures integration as input. Speech recognition in this system utilizes Microsoft Speech SDK. On the other hand, two oculus touch controllers are used for gesture control.

Fig. 3. Paint That Object Yellow [14]

Table 1 shows the comparison between the two applications.

Table 1
Comparison between existing applications

Application	Human-Robot Interaction	Paint That Object Yellow
Platform	GOOGOL GRB3016	Oculus Rift HMD
Environment	Physical World	Virtual Reality
Input Device, Technology	LMC, Microsoft Speech SDK	Oculus Rift HMD, Oculus Touch Controller, Microsoft Speech SDK
Manipulation	Translation, Placing Peg, metal blocks	Change Color, Texture Size
Speech Command Interaction	Yes	Yes
Hand Gestures Interaction	Yes	Yes

3. Multi-user Interaction

The project developed in this work follows four phases namely data collection and analysis, application design, implementation and testing.

In the first phase, analysis was made on works related to the project was conducted to have a better insight and understanding. The focus of this project includes speech recognition and hand gestures recognition in object manipulation. Besides, the related works similar to this project have been explored to have a better understanding and expectation about the output of the project and issues that may face in the future.

In the implementation phase, the Unity Game Engine is used as the platform for the integration of both speech and hand gesture recognition in 3D object manipulation. Then, the manipulation techniques for hand gestures have been developed. On the other hand, a new app has been created in LUIS to build and train a prediction model. After that, STT Services SDK has been set up in Unity game Engine.

In testing phase, evaluation about the use of speech-gesture, in object manipulation has been carried out to ensure that the project meets the requirements. Test cases will be conducted to assess the functionality of the entire application. This is done to verify that the application behaves as expected and meets its objectives.

3.1. Hand Gestures Design

For this application, LMC has been used for hand gesture tracking. The hand gesture recognition process is shown in Figure 4. Table 2 shows the types of gestures that have been designed for the user to interact with.

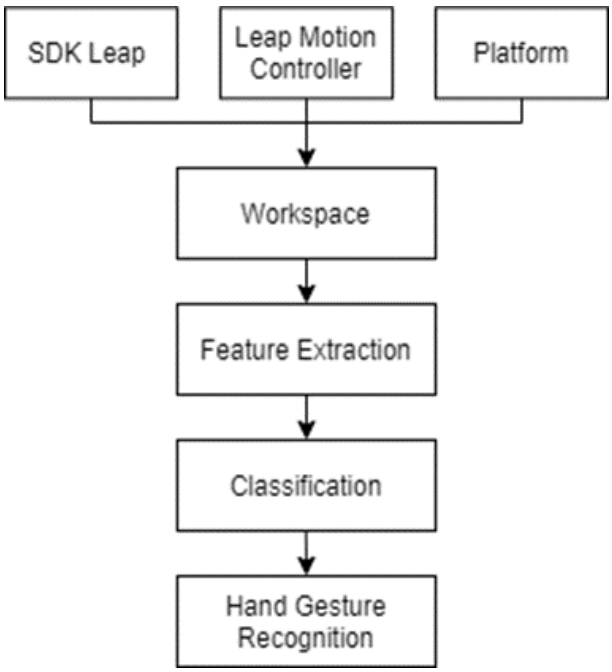


Fig. 4. Hand recognition process

Table 2
Types of events supported by hand gestures

Hand Gestures	Events
Left Palm Up	Menu Display
Left Hand Pinch	Zoom In/Out
Right hand Fist	Camera Navigate along X, Y-axis
Right Hand raised to around 90° / put down	Start/Stop Speech Command Control
Both Hand interact with object	Form Creation

3.2. Speech Command Design

The technology used for speech recognition in this application is STT and LUIS. The STT, also known as speech recognition, a service enables the machine to accurately transcribes up to 85 languages of spoken audio into written text in real-time.

The user needs to ensure that the microphone is enabled, and the speech-language setting of the Windows Operating System is set to English. Then, the user can perform the gesture to allow the application to listen to the command.

STT service has been used to convert user's speech command into text and send to the LUIS system for interpretation and prediction. After that, the LUIS system will return JSON responses, which contain the prediction for the command, as shown in Figure 5. The application will parse the JSON response, extract the useful information, which is the action that the user wants to perform, and then execute the respective action. Table 3 shows the events that supported by speech commands.

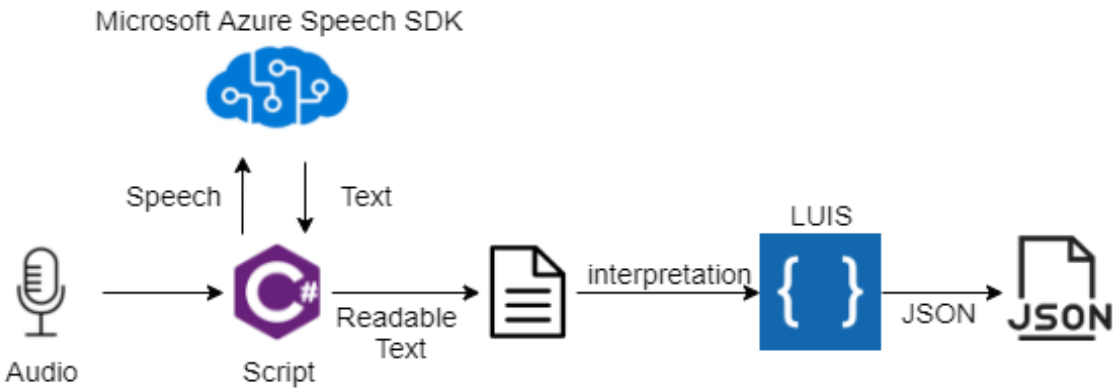


Fig. 5. Natural language speech recognition process

Table 3
Types of events supported by speech command

Intents	Events
PlaceClay	Placing the 3D object
Start Rotation	Start the rotation of the 3D object
Stop Rotation	Stop the rotation of the 3D object
ChangeTexture	Change the texture of the 3D object

3.3. User Interface Design

The user interface design for this application menu is an arc-shaped, hierarchical structure menu, which provided by Hover-UI-Kit library. It is a tool for creating a beautiful, customizable, and dynamic user interface.

3.4. Requirements and Specifications

Table 4 shows the software and hardware requirements in this project respectively

Table 4
Software and hardware requirements

Software	Hardware
Unity 3D Game Engine	X64 CPU Processor
Microsoft Visual Studio	Random Access Memory 4 GB
Leap Motion Unity SDK	NVIDIA GeForce GTX
Blender	Leap Motion Controller
Microsoft Azure Speech-to-Text Cognitive Service	Microphone
Microsoft Azure Language Understanding Intelligence Service	

4. Implementation

The main processes involved in this phase are the implementation and integration of both hand gesture and speech recognition in manipulate 3D object. The phase start with hand gesture implementation, followed by speech recognition implementation, and lastly integration. The setup and supported events are based on the guideline and designs discussed in previous section.

4.1. Hand Gestures Implementations

This section discusses the implementation of hand gesture interaction, which involved the setup of the LMC, and the events supported by hand gestures.

4.1.1. Hand gestures implementations

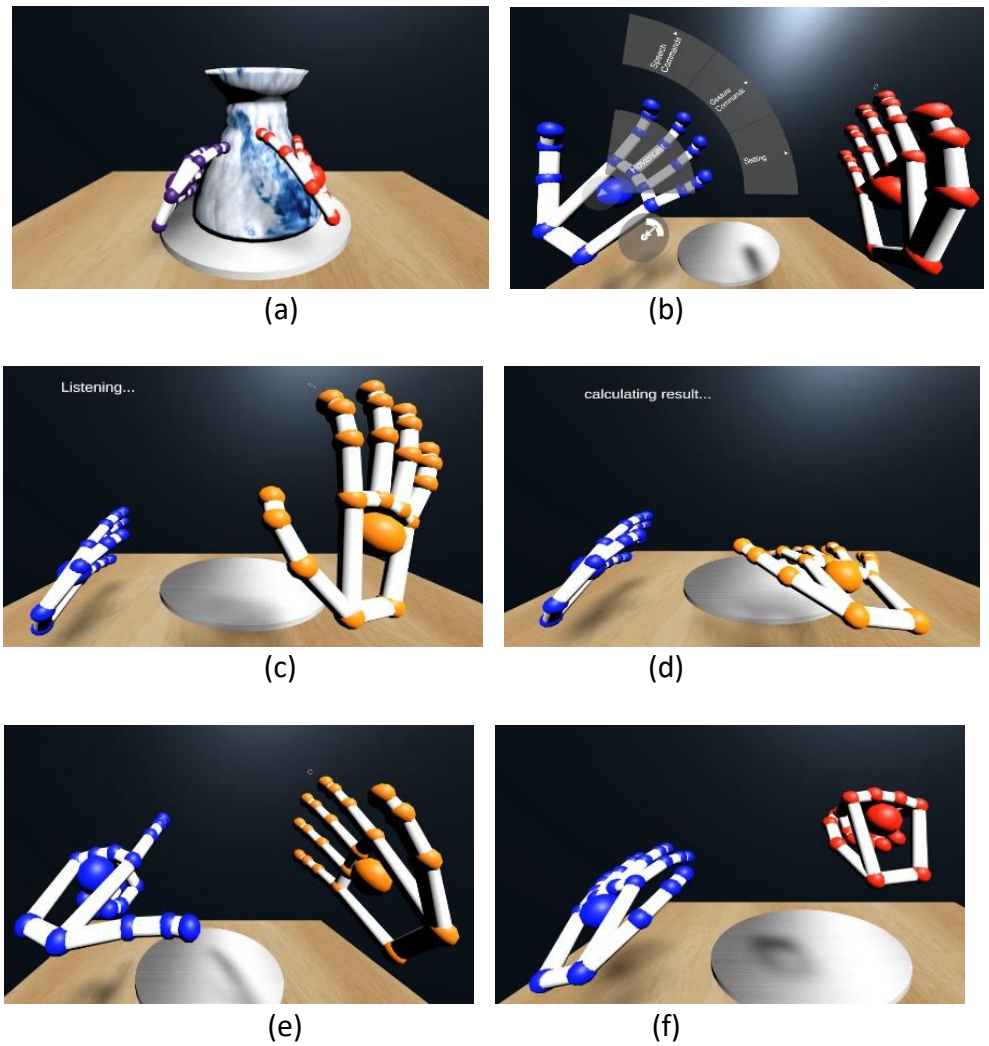
Ultraleap Hand Tracking Unity SDK V4 Orion and Unity Modules Package need to be downloaded and import into Unity. The Interaction Manager needs to be included in the application, as it is in charge of handling all the internal logic that makes interaction happen. Then, add the InteractionBehaviour component to the 3D object that will interact with.

4.1.2. Events supported by hand gestures

The events that supported by hand gestures are as follows:

Form Creation

The object is first created using Blende. Shape keys are added to the cylinder object (pottery object) during object modeling in Blender to enable deformation. Then, a simple cylinder (cylinder collider) is modeled using Blender, which will be used as collider for each Shape Key. Both 3D objects were imported into Unity.



(a) (b)

(c) (d)

(e) (f)

Fig. 6. Hand Gesture type

When the pottery object is imported, there is a SkinnedMeshRenderer component that contains the same blend shape keys created in Blender. Each blend shape key is linked with one-cylinder collider through the Blend Shape Index in respective script. Besides, the InteractionBehaviour Script has been added to each cylinder collider. When the Hand Model contacts each collider, a function will check on the interacted collider id, get the blend shape index and set a new blend shape weight to it, as shown in Figure 6 (a).

Menu Display

For the menu implementation, Hover-UI-Kit was used. Therefore, it has been customized to fit with the application. Customization is done in HoverInterface script component. Therefore, it has been customized to fit with the application The content of the menu including the demo guidance video to provide instruction for user, background music volume setting, restart game and quit game button, as shown in Figure 6 (b).

Speech Command Control

The gesture for Speech Command Control is to raise the right palm to around 90° to activate the speech event listener and put down the right palm when completing the command. Palm Direction Detector has been used to detect the direction of the right palm. By specifying the desired Hand Model, it will detect if the selected palm is pointing toward the specified direction. Since the gesture in controlling speech command is set to the right palm, the Right Hand Model is linked to the Hand Model properties. Besides, the detector will only activate when the palm direction is within the On Angle degrees of the desired direction (refer Figure 6 (c)) and deactivate when it becomes more than Off Angle degrees (refer Figure 6 (d)). Thus, when the detector is activated, the Speech Command Recorder will be hooked up, and vice versa.

Zoom In and Out

Since the gesture is designed on the left hand, the respective script component is added to the Left Hand Model. Besides, the Left Hand Model need is linked to the Hand Model property of the script. If the user performs the Zoom In gesture, the Field of View will decrease and vice versa. The distance between the tip position of the Index Finger and Thumb is used to determine if the user performs the Zoom In or Zoom Out gesture (refer Figure 6 (e)).

Camera Control

The hand gesture for controlling the camera movement is the right hand fist. Since the gesture is designed on the right hand, the respective script component is added to the Right Hand Model. Besides, the Right Hand Model need is linked to the Hand Model property of the script. When the Leap Motion device detects the performed gesture to be a right hand fist, the application will check the hand position to determine the direction of the camera movement, as shown in Figure 6(f). The application allowed the camera to move along the X and Y-axis.

4.2. Speech Recognition Implementations

This section discusses the implementation of speech recognition in the application. In this phase, there are 3 main processes, which include the setup of Speech-to-Text Recognition, Language Understanding Intelligence Service, and the events supported by speech command.

4.2.1. Speech-to-text recognition setup

First, Speech SDK needs to be installed and imported for Unity as an asset package. Then, it is required to subscribe to the service on the Microsoft account. After subscription, an endpoint and subscription key are provided, which is needed to connect with the service. A script is used to check if the Palm Direction Detector is activated. If it is activated, the application should start on recording audio, and convert audio into text. Once the conversion is finished, text will be sent to LUIS endpoint

4.2.2. LUIS setup

The process of creating a LUIS model is to define a model first. This step includes defining intents, entities. Besides intent and entity, utterances are also required to be added to each intent. Adding utterances to each intent is important as it assists and trains LUIS understanding. After that, train and retrain the model based on the experience. Then, publish the model and use it in the application.

First, LUIS application needs to be created in the LUIS portal and select Culture, which is the language that the application targets. Then, the endpoint URL and primary key to link to this API can be found at the Azure Resources tab. After that, 4 intents have been created, which include "PlaceClay", "StartRotation", "StopRotation", and "ChangeTexture". Each of the intents is associated with more than 15 utterances. Then, come to the process of training the LUIS application to understand natural language. This process is iterative until the result is satisfied. Next, publish the application to the endpoint.

From the previous section, the speech command provided by the user has successfully converted into text format. Then, the text data needs to be sent over to the LUIS API to calculate and make predictions on the intention of the command. After completing the prediction, LUIS API will return data in JSON format, and the Unity application needs to convert it into readable format. Therefore, a LUIS manager object with the respective script is created to perform all these actions.

There is one function is used to send the string command through web request to LUIS API, handle the response back from the site, and perform JSON to object conversion. The JSON data will then be deserialized into an object. After getting the prediction result, the application should check the intent and entity value.

There are 4 types of events that are supported by respective speech commands. The events including placing the pottery 3D object on the spinning wheel, start the rotation of the object, stop the rotation of the object, and texture selection for the object. Since the application is implemented LUIS in speech recognition, thus, there is no predetermined command to trigger those events.

5. Testing Results

This application only supports a single user. The user needs to make sure that the Leap Motion device is connected and functions properly. The device needs to be placed at a flat surface with the appropriate position, in which the camera is facing the user, as shown in Figure 7. Besides, the user needs to enable the microphone for speech input.

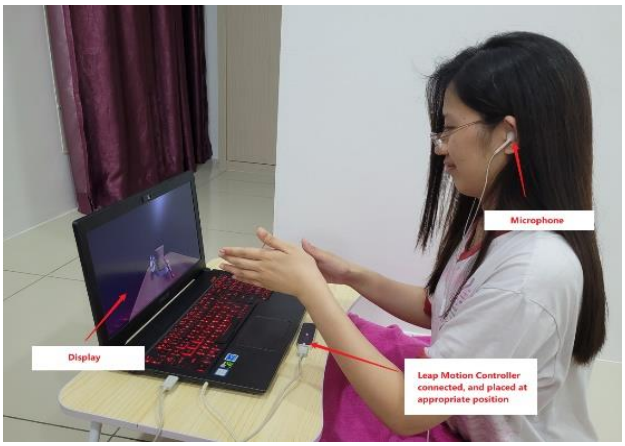


Fig. 7. Experimental setup

For evaluation, usability testing and black box testing has been carried out using questionnaires and video recordings. There are total of 10 participants involved in the testing, and the number of participants was referring to the experiment that has been carried out by Alibay and his team [16].

Pre-experiment and post-experiment questionnaires were distributed to testers in order to collect their background, experience, feedback, and suggestions for this application. Then, each participant is given fifteen minutes to explore and familiarize with the application individually before starting on the experiment, and another fifteen minutes to complete the tasks.

From pre-experiment questionnaire result, only 60% of testers has experience in using LMC, while 50% of testers has experience with using speech commands in interacting with digital content.

From the post-experiment questionnaire result, up to 70% of testers strongly agree that the instruction provided in user menu is clear and easy to interact with. This shows that most of the testers are satisfied and do not face difficulty in interacting with the user interface. For hand gestures recognition part, up to 60% of the testers strongly agree that the application can recognize the predetermined gestures easily, and the gestures are intuitive. While 40% of the testers agree with that statement. For speech recognition, up to 70% testers strongly agree that the application can respond to the command in a short time, understand the speech command in natural language and perform the respective action. This shows that most of the testers can interact with the application using speech commands in natural language, and the application able to respond with the respective events.

As for black box testing, the average of recognition for all 15 gestures interaction recognition achieved 92%. While for a total of 80 speech command phrases recognition rate is 83%. Overall, the recognition accuracy for both gestures and speech command are within satisfaction of the tested users.

6. Conclusions

This paper presented an application that support hand gesture and speech recognition in 3D object manipulation in virtual environment. The testing results also shown that the intuitive hand gestures interaction, and natural language recognition, speech and gesture can become well-coordinate input in HCI.

Even though the results also indicate that some of the testers required some time to get used on interaction using mid-air gestures, however, after some time of practice, most of the testers still able to perform well and interact naturally with the computer. Hence, it is suggested for future works that the commands can be more intuitive and intelligent without having specifically set the speech phrases or gesture command which will help in shorten user learning curve.

References

[1] Wigdor, Daniel, and Dennis Wixon. *Brave NUI world: designing natural user interfaces for touch and gesture*. Elsevier, 2011. https://doi.org/10.1016/B978-0-12-382231-4.00002-2

[2] Fu, Limin Paul, James Landay, Michael Nebeling, Yingqing Xu, and Chen Zhao. "Redefining natural user interface." In *Extended abstracts of the 2018 CHI conference on human factors in computing systems*, pp. 1-3. 2018. https://doi.org/10.1145/3170427.3190649

[3] Kurniawan, Sri. "Interaction design: Beyond human–computer interaction by Preece, Sharp and Rogers (2001), ISBN 0471492787." *Universal Access in the Information Society* 3 (2004): 289-289. https://doi.org/10.1007/s10209-004-0102-1

[4] Kaushik, Dr Manju, and Rashmi Jain. "Natural user interfaces: Trend in virtual interaction." *arXiv preprint arXiv:1405.0101* (2014).

[5] Falcao, Christianne, Ana Catarina Lemos, and Marcelo Soares. "Evaluation of natural user interface: a usability study based on the leap motion device." *Procedia Manufacturing* 3 (2015): 5490-5495. https://doi.org/10.1016/j.promfg.2015.07.697

[6] Martinez, Jude Joseph Lamug, and Sindy Senorita Dewanti. "Multimodal Interfaces: A Study on Speech-Hand Gesture Recognition." In *2019 International Conference on Information and Communications Technology (ICOIACT)*, pp. 196-200. IEEE, 2019.

[7] Hansberger, Jeffrey T., Chao Peng, Victoria Blakely, Sarah Meacham, Lizhou Cao, and Nicholas Diliberti. "A multimodal interface for virtual information environments." In *Virtual, Augmented and Mixed Reality. Multimodal Interaction: 11th International Conference, VAMR 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part I 21*, pp. 59-70. Springer International Publishing, 2019. https://doi.org/10.1007/978-3-030-21607-8_5

[8] Baig, Muhammad Zeeshan, and Manolya Kavakli. "Qualitative analysis of a multimodal interface system using speech/gesture." In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2811-2816. IEEE, 2018. https://doi.org/10.1109/ICIEA.2018.8398188

[9] Williams, Adam S., Jason Garcia, and Francisco Ortega. "Understanding multimodal user gesture and speech behavior for object manipulation in augmented reality using elicitation." *IEEE Transactions on Visualization and Computer Graphics* 26, no. 12 (2020): 3479-3489. https://doi.org/10.1109/TVCG.2020.3023566

[10] Clough, Sharice, and Melissa C. Duff. "The role of gesture in communication and cognition: Implications for understanding and treating neurogenic communication disorders." *Frontiers in Human Neuroscience* 14 (2020): 323. https://doi.org/10.3389/fnhum.2020.00323

[11] Sodhi, Rajinder, Ivan Poupyrev, Matthew Glisson, and Ali Israr. "AIREAL: interactive tactile experiences in free air." *ACM Transactions on Graphics (TOG)* 32, no. 4 (2013): 1-10. https://doi.org/10.1145/2461912.2462007

[12] Ismail, Ajune Wanis, Mohamad Yahya Fekri Aladin, and Muhammad Nur Affendy Nor'a. "Real Hand Gesture in Augmented Reality Drawing with Markerless Tracking on Mobile." *International Journal of Computing and Digital Systems* 12, no. 1 (2022): 1071-1080. https://doi.org/10.12785/ijcds/120186

[13] Panda, Soumya Priyadarsini. "Automated speech recognition system in advancement of human-computer interaction." In *2017 International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 302-306. IEEE, 2017. https://doi.org/10.1109/ICCMC.2017.8282696

[14] Yongda, Deng, Li Fang, and Xin Huang. "Research on multimodal human-robot interaction based on speech and gesture." *Computers & Electrical Engineering* 72 (2018): 443-454. https://doi.org/10.1016/j.compeleceng.2018.09.014

[15] Wolf, Erik, Sara Klüber, Chris Zimmerer, Jean-Luc Lugrin, and Marc Erich Latoschik. "" Paint that object yellow": Multimodal Interaction to Enhance Creativity During Design Tasks in VR." In *2019 International Conference on Multimodal Interaction*, pp. 195-204. 2019. https://doi.org/10.1145/3340555.3353724

[16] Alibay, Farzana, Manolya Kavakli, Jean-Rémy Chardonnet, and Muhammad Zeeshan Baig. "The usability of speech and/or gestures in multi-modal interface systems." In *Proceedings of the 9th international conference on computer and automation engineering*, pp. 73-77. 2017. https://doi.org/10.1145/3057039.3057089