



# HoloGCS: mixed reality-based ground control station for unmanned aerial vehicle

Daniar Estu Widiyanti<sup>1</sup> · Krisma Asmoro<sup>1</sup> · Soo Young Shin<sup>1</sup>

Received: 28 February 2022 / Accepted: 9 November 2023 / Published online: 9 February 2024  
© The Author(s) 2024

## Abstract

Human–robot interaction (HRI), which studies the interaction between robots and humans, appears as a promising research idea for the future of smart factories. In this study, HoloLens as ground control station (HoloGCS) is implemented, and its performance is discussed. HoloGCS is a mixed reality-based system for controlling and monitoring unmanned aerial vehicles (UAV). The system incorporates HRI through speech commands and video streaming, enabling UAV teleoperation. HoloGCS provides a user interface that allows operators to monitor and control the UAV easily. To demonstrate the feasibility of the proposed systems, a user case study (user testing and SUS-based questionnaire) was performed to gather qualitative results. In addition, throughput, RTT, latency, and speech accuracy were also gathered and analyzed to evaluate quantitative results.

**Keywords** Mixed reality · Unmanned aerial vehicle · Ground control station · Speech control · Video streaming · Microsoft HoloLens · Human–robot interactions

## 1 Introduction

The 4.0 industrial revolution has introduced a paradigm shift toward a smart factory that utilizes a new approach to achieve mass scale and cost-effective production (Zuehlke 2010). Smart factories depend on the auto-decision-making that is performed over a massive interconnected device; thus, artificial intelligence (AI) is utilized to aid it (Lee et al. 2018). Moreover, to enhance massive production, a robot is deployed to lower defective production rates (Wan et al. 2018) (Table 1).

In Luque Vega et al. (2020), an unmanned aerial vehicle (UAV) as a mobile robot in smart factories has gained

huge popularity owing to its capabilities to hover and allow 3-dimensional movement. Therefore, it can minimize delays occurring in smart factories. Although UAVs can work autonomously in a smart factory, an operator must monitor and observe to take over control if it is misbehaving. Currently, controlling UAVs using the conventional remote control (RC) which potentially causes a problem because of its complexity for first-time-user (Graff 2016). As a result, a diverse approach regarding the study to replace conventional RC. Furthermore, in Fayjie et al. (2017); Landau and van Delden (2017) voice-based and gesture control system was proposed to aid conventional RC. Moreover, in Bokade and Ratnaparkhe (2016) developed a web-page-based application integrated with an android operation system to jointly control and stream live video from UAV.

The problem arises when human operators have to monitor smart factories because of a lot of interconnected machines and robots. Recently, augmented reality (AR) and mixed reality (MR) as head-mounted displays (HMDs) with intuitive interactions are the most likely strong candidate devices in the future to aid humans in smart factories (Gatullo et al. 2019). Several portable devices that exploit the MR principle have been developed to enhance the ease of human–robot interaction (HRI), which are conventional smart glasses (Sutar and Pawar 2021) and Microsoft HoloLens. HoloLens has numerous advantages compared to

---

Daniar Estu Widiyanti, Krisma Asmoro and Soo Young Shin have contributed equally to this work.

✉ Soo Young Shin  
wdragon@kumoh.ac.kr

Daniar Estu Widiyanti  
daniarestuw@kumoh.ac.kr

Krisma Asmoro  
krisma@kumoh.ac.kr

<sup>1</sup> Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, Gyeongsangbuk, South Korea

**Table 1** Previous work comparison

Reference	Device	Vid. Stream	Controller type		
			RC/ Joystick	GUI	Speech
(Gong et al. 2017)	Moverio BT-200	✓	✓	x	x
(Jeong et al. 2018)	HMD	✓	x	x	x
(Hedayati et al. 2018)	HoloLens1	✓	x	x	x
(Kot et al. 2018)	HoloLens1	✓	✓	x	x
(Widiyanti and Shin 2021)	HoloLens1	✓	✓	x	x
Proposed	HoloLens2	✓	x	✓	✓

smart glass. The device provides a broader field of view (FoV), better battery lifespan, and faster clock speed of the central processing unit (CPU). HoloLens is capable of recognizing and processing human interaction inputs (gaze, gesture, voice) (Microsoft HoloLens 2021). The implementation of HoloLens to assist HRI has been done before, such as gesture tracking with precise and fast fingertip tracking to get a two dimension position (x,y coordinates) of a mobile robot and mapping it to a 3D space to mark the location of the robot (Sun et al. 2019).

Motivated by UAV controlling and HRI through HoloLens, an MR-based GCS is proposed in this study. Without loss of generality, this paper utilized a robot operating system (ROS), which can be applied to various types of UAVs. In addition, this paper also designs a user interface (UI) for a GCS application that contains three key features that can run simultaneously, namely: video stream, speech recognition control, and manual control using the designed UI. Furthermore, the contributions of this study can be summarized as follows:

- A speech command system in HMD for controlling UAV. In this study, the UAV commands are as follows: take off; land; bank left or right; pitch forward or backward; yaw left or right; ascend; descent.
- Design a video streaming system that forwards the camera attached to the UAV to HMD.
- Design a UI to control UAV manually with similar commands with speech recognition.

In addition, throughput, round trip time (RTT), latency (Zhao et al. 2018), and user study were evaluated to benchmark the proposed system (Steinfeld et al. 2006). Moreover, we consider speech accuracy as a performance evaluation of the reliability of MR-based GCS.

This study is structured as follows. Section 2 examines previous research on the current remote controllers and video streaming methods. Section 3 focuses on the details of the proposed system. Section 4 explains the implementation process in detail. Section 5 discusses the measured throughput, round-trip time, latency, speech accuracy, and

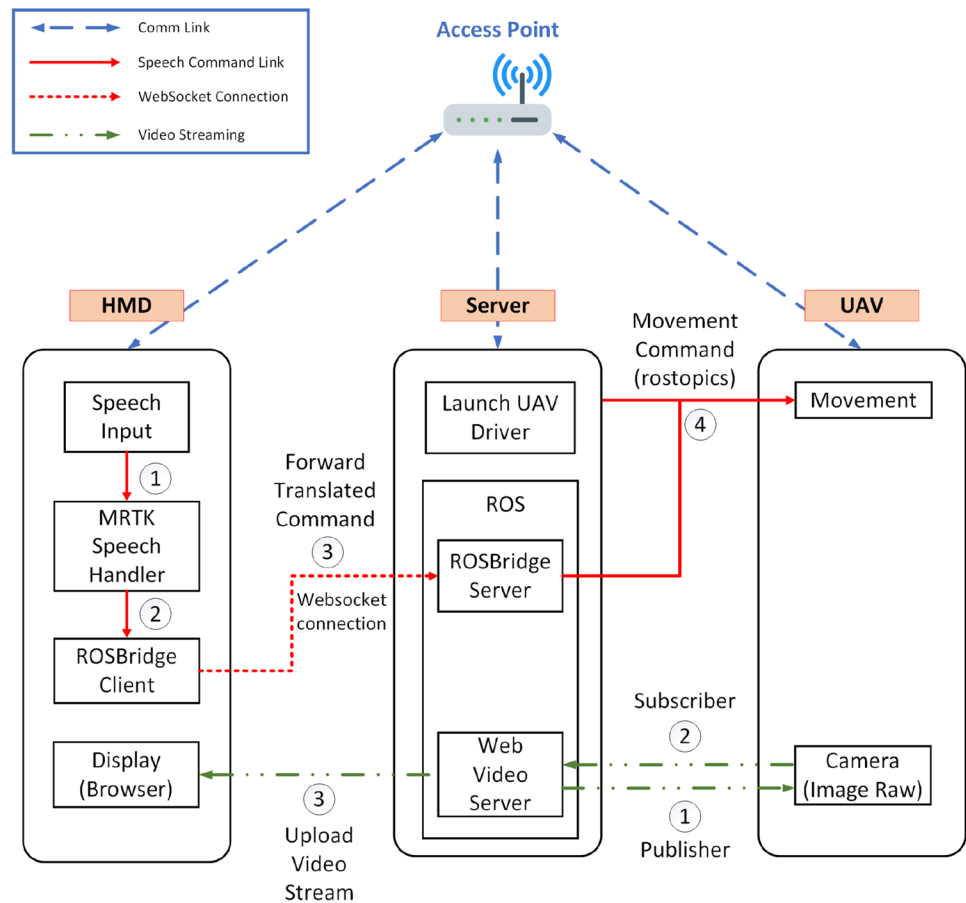
user study details. Finally, Sect. 6 concludes the conclusion of this study.

## 2 Related work

Initial work of controlling a robot via HMD has been done in Stoker et al. (1995), where humans use teleoperation to control a robot submarine or unmanned underwater vehicle (UUV). In Higuchi et al. (2013), a VR device was integrated to control the movement of a UAV. In Bennett et al. (2017), design an exo suit to control the humanoid robot and stream video from a camera mounted on the robot. Furthermore, in Bosch et al. (2022), the authors proposed a VR-based system to stream video from an omnidirectional camera mounted on UUV. In addition, an external device was utilized to control the UUV. In Ibrahimov et al. (2019), they designed an HMD application to simultaneously control UAV movement and wearable-controlled teleportation to pick up the object that UAV carried out.

Numerous studies to develop VR/AR-based GCS have been done before. In Kot and Novák (2014) proposed a system to control a robot using a VR device, namely Oculus Rift. In García et al. (2017) proposed a VR system to precisely control UUV movement with an external joystick. In (Lipton et al. 2018) proposed a system to control manufacturing robots using a VR device. Furthermore, Whitney et al. (2020) proposed a VR-based system to control a robot to grasp an object from several miles away. Furthermore, to reduce system complexity, Krupke et al. (2018) designed an AR-based UI to combine hand gestures and speech commands to control a virtual robot arm. Herrmann and Schmidt (2018) proposed a natural user interface with gesture or speech commands to control a UAV without real-time video streaming and GUI. Natural language-based control for the UAV was proposed by Huang et al. (2019). In Chuixin and Xinzheng (2021), combined speech recognition with a sonar sensor was proposed to develop a car control system with obstacle avoidance. In (Zhou et al. 2020) proposed a voice control Ground Control Station (GCS) to reduce the complexity of a UAV mission execution.

Fig. 1 Proposed method



Several studies have shown that it is possible to stream feed to AR devices. In Hedayati et al. (2018) proposed an AR-based system to stream the robot's video feed to monitor its movement. In Gong et al. (2017), design an AR system to display video from a Raspberry Pi camera mounted on a humanoid robot. Finally, Kot et al. (2018) proposed a system to view robot perspectives using HoloLens with a separated joystick to control robot movement. In Hoppenstedt et al. (2019) proposed an (MR) based system to visualize the flight data of UAV that is controlled directly by the ROS platform.

### 3 Proposed system architecture

The architecture of the whole proposed system is summarized in Fig. 1, consisting of an intractable MR application that fully controls and accesses the view from the UAV. In particular, two systems have been made; a video streaming system, where the view from the UAV can be visualized on the HoloLens, and a UAV speech control system, where the HoloLens user can perform a speech control to the UAV. The proposed system purpose is to enable the HoloLens user to observe the UAV view from video streaming and make

decisions based on the streaming to control the UAV using a speech command controller.

To coordinate the whole system, three devices are included in the process. Devices that are used are a Server, a Microsoft HoloLens, and a Parrot Bebop 2 as the clients. All the devices are connected to the same Wi-Fi connection as an access point, where all the devices are accessing the same access point. The detailed information of every system is shown in Sects. 3.1 and 3.2.

#### 3.1 Video streaming system

The details about video streaming are explained in this section. HoloLens, as a client, should access a specific Bebop streaming IP address in order to obtain the UAV view. Since both HoloLens and the Bebop are clients that need to exchange data, they must be in the same connection. The details of the connection are explained in Sect. 4.

Once both clients are connected to each other and the server has previously determined the IP access for video streaming by running a `web_video_server`, video streaming can proceed. While the `web_video_server` is fully operated, it will request raw image data from the Bebop 2. After ROS subscribes to the Bebop 2 image raw

data info and forwards it to the `web_video_server`, the `web_video_server` will forward the video streaming that will be accessible by the HoloLens when the user accesses the determined IP Address from the browser of the HoloLens. After getting access to the UAV perspective, the HoloLens could confirm the UAV steering direction.

### 3.2 Speech control system

The details of the speech command system are explained in this section. It starts when the speech command application on the HoloLens is launched. By communicating with the server, ROSBridge-Client is bridged to the ROSBridge-Server. As soon as both connections are established, the server must launch the Bebop driver, at the same time, the movement script must be translated into a rostopic in order for the Bebop 2 to move in compliance with the assigned rostopic. To translate the command to a rostopic the process starts when the HoloLens user delivers a designed speech command through the HoloLens. The MRTK “Speech Handler” package starts and recognizes the command. The successfully readable command then will be processed in NPM, which in NPM the detected speech command will translate as a nodeJS format. However, if the command is not suitable with the designed speech, then the system will ignore the command.

Since it is connected to the server side via the ROSBridge-Server, the system will proceed by sending the command to the ROSBridge-Client after it has been converted to nodeJS format. When the ROSBridge-Server receives data, it must also operate the Bebop driver node in order to connect to the Bebop 2 as a client. When a connection is established between the server and the Bebop 2, the data on the ROSBridge-Server are processed by transforming the data into a rostopic. The rostopic will then be accessed, and the bebop will move in accordance with the rostopic that has been selected. The list of rostopics and commands is shown in Table 2.

## 4 Implementation

Once the system analysis has been completed and all proposed diagrams have been created, this section will go through the implementation component in depth. The implementation will be conducted out using a HoloLens, a Server, and a Bebop 2. The HoloLens will emphasize how control is optimized and how streaming from the Bebop 2 is displayed. The middleware between HoloLens and Bebop 2 will be the server. Bebop 2 will be the client that can move in response to the movement given by the HoloLens.

Figure 2 generally explains the implementation of this proposed system. The main platform for the implementation

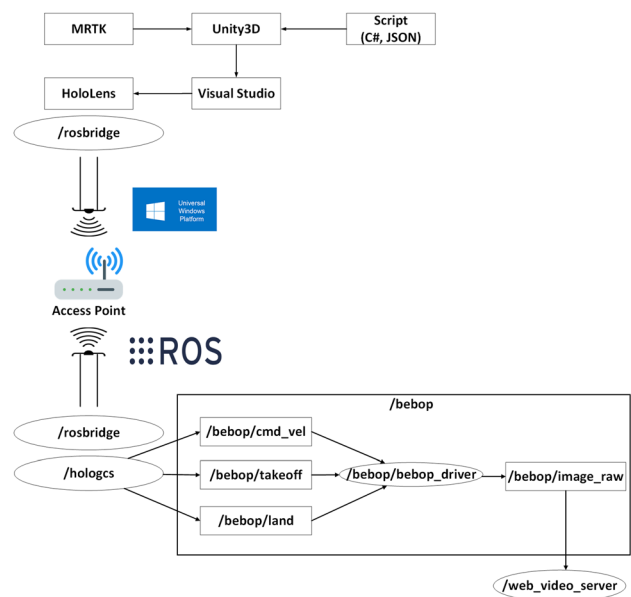


Fig. 2 Development diagram

Table 2 ROS topics and command list

Movement	Speech	ROS Topics	X,Y,Z,(Angular) Axis
Take off	Take off	Bebop/takeoff	–
Land	Land	Bebop/land	–
Pitch forward	Front	Bebop/twist	0.3, 0, 0
Pitch backward	Back	Bebop/twist	–0.3, 0, 0
Left	Left	Bebop/twist	0, 0.3, 0
Right	Right	Bebop/twist	0, –0.3, 0
Rotate left	Rotate Left	Bebop/twist	0, 0, 0, (–0.3)
Rotate right	Rotate right	Bebop/twist	0, 0, 0, (0.3)
Increase altitude	Up	Bebop/twist	0, 0, 0.3
Decrease altitude	Down	Bebop/twist	0, 0, –0.3

is the Universal Windows Platform (UWP) and Robot Operating System (ROS). Each platform is explained in structure as follows. Section 4.1 will explain the HoloLens development software, Sect. 4.2 will explain the server and UAV development kit, Sect. 4.3 will explain the connection and communication between each different platform, and Sect. 4.4 explain the user interface design using MRTK.

### 4.1 HoloLens development software

For developing the whole system on the HoloLens, development software is needed. Unity3D (Uni 2021) is a popular development tool for creating HoloLens applications. Unity3D is used on the system since the proposed system is an MR-based system under the Windows platform, which

use C# as the scripting language. This scripting combines with .NET libraries to support XML files, databases, and networking. Scripting is frequently considered limited and slow, but scripts in Unity3D are compiled to native code and run nearly as fast as C++. The ease of use is ideal for scripting languages; additionally, these languages can be mixed and matched.

Additional HMD development tools are used in this paper. These tools are used to handle some specific purposes while developing the application. In this paper, the tools are used to develop the speech input handler and UI handler before the application is deployed on the HoloLens device. The tool is called Mixed Reality Tool-Kit (MRTK) (Jerald et al. 2014). The MRTK is a library of scripts and components that developers may use to build applications for HoloLens and other Windows-based headsets. This subsection will only explain the use of the MRTK speech input handler, the MRTK UI handler will be discussed in the next subsection. The MRTK speech input handler handles the HoloLens user input as a voice command and translates it as data that can be forwarded to the server and could get the input data from the HoloLens user.

## 4.2 Server and Bebop 2 development kit

The ROS has gained a lot of popularity for the development of robotic systems. ROS is an open-source flexible system to operate a robot (ROS 2021a). It includes a set of tools, a library, and packs that help to simplify the operation of a robust robot. Controlling code in robot operating systems can also be written in a more abstract language, allowing it to be easily adapted to a different controlling microprocessor. ROS can adapt to many kinds of robots and uncrewed vehicles, such as Unmanned aerial vehicles (UAV), Unmanned ground vehicles (UGV), Unmanned surface vehicles (USV), Unmanned underwater vehicles (UUV), and more. ROS is utilized in the system as it is simple to use, adaptive, easy to install and maintain, and it can be modified to meet the needs of the system.

Some ROS resources that must be accessed for the proposed system are the ROS Package and messages types. For processing data from the server to Bebop 2 client, rosmaster, rosnodes, rostopics, rosmessages are required. For implementation in a video streaming system, rostopics called `web_video_server` is accessed. The topics will manage the video subscribing and publishing from Bebop 2 to the server and vice versa. For implementation in the control system, accessed rostopics depend on the control. The detail of the control can be seen in Table 2, while the X, Y, Z, and Z (angular) axis in the table explains the movement length set for UAV to move within a command at a time. To highlight, the Take Off and Land commands do not require any axis setup in order to move. Based on the

table, to make the Bebop take off a `bebop/takeoff` topic is accessed. For making the bebop land, a `bebop/land` is accessed. Meanwhile for Bebop 2 movement `bebop/cmd_vel` is accessed. These three topics have their own rosmessages, while rosmessages is a data structure to communicate between rosnodes.

## 4.3 Connectors and communication

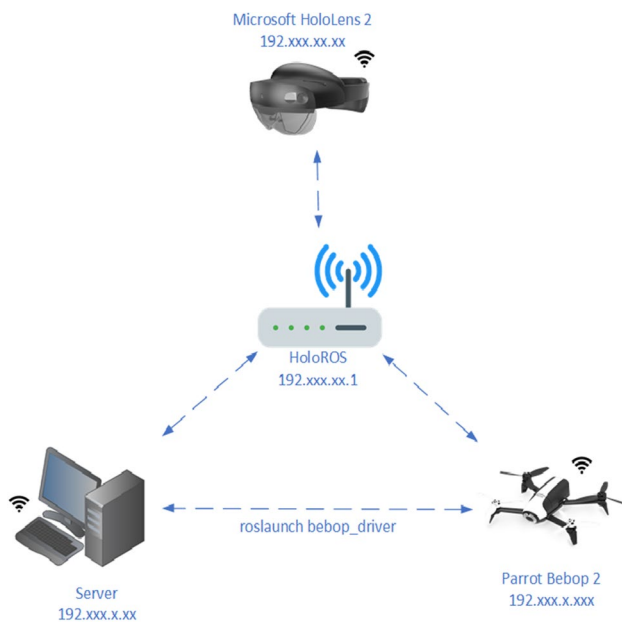
This connectors and communication implementation section are focusing on the ROS and Unity3D connectors and network communication. Connectors are a crucial component of the implementation, and this section focuses on the implementation of connecting the application inside the client device HoloLens, with the server as a rosmaster to deliver commands to the UAV. ROSBridge is a middleware to communicate ROS-based programs and non-ROS-based programs. ROSbridge consists of a JSON-based API to access ROS functionality which makes the non-ROS-based programs communicate and interact with ROS-based programs.

ROSBridge allows the ROS system to connect to other systems using WebSocket connection (ROS 2021b). ROSBridge operates in a single TCP/IP socket using WebSocket that is able to support upstream and downstream communications over a single connection. ROSBridge is divided into two device needs: ROSBridge-Server and ROSBridge-Client. The ROSBridge-Client and the ROSBridge-Server are set up to be connected in the Unity3D application before the application is deployed on the HoloLens, and the ROSBridge-Server is installed on the server.

When creating an implementation-based system with several clients, the communication component must be considered; features of the communication link are shown in Fig. 3. In this work, an internet connection is not needed. But, Wi-Fi as an access point connection is used to connect all the devices. After HoloLens and the server are connected to the same access point, Bebop 2 has to connect to the preferred access point. In this case, IP pushing toward the bebop is needed as generally, bebop has its own access point.

## 4.4 User interface

The user interface is the visual representation of the application that the user sees. User interface (UI) design has increased in popularity over the years, becoming one of the most creative, innovative, and fascinating fields in technological innovations. The UI was created using the Unity3D 2019.4.12f1 version in collaboration with the MRTK which has been explained in the previous subsection. The UI is divided into Video streaming UI and Speech Command & Control UI. General UI can be seen in Fig. 4. Video streaming UI can be seen in Fig. 4a and b. Speech Command & Control UI can be seen in Fig. 4c.



**Fig. 3** Communication link

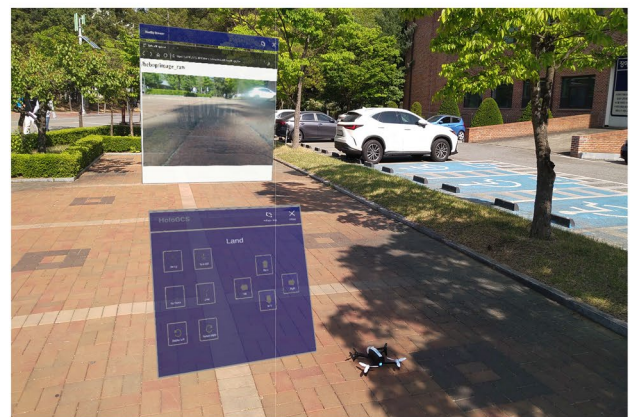
In the proposed system, the UI is made by holograms created by using MRTK as it is used as the primary speech input handler as well as the UI handler. MRTK UI handler combined with spatial mapping, gaze handlers, and gesture handlers is used to construct the UI. The controller interface is automatically identified in the users real environment through spatial mapping. All scripting is handled in Visual Studio 2019 prior to the implementation of the GUI system. The script can then be assigned to the button in Unity3D. The system could be built and deployed on HoloLens once all of the system components have been created. The deployment procedure in Visual Studio 2019 must be completed before it can be shown on the HoloLens device as an application. After it is deployed to the HoloLens when users operate the application, the user can enlarge or reduce the UI display size according to their preferences.

### 5 Experiment and result

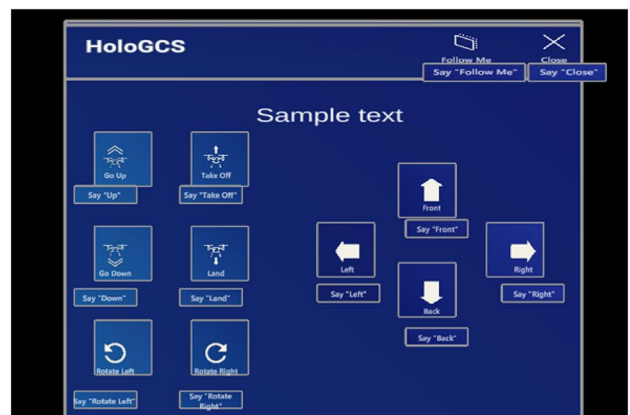
The experiment of the development of MR as a GCS for UAV has been conducted. The experiment calculation uses TCP protocol to calculate throughput, round-trip time (RTT), and latency for Video Streaming (VS) and combined video streaming and command and control systems (VSC &C). To highlight, as the video streaming is uploaded in a Hypertext Transfer Protocol (HTTP) from the server, the HoloLens access it as a TCP protocol instead of UDP. A speech accuracy analysis was also measured to assess the users speech capabilities using the application. Experiment for Command and Control (C &C)



(a)



(b)



(c)

**Fig. 4** a HoloLens Video streaming and control view after UAV takes off, b HoloLens Video streaming and control view when UAV lands, and c Speech Command & Control User Interface

only will not be counted since the protocol utilized is UDP. To conclude, a user study was also conducted to show how the system can be utilized by the participants.

For measuring the throughput, round-time trip, and latency, a total of 20 trials have been done for each performance metric. In this experiment, the distance is set as follows: HoloLens 2 at (0, 0) meters; UAV at (0, 40) meters as a starting point; AP in the middle at (0, 20) meters. The proposed system was tested with a set of tasks in Table 2, namely: the Take Off; Front; Right; Left; Back; and Land, each one command making a 0.3-meter movement. The evaluation for these metrics is measured in two categories: the first is VS-only performance, and the second is combined VS and C & C performance.

Firstly, for VS-only performance evaluation, users were given a single command to the UAV within a 5-second interval period. Secondly, for the combined VS and C & C performance evaluation procedure (VSC & C), users were given a single command to the UAV with a 15-second interval period and streamed a video simultaneously. Lastly, to note, for counting the latency for VS and VSC & C, servers are used to count the data transfer transmission. When the user gives a single command to the UAV, the server will check the latency time toward both the UAV and the HoloLens.

### 5.1 Throughput

Throughput is the average rate at which data are obtained by a node over a given measurement interval. Throughput measurement is conducted in VS only and VSC & C scenario. Equation 1 is used to measure the system throughput in both cases.

$$\chi = \frac{N^{rx}}{t} \tag{1}$$

Where  $N^{rx}$  is the received packets for each node and  $t$  is the time(s) that is received from the total packets trials.

The first scenario for throughput performance is the VS system. The trials have been done in 20 trials. Every  $i^{th}$  trial is conducted with a total of 125 s of simultaneous streams for VS and 600 s for VSC & C. Figure 5 shows the throughput sampling for VS, and Fig. 6 shows the throughput sampling for VSC & C. There is a possible saturation throughput or exhausted system resource (Device CPU utilization) which causes the decrease in 110 s for VS and 590 s for VSC & C.

### 5.2 Round-time trip (RTT)

The round-time trip is the time it takes for a signal to be transmitted plus the time it takes for that signal to be acknowledged as having been received. RTT measurement combined both servers to HoloLens and HoloLens-to-server communication. RTT measurement is conducted in both VS and VSC & C scenarios. Every  $i^{th}$  trial is conducted with a total of 250 s of simultaneous streams for VS and 600 s for VSC & C.

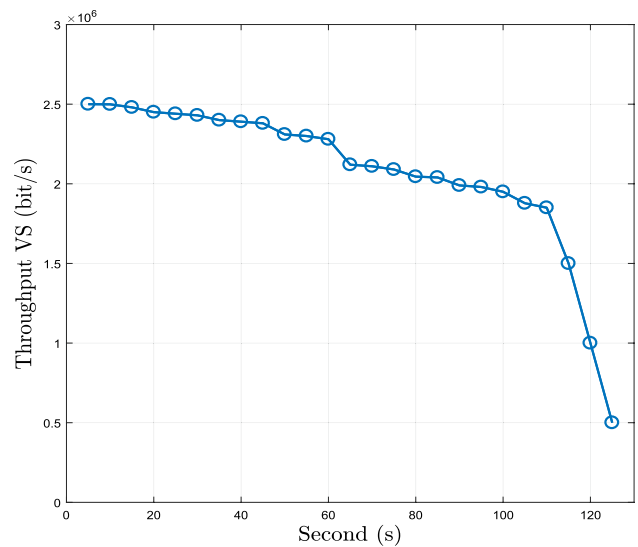


Fig. 5 Server to HoloLens video streaming throughput

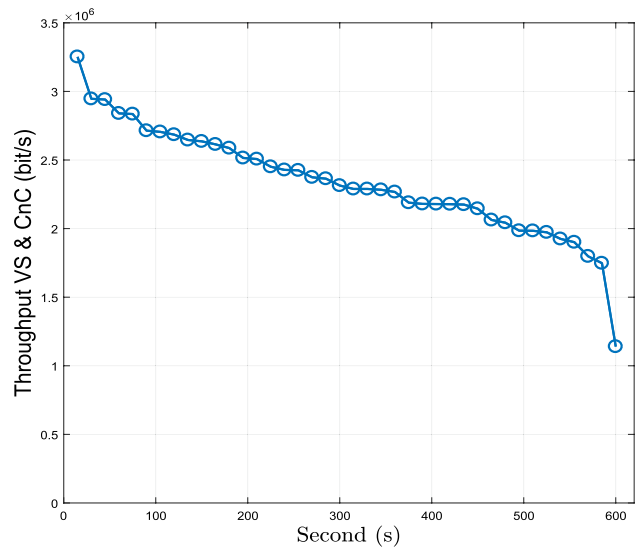


Fig. 6 Server to HoloLens video streaming and command and control throughput

The result is shown in Figs. 7 and 8 that the RTT for VS and VSC & C both went higher as the streaming time increased, the increasing value is caused by the network traffic, the distance between the source and destination, and the infrastructure component. To calculate the average RTT, it can be written as:

$$\overline{\mu_x} = \frac{1}{T} \sum_{s_x=0}^T \mu(s_x), \tag{2}$$

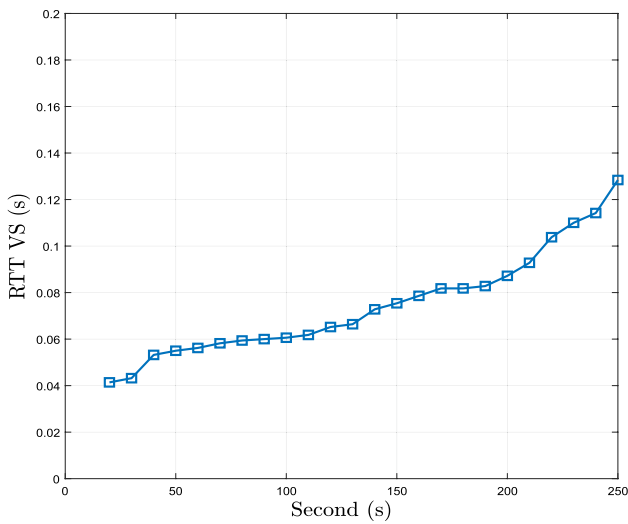


Fig. 7 Server to HoloLens video streaming RTT

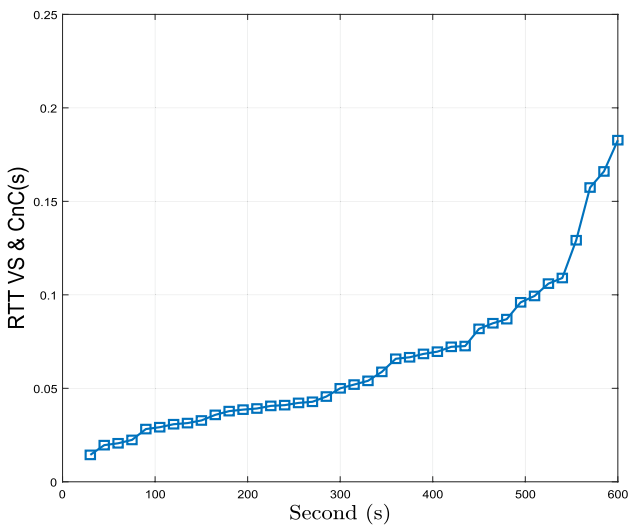


Fig. 8 Server to HoloLens video streaming and command and control RTT

where  $T$  denotes a total observation time,  $s$  denotes the second interval, and  $\mu_x$  denotes the RTT obtained on  $s^{th}$  interval for  $x : VS \text{ or } VSC \& C$  scenario.

For VS scenario, with  $T=250$  and  $s = [0, 5, 10 \dots, T]$  the value obtained from Eq. (2) for  $\overline{\mu_{VS}}$  is 0.366 s. While for the  $\overline{\mu_{VSC\&C}}$ , where the  $T=600$  and  $s = [0, 15, 30 \dots, T]$ . The value obtained for  $\overline{\mu_{VSC\&C}}$  is 0.332 s.

### 5.3 Latency

The network end-to-end latency is a packet delay from one client to another client and in this paper, the result was obtained from  $L = \|t_o - t_1\|$ . Here,  $t_o$  denotes the time of the network packet departing from HoloLens, and  $t_1$  denotes the

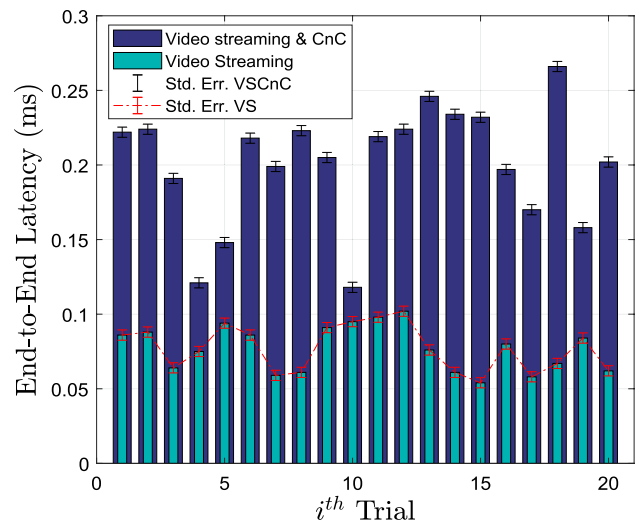


Fig. 9 The result of end-to-end latency devices

time of packet arrival on UAV. Latency measurement has been done in both VS and VSC & C scenarios.

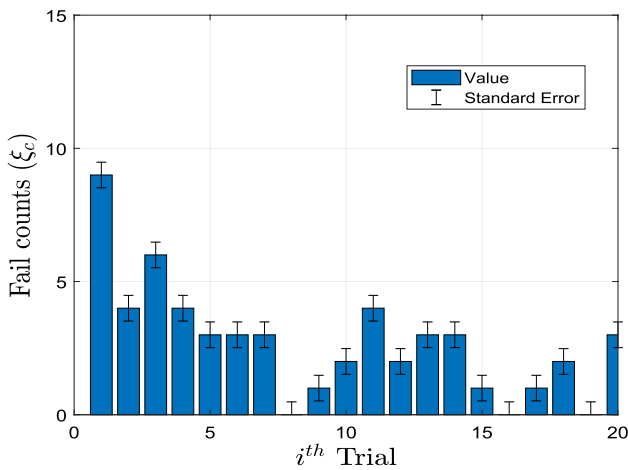
Figure 9 represents the latency that occurred in VS and VSC & C scenarios. Experiments were conducted in a total of 20 trials. By counting the average for a total of  $i^{th}$  60 s each. The total average latency for VS measurement is 0.077s. In VSC & C latency measurement, the average latency is 0.244s. In addition, the calculated standard deviation for VS latency is  $\approx 0.015$ , while the standard error is  $\approx \pm 0.003$ . Furthermore, the standard deviation for VSC & C latency is  $\approx 0.03$ , while the standard error is  $\approx \pm 0.008$ .

The result shown by Mellinkoff et al. (2018) shows the minimum thresholds of low-latency teleoperation range from 0.4 to 2.4s. Based on this threshold, the current work falls short of meeting this threshold, which can lead to a notable decrease in operator efficiency, and it is the limitation of this current work. This latency problem can be minimized in the future iteration of this work, especially for video streaming by using e.g., WebRTC, Nimbro, etc.

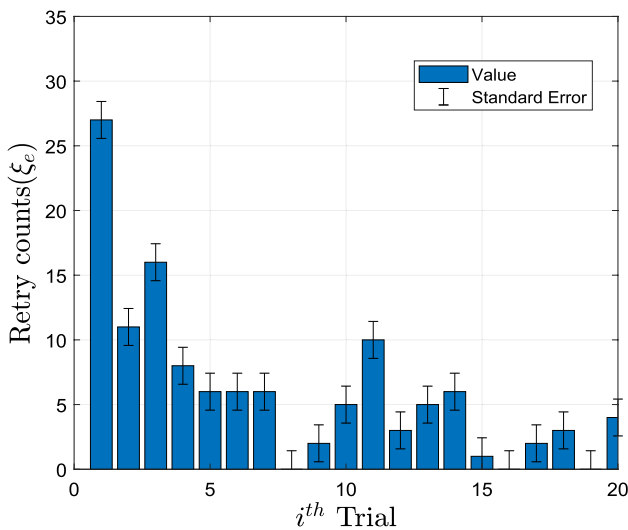
### 5.4 Speech accuracy test

In this section, the analysis of the system performance was performed. The result of the performance tests is the number of retries and count of errors that occur while giving speech commands to the UAV. For this experiment, multiple users who have different native language backgrounds are involved. Experiments were conducted in a total of 20 trials, where every  $i^{th}$  trial contains 102 random speech commands from Table 2. Additionally, in order to achieve reliability





**Fig. 10** Number of wrong actions that UAV takes while receiving speech command



**Fig. 11** Number of retry command that has to be repeated in order to obtain correct UAV action

of the system, it is considered that the fail threshold is  $\epsilon_x \approx 0.9 \times Trials^1$

Figure 10 represents the number of  $\xi_c$  of each  $i^{th}$  trial, where  $\xi_c$  denotes number of fail counts. The fail counts are calculated when UAV action is not suitable with the speech command. It is considered that the system is considered reliable since the fail counts from all  $i$  trials fall below  $\epsilon_c$ .<sup>2</sup> In addition, the calculated standard deviation from all trials is  $\approx 2.1$ , and the standard error is  $\approx \pm 1.43$ . Figure 11 shows the total number of speech command retries that have to be

<sup>1</sup> The  $\epsilon_x$  represents the threshold for different scenarios and  $Trials$  is the total of trial test performance.

<sup>2</sup> Obtain fail count threshold with calculation  $\epsilon_c \approx 0.5$ .

**Table 3** Questionnaire item

No.	Statements
1	I can control the UAV properly with this system.
2	I found the latency interferes with the control operation.
3	I found the functions in this system were well integrated.
4	I feel confident controlling the UAV using button control.
5	I think the speech command mechanism is easy to control.
6	I found the system unnecessarily complex.
7	I think that most people would adapt to this service very quickly.
8	I think that I would need the support of an experienced person to be able to use this system.
9	I felt comfortable doing the experiment.
10	I felt tired during the experiment.

made until the UAV takes an action based on it, and it is denoted with  $\xi_e$ . Based on this condition, it can be concluded that the average retries that have to be done are  $\approx \frac{102}{27} \approx 3$ . In addition, the standard deviation from  $\xi_e$  is  $\approx 6.3$ , and the standard error is  $\approx \pm 0.48$ .

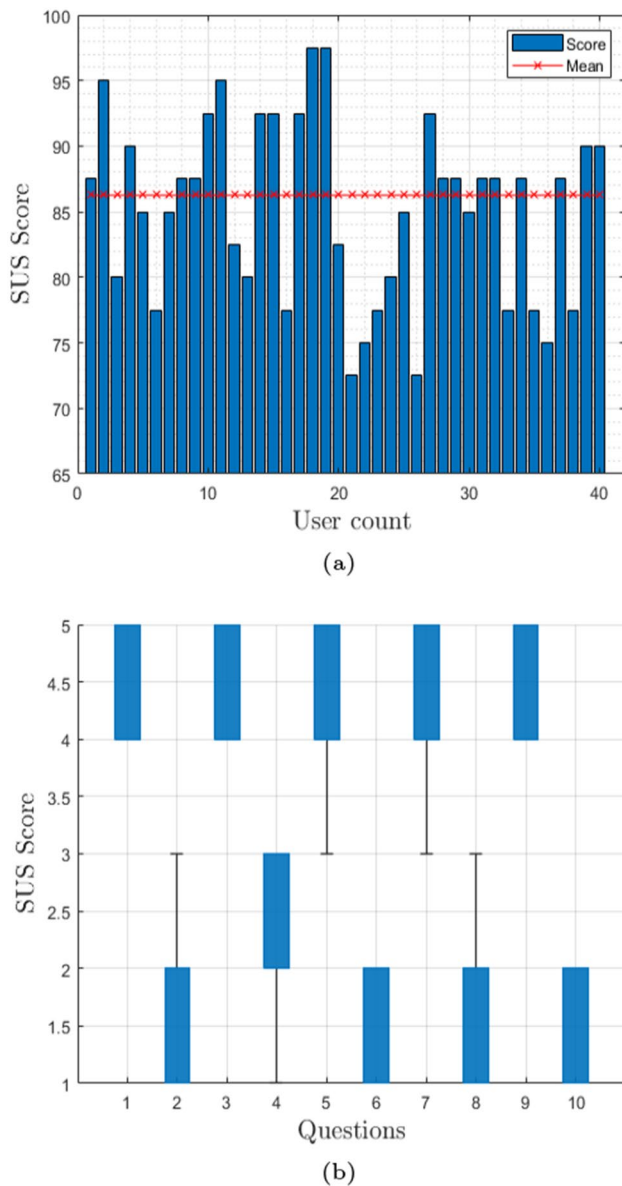
$$SuccessRate = 1 - \frac{E}{TotalTrial} \times 100\% \tag{3}$$

The accuracy success rate is the percentage of attempts that result in success. Equation(3) shows a formula to count the success rate of the proposed system, where  $E$  is the error trial amount, and  $TotalTrial$  is the total of the command trial, which can be calculated as  $20 \times 102 = 2,040$  random commands.

### 5.5 User study

A user study was conducted with 40 participants, including 24 males and 16 females aged 18 to 40, who joined the experiment and provided feedback through a questionnaire after trying the applications. Three of the participants had prior experience with HMD, and the other 37 were unfamiliar with it. Eight participants (20%) are fellow laboratory members, and the other 32 (80%) are students enrolled in the school. System Usability Scale (SUS) has been chosen to evaluate the effectiveness of the system as SUS become a standard for system analysis (John 1996). SUS is composed of 10 statements, and users answer on a five-point scale ranging from “strongly disagree” (1) to “strongly agree” (5). The statement list shown in Table. 3.

In order to assess the SUS score, the result of each statement is calculated and added together. The result can range from 0 to 4, depending on whether the statement is worded positively or negatively. For positively worded statements (Q1, Q3, Q5, Q7, and Q9), the result equals the scale position minus by 1. For negatively worded statements (Q2, Q4,



**Fig. 12** **a** System Usability Scale (SUS), **b** Boxplot for the ten SUS questions

Q6, Q8, and Q10), the result equals 5 minus the scale position. The final SUS score is obtained by multiplying the total score by 2.5, which results in a score between 0 and 100, indicating the overall usability of the system.

The SUS result depicted in Fig. 12 reveals an average score of 85.25, falling within the “good” to “excellent” range for system evaluation (Bangor et al. 2009). Figure 12b shows the Box-and-Whisker diagram of the scores of the ten SUS statements. It shows the median and the distribution of the scores of the ten SUS statements. Based on the result, the highest average scores are for Q1, Q7, and Q9. This implies that the participants can control the UAV properly with this system, this system is easy to adapt, and the participants are

comfortable with the experiment that has been conducted. On the other hand, the lowest scores are measured for Q4. This implies that the participants were not confident in controlling the UAV using the button control and were more confident using speech commands instead.

## 6 Conclusion

This paper presents an implementation of MR-GCS to support HRI on UAV teleoperation. The proposed system comprises a speech command-based UAV control system, video streaming, and a designed user interface. A user study with 40 participants was conducted to measure the effectiveness of the system. The qualitative result suggests that HoloGCS has the potential for teleoperation in HRI, with speech commands being easier to use than non-speech interfaces. Based on the quantitative result, VSC & C scenarios show lower throughput, higher RTT, and higher latency. However, based on the SUS score that represents the qualitative result of the proposed system, the system got an 85.25 SUS score, and it does not interrupt the control mechanism in terms of drone response and user experience. Future studies in network optimizers for HoloGCS latency minimization using, e.g., WebRTC and Nimbro, can be implemented to reduce these network limitations. Moreover, by decreasing latency, the current system can be extended for more complex commands, and the SUS score can be improved.

**Acknowledgements** This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2023-2020-0-01612) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation) and this research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2023-00259061) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

**Data availability** Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** All the authors of the paper make the following statement: No potential conflict of interest was reported by the authors.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Bangor A, Kortum P, Miller J (2009) Determining what individual SUS scores mean: adding an adjective rating scale. *J Usabil Stud* 4(3):114–123
- Bennett M, Williams T, Thames D, Scheutz M (2017) Differences in interaction patterns and perception for teleoperated and autonomous humanoid robots. In: 2017 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp 6589–6594
- Bokade AU, Ratnaparkhe VR (2016) Video surveillance robot control using smartphone and raspberry pi. In: 2016 International conference on communication and signal processing (ICCSP), pp 2094–2097
- Bosch J, Ridao P, García R, Gracias NR (2022) Towards omnidirectional immersion for rov teleoperation. *Actas de las XXXVII Jornadas de Automática 7, 8 y 9 de septiembre de 2016*, Madrid
- Chuxin C, Xinzhen J (2021) Speech recognize mobile robot. In: 2021 5th International conference on robotics and automation sciences (ICRAS), pp 89–93
- Fayjie AR, Ramezani A, Oualid D, Lee DJ (2017) Voice enabled smart drone control. In: 2017 Ninth international conference on ubiquitous and future networks (ICUFN), pp 119–121. IEEE
- García JC, Patrão B, Almeida L, Pérez J, Menezes P, Dias J, Sanz PJ (2017) A natural interface for remote operation of underwater robots. *IEEE Comput Graph Appl* 37(1):34–43
- Gattullo M, Scurati GW, Fiorentino M, Uva AE, Ferrise F, Bordegnoni M (2019) Towards augmented reality manuals for industry 4.0: a methodology. *Robot Comput-Integr Manuf* 56:276–286
- Gong L, Gong C, Ma Z, Zhao L, Wang Z, Li X, Jing X, Yang H, Liu C (2017) Real-time human-in-the-loop remote control for a life-size traffic police robot with multiple augmented reality aided display terminals. In: 2017 2nd International conference on advanced robotics and mechatronics (ICARM), pp 420–425
- Graff C (2016) Drone piloting study. University of Italian Switzerland, Lugano
- Hedayati H, Walker M, Szafir D (2018) Improving collocated robot teleoperation with augmented reality. In: Proceedings of the 2018 ACM/IEEE international conference on human–robot interaction, HRI '18, pp 78–86, New York, NY, USA. Association for Computing Machinery
- Herrmann R, Schmidt L (2018) Design and evaluation of a natural user interface for piloting an unmanned aerial vehicle. *i-com* 17(1):15–24
- Higuchi K, Fujii K, Rekimoto J (2013) Flying head: A head-synchronization mechanism for flying telepresence. In: 2013 23rd international conference on artificial reality and telexistence (ICAT), pp 28–34
- Hoppenstedt B, Witte T, Ruof J, Kammerer K, Tichy M, Reichert M, Pryss R (2019) Debugging quadcopter trajectories in mixed reality. In: De Paolis LT, Bourdot P (eds) *Augmented reality, virtual reality, and computer graphics*. Springer International Publishing, Cham, pp 43–50
- Huang B, Bayazit D, Ullman D, Gopalan N, Tellex S (2019) Flight, camera, action! using natural language and mixed reality to control a drone. In: 2019 International conference on robotics and automation (ICRA), pp 6949–6956
- Ibrahimov R, Tsykunov E, Shirokun V, Somov A, Tsetserukou D (2019) Dronepick: Object picking and delivery teleoperation with the drone controlled by a wearable tactile display. In: 2019 28th IEEE International conference on robot and human interactive communication (RO-MAN), pp 1–6
- Jeong M, Lee H, Bae M, Shin D-B, Lim S-H, Lee KB (2018) Development and application of the smart helmet for disaster and safety. In: 2018 International conference on information and communication technology convergence (ICTC), pp 1084–1089
- Jerald J, Giokaris P, Woodall D, Hartbolt A, Chandak A, Kuntz S (2014) Developing virtual reality applications with unity. In: 2014 IEEE Virtual Reality (VR), pp 1–3
- John B (1996) Sus: a “quick and dirty” usability scale. *Usability evaluation in industry*, 189–194
- Kot T, Novák P (2014) Utilization of the oculus rift hmd in mobile robot teleoperation. In: Modeling and optimization of the aerospace, robotics, mechatronics, machines-tools, mechanical engineering and human motricity fields, volume 555 of *Applied Mechanics and Materials*, pp 199–208. Trans Tech Publications Ltd
- Kot T, Novák P, Bajak J (2018) Using hololens to create a virtual operator station for mobile robots. In: 2018 19th International carpathian control conference (ICCC), pp 422–427
- Krupke D, Steinicke F, Lubos P, Jonetzko Y, Görner M, Zhang J (2018) Comparison of multimodal heading and pointing gestures for collocated mixed reality human-robot interaction. In: 2018 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp 1–9
- Landau M, van Delden S (2017) A system architecture for hands-free uav drone control using intuitive voice commands. In: Proceedings of the companion of the 2017 ACM/IEEE international conference on human–robot interaction, pp 181–182
- Lee J, Davari H, Singh J, Pandhare V (2018) Industrial artificial intelligence for industry 4.0-based manufacturing systems. *Manuf Lett* 18:20–23
- Lipton JJ, Fay AJ, Rus D (2018) Baxter’s homunculus: virtual reality spaces for teleoperation in manufacturing. *IEEE Robot Autom Lett* 3(1):179–186
- Luque Vega LF, Lopez-Neri E, Arellano-Muro CA, González-Jiménez LE, Ghommam J, Carrasco-Navarro R (2020) Uav flight instructional design for industry 4.0 based on the framework of educational mechatronics. In: IECON 2020 The 46th annual conference of the IEEE industrial electronics society, pp 2313–2318
- Mellinkoff BJ, Spydell MM, Bailey W, Burns JO (2018) Quantifying operational constraints of low-latency telerobotics for planetary surface operations. In: 2018 IEEE aerospace conference, pp 1–10. IEEE
- Microsoft HoloLens 2 (2021). <https://www.microsoft.com/en-us/hololens/hardware>. Accessed on May. 25, 2021
- Robot Operating System—ROS (2021a). <https://www.ros.org/>. Accessed on May. 25, 2021
- ROSbridge (2021b). [http://wiki.ros.org/rosbridge\\_suite/](http://wiki.ros.org/rosbridge_suite/). Accessed on May. 25, 2021
- Steinfeld A, Fong T, Kaber D, Lewis M, Scholtz J, Schultz A, Goodrich M (2006) Common metrics for human–robot interaction. In: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction, pp 33–40
- Stoker C, Burch D, Hine B, Barry J (1995) Antarctic undersea exploration using a robotic submarine with a telepresence user interface. *IEEE Expert* 10(6):14–23
- Sun A, Shi J, Li W (2019) A precise fingertip tracking algorithm based on 3D motion mapping for hololens. In: 2019 11th International conference on intelligent human-machine systems and cybernetics (IHMSC), volume 1, pp 229–232
- Sutar T, Pawar S (2021) Smart glasses: Digital assistance in industry. In: Merchant SN, Warhade K, Adhikari D (eds) *Advances in signal and data processing*. Springer Singapore, Singapore
- Unity3d (2021). <https://unity.com/>. Accessed on May. 25, 2021
- Wan J, Tang S, Hua Q, Li D, Liu C, Lloret J (2018) Context-aware cloud robotics for material handling in cognitive industrial internet of things. *IEEE Internet Things J* 5(4):2272–2281

- Whitney D, Rosen E, Phillips E, Konidakis G, Tellex S (2020) Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality. In: Amato NM, Hager G, Thomas S, Torres-Torriti M (eds) *Robotics Research*. Springer International Publishing, Cham, pp 335–350
- Widiyanti DE, Shin SY (2021) Holo-drone streaming system. *Korean Inst Commun Inf Sci* 2021:1054–1055
- Zhao J, Allison RS, Vinnikov M, Jennings S (2018) The effects of visual and control latency on piloting a quadcopter using a head-mounted display. In: 2018 IEEE International conference on systems, man, and cybernetics (SMC), pp 2972–2979
- Zhou Y, Hou J, Gong Y (2020) Research and application of human-computer interaction technology based on voice control in ground control station of uav. In: 2020 IEEE 6th International conference on computer and communications (ICCC), pp 1257–1262. IEEE
- Zuehlke D (2010) Smartfactory-towards a factory-of-things. *Annu Rev Control* 34(1):129–138

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.