A QoE Model to Evaluate Semi-Transparent Augmented-Reality System

by

Longyu Zhang

Thesis submitted to the University of Ottawa

in partial fulfillment of the requirements for the

Doctorate in Philosophy degree in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

With the development of three-dimensional (3D) technologies, the demand for high-quality 3D content, 3D visualization, and flexible and natural interactions are increasing. As a result, semi-transparent Augmented-Reality (AR) systems are emerging and evolving rapidly. Since there are currently no well-recognized models to evaluate the performance of these systems, we proposed a Quality-of-Experience (QoE) taxonomy for semi-transparent AR systems containing three levels of influential QoE parameters, through analyzing existing QoE models in other related areas and integrating the feedbacks received from our user study. We designed a user study to collect training and testing data for our QoE model, and built a Fuzzy-Inference-System (FIS) model to estimate the QoE evaluation and validate the proposed taxonomy. A case study was also conducted to further explore the relationships between QoE parameters and technical QoS parameters with functional components of Microsoft HoloLens AR system. In this work, we illustrate the experiments in detail and thoroughly explain the results obtained. We also present the conclusion and future work.

Acknowledgements

I would like to give my sincerest gratitude and appreciation to my supervisor Prof. Abdulmotaleb El Saddik for his continuous guidance and support, not only in academic level but also in my personal life.

Special and sincere thanks go to Dr. Haiwei Dong for his invaluable assistance, guidance, and feedback, throughout my research.

I would also like to thank all colleagues in Multimedia Communications Research Laboratory and all my friends, for their contributions throughout the research and their help on my life.

Finally, I am grateful to my family. My parents and my wife Lei Zhou, whose consistent inspiration, understanding, support, and endless love, help me go through all hardships during this journey. This work is dedicated to them.

Contents

List of Tables

List of Figures

List of Abbreviations

3D : Three-Dimensional

AR : Augmented-Reality

CT : Computed Tomography

HMD : Head-Mounted Display

VR : Virtual-Reality

FOV : Field-of-View

OS : Operating System

ToF : Time-of-Flight

CPU : Central-Processing Unit

HPU : Holographic-Processing Unit

QoE : Quality-of-Experience

QoS : Quality-of-Service

HCI : Human-Computer Interaction

HAVE : Haptic-Audio-Visual Environments

FIS : Fuzzy-Inference System

SoC : System-on-Chip

GIS : Geographic Information System

MTM : Modified Traverse Mercator

MVS : Multi-View Stereo

MR : Mixed Reality

PID : Proportional-Integral-Derivative

ICP : Iterative-Closest-Point

IMU : Inertial-Measurement Unit

DC : Direct Current

MF : Membership Function

UX : User Experience

Chapter 1

Introduction

1.1 Background

The vision of a digital twin is a digital replication of a living or non-living physical entity, introduced by El Saddik [13]. By bridging the physical and the virtual worlds, data are transmitted seamlessly, allowing the virtual entity to exist simultaneously with the physical entity. A digital twin facilitates the means to monitor, understand, and optimize the functions of the physical entity and provides continuous feedback to improve quality of life and well-being. A digital twin is hence the convergence of several technologies such as AI, AR/VR and Haptics, IoT, Cybersecurity and Communication networks.

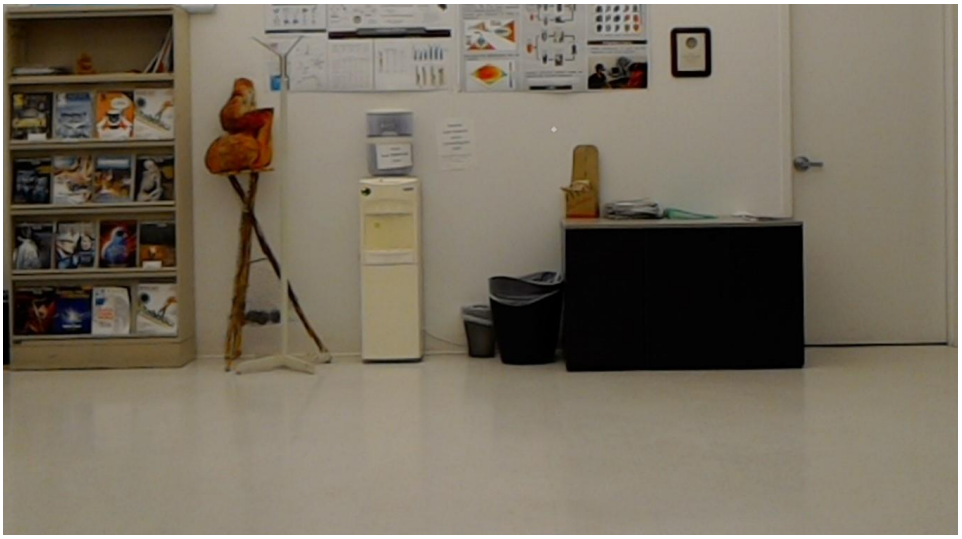
A component of the digital twin vision is three-dimensional (3D) technology, which has been developing rapidly recent years, and has influenced the industrial, medical, cultural, and many other fields. As a result, the demand for high-quality 3D content, 3D visualization, and flexible and natural interactions is growing.

One important method used to display and interact with 3D contents is through an augmented-reality (AR) system, which enables users to simultaneously visualize real and virtual contents, and interact with 3D holograms. The first appearance of AR system

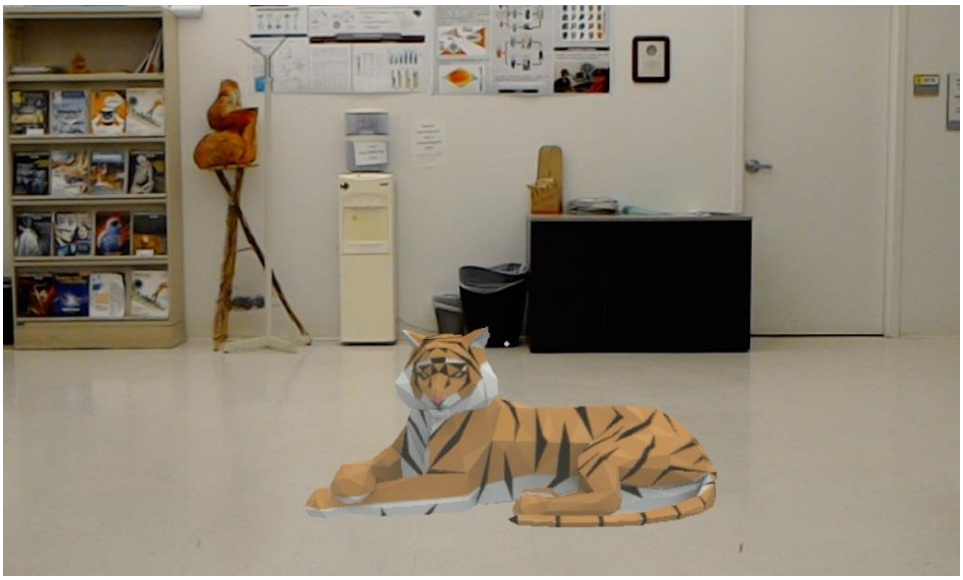
was in 1968, when Sutherland developed the first optical see-through head-mounted display (HMD) system [14]. In 1975, Krueger et al. built Videoplace, a lab containing video cameras and projectors to offer users an interactive environment [15]. Later, Caudell, a former Boeing researcher, coined the term "Augmented Reality" [16]. Rosenberg developed Virtual Fixtures, one of the first functioning AR systems, a full upper-body exoskeleton that allowed Air Force soldiers to perform remote tasks with virtually guided machinery [17].

AR technology enables users to see a combination of the real and virtual things in real-time through augmenting virtual holograms on the real environment. Figure 1.1 shows the comparison between the real environment view and the augmented-reality view. Figure 1.1(a) is the view of the real environment, while Figure 1.1(b) shows the view of integrating a virtual 3D "tiger" hologram with the real environment using AR technology, which enables users to see real and virtual contents at the same time. AR differs from virtual-reality (VR) in the way that VR blocks users from the real world to provide them more immersive experience [18, 19].

One important aspect of AR technology is the creation of virtual text or models. When text contents are easily generated, 3D models can be created in many ways. One method is using graphics and animation software, such as 3DMax, SketchUp, and Maya, to build the digital models, while another method is 3D reconstruction from real objects or human [20, 21]. The 3D reconstruction process usually involves acquiring data, building the point cloud, and converting the 3D model into a triangulated network (mesh) or textured surface [22]. Because different 3D sensing techniques have different requirements for light conditions (visible or invisible lights), result accuracy (for entertainment or medical use), and sensor configuration (a moving sensor or multiple sensors), users should choose an appropriate technique according to their specific requirements [23]. The field of 3D reconstruction has been the subject of intensive and long-term research by



(a) Real environment



(b) Augmenting a virtual “tiger” on the real environment

Figure 1.1: Comparison of real environment view and augmented-reality view. (a) real environment (b) augmenting real environment with a virtual “tiger” hologram.

the graphics, vision, and photogrammetric communities, and is fundamental for AR and VR development.

AR contents can be augmented on the real environment using a variety of different methodologies. Popular mobile AR games, such as Pokemon Go, have attracted thousands of players by placing virtual creatures on their smartphone screens, which simultaneously display the surrounding real-world environment. Users' reactions to these mobile AR games are also studied from the aspects of hedonic, emotional and social benefits, as well as in terms of social norms and physical risks [24]. Though augmenting contents on phone screens is easy and popular, due to device limitations, the surrounding environment cannot be accurately reconstructed to augment 3D contents precisely and vividly. Therefore, semi-transparent AR devices in the shape of glasses are designed specifically for AR applications, such as Google Glass and Microsoft HoloLens, providing users with a comparatively novel way to interact with AR contents [25, 26]. As a combination of fashion and technology, glasses-like semi-transparent devices have drawn increasing attention in various disciplines, such as medicine, tourism, education, social and marketing [27]. For instance, Ro et al. discussed the potential values and barriers to the use of such devices [25]; Orts-Escolano et al. presented Holoportation, a system that allows the user to interact with the augmented avatars of remote users in real time [28]; while Kalantari et al. developed a model to understand individuals' acceptance of and reactions to these glasses-like semi-transparent devices in a social environment [29].

Besides differences in visual aspects, semi-transparent AR applications also differ from mobile phone AR games (e.g., Pokemon Go) in several other ways, such as 3D holographic displays, enabling more natural gesture control, and augmenting content based on the detailed surrounding environment instead of directly superimposing content on the screen.

AR technology can be utilized in many fields, such as gaming, education, machine

operation, and medical treatments [30]. Since AR does not block users from the real surrounding environment, as does VR technology, AR is mainly used to develop applications that interact with the real world, while VR is mostly deployed to design applications requiring a high degree of immersion.

Significant progress has recently been achieved in the area of AR, and many advanced AR devices have been developed. For example, Google introduced Google Glass, which can be worn like conventional glasses, but have many integrated components, such as CPU, touchpad, microphone, display screen, wireless connectivity, and others [31]. Epson Moverio BT-300 also uses a pair of eyeglasses to display augmented contents superimposed on the real surrounding environment [32]. A recently released AR device HoloLens, which is developed by Microsoft, differs from most other similar devices in that it itself is a complete AR system, running the Windows 10 operating system (OS) and containing a central-processing unit (CPU), a custom-designed holographic-processing unit (HPU), various types of sensors, and see-through optical lenses with a holographic projector. Semi-transparent AR systems have attracted considerable attentions, and most research and development highly depend on the devices themselves. However, currently there are no well-recognized models to evaluate the performance of semi-transparent AR system.

Therefore, in this thesis, we proposed a Quality-of-Experience (QoE) taxonomy for semi-transparent AR systems, through analyzing QoE parameters in related fields and integrating user feedback. We designed a user study to collect training and testing data for our QoE model, and selected high-level QoE parameters as inputs to build a Fuzzy-Inference-System (FIS) model to estimate the QoE evaluation and validate the proposed taxonomy. We also conducted case study to further explore QoE parameters with functional components from the Microsoft HoloLens AR system.

1.2 Motivation

Traditionally, aspects of system performance, such as network conditions and video quality, are usually evaluated with quality-of-service (QoS) models, proposed to capture the qualitatively or quantitatively defined performance contract between the service provider and the user applications. Although QoS models successfully measure the technological quality and functionality of systems with little human involvement, such models fail to evaluate the user's satisfaction with interactive multimedia applications or devices, including semi-transparent AR systems [33]. Compared to system-centric QoS evaluation methodologies, quality-of-experience (QoE) metrics are more human-centric and take into account user-involved interactions, thus also providing a comprehensive overview of the multimedia devices.

Many research efforts have been undertaken to develop user-centric QoE evaluation models [12]. For instance, the European Network on Quality of Experience in Multimedia Systems and Services (QUALINET) has extended the notion of network-centric QoS to QoE in multimedia systems to develop subjective and objective quality metrics [34], and defined QoE as "the degree of delight or annoyance of the user of an application or service". In addition, Wu et al. also presented a conceptual framework for QoE in a distributed interactive multimedia environment and developed a mapping methodology to demonstrate the correlations between QoS and QoE [11]. However, QoE metrics vary depending on the systems, and the QoE models for other fields cannot be directly used on semi-transparent AR systems.

To the best of our knowledge, there are currently no well-recognized QoE models for measuring semi-transparent AR systems. Therefore, we propose a QoE model to evaluate these devices.

1.3 Existing Problems

Since a semi-transparent AR system involves large amounts of human-computer interactions (HCI), it cannot be accurately evaluated with QoS models that have pure technical parameters. Therefore, more user-centric QoE model is needed to describe the user's satisfaction level. Compared with traditional system-centric QoS models, human-centric QoE evaluations can more accurately reflect users' AR experiences.

Various multimedia applications and devices are designed to satisfy users' varying needs, and although there are QoE models for other related fields, there are no unified QoE models that are applicable to all multimedia devices. Currently, there are no well-recognized models to evaluate the performance of semi-transparent AR systems.

Therefore, the main problem is the generation of a QoE model for semi-transparent AR systems to accurately evaluate their AR experience. We start by deriving a QoE taxonomy containing three levels of influential QoE parameters, and model it with a fuzzy-inference-system (FIS) using data from our user study to quantitatively evaluate the semi-transparent AR system. We validate our FIS model by comparing its outputs (the general user experience) with ground truth user ratings, to prove that our proposed QoE framework can represent a user's general AR experience. Details of the QoE taxonomy and of the FIS model are described in our work, and we also provided validations and an analysis of the model.

1.4 Contribution

In this thesis, we proposed a QoE taxonomy for semi-transparent AR systems, designed a user study to collect training and testing data for a FIS model, selected high-level QoE parameters as inputs to build a FIS model, and validated the proposed model. We also conducted a case study to further explore QoE parameters with AR functional

components. Detailed contributions are as follows:

- Proposed a QoE taxonomy for semi-transparent AR systems, through both analyzing QoE metrics for other related fields and integrating our own user feedbacks;
- Designed user study based on the proposed QoE taxonomy to collect training and testing data, and selected high-level parameters as FIS inputs, by evaluating their relationships with low-level parameters;
- Built a FIS model to estimate the QoE evaluation, and validated the proposed taxonomy;
- Designed and conducted several case studies to further explore the relationships between QoE parameters and technical QoS parameters, through designing experiments to evaluate functional components of AR systems.

1.5 Scholarly Achievements

In the process of completing this work, the following publications have been published or accepted:

- **Refereed journal papers**:
    1. *L. Zhang*, H. Dong, A. El Saddik, "Towards a QoE model to evaluate holographic augmented reality devices: A HoloLens case study." *IEEE Multimedia*, (accepted)
    2. Y. Liu, H. Dong, *L. Zhang*, A. El Saddik, "Technical evaluation of HoloLens for multimedia: A first look." *IEEE Multimedia*, 25.4, (2018): 8-18
    3. *L. Zhang*, S. Chen, H. Dong, A. El Saddik, "Visualizing Toronto city data with HoloLens." *IEEE Consumer Electronics Magazine*, 7.3 (2018): 73-80.

4. *L. Zhang*, B. Han, H. Dong, A. El Saddik, "Development of an automatic 3D human head scanning-printing system." *Multimedia Tools and Applications*, 76.3 (2017): 4381-4403.
5. *L. Zhang*, H. Dong, A. El Saddik, "From 3D sensing to printing: A survey." *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12.2 (2016): 27:1-23.
6. L. Yang, *L. Zhang*, H. Dong, A. Alelaiwi, A. El Saddik, "Evaluating and improving the depth accuracy of Kinect for Windows v2." *IEEE Sensors Journal*, 15.8 (2015): 4275-4285.
7. *L. Zhang*, J. Saboune, A. El Saddik, "Development of a haptic video chat system." *Multimedia Tools and Applications*, 74.15 (2015): 5489-5512.

- **Refereed conference papers**:

1. *L. Zhang*, H. Dong, A. El Saddik, "A multisensory datafusion-based 3D plank-coaching system." *Digital Media Industry & Academic Forum (DMIAF)*, Santorini, 2016, pp. 154-157.
2. F. Arafsha, *L. Zhang*, H. Dong, A. El Saddik, "Contactless haptic feedback: state of the art." *IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, Ottawa, 2015, pp. 1-6.

1.6 Thesis Organization

This thesis is organized as follows:

- **Chapter 2:** This chapter presents a review of related literature, including AR, QoE, and FIS.

- **Chapter 3:** In Chapter 3, we explain our proposed methods. Details of the system design, proposed framework, implementation, experiments design and process, are illustrated.

- **Chapter 4:** The experimental details and results are exposed in this chapter. We display and analyze all of the results.

- **Chapter 5:** In this chapter, we comprehensively discuss and review our QoE evaluation for semi-transparent AR system, as well as the experiments to explore functional components and QoE parameters.

- **Chapter 6:** In this final chapter, we summarize the thesis and present the future work to be completed.

Chapter 2

Background and Related Works

2.1 Augmented Reality History

Augmented-Reality (AR) is a subclass of Mixed-Reality (MR), which is generally used to refer to the merging of the real and virtual worlds. Milgram and Kishino presented the concept of "virtuality continuum" for different display situations, including real environment, augmented reality, augmented virtuality, and virtual environment, as show in Figure 2.1 [1]. Each situation can be linked to a point on the continuum, depending on the degree of its "real" and "virtual" aspects. In this figure, the left extrema real environment consists solely of real objects, while the right extrema virtual environment displays solely virtual objects, such as a conventional computer graphic simulation. MR refers to any point between the extrema of the virtuality continuum, which means real world and virtual world objects are presented together using a single display. Based on the ratio of real and virtual contents, MR can be further divided into AR and augmented virtuality.

The development history of AR technology began in the 1960s, when Sutherland created the first AR head-mounted system with 3D display and head position sensors

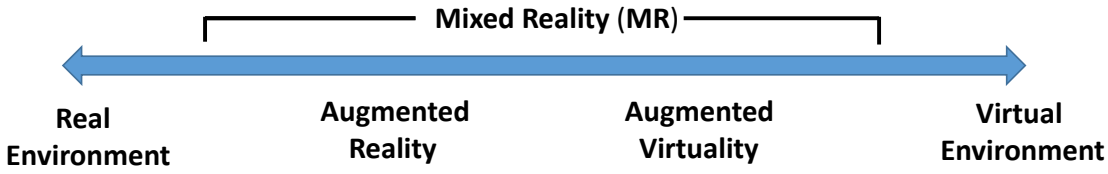


Figure 2.1: A representation of the "virtuality continuum" [1].

[14]. Since then, AR has attracted a great deal of attentions, and has been used to enhance users' perception of and interaction with the real world.

A well-accepted definition of AR was proposed by Azuma in 1997, who used three key characteristics to define the AR system [30]:

- It combines virtual and real contents;
- It is able to interact with users in real-time;
- It is registered in three-dimensional (3D).

These three key elements also represent the AR technical requirements, namely, it should have a screen to display both real and virtual contents, it must track the position and view-point of the user, and it needs to generate interactive 3D contents in real-time.

Based on AR related publications, Zhou et al. divided AR research areas into two groups: the first group contains five core topics, which are tracking, interaction, calibration and registration, AR applications, and display technology; while the second group reflects more emerging research interests, including evaluation/testing, mobile/wearable AR, AR authoring, visualization, multimodal AR, and rendering [35]. According to their report, tracking attracted the most attention as one of the fundamental AR enabling technologies, and interaction became the second most influential topic, which reflects the progress made from exploring fundamental AR techniques to apply them on real world applications. Many recent AR research projects focus on enhancing user interactions with

the real world and making computer interfaces invisible [19, 36]. The latter is the same as virtual-reality (VR), which usually uses a HMD device to separate the user from the real world by showing computer-generated contents, such as Oculus Rift [37]. However, compared with VR, AR requires higher accuracy to "place" the holographic contents at appropriate positions in the real world, but demands less realistic image rendering and a smaller field-of-view (FOV) to create an immersive experience [38]. Based on the key components from the AR definition, we introduce the development of AR in following areas:

2.2 AR Visual Displays

There are three main methods for AR visual displays: video see-through, optical see-through (semi-transparent glasses), and projection on real objects, based on the paper by Van Krevelen and Poelman [39]. Each method has its own advantages and disadvantages, and can be selected based on the specific requirements of users.

The video see-through method replaces the reality with a video feed of reality, and then overlays AR content on it. Since the reality has been transformed to a digital format, the system can easily mediate or remove contents from reality. In addition, the brightness and contrast can also be adjusted for both indoor and outdoor use, simply by evaluating the light conditions. With proper adjustments, the computer generated objects can blend smoothly into reality [40]. The disadvantages of this method are low-resolution reality, limited field-of-view, user disorientation, and poor eye accommodation. User disorientation is mainly caused by eye-offset since the camera's position is generally not at the exact same position as the human eyes are [41]. Discomfort display causes eye strain and fatigue [42]. An example of this type of device is the Pokemon Go game, which has attracted thousands of players through superimposing their animated creatures on players' phone screens when they scan the real surrounding environments [43].

Figure 2.2: Projecting a virtual "maple leaf" and red color on Parliament Hill (Canada).

The optical see-through (semi-transparent glasses) technique uses optical semi-transparent material to allow users to see the real-world in tact, and overlays holograms on the real environment. Semi-transparent systems work by placing optical combiners in front of users' eyes. The combiners are partially transmissive and partially reflective, so that the user can both look directly through them to see the real world, and the virtual images bounced off the combiners. With this technology, the virtual contents appear ghost-like and semi-transparent [30]. The first AR system created by Sutherland adopted this technique. Google Glass, a pioneering AR device, also uses a pair of eyeglasses to show augmented contents upon the real environment [44]. One drawback is the reduced brightness and contrast of both the real and virtual contents, making such technique less suitable for outdoor use. The field-of-view range depends on the design of the device itself.

Projecting AR contents directly onto real objects has the advantages of not requiring

special eye-wear and covering a large surface, enabling a wide field-of-view. As shown in Figure 2.2, the light show of Parliament Hill in Ottawa consists of virtual contents, such as a maple leaf, being projected directly onto the wall, and hundreds of visitors can experience the show together without wearing any special equipment. The drawback is that the projectors need to be calibrated based on the distance and the surface of the environment. It is also mainly used indoors or at night to increase brightness and contrast effects.

Besides visual displays, realistic and high-quality audio can also enhance a user's experience. For instance, Blessenohl et al proposed a set of context-specific cures that only require the systems to provide minimal depth-based audio feedback, with reduced masking of natural sounds, for a general-purpose sensing device [45]. Choi et al. used the spatial sounds to realize a holographic reconstruction of sound and generate realistic audio experience for users [46].

2.3 3D Content Generation

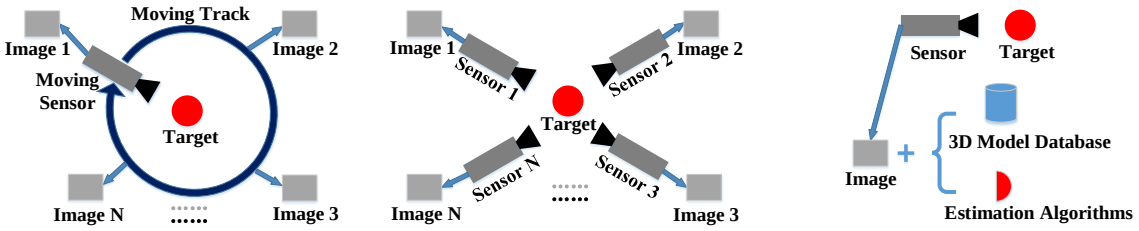
Before being displayed, AR contents first need to be generated. 3D models can be created in many ways. One method is by using graphics and animation software, such as 3DMax, SketchUp, and Maya, to build the digital models, while another method is 3D reconstruction based on real objects or humans [20, 21].

In the past few years, developments in the field of precisely measuring and reconstructing 3D models have attracted increasing attention, especially with the release of the consumer-grade RGB-depth sensor Microsoft Kinect [47]. Large projects, such as the *Digital Michelangelo Project*, which created a 3D computer archive of many of Michelangelo's statues and architectural works, and the *Great Wall of China in 3D Project*, which aimed to recreate the whole 6,000km length of the Great Wall of China using high resolution 3D models, are all practical applications of such 3D sensing technologies [48]. 3D

content reconstruction results are mainly affected by the sensing process and the sensors.

2.3.1 3D Reconstruction Process

The 3D reconstruction process mainly consists of capturing target information with sensors, merging the obtained information based on the sensing methods, and then reconstructing the 3D model. Based on the configurations of 3D sensing, it can be divided into three different categories: adopting one moving sensor, adopting multiple sensors with different views, and adopting one sensor with a limited view, as shown in Figure 2.3. Each configuration has its own advantages and disadvantages depending on its attributes, for instance speed, robustness, flexibility, computational cost, and completeness of the reconstruction [49]:



(a) One moving sensor. (b) Multiple sensors with different views. (c) One sensor with limited view.

Figure 2.3: Configurations of different 3D sensing processes

- Adopting One Moving Sensor: although sensing the target with a fixed sensor from a single view is not sufficient to generate detailed 3D models, users can obtain the full set of 360-degree views of the physical scene or object by moving the sensor around the target or by rotating the object. Through fusing all of the sensed information, a single representation can then be reconstructed [50, 51]. Figure 2.3(a) shows the detailed configuration of this sensing process. This method enables the use of portable 3D sensors to scan large objects or scenes conveniently without complex configurations [52]. the operating steps can be easily understood, and the

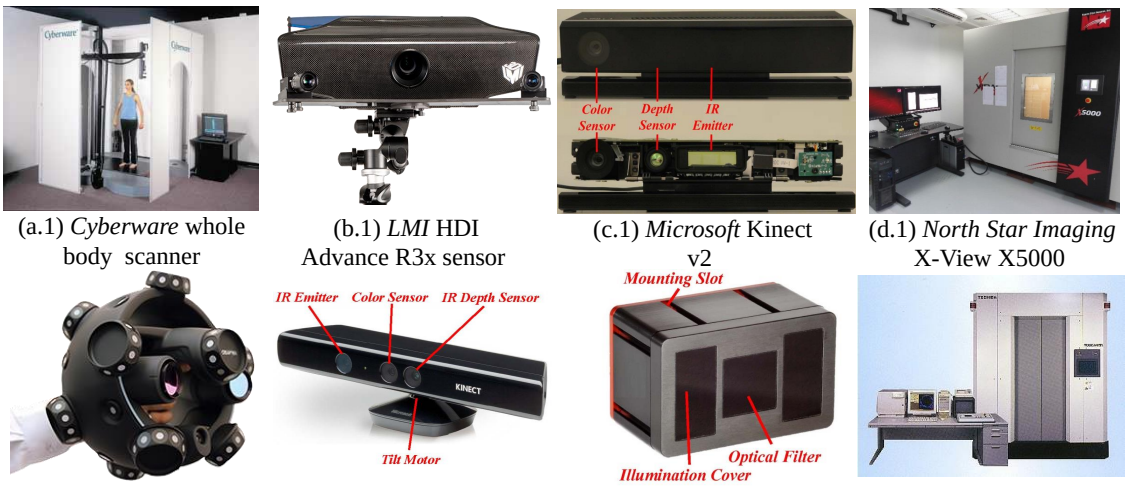
sensor's or turntable's rotating track, speed and angles can be carefully controlled [53, 54]. One major drawback of this sensing process is the requirement that the target remain still for a long time, which is not suitable for certain situations where there is a lot of movement, for example when sensing infants. Another drawback is the loop closure problem, caused because the sensing system sometimes fails to detect the completion of the scanning. As a result, the system cannot perfectly match the starting scene with the ending scene to reconstruct the complete 3D model [55, 56]. Moreover, various noises and an inappropriate operation of the sensor may degrade the reconstruction algorithms' performance. For example, individual pairwise errors would cause Iterative Closest Point (ICP) failure [57].

- Adopting Multiple Sensors with Different Views: setting two or more fixed sensors around the target to capture a full 360-degree view concurrently is another widely used 3D sensing process, as shown in Figure 2.3(b). It can be further divided into two categories: with sensor calibration (e.g., Photogrammetry) and without sensor calibration (e.g., Multi-View Stereo, known as MVS). Photogrammetry usually consists of camera calibration and orientation, image point measurements, and 3D model generation [58]. Among these steps, sensor calibration is crucial to obtain accurate models, and reliable packages are commercially available to complete this phase, such as *Photomodeler* and *Menci*. Conversely, MVS is more straightforward and cost-effective because its reconstruction is based on the identification of the common points within the image pairs. Generally, MVS algorithms can be classified into four categories based on the underlying object models: voxel-based, deformable-polygonal-meshes-based, multiple-depth-maps-based, and patch-based [59]. With this configuration, a relatively high accuracy can be achieved. For example, Rau and Yeh realized reconstruction results with an accuracy of 0.26 mm by carefully calibrating the digital single lens reflex cameras' exterior and interior ori-

entation parameters. Furukawa and Ponce also obtained considerably compelling results using MVS [60, 59]. The disadvantages are that this method always requires large mounts of computational resources, is limited to well-defined scenes, and may cause a certain level of interference and accuracy degradation [61, 62, 58].

- Adopting One Sensor with Limited View: despite the rapid development of 3D sensing technologies, there are still situations in which users cannot move the sensor or use multiple sensors to obtain the full views of the target. If a 3D reconstruction is still required in this case, appropriate estimations or 3D model database matching may need to be applied to reconstruct rough and approximate 3D models rather than detailed ones, as shown in Figure 2.3(c). Some techniques, such as shape from focusing, shape from shadows, shape from shading, and shape from photometry, are indirect, simple, and low-cost ways to solve this problem [63, 64]. Utilizing a collected or learned 3D database to match the acquired 2D images is also widely adopted to reconstruct objects or scenes. [65] reconstructed 3D human head models by extracting features from the 2D detected face and then combined the features with the matched 3D head model from the database. Although this 3D sensing process is capable of conveniently and inexpensively generating 3D models from limited views, the results obtained are usually subject to the estimation algorithms or matched database and are not suitable for precise reconstructions. Therefore, this process is used as a compromise when the other two sensing processes cannot be implemented.

The real environment can also be reconstructed as a 3D model by means of the depth camera, the environmental understanding cameras, and the KinectFusion algorithm, the last of which was originally developed for 3D reconstruction using the Kinect depth camera [52].



(a.1) *Cyberware* whole body scanner

(b.1) *LMI* HDI Advance R3x sensor

(c.1) *Microsoft* Kinect v2

(d.1) *North Star Imaging* X-View X5000

(a.2) *Creatform* MetraSCAN 210 handheld sensor

(b.2) *Microsoft* Kinect v1

(c.2) *MESA Imaging* SR4500

(d.2) *Toshiba* TOSCANER 20000AV

(a) Triangulation-based laser sensors

(b) Structured-light sensors

(c) Time-of-flight sensors

(d) X-ray Computed Tomography scanners

Figure 2.4: Several commercially available 3D sensing devices [2, 3, 4, 5, 6, 7, 8]

2.3.2 3D Sensors

Sensors are also widely used by AR systems to reconstruct 3D models of the surrounding environment and objects. Several types of sensors adopting various technologies were developed and/or marketed to satisfy the 3D sensing demands. Sensors are usually used to measure the shape and appearance of physical objects or the environment and then generate dense point clouds or polygon meshes to reconstruct the target. Traditional passive-image-based cameras, which are only capable of capturing 2D images without depth information, can be used as 3D sensors with careful calibration, feature matching among images, and/or depth estimation algorithms [60, 65, 66]. Additionally, active 3D sensing devices have a variety of working principles [67, 68]. Figure 2.4 shows several commercially available active 3D sensing devices that adopt different working principles, including triangulation-based laser, structured-light, time-of-flight, and X-ray computed tomography:

- Triangulation-based Laser Sensing Devices: triangulation-based laser sensors usually shine a laser on a subject and utilize a camera to measure the position of the laser's dot. Depending on the distance of the object that the laser strikes, the laser dot appears at different places in the camera's field of view. This method is called triangulation since the laser dot, the camera, and the laser emitter form the shape of a triangle [69]. These types of sensors are usually able to acquire high-quality information to build precise 3D object models, but they are expensive compared with other sensors and require expert knowledge to be operated. Examples are Cyberware Whole Body Color 3D Scanner, NextEngine desktop 3D scanner, and Creaform's handheld HandyScan scanner [2]. Moreover, targets always need to stand still during the whole capturing process, which is difficult in certain situations, such as when creating 3D models of an infant [70].

- Structured-Light Sensing Devices: structured-light devices usually project patterns of light containing many stripes at once or of arbitrary fringes, which allows the acquisition of several samples simultaneously. Working as 3D sensors, they offer several advantages at affordable prices and have attracted a large amount of attention all over the world [54, 71, 72]. Structured-light devices have several capabilities: (1) capturing depth images at a video rate under low light conditions; (2) operating safely and easily for both the scanned object and the user with a similar operation to that of video cameras; (3) being able to solve silhouette ambiguities in pose; (4) simplifying the process of background subtraction; and (5) easily synthesizing realistic depth images of humans [73, 74, 75, 76]. Structured-light devices have not yet begun to dominate the 3D scanning market because they were not originally designed as high-quality 3D sensors but were instead developed for object detection purpose and as part of natural user interfaces [77]. As a result, they typically have low X/Y resolution and a high noise level, which affects their accu-

racy. For example, the first consumer-grade structured-light Kinect v1 had a low resolution (640×480 pixels for RGB images and 640×480 pixels for depth images), which usually results in only acceptable accuracy when reconstructing 3D models. However, researchers are still working on improving that performance. Wijenayake et al. proposed an error-correcting technique to improve the 3D scanning results of the structured-light method [78], and Smisek et al. suggested an algorithm that allowed Kinect v1 to outperform SwissRanger ToF SR4000 in accuracy when measuring planar targets [79].

- Time-of-Flight Sensing Devices: Time-of-Flight (ToF) sensors work differently than structured-light sensors. ToF sensors use active sensors to measure the distance of a surface by calculating the round-trip time of the emitted infrared light, and commercially available ToF cameras usually employ homodyning methods and operate in a continuous mode [80]. ToF sensors do not interfere with the scene in the visual spectrum since they use infrared light. They are usually more expensive than structured-light cameras but are cheaper than triangulation laser type sensors. ToF sensors emerged around the year 2000 because the semiconductor process became fast enough to deal with such devices and were first introduced by Lange and Seitz, who successfully realized an all-solid-state 3D ToF range sensor [81]. Later, Gokturk et al. introduced ToF sensors to the graphics and vision community through integrating a complete ToF sensor with a complementary-metal-oxide-semiconductor chip to develop 3D sensors that are highly cost-effective [82]. One application based on ToF sensor is detecting heart rate through monitoring user's face, which can obtain similar results as traditional ECG device [102]. ToF sensors have capabilities that are similar to those of structured-light sensors, such as the capturing of depth images at normal video rates in low light conditions, color and texture invariance, and the ability to resolve silhouette ambiguities in

poses. However, ToF sensors also have certain drawbacks: (1) they suffer from signal-dependent shot noise, because the process of measuring instantaneous light power with semiconductor substrates involves a conversion from photon energy to electron displacement [83]; (2) they have random noise and notable systematic measurement bias [84]; (3) the images are occasionally compromised by scattering and motion blur problems [85]; and (4) they usually capture depth images at a rather low resolution [77]. Among ToF companies, MESA Imaging produced the Swiss Ranger SR4k family [6], PMD developed the PhotonlCs series [86], Canesta developed the CanestaVision, and Microsoft created the Kinect v2 [47].

- X-ray Computed Tomography Sensing Devices: although conventional Computed Tomography (CT) is a medical imaging technology used to generate a 3D image of the inside part of an object, industrial X-ray CT is able to reconstruct a 3D model of both the internal and external structures of the scanned target using a number of 2D images. These images are obtained with X-ray radiation in several positions around an axis of rotation [87]. As a non-destructive sensing technique, X-ray CT is able to measure both the inner and outer geometries of a solid object without the need to destroy or cut through it. X-ray CT can also have a high resolution or density, for example, X-View X5000, shown in Figure 2.4 (d), has a best resolution of $500nm$. Another advantage is that it can scan several types of surfaces, shapes, colors, and materials with certain densities and penetrable thicknesses [88]. X-ray CT has the following limitations: (1) it can only sense objects within its maximum penetrable thickness, otherwise, the resulting X-ray images will be of low-quality, since the object absorbs too much energy; (2) X-ray is inherently noisy, as is the detector and its amplification, which limits the X-ray CT's performance; (3) most industrial X-ray CTs do not function with live body scans; and (4) scanning a multi-material object could fail if the sensing device cannot detect the changes in

the material during the scanning process [89, 88].

2.4 User Interaction

User Interaction refers to the interactivity between users and the computing device. It can be realized in several ways, such as gesture control, voice commands, and user movements.

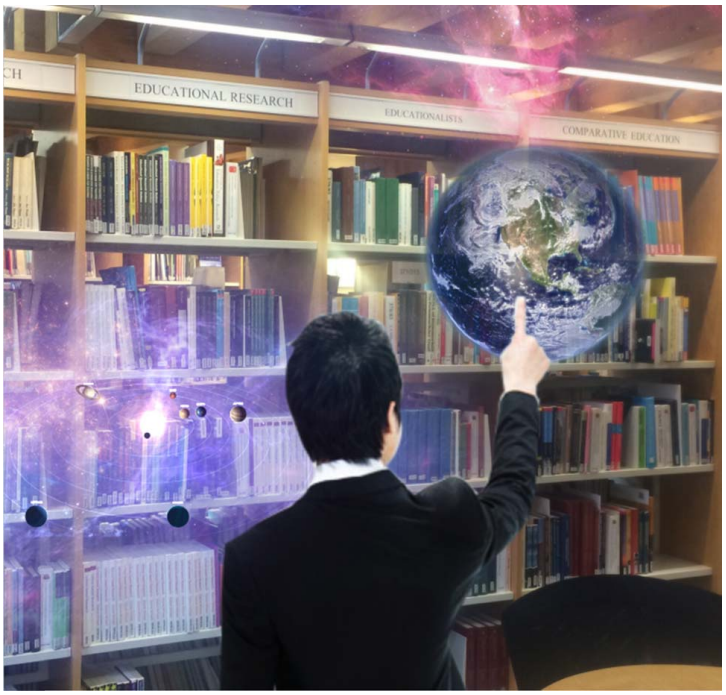


Figure 2.5: Immersive user interaction [9].

Gesture control has attracted a great deal of attention as an important method for user interactions [90]. These methods are mainly based on detecting the hand or finger from the captured image or video feed during the interaction, and estimating or tracking its position to match pre-defined commands [91, 92]. With accurate hand or gesture detection and recognition, immersive user interactions can then be realized, such as in the example of Figure 2.5 [9], which shows a user using his/her bare hands to interact

with a hologram naturally and immersively.

A lot of research has been conducted in this field. For instance, Zhang et al. proposed a "visual screen" by putting a webcam in the center of the computer monitor, and using different colors to segment a hand from the background and enable further finger features matching [93]; Cheng et al. realized a 3D pointing system which allows users to interact with large-size displays using their bare hands and a camera [94]; Hoang et al. positioned one camera to determine if the user's finger was in the detection region, and used another camera to recognize the gesture commands; Wang et al. proved that the finger tips always have the highest ratio curve in a hand contour after Fourier Transform, which could be used for finger tips extraction [95]; Anagnostopoulos et al. found all object contours in the image captured by a webcam, and used hand features like dimensions and shape to wipe out non-hand objects [96]; Zhang et al. mounted a camera on top of a screen to realize touch control, and linked it to a haptic video chat system [66]; Wang et al. proposed a chroma-keying method to extract key information from the background [97, 98]; Gao et al. proposed a way to reduce noise during detection, using low pass spatial filtering [99]; Kang et al. assumed that skin color varied inside a certain range, and indicated that the red component of skin color was in the range of 37 to 60, whereas the green component of the skin color was between 28 to 34. They then used *OpenCV* to perform a faster RGB to YUV conversion in order to make the color ranges more robust to changes in brightness and intensity [100]. The development of the depth camera also brings new methods of gesture interaction, and is widely used for semi-transparent AR systems [101].

While gesture control is not accessible or does not work efficiently with certain tasks, voice control offers an alternative way to interact with computing devices. James and Gurram proposed a system including a first user interface and a voice extension module to realize voice control functions [103]; to ensure high-resolution and deal with large-

size display, Jacobsen et al. presented a method to remotely control devices with a combination of voice commands, hand motions, and/or head movements, providing access to application software that can work on a remote host device, and can successfully overcome physical limitations [104].

2.5 Head Pose Estimation

Head pose estimation is crucial for an AR system, since it plays a key role in seamlessly blending virtual objects with the real environment. Back in 1997, this registration issue was considered "one of the basic problems currently limiting augmented reality" [30].

Various kinds of methods and sensors, such as ultrasonic devices, mechanical devices, inertial devices, magnetic sensors, GPS, optical sensors, and compasses, have been used to solve this problem, but every method has its own limitations [105]. Vision-based marker or markerless methods are also being developed quickly for end-user or industrial applications [106]. One advantage of a markerless head pose estimation is that it allows users to move freely without wearing special markers, which enables it to be used in normal life, instead of just for research or testing purposes.

The head posture of a AR user can be determined from the position and orientation of the device, and estimated by the inertial-measurement unit (IMU) and the iterative-closest-point (ICP) algorithm [101]. The accuracy of this posture estimation is in large part responsible for the tracking performance of the system [107].

Along with the head pose estimation is the virtual environment processing, which receives information from the head posture, the real environment, and the user control components, to process the obtained data and calculate the desired locations in order to augment the real environment with computer-generated holograms.

2.6 Hardware Components

As opposed to most other AR devices, which need to be connected to an external computing machine with a cable or by a wireless connection, recently launched independent AR system Microsoft HoloLens, which greatly advanced this field [108]. As a complete AR system, HoloLens itself integrates all required components, running the Windows 10 operating system with cutting-edge AR technologies. The hardware components of HoloLens are shown in Figure 2.6, and the components are (1) processing units; (2) two pairs of environment understanding cameras; (3) infrared laser projector; (4) depth camera; (5) HD video camera; (6) ambient light sensor; (7) holographic projector; (8) see-through optical waveguide lenses; (9) microphones; (10) built-in speaker; (11) battery [109]. There are also other components, such as an inertial-measurement unit (IMU). The processing units of HoloLens consist of a central-processing-unit (CPU) and a holographic-processing-unit (HPU). Its CPU is a 14nm Intel 32-bit system-on-chip with 1GB of RAM, and its custom-designed HPU is a TSMC-fabricated 28nm coprocessor with 24 Tensilica DSP cores arranged in 12 clusters and occupying another 1GB of DDR RAM and 8MB of SRAM [110]. Therefore, HoloLens can process all required tasks itself, such as detecting the surrounding environment, and interacting with users through gesture and voice commands. HoloLens has attracted considerable attention. For example, Avila et al. presented the basic capabilities of HoloLens [111], Furlan provided a general data flow pipeline for the hardware and user control system of HoloLens to conduct an intuitive experiment [108], and Lu et al. illustrated several examples of the use of HoloLens for immersive analysis and discussed the new opportunities and challenges the system presents for visualization and visual analytics [112].

Figure 2.6: HoloLens hardware components.

2.7 Computing Performance

As previously introduced, an AR system contains several functional components and needs to deal with a large amount of information while operating. Therefore, it requires a rather high computing performance. If it is connected to a computing machine with a cable, it limits the user's movement; if it is portable, the computing performance may be compromised.

As part of our evaluation of the semi-transparent AR system, we developed an AR city data visualization application and measured its computing performances when users were completing required tasks [113].

2.7.1 City Data Visualization

Generating 3D content to visualize an abstract database is an interesting, prevalent, and appealing topic. It helps people better comprehend and manipulate the data. City data visualization can contribute to a quick understanding of what the data stands for, and therefore reduce the work and complexity related to city management and decision-making. As a consequence, a digital, smart, efficient and sustainable city can be expected

to improve and to facilitate the life of its citizens [114].

Without a doubt, city data plays a crucial role in the construction of a smart city. It contains various types of data from the city, and can be used to visualize, analyze, and predict the state of the city. For example, Dong et al. introduced available open data sets from seven Canadian cities, and listed several applications utilizing these data sets for visualization, localization, and analysis [115]. Lv et al. built a 3D Shenzhen city web platform, based on a geographical information system, to enable the visualization of different types of city data, such as 3D building model data, resident information, and real-time and historical traffic data [116]. This demonstrates that the progress of 3D technologies further facilitates the process of 3D city data management [117].

To explore the computing performance of a semi-transparent AR system for city data visualization, we developed a 3D Toronto city data visualization application to evaluate the computing power of HoloLens.

2.7.2 Combining AR and Data Visualization

Most city data visualization applications, such as the two aforementioned examples, are implemented with traditional desktops or laptops, and are shown through daily-use computer monitors. Though this method can satisfy the requirements for city data visualization, it is hard to improve the moving flexibility, the visual quality, the natural level of user interaction, and the grouping with other real devices or objects. However, recent developments in AR technology bring a new way to visualize city data, providing several advantages in city data visualization, especially when compared to traditional display methods:

- *Flexibility:* traditional monitors are usually fixed at one position with limited rotation angles, while semi-transparent AR systems can manipulate and demonstrate large-scale city data (up to 1.4m×1m in our implementation) with more flexibility,

such as on a table, on the wall, on the ground, or even in the air, and with either horizontal or vertical views.

- *Natural interaction:* Semi-transparent AR systems enable a more natural interaction experience than that using indirect interfaces, such as a keyboard or a mouse. For instance, if a user wearing an AR system wants to see more details of a region in a city, he/she can simply move closer to it. They can also interact with the city data using gestures or voice commands.

- *Combination with real devices/objects:* Since AR technology is able to display virtual contents upon the real environment, it can easily be combined with real devices or objects to offer users a special experience. For example, real haptic feedback devices can be placed to geographically overlap with augmented city buildings, and then provide different levels of force feedback that represent the crime rates of corresponding buildings when users touch these buildings. In addition, city data can also be combined and displayed on a real paper map to augment it.

- *Visual quality:* With an advanced holographic projection system, AR systems can also provide vividly augmented contents for users.

Since HoloLens integrates all components into one head-mounted device, it processes city data visualization with its own CPU and HPU, as mentioned previously. Therefore, we conducted experiments to track its computing performance, including CPU usage, HPU usage, memory occupation, SoC (system-on-chip) power consumption, and system power usage. This evaluation is of great importance to enable AR developers to be aware of the computing capabilities while developing their AR applications.

2.8 Quality-of-Service

Many Quality-of-Service (QoS) models have been used in traditional fields, such as network conditions, to indicate the quality and functionality of various technologies used in applications with little human involvement [118]. QoS models mainly use technical parameters to evaluate the quality and functionality of systems. For instance, QoS parameters, such as capacity, data rate, end to end latency, number of connects, cost, spectral efficiency, and channel bandwidth, are used to define network communications [119, 120].

Many works related to network QoS have been conducted. For example, Banerjee et al. used a QoS access point to allow the transmission between Bluetooth units and WLAN client stations [121]. To overcome the diminished QoS by wireless network service providers, such as cell terminal changes, Agrawal et al. proposed a system to provide and maintain high-level QoS [122]. To improve the performance of network intrusion detection and protection functions with high-speed networks, Bulajoul et al. used QoS configuration and parallel technologies to overcome the weakness of such networks, including dropping packets under heavy traffic and high-speed, and being unable to deal with multiple packets simultaneously [123].

Network QoS are also widely used for smart cities, Internet of Things, and cloud computing. With the development of the smart city, different QoS parameters have been proposed to suit aspects of human life, such as transportation, infrastructure, buildings, governance, energy control, communications, and health care [124, 125]. Since emerging Internet-of-Things requires low-power remote wireless sensors as a key technology, its requirements for networks are rather high. Therefore, Gaddour et al. proposed QoS-aware routing for both static and mobile IPv6-based low-power [126]. Chen et al. proposed measurement techniques to evaluate the cloud gaming systems to decide if the systems deliver good user-perceived QoS [127]. Their results show that cloud gaming

systems with adaptable frame rates, better graphic quality, shorter server processing delays, and lower network bandwidth consumption, can receive higher QoS scores. In addition, Abdelmaboud et al. used a systematic mapping study to address the QoS of cloud computing [128].

Besides network areas, QoS are also widely used in other fields, such as power consumption and smart grids. For instance, You et al. selected frame-per-second as the key QoS parameter while using dynamic voltage and frequency scaling techniques to balance the requirements of QoS and the power consumption of GPUs embedded in mobile systems [129]. Sahin et al. applied wireless sensor network concepts with a smart grid, which is a modern power grid system with advanced sensing, monitoring, control, and communication capabilities [130]. They evaluated harsh smart grid environmental conditions with multi-path and single-path QoS-aware routing algorithms to estimate the capability differences.

2.9 Quality-of-Experience

Though with QoS models it is easy to develop measurable standards to indicate the quality and functionality of technological systems with little human involvement, they fail to evaluate user's satisfaction level with interactive multimedia applications or devices [33]. Thus, many research efforts have been conducted to evolve user-centric QoE evaluation models.

In the field of virtual reality, Steuer proposed vividness and interactivity as the two essential elements influencing the user experience [10]. According to his paper, vividness is defined as the representational richness of a mediated environment, which is then further divided into sensory breadth and sensory depth, referring to the number and resolution of the presented information, respectively. The second element, interactivity, is used to describe the extent to which users can participate in modifying the content

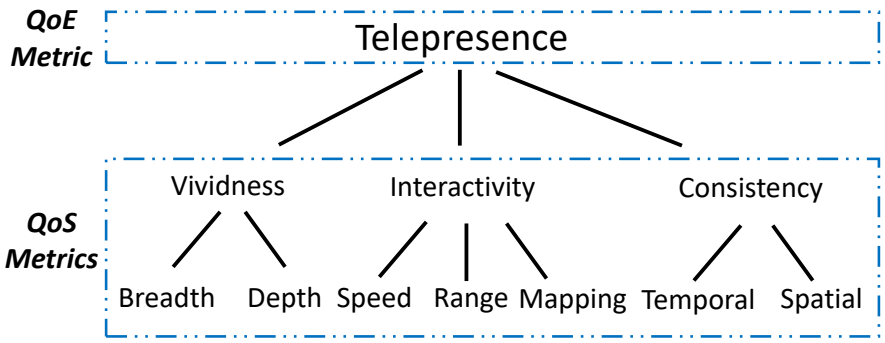


Figure 2.7: Telepresence QoE and QoS metrics [10, 11].

and form of a mediated environment in real time, and is evaluated with speed, range, and mapping. Later, consistency is added as another essential element for distributed interactive multimedia environments, and is further classified into temporal and spatial aspects [11]. Figure 2.7 illustrates the dimensions of telepresence with these three essential elements and their sub-elements.

Considering haptics (the sense of touch) as an important aspect of a virtual environment, Hamam et al. proposed a QoE evaluation taxonomy associated with Haptic-Audio-Visual Environments (HAVE) [12]. They define QoS as a part of QoE, since it affects a user's general experience. Therefore, as shown in Figure 2.8, they adopt QoE as the integration of QoS and the User Experience (UX): QoS is divided into network/standard QoS (e.g. Latency and response time) and HAVE rendering qualities (e.g. graphics, haptics, and cross modality); and the UX is classified as user state measures and performance measures.

Because different applications and devices are designed to fulfill users' various needs, there are currently no unified QoE models to suit all of the applications and devices. The European Network on Quality of Experience in Multimedia Systems and Services (QUALINET) has been extending the notion of network-centric QoS to QoE in multimedia systems to develop both subjective and objective quality metrics [34]. They distinguish

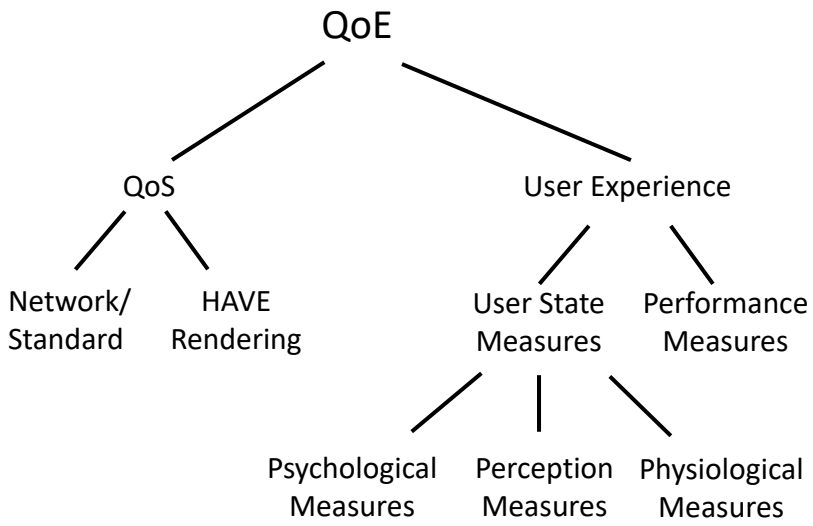


Figure 2.8: QoE taxonomy for haptic-audio-visual environments [12].

one perception path and one reference path to explain the actual quality formation process, define QoE as "the degree of delight or annoyance of the user of an application or service", and emphasize that QoE for visual and audio feedback needs to be combined with haptic feedback to provide a joint multi-sensorial and multi-dimensional QoE metric [34].

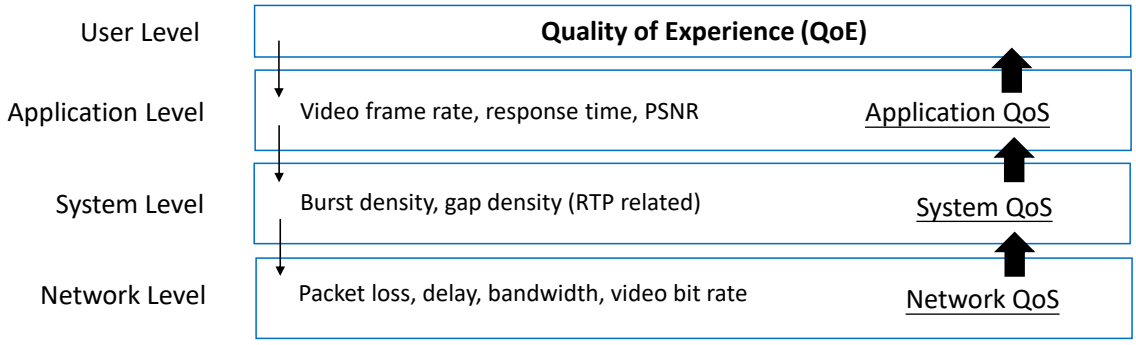


Figure 2.9: Conceptual relations of QoE and QoS at different levels of the communication model [11].

Based on the layers of the communication network model for a distributed interactive multimedia environment, Wu et al. illustrated the conceptual relations with the QoS and QoE of different levels [11]. Figure 2.9 shows that network levels mainly consist of QoS packet loss, delay, bandwidth and video bit rate; system levels contain burst density and gap density; application levels have important QoS parameters such as video frame rate, response time, and others; while user levels are generally evaluated with QoE. Each high level is affected by the QoS of its lower levels, and the final QoE evaluation can be affected by all lower level QoS parameters.

There are also several QoE models proposed for other fields, such as http streaming and 4K video delivery. For example, Ge et al. presented a system architecture for QoE-assured 4K video delivery [131]. In addition, Mok et al. investigated the relations among HTTP video streaming's network QoS, application QoS, and QoE [132]. Bentaleb et al. utilized a software-defined-network in their architecture to optimize QoE in HTTP adaptive streaming, while maximizing per player QoE by addressing scalability issues such as network resource underutilization, video instability, and QoE unfairness [133]. There is also some research related to AR QoE evaluations. For instance, Keighrey et al. presented a QoE model for AR and VR speech and language evaluation; and Evans et al. evaluated HoloLens' computing capabilities and spatial mapping accuracy for their assembly instruction application [134, 135].

Different multimedia applications and devices have various structures, and are specifically targeted at users' different needs. Currently, there are no well-recognized QoE models to suit all applications and devices. In addition, QoE parameters in the field of AR are not yet well defined and selected. To develop a framework for general AR QoE evaluation, we propose a QoE taxonomy for semi-transparent AR system, in order to comprehensively measure its performances from a real user's perspective, and build a fuzzy-inference system for it.

Chapter 3

The Proposed QoE Evaluation Framework for Semi-Transparent AR System

In this chapter, a detailed description of the proposed Quality-of-Experience (QoE) framework for semi-transparent AR systems is given. Our proposed AR QoE framework is composed of one taxonomy and one Fuzzy-Inference-System (FIS) model, and the FIS model is based on the taxonomy. We first introduce the method adopted to obtain the proposed QoE taxonomy, then talk about the user study we designed to collect training and testing data for our QoE model, and finally select high-level QoE parameters as inputs to build a Fuzzy-Inference-System (FIS) model to estimate the overall QoE rating scores. We also conducted case studies to further explore the relationships between QoE parameters and technical QoS parameters with functional components of Microsoft HoloLens AR system. Details of each component's experiment design, testing environment, user information, and results are explained in the following parts.

3.1 Proposed Quality-of-Experience (QoE) Taxonomy

As previously mentioned, several QoE models have been proposed for virtual environments, haptic-audio-visual environments, networks, video streaming, and many other fields, as shown in Figure 2.7, Figure 2.8, and Figure 2.9. Since different fields use various key parameters for their QoE evaluation, we selected the QoE parameters that affect semi-transparent AR experience from some existing models [34, 10, 12], and integrated the feedbacks from our designed user study, to propose a conceptual QoE taxonomy to evaluate such devices. For instance, our QoE influential parameters "content quality" and "user interaction" are derived from "vividness" and "interactivity" in Steuer's QoE model for telepresence, and are explored with more detailed lower-level parameters [10]. These parameters and their levels in the taxonomy are also further adjusted based on feedback from our user study, to ensure they can reflect users' true experience of semi-transparent AR system.

Based on the special attributes of semi-transparent AR systems, our taxonomy is mainly composed of four high-level (1st-level) parameters: content quality, hardware quality, environment understanding, and user interaction. To clearly illustrate and extend our proposed taxonomy, we further explore each high-level parameter's lower-level (2nd- and 3rd-level) aspects (e.g. speed and precision). Figure 3.1 demonstrates the entire QoE taxonomy including all parameters for semi-transparent AR system evaluation [136]. More details are given in the following parts.

3.1.1 Content Quality

Content quality is the core of an AR system. Two significant factors that influence the content quality are: information realistic level and required user focus level.

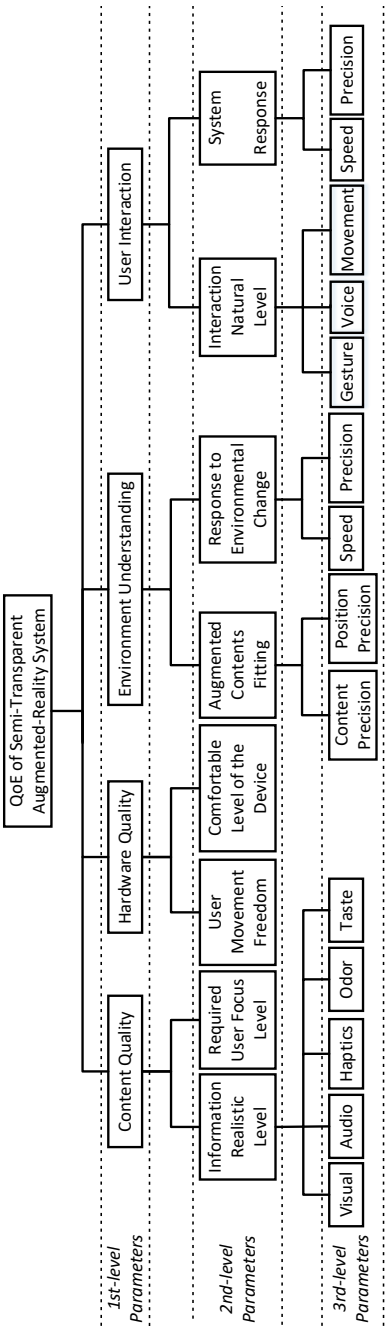


Figure 3.1: Proposed QoE taxonomy for semi-transparent AR system evaluation.

Firstly, the realistic level of the contents, generated by an AR system, should at least meet a user's basic requirements. For instance, an augmented "cup" on the table should not look like a "book", and a voice from the front should not sound like it comes from the back. Generally, AR usually has lower realistic content requirements than VR, which demands vivid contents to totally immerse users in the virtual world. Besides commonly mentioned visual and audio contents, other properly integrated sensory contents, such as haptic feedback, odor, and taste, can also increase a user's interaction experience [137]. Haptics, which is related to the sense of touch, can greatly improve the interactions through proper force feedback. Users will feel as though they were interacting with "real" objects or humans [138, 139]. Digital odor and taste have also attracted lots of attention, since they can make the AR experience more realistic, and therefore enhance the user's overall experience. For instance, when a user wants to interact with an augmented "flower", he/she can touch it with haptic feedback, smell the fresh scent and taste it with digitally generated odor and taste. For these reasons, we also listed them on our QoE evaluation taxonomy. However, to focus on the general AR experience, we did not put them in our QoE user study questionnaire, as the semi-transparent AR system we selected does not provide these options.

Secondly, different applications demand various levels of user focus, which can have an impact on the required realistic level of the content. For example, intense shooting games make users mainly focus on the target, instead of the surrounding environment. As a result, the realistic level of the target should be higher than that of the background. However, for AR museum tourism, the user will usually be relaxed, and may look around each hologram, which means that high-quality AR contents are necessary to improve the user experience. Similarly, if augmented text or words can deliver precise information to a user, its realistic level is considered acceptable, regardless of the font, size, or position. An example is an AR navigation system, if the "direction arrow" can accurately show

the way users need to go, it will be considered as acceptable.

3.1.2 Hardware Quality

Since users usually need to move around wearing an AR device to interact with the augmented contents and the surrounding environment, AR hardware quality can greatly influence the user experience.

The first concern should be the user's freedom of movement. Traditional semi-transparent AR devices usually require cables connected to powerful external computing machines to transmit and process corresponding contents, which largely restricts the movements of users. With the recent developments of microprocessors and increased computing capabilities, HoloLens itself integrates all required components to become a complete AR system running the Windows 10 operating system with cutting-edge AR technologies. It processes data and computing with an embedded CPU and a specifically designed holographic-processing-unit (HPU), which dramatically increase its capability to process AR holograms. Since it is a single integrated device, users can move freely while wearing it, and can thus better interact with both the real environment and the augmented virtual holograms, instead of being restricted by the cables.

Another evaluation criterion is the comfortable level of the device. As most AR devices adopt head-mounted display (HMD) that is designed to enable the user to promptly adjust the display screen's position based on their movements and facing directions, the user's head needs to bear the whole weight of the device at all times. For hand-held AR devices, e.g. cellphones with Pokemon Go, though they can be carried around easily, they usually do not offer 3D visualization, may cause hand fatigue, and have limited interactions with the surrounding environment. Therefore, the quality of the design and materials closely affect the user experience.

3.1.3 Environment Understanding

As previously mentioned, compared to VR devices, AR devices have a special requirement to understand the real surrounding environment, in order to lay a foundation before "augmenting" the reality. AR devices need to detect the locations of real-world objects around the user, estimate the user's relative position and facing direction, and occasionally, even recognize the objects or humans. Therefore, we consider environment understanding to be a unique aspect of the semi-transparent AR system QoE evaluation model.

Though AR contents can be augmented on the environment based on simple depth detection, some AR devices still require users to reconstruct the 3D models of the room or place in advance, to enable faster interactions.

We divide this category into two sub-categories. Firstly, the augmented elements need to fit the surrounding environment in both content and position to make them more meaningful. This matching between the real environment and the augmented components will make the application realistic and natural for the user. For instance, the generated virtual object on a real table should be a "cup" instead of a "car", and the "cup" should be at a proper location of the table, rather than floating in the air. Secondly, since users frequently change their positions and looking directions, AR applications should be able to respond quickly and precisely to environment changes to avoid mismatches between the real and augmented environments.

3.1.4 User Interaction

Unlike applications that only display pre-designed contents, AR applications involve a certain amount of user interactions. Thus, evaluating user interactions becomes an important part of our QoE taxonomy.

The first component is the natural level of interaction, meaning the definitions of

the ways that users interact with the application. For instance, wave hand right or left to switch content, press finger forward to push a "button", and say "close" to shut down the application. If users want to shake hands with the hologram of another user, they can simply move their hand forward to do it. In our taxonomy, we summarize these defined interactions into commonly utilized methods such as gesture, voice, and movement commands.

For gesture commands, users can simply perform the pre-defined gesture to activate the corresponding action; voice commands can be used to match stored voices, or can be further increased and combined with natural language processing technique to create more natural user interactions; body or head movements can be tracked by IMU sensors embedded in the AR device and the ICP algorithms.

Another inevitable component is the precision and speed of the system's response to user interactions. Taking much time to respond to a user's interaction commands, or incorrectly interpreting certain commands, can degrade a user's experience with the semi-transparent AR system, or even make them lose interests.

This task is challenging because an AR system interacts with various complex environments. Gestures may blend with different backgrounds, users can have different hand sizes and gesture styles, voice commands may be affected by other noise, and movement tracking can also be degraded by mismatching and inaccuracy.

3.2 Fuzzy-Inference-System (FIS) Model Design

We conducted a user study to evaluate the QoE parameters in our proposed taxonomy for semi-transparent AR system evaluation. We asked the participants to fill out a questionnaire regarding all of the QoE parameters in the taxonomy (except haptic feedback, odor, and taste), and give an overall rating score (0-100) for the quality of their AR experience. Based on their ratings, we selected high-level QoE parameters as the inputs

for the Fuzzy-Inference-System (FIS). Details of the user study will be given in the next section, where we will first introduce the design of our FIS model, the general architecture of which is shown in Figure 3.2.

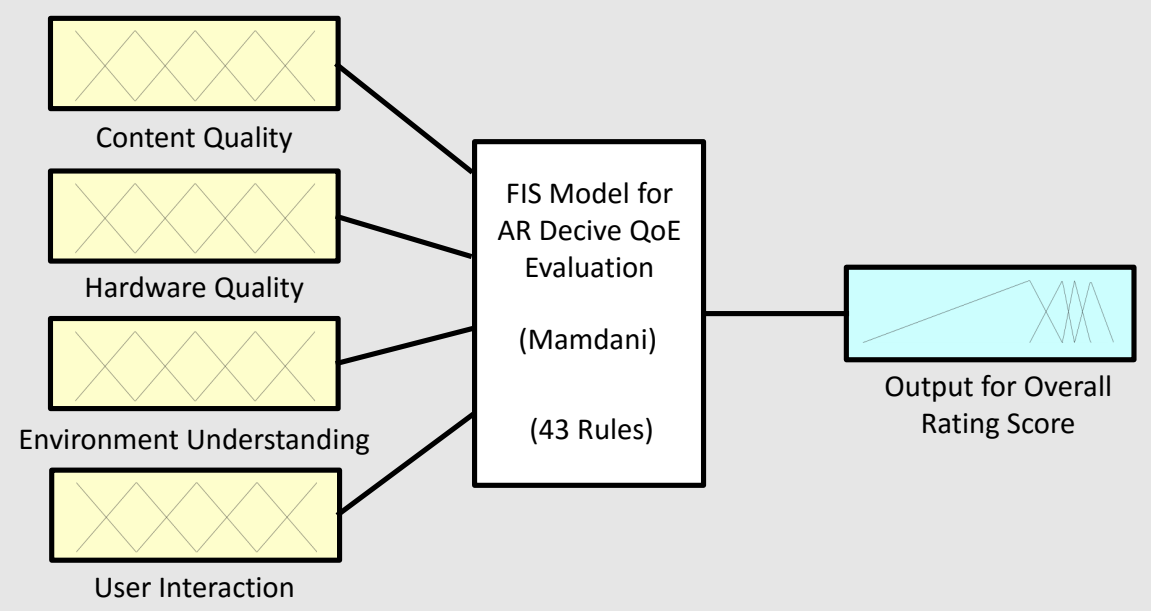


Figure 3.2: General architecture of the FIS evaluation system.

A QoE evaluation focuses on the subjective feeling of the users, and its parameters are also subjective and fuzzy in nature, which makes it too complicated to represent them with classic linear approaches. For example, if we use a three-dimensional (3D) curve to represent the score mapping from two high-level parameters (content quality and environment understanding) to the overall rating score, when setting the other high-level parameters as average values (50), we obtain their nonlinear relations. Therefore, we designed a FIS model based on part of the user data, to map the input parameters into the output using fuzzy logic, namely, building fuzzy IF-THEN rules with natural language to formalize the reasoning process of human language, and generating a decision based on the evaluation questionnaires.

3.2.1 Selecting the Fuzzy Inference System Type

Fuzzy logic deals with the imprecise terms in a way that is similar to how our brain assimilates information (e.g. the temperature is hot, not the temperature is 23 degrees), and then responds with precise actions. In this way, it is able to reason with uncertainties, vagueness, and judgments.

Generally, a FIS consists of four modules: fuzzification module, knowledge base, inference engine, and defuzzification. It is flexible to add or delete rules, which is built on top of the acquired knowledge. In our design, we selected the well-known and commonly-used Mamdani FIS [140]. Mamdani's method first applies input fuzzification to the inputs with fuzzy operators, and then obtains a new fuzzy set with a fuzzy implication operator. Outputs of all rules are fused into a single fuzzy set using a fuzzy aggregation operator. To calculate a crisp output value for the decision problem (e.g. final score of the system evaluation), a defuzzification method called Centroid can be used.

3.2.2 Choosing the Input and Output Variables

In the presented taxonomy, we have introduced four high-level parameters and their corresponding low-level ones. To make the model more manageable and reduce the number of possible combinations, we chose the high-level parameters, including content quality, hardware quality, environment understanding, and user interaction, as the input variables of the FIS. The model then generates a crisp overall rating score ranging from 0 to 100 as the output variable.

To unify the measurements for input and output variables, we transfer all input answers I_{in}, which are from 1 to 5, into percentage values I_p, ranging from 0 to 100 as

$$I_p = \frac{I_{in} - 1}{N - 1} \times 100 \tag{3.1}$$

where N represents the number of answer options, which is 5 in our case. Therefore, the

raw input data is divided into five groups: 1 mapped to 0, 2 mapped to 25, 3 mapped to 50, 4 mapped to 75, and 5 mapped to 100. All following contents mentioning input values are represented with this percentage format.

3.2.3 Defining Membership Functions

Fuzzy set theory deals with degrees of truth and degrees of membership, and membership functions (MFs) represent the fuzzy subsets of each variable. Here we first defined MFs for variables of high-level parameters. Since each question in our questionnaire regarding high-level parameters has five answers (1 to 5) to choose from, we divided each input variable into five equal MFs with triangular shapes, namely, very poor, poor, fair, good, and excellent. Each input value was then transformed to the corresponding MF based on its degree of membership. For instance, the value 50 is "Fair", and the value 75 is "Good". In fact, our user study results show no rating scores in the "very poor" range, which was thus omitted.

As the overall rating score is more dispersive than the input score, we used the Fuzzy c-means (FCM) clustering method to define the output's fuzzy sets before generating MFs, similar to the approach used in [12]. Since all of the scores from the questionnaires were larger than 50, we divided the outputs into 4 groups. The detailed results are given in the next chapter.

3.2.4 Deriving Fuzzy Rules

Since we have defined MFs for each high-level parameter as well as the overall rating score, user data from questionnaires can then be transferred to the corresponding MFs to build fuzzy rules. For example, if a user rates the values for content quality, hardware quality, environment understanding, user interaction, and overall rating as 75, 50, 100, 75, and 85, respectively, these values can then be interpreted into MFs as Good, Fair,

Excellent, Good, and Good, respectively, and a fuzzy rule can be derived as:

IF content quality is *Good*,

AND hardware quality is *Fair*,

AND environment understanding is *Excellent*,

AND user interaction is *Good*,

THEN the overall rating is *Good*.

Based on this method, each sequence of user data can then be transformed into the format of a fuzzy rule, as listed above, and can be integrated into the fuzzy rule sets to influence the system's output. With several fuzzy rules, when the FIS system is given a set of inputs, it can automatically judge if the final rating should be poor, fair, good, or excellent, and output the result.

3.2.5 Generating Output

Given a set of inputs, based on the fuzzy rules, the FIS model can generate corresponding output. As our system is a Mamdani-type FIS, a defuzzification of the results is required to obtain a crisp value as the output.

Defuzzification is the process of generating a quantifiable result in crisp logic, with given fuzzy sets and corresponding membership degrees. It maps a fuzzy set to a crisp set. In our implementation, we utilized a centroid calculation to calculate the center of gravity of the curve describing the output. Therefore, the FIS output becomes a crisp value representing the overall rating score obtained from our model.

3.3 Validation of QoE Parameters with Technical QoS Parameters

To further explore the relationships between QoE parameters and technical QoS parameters, we designed and conducted several case studies to validate the QoE parameters with technical QoS parameters. With the AR functional components, we can find the technical QoS parameters that affect specific QoE parameters in order to improve them.

Based on the operational mechanism of a semi-transparent AR system, we divided the system into seven main functional components, and designed a series of experiments to evaluate the performance of each component: a head localization experiment, to compare the head posture estimation results from HoloLens with a ground-truth record from OptiTrack; a 3D content reconstruction system, to reconstruct 3D models from real humans for the AR system; a city data visualization application, to test the computing performance of HoloLens; a real environment reconstruction experiment, to evaluate the differences between the reconstructed model and the real environment; a spatial mapping experiment, to measure the gap or overlap between an augmenting hologram and the target mapping surface; a hologram visualization experiment, to calculate the deviation between a visualized hologram and its corresponding real object; and a speech recognition experiment, to test the reliability of user control through voice commands.

3.3.1 Head Localization

The head localization experiment was designed to evaluate the accuracy and stability of HoloLens' head posture estimation. Since AR systems usually need to reconstruct a 3D model of the surrounding environment before augmenting holograms onto it, they constantly track the user's position and posture. This functional component mainly affects the environment understanding parameters, especially its augmented contents

fitting. For instance, if the head localization results are not accurate, the augmented content, which is supposed to be displayed three meters from the user, is displayed four meters away instead. It may also influence other QoE parameters, such as user interaction, because there will be errors in matching user commands with AR contents.

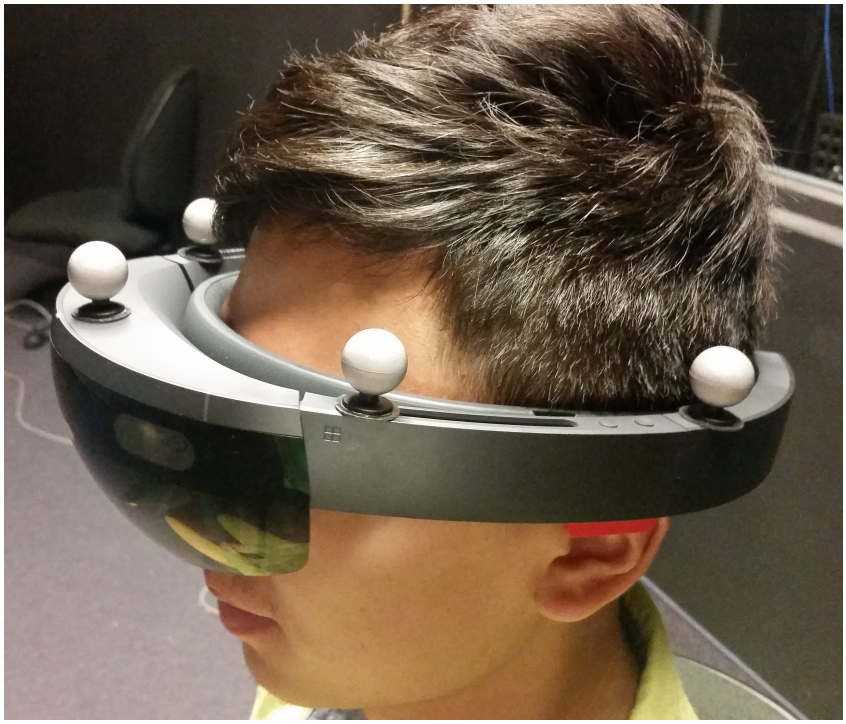
To explore the drifting effects of the inertial-measurement unit (IMU) and iterative-closest-point (ICP) evaluation, participants wearing HoloLens were asked to test various conditions: moving or rotating the head at high or low speeds in the x, y, and z dimensions separately. The head localization performance was then evaluated in terms of Euclidean distances, which represent the distance between two points in a metric Euclidean space, by comparing the results from HoloLens with the ground-truth from OptiTrack.

In this experiment, the OptiTrack system was employed to record the ground truth for head localization. OptiTrack is an accurate marker-based motion capture system, with a distance error of ± 0.3 mm and a rotational error of ± 0.05 degrees. In this experiment, six OptiTrack sensors were placed in a circle with a 2 m radius to track the positions of IR markers mounted on the HoloLens system. The OptiTrack system (Flex V100 camera with the software Arena v1.7) was set up in the room, as depicted in Figure 3.3(a), and its tracking markers were attached to the HoloLens unit as shown in Figure 3.3(b).

3.3.2 3D Content Generation

Since 3D content is the basis of an AR application, it can also affect QoE ratings through parameters, such as the information realistic level and augmented contents fitting precision. Therefore, we developed a system to scan, reconstruct, select, and generate 3D human models from reality. The general architecture of the system is shown in Figure 3.4. Our developed scanning device rotates around a subject to obtain 360-degree views, and generates the reconstructed 3D model. After post-processing, the 3D model can then be integrated with the HoloLens AR system.

(a)



(b)

Figure 3.3: Experimental setup for the head localization experiment. (a) OptiTrack cameras placed in a circle with a 2 m radius. (b) User wearing HoloLens with IR markers.

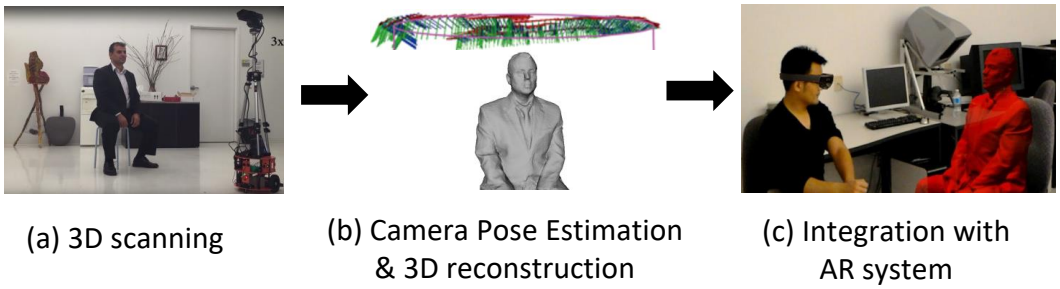


Figure 3.4: Generating and integrating reconstructed 3D content with an AR System

The proposed scanning device is shown in Figure 3.5, and contains a composite sensor, a tripod, a Microsoft Surface tablet, and a robot. We chose the Microsoft Kinect v2 sensor as our depth sensor, which adopts time-of-flight range detection technology to observe depth images of the subject with detailed distance information. Because Kinect v2 requires an extra power source, we cut off its power adaptor cable and added an extra portable battery to help increase mobility.

In order to realize the automatic scanning of a human subject, we mounted our composite sensor on a tripod and fixed both of them to a robot, programmed to rotate around the subject with an approximate radius 1 meter. The robot we adopted is the DFRobot's HCR Mobile Robot[1], a two-wheel drive platform. We used the Arduino Mega microcontroller to control the robot's movements. The robot comes with two 12V direct-current (DC) geared motors, and its reduction ratio is 51:1, with an encoder resolution of 663 pulses per round. Since DC motors may generate inconsistent motion, we utilized a proportional-integral-derivative (PID) controller to improve the precision. After several rounds of testing, we set the left-wheel speed as 13 rounds per minute and the right-wheel speed as 16 rounds per minute. The microcontroller communicates with the motor driver through serial communications, and receives data from the composite sensor by an inter-integrated circuit interface.

[1]HCR Mobile Robot: http://www.dfrobot.com

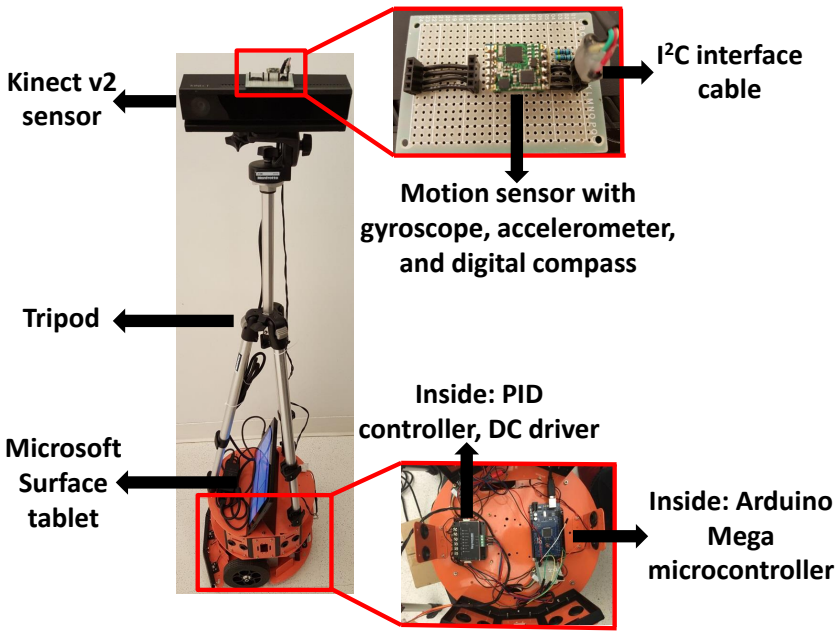


Figure 3.5: Developed scanning device with a composite sensor, tripod, Microsoft Surface tablet, and robot.

For mobility and proper payload, we used a tablet as our computational resource, instead of a laptop. We decided on the Microsoft Surface tablet to fulfill Kinect v2's high demand for graphic processing units and USB connection. This tablet can also easily be placed on the robot, as shown in Figure 3.5. Since both the Kinect v2 and the robot microcontroller interact with the tablet through USB serial connections, we used a USB 3.0 four-port hub from Unitek to extend the tablet's single USB port into multiple ones. Thus, the tablet could successfully receive the data streams and process them [117].

To evaluate the 3D model reconstructed with our system, we compared the scanning results with the ones obtained from Cyberware, an expensive laser-scanning system. The comparison results are given in next chapter.

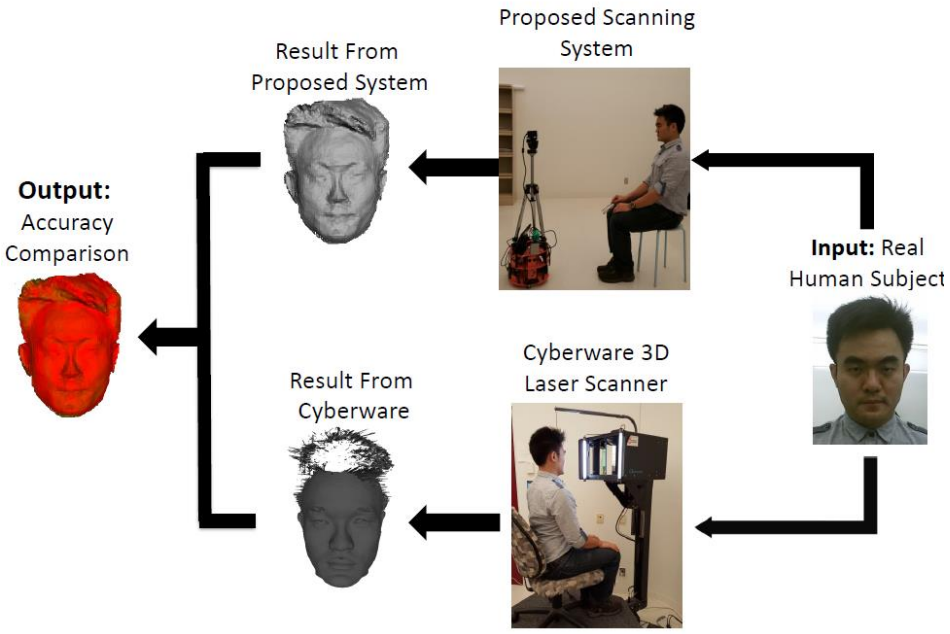


Figure 3.6: Experimental setup for the evaluation of proposed 3D scanning system.

3.3.3 Computing Performance

Since many QoE parameters, such as content quality, environment understanding, and user interactions, all require the computing power of an AR system to process, the computing power is crucial to the user experience of an AR system. Quick and precise processing can greatly improve the AR experience. As previously mentioned, to test the computing performance of HoloLens, we designed a Toronto city data visualization application to track its performance.

A. City Data Pre-Process

We first obtained a city of Toronto 3D model and 220 open city data sets from the City of Toronto website. The original city model was composed of several 3D massing AutoCAD files in .dwg format, which is not suitable for HoloLens, so we converted them to .fbx 3D files. The open data sets cover a variety of city areas, including transportation, city

Table 3.1: Data table description from a red-light camera file

Name	Description
Road 1	First road of intersection
Road 2	Second road of intersection
X	Easting in MTM NAD27 3 degree Projection
Y	Northing in MTM NAD27 3 degree Projection
Longitude	Longitude in WGS84 Coordinate System
Latitude	Latitude in WGS84 Coordinate System
Object ID	Unique system identifier

government, business, health, and so forth. Since they were stored in several file formats, such as XML, CSV, and XLS, we first converted them all into XML files by extracting the meaningful information. We then selected the data sets with complete information, and processed them for visualization and interaction.

The processing steps vary depending on the contents of each data set. The read-me file of each data set usually contains detailed descriptions, and we need to process this information to geographically map the related data with the 3D city model. Table 3.1 is an example that shows the description we extracted from a red-light camera file. Since this file contains position data with x and y coordinates in a Modified-Traverse-Mercator (MTM) coordinate system, we can directly and accurately map them with the reference frame of the city model.

However, some files do not contain straightforward MTM coordinate information, and require further transformation. Take Table 3.2 as an example, which stores the location data of washroom facilities in a Geographic Information System (GIS) with latitude and longitude. The solution we adopted is to apply a transverse Mercator map projection on the GIS coordinate to transform it into a MTM coordinate with the related parameters

Table 3.2: Data table description from a washroom facilities file

Column name	Value
Washroom building asset ID	Number of the id
Park name	Name of a park
Building name	Name of a building
GIS coordinate	(longitude, latitude)
Access	Public, private or none

listed in Table 3.3 [141]. Through this process, we successfully obtained the location of the buildings on the city model's coordinate system.

Table 3.3: Parameters of transverse Mercator map projection

Name	Symbol	Value
Equatorial radius	a	6378.2064 km
Flattening factor	f	1 / 294.9786982
Latitude of origin	ϕ_0	0 degree
Central meridian	λ_0	79.5 degree
Scale factor	k_0	0.9999
False easting	E_0	304.8 km
False northing	N_0	0 km

In addition to location information, many other types of data are also used to visualize Toronto city data. For instance, when a data file containing the locations of taxicab-stand is loaded, the maximum number of taxicabs permitted at each data point can be used in comparison with others to allow them to be visualized differently. Moreover, the data set of bicycle shops in Toronto can be divided into two groups with different colors, based on whether or not the bicycle shop provides a rental service. We stored the

location and other related information as readable XML files, to be used as the source files for our Toronto city data visualization.

B. Integration with HoloLens

We designed a HoloLens application based on the previously processed Toronto city model and data sets. Since HoloLens runs a Windows 10 operating system, we utilized Unity engine, C#, and the library HoloToolKit for our implementation.

Since the source files of the Toronto city model are rather large, we scaled them down before importing, to optimize the system's performance. The complete city model is made of several tiles, so we can also select a few tiles to build a sub-region of Toronto. In addition, a grey plane is placed underneath the 3D city model to improve visibility. Figure 3.7 shows the visualization results of the city of Toronto 3D model, and the menu to load the city data files. Once we select "Load File List", we can view a list of city data types for visualization.

All open data sets have been stored in readable XML files with MTM coordinate and other related information. We designed a menu above the city model for users to choose the desired data set for visualization. Once a data set is selected, its data can be easily mapped on the corresponding locations on the 3D city model. We also attached them together to enable the data positions to change with scale adjustments of the city model. Then, cuboids with various heights and colors are placed to represent different data properties. All stations are represented with cuboids, whose heights depend on the chosen value type, as shown in Figure 3.8. Red means high values, while blue means ordinary values. We can also select other data sets from the menu to visualize, such as places of interest, taxicab stations, and bicycle shops.

Interactions between users and augmented content are crucial to the AR experience [142]. Thus, we integrated several interaction designs into our implementation. Holo-

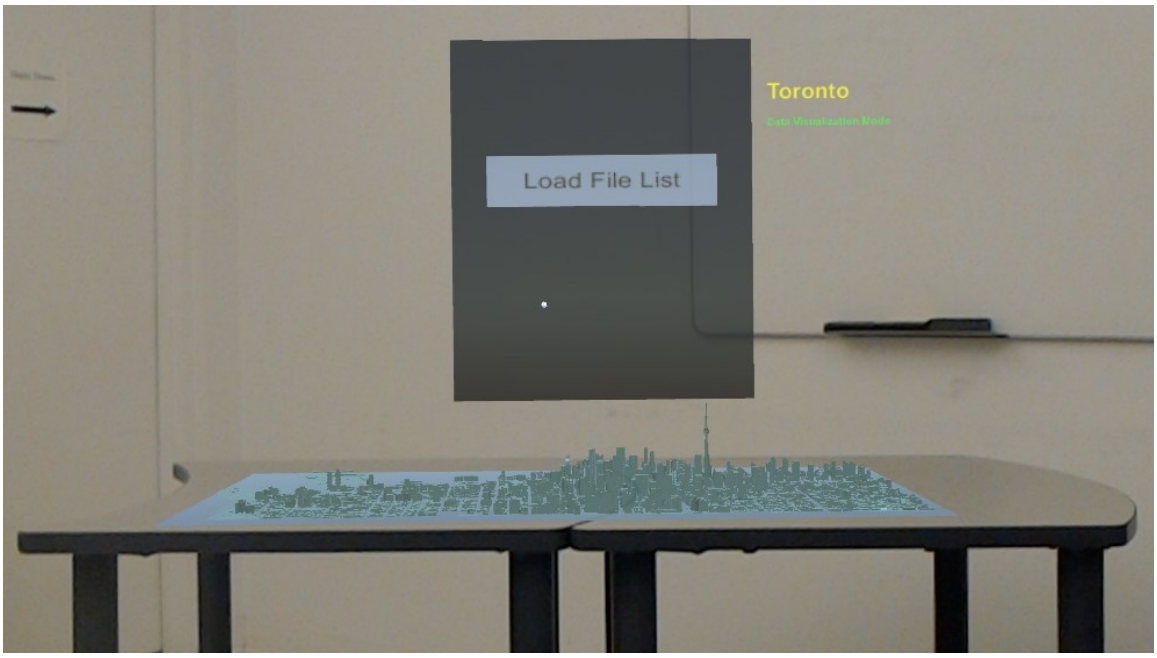


Figure 3.7: Visualizing the city of Toronto 3D model (without city data) on a grey plane using HoloLens

ToolKit, the official library provided by Microsoft, has already defined several interaction types, such as gaze, gesture, and voice. We utilized them to facilitate the development of our project.

Gaze input can be used to select an object on HoloLens. It displays a cursor at the center of the vision scope to determine the direction of a ray to hit the selected object. The option on a menu selected by gaze input will usually change size or color to provide feedback. We can also use gaze input to select a specific building on the Toronto city model.

When it comes to gesture input, the gestures that we have applied to interact with the city model and the data objects are called air-tap, manipulation, and navigation gestures. Since gesture input is detected by the camera embedded in the front part of HoloLens, the user needs to put his/her hand in front of HoloLens to interact. The

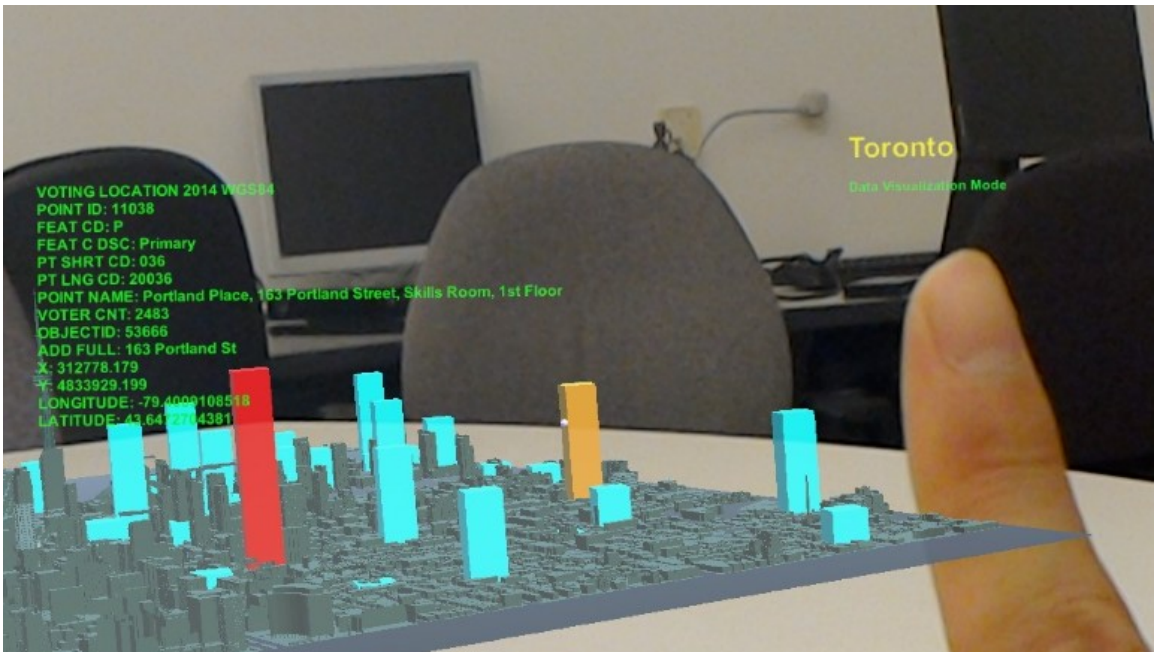


Figure 3.8: Interactions between a user and augmented content

real finger in Figure 3.8 shows a user making gestures to interact. When an option is selected, the air-tap gesture can be applied to confirm the choice. For example, when we gaze at a building on the Toronto city model, we can use air-tap to "click" it, and then the building will change to an orange color and display detailed information with green sentences, shown in Figure 3.8. Manipulation and navigation gestures can be used to rotate the city model, change its positions, adjust its size, and so on.

Voice commands are also used in our implementation, and they can replace certain gesture inputs. For example, the user can say "select" to confirm a selection instead of using the air-tap gesture. Besides, Figures 3.7-3.9 are all captured with the voice command "Hey Cortana, take a photo".

In this part, we use the example of visualizing Toronto voting data to illustrate our visualization results. As aforementioned, after the user gazes at the "Voting Location 2014" option on the menu, and selects it with an air-tap gesture, all voting data is shown

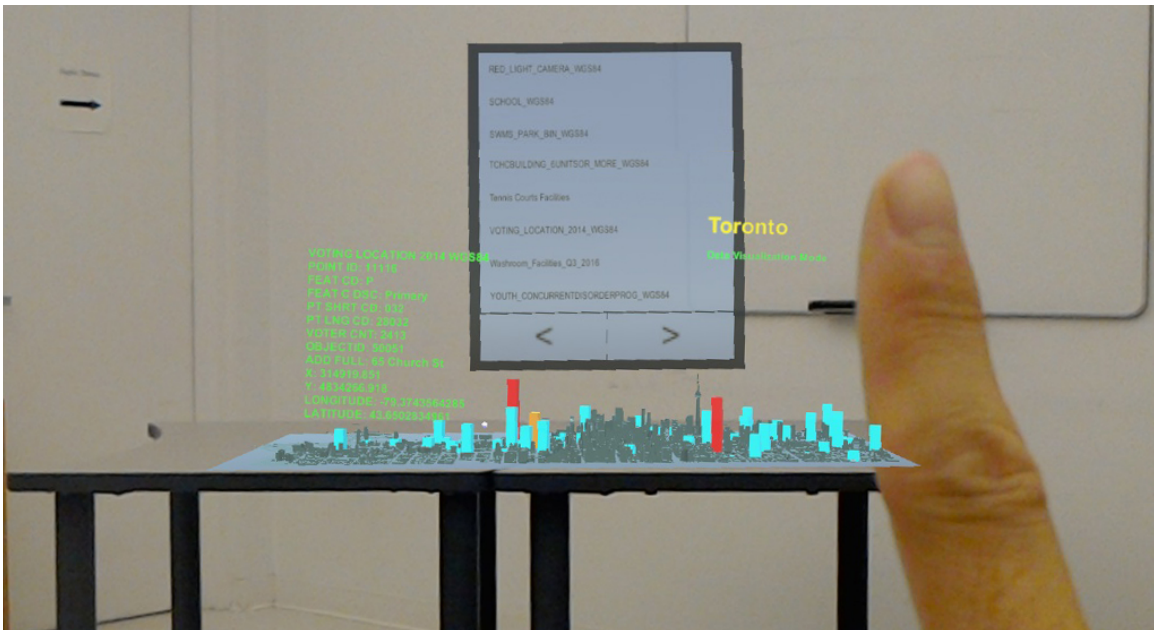


Figure 3.9: Visualizing Toronto voting data, option menu, and a specific building through interacting with HoloLens.

on the 3D Toronto city model with red and blue cuboids of different heights, as shown in Figure 3.9. Then the user can move his/her head to gaze at certain buildings and air-tap with finger to obtain more detailed information. As we can see from the figure, the selected building is changed to an orange color, and shows its detailed information, including that the voter count is 2413, the address is 65 Church, the latitude is 43.65, and so forth. The user can also move around the augmented model to view it from different angles, and select other options in the menu (e.g. tennis courts) to visualize a new data set.

C. Computing Performance Evaluation with City Data Visualization

HoloLens itself is an independent head-mounted semi-transparent AR system that processes all tasks with its own CPU and HPU with 2GB of RAM. We tracked its computing performance while users were conducting the desired interactions. The six performance

metrics, defined and tracked with HoloLens' own Device Portal, are CPU usage, HPU usage, memory usage, frame rate, SoC (system-on-chip) power usage, and system power usage.

The interactions we conducted are shown in Figure 3.10. The user first used "blossom" gesture command to display the HoloLens main menu, then moved their head to aim our developed Toronto city data visualization application, and used their index finger to select and open this application. After loading, the pure city model appears, along with the menu to select the city data type. Finally, the user can choose various city data types to visualize. The computing performance during this whole process is tracked for evaluation.

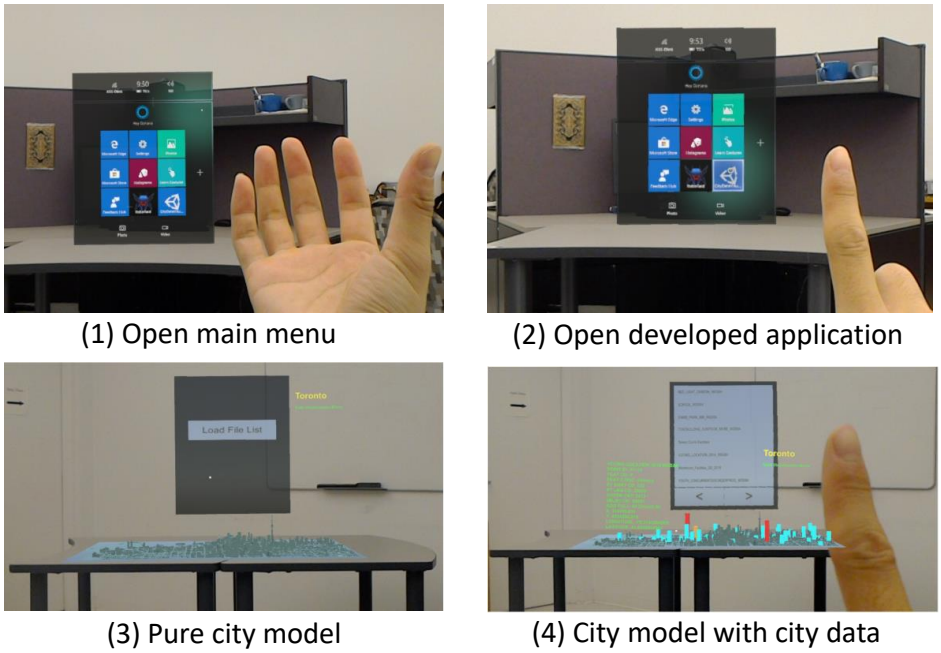


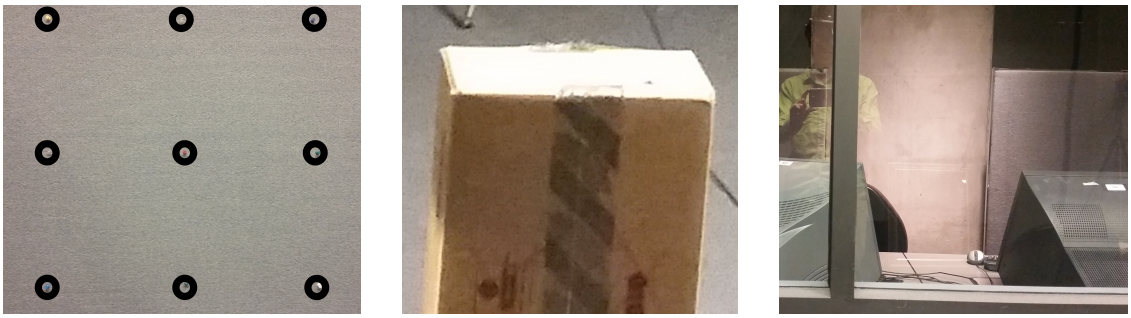
Figure 3.10: Tracking HoloLens computing performance when conducting designed operations.

3.3.4 Real Environment Reconstruction

As previously mentioned, AR systems usually reconstruct a 3D model of the surrounding environment before interacting with it. Then, during the interaction, the real environment information and virtual content matching process will depend solely on the previously reconstructed model, to avoid the high-demand and requirements of real-time processing of the surrounding environment data. Therefore, if the reconstructed 3D model of the room or place is not accurate, the AR contents may not be accurately placed on the desired locations. For instance, if the 3D model of a coffee table is generated at a wrong place, and a "cup" is augmented on it, the user will see the real coffee table at its original place, but the "cup" will be floating in the air somewhere else, which obviously degrades the user's AR experience.

The real environment reconstruction capability of HoloLens was evaluated through comparing a real environment with its reconstructed 3D model. Such a reconstructed model is usually influenced by the complexity of the real environment [117]. Lighting conditions, surface textures, and the distances between HoloLens and the real objects, also influence the performances of the sensors. Therefore, in this experiment, the differences between a real environment and its reconstructed model were separately measured for different influencing factors: object shapes (flat, convex or concave angles), lighting conditions (bright or dark).

In this experiment, the reconstruction performance for object shapes was evaluated using a flat surface (Figure 3.11(a)) and a box with convex and concave angles (Figure 3.11(b)). The brightness of the environment was controlled using ten brightness-adjustable incandescent lamps. The distance between the lamps and the object to be measured was 3.5 m. For the bright lighting condition, the output power of the lamps was set to 25 W, whereas their power was reduced to 5 W for the dark lighting condition. When the participants recorded markers through a glass pane (Figure 3.11(c)), the mark-



(a) (b) (c)

Figure 3.11: The marked objects used for testing the accuracy deviation σ_A^R of the reconstructed models obtained from HoloLens. (a) Flat surface with markers. (b) Box placed at a convex angle. (c) Glass surface separating the participant wearing the HoloLens system from objects under bright lighting condition.

ers were placed under the bright lighting condition. In the measurement implementation, the gaze point is visualized as a red circle, and its corresponding position information is also displayed nearby.

3.3.5 Spatial Mapping

The spatial mapping function mainly deals with the QoE parameters augmented contents fitting. The spatial mapping experiment, which investigated the anchoring of holograms to the real environment, was performed to test the virtual environment processing component. For instance, if the user opens a web browser, it will automatically be attached to a detected wall. Through detecting the accuracy of this attachment, we can evaluate the spatial mapping process.

In our experiment, we measured the gap or overlap between a hologram and the target surface to which it was attached, and the spatial mapping performance was evaluated using a measure denoted by σ_A^S (calculated in a manner similar to Equation 4.6). For this experiment, we created an application that allows a holographic box (0.3 m×0.3 m×0.1

m) to be attached to a target surface through a spatial mapping function, to check the accuracy.

3.3.6 Hologram Visualization

Since the content quality parameter is important in an AR QoE evaluation, we tested the visualization performance of the generated holograms.

The hologram visualization experiment was performed to evaluate the visual perception effect for HoloLens users. Since the holographic models built in HoloLens applications are created using metric values, users can visualize holograms of the same scale as real objects through the optical lenses. In this experiment, the hologram visualization performance was evaluated by calculating the accuracy deviation σ_A^V (in a manner similar to Equation 4.6) representing the visual deviation between a visualized hologram and a corresponding real object in recorded photographs.

To evaluate the hologram visualization performance, we created a 0.25 m×0.2 m×0.1 m holographic box in a self-developed HoloLens application and built a real box of the same size. Then we tested to see if users could overlap them.

3.3.7 Speech Recognition

Though gesture control is convenient for user interactions under certain conditions, it may be too complex to be performed in other conditions. For example, saying "select" to select a very small target is sometimes easier than trying to tap it. Therefore, voice commands are occasionally utilized to facilitate user interactions with AR systems. Since users with different backgrounds have various accents, and the surrounding noises also influence the detection accuracy, we designed experiments to test the speech recognition capabilities.

The speech recognition experiment was conducted to evaluate the reliability of con-

trolling HoloLens using voice commands. Considering the fact that the original system-defined voice commands may have been optimized, we also derived non-system defined voice commands. Therefore, we tested both system-defined and user-defined commands to obtain an overall evaluation. The user-defined commands (typed into the measurement application) were selected from Wobbrock's paper ("move", "rotate", "delete", "zoom in/out", "open", "duplicate", "previous" and "help") [143], and several system-defined commands are also chosen ("select", "place", "face me", "bigger/smaller", "adjust", "remove", "Hey Cortana, shut down" and "Hey Cortana, take a picture").

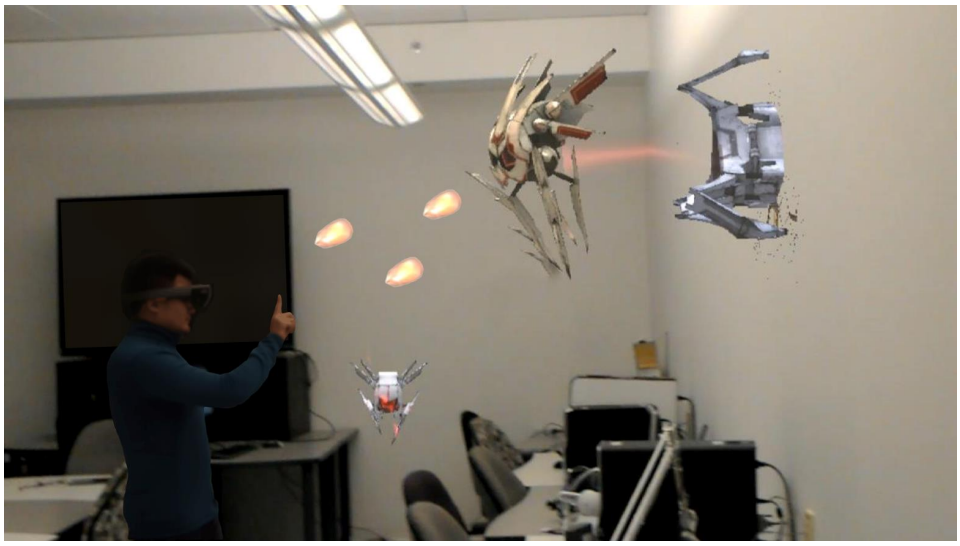
Chapter 4

Experiments and Results

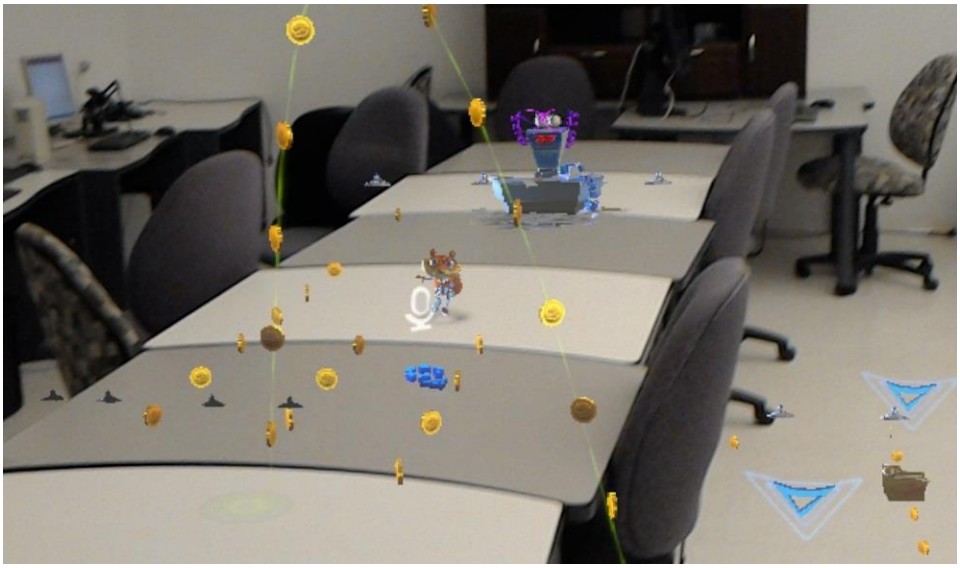
In the previous chapter, we introduced our proposed framework and the designed experiments to evaluate semi-transparent AR systems. This chapter provides more details of the experiment, and illustrates the experimental results.

4.1 FIS Model Build

Based on our proposed QoE taxonomy for semi-transparent AR system evaluation, we designed a user questionnaire, conducted a user evaluation study, and then used the collected user data to build the FIS model. Users were selected with various ages, education levels, and races. During the user study, users were first trained to get familiar with a semi-transparent AR system, then played selected AR applications with it, and finally filled out the questionnaire. Our user study procedures are similar to Steinmetz's and Silva's user study [144, 145] Details of application selection, questionnaire design, experimental setup, results, and analysis are as follows.



(a) A user is playing "*RoboRaid*"



(b) A user is playing "*Young Conker*"

Figure 4.1: The scenarios of users playing with AR applications "RoboRaid" and "Young Conker".

4.1.1 AR Application Selection

The AR applications we used for testing are a first-person shooting game "RoboRaid"[1] (Figure 4.1) and an adventure game "Young Conker"[2]. We chose these two applications because they both cover all 1st-, 2nd-, and 3rd-level parameters (except for haptic feedback, odor and taste) of the framework. The selected AR smart glass applications differ from those of mobile AR games (e.g., Pokemon Go) in several ways, such as 3D holographic displays, enabling more natural gesture control and augmenting content based on the detailed surrounding environment instead of directly superimposing content on the screen. To better experience the special attributes of the glass-like AR system during the test, participants were required to perform the following actions (we use "RR" to denote "RoboRaid" and "YC" to denote "Young Conker"):

- Content quality: for "RR", watch the augmented enemies and their fire and listen to the noise direction to decide which part of the real-world wall would be "breached" by battleships; for "YC", browse coins, chase or avoid enemies, follow commands of NPCs, etc.

- Hardware quality: for both "RR" and "YC", users need to walk and run around the room wearing HoloLens to adjust the distance from enemies.

- Environment understanding: "RR" needs users to find enemies "climbing" on the wall or "flying" in the air and to "destroy" the real-world wall to reveal the hidden aliens; "YC" requires participants to control the avatar to jump on a real-world desk or virtual coins with a springboard on the actual ground.

- User interaction: "RR" uses an air-tap to shoot, moving body to dodge attacks, and saying "X-Ray" to activate the special see-through weapon; "YC" defines the

[1]https://www.microsoft.com/en-us/hololens/apps/roboraid

[2]https://www.microsoft.com/en-ca/hololens/apps/young-conker

gazing direction as the avatar's direction of movement, saying "let's go" to start, etc.

4.1.2 Questionnaire Design

A questionnaire was designed for the user study based on our proposed framework, as show in Figure 4.2 and 4.3, and also publicly posted on GitHub[3]. The questions for each application are identical, but the examples differ to help participants understand questions specifically selected based on the game. The questionnaire first collects several aspects of basic user information, including name, gender, age range and previous AR and non-contact gesture control experience. For each related QoE parameter, we designed a Likert-scale question with the answer provided on a five-point Likert scale, namely, with values from "1" to "5" representing different responses (e.g., from "not at all" to "completely"). Questions regarding low-level parameters are asked first before a general rating for the corresponding high-level parameters is required. For instance, Question 1 (Q1) asks the participant to rate the realism level of the visual information on a scale from 1 to 5, namely, from "not at all" to "completely realistic"; Q2 concerns the realism level of the audio, while Q3 asks for the required focus or attention level. After these three questions, a general question (G1) is presented to rate the high-level parameter content quality based on Q1-Q3. As a result, the participant has a better understanding of high-level parameters. Finally, a rating score (0-100) for the overall quality of this AR experience is also required.

Though The questionnaire contains redundant information, and not all data are required to further build the FIS model, we use these additional data to validate the taxonomy through exploring the correlations between high-level parameters and their corresponding low-level parameters.

[3]https://github.com/HoloLensQoE/HoloLens_QoE_User_Study

Questionnaire for AR QoE Evaluation (Page 1)

Tester ID: ___ Name: _____ Gender: ☐M ☐F
Email: _____ Application Name: ***RoboRaid***
Your age range: ☐10-19 ☐20-29 ☐30-39 ☐40-49 ☐50+

(1) Did you ever experience Pokemon Go or other AR applications? ☐Y ☐N

(2) If yes, how many times? ☐ 1-5 ☐ 5-10 ☐ 10+

(3) Did you ever experience non-contact gesture control applications, e.g. Xbox Kinect? ☐Y ☐N

(4) If yes, how many times? ☐ 1-5 ☐ 5-10 ☐ 10+

1. To what extent did the device provide realistic visual information?
   Not realistic Completely realistic
   1 2 3 4 5
2. To what extent did the device provide realistic audio information?
   Not realistic Completely realistic
   1 2 3 4 5
3. To what extent did the application require your focus/attention level?
   Not at all Completely focus
   1 2 3 4 5

G1: Based on your ratings for Q1-3, please give a general rating for the content quality?
Very bad Very good
1 2 3 4 5

4. To what extent could you move freely wearing the device?
   Not at all Completely free
   1 2 3 4 5
5. To what extent do you feel comfortable wearing the device?
   Not comfortable Very comfortable
   1 2 3 4 5

G2: Based on your ratings for Q4-5, please give a general rating for the hardware quality?
Very bad Very good
1 2 3 4 5

6. To what extent did the meaning of the augmented contents fit the real environment?
   Not at all Completely fit
   1 2 3 4 5
7. To what extent did the position of the augmented elements fit the real environment?
   Not at all Completely fit
   1 2 3 4 5
8. To what extent could the application contents precisely change with environmental changes?
   Not at all Completely precisely
   1 2 3 4 5

Figure 4.2: Questionnaire for the AR QoE Evaluation Page 1.

Questionnaire for AR QoE Evaluation (Page 2)

9. To what extent could the application contents quickly change with environmental changes?
   Not at all Very quickly
   1 2 3 4 5

G3: Based on your ratings for Q7-10, please give a general rating for the environmental understanding?
Very bad Very good
1 2 3 4 5

10. To what extent do you think the gesture interaction design (e.g. air-tap gesture is shooting) is natural?
    Not at all Very natural
    1 2 3 4 5
11. To what extent do you think the voice interaction design (e.g. say "X-Ray" for see-through superpower) is natural?
    Not at all Very natural
    1 2 3 4 5
12. To what extent do you think the body movement interaction design (*e.g.* move to avoid enemy attack) is natural?
    Not at all Very natural
    1 2 3 4 5
13. To what extent do you think the system respond precisely to your interaction instructions?
    Not at all Completely precise
    1 2 3 4 5
14. To what extent do you think the system respond quickly to your interaction instructions?
    Not at all Very quickly
    1 2 3 4 5

G4: Based on your ratings for Q11-15, please give a general rating for the user interaction?
Very bad Very good
1 2 3 4 5

Please give a grade, over 100, for the **overall quality** of this AR experience: ___/100
Any other comments about this user study?
Signature: _____
Thank you for participating in this study!

Figure 4.3: Questionnaire for the AR QoE Evaluation Page 2.

4.1.3 Using High-Level Parameters to Represent Low-level Parameters

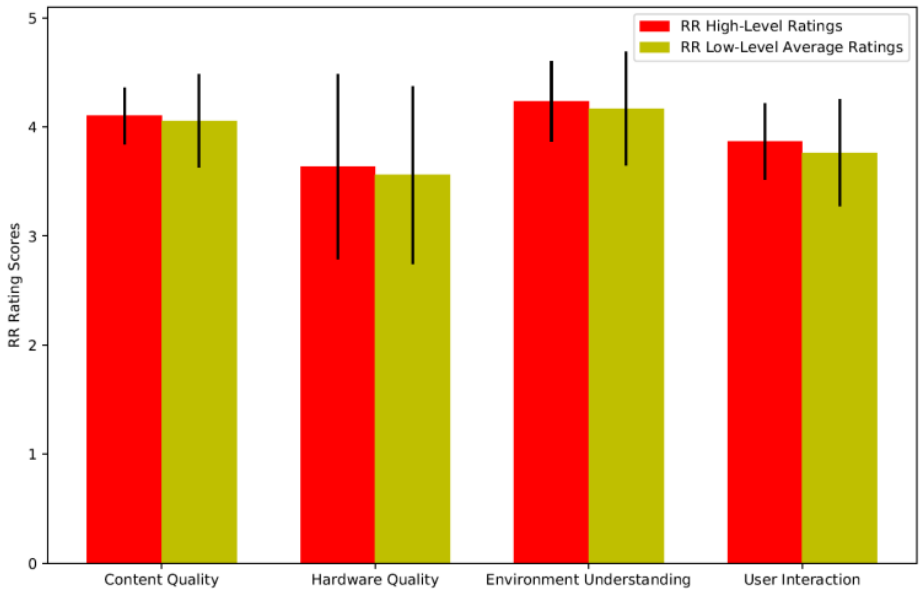
Since there are many low-level parameters in the proposed AR QoE taxonomy, if we take the user ratings for all these parameters as inputs for the evaluation model, there will be too many combinations and the model will not be manageable. Therefore, we used just the high-level ratings as the inputs for our model.

In our user study, users were asked to rate both high-level and lower-level parameters. We further used bar tables to demonstrate the comparisons of high-level ratings and low-level average ratings with error bars, as seen in Figure 4.4, where (a) is the result of "RoboRaid" testing and (b) is the result of "Young Conker" testing. From this figure, we can see that users' direct input ratings for the high-level parameters are similar to the averages of users' low-level parameter ratings, which validates our method of using high-level parameters as the inputs.

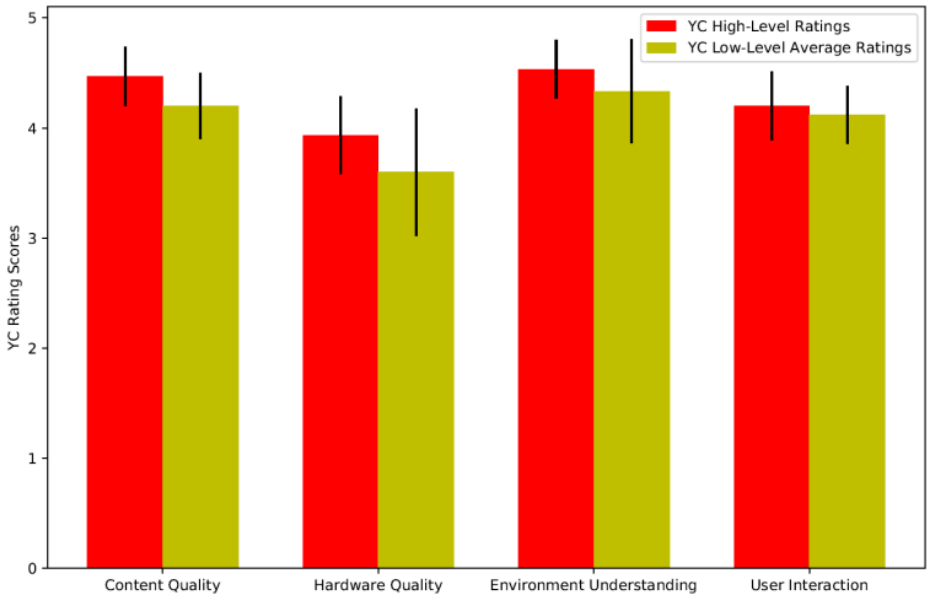
In addition, this also means that in the situations where the ratings of high-level parameter are not available, we can use low-level parameters to estimate them. Since low-level QoE parameters connect more closely with QoS parameters, this can also be used to further explore the relationship between the QoE model and the QoS model.

4.1.4 FIS Model Results

With the user data obtained from our experiments and questionnaire, we built a FIS model with them. The general architecture of our implemented FIS model is shown in Figure 4.5, which contains MFs defining results for four inputs (content quality, hardware quality, environment understanding, and user interaction) and one output (overall rating score), as well as the derived 43 fuzzy rules.



(a) RoboRaid (RR) high-level ratings and low-level average ratings



(b) Young Conker (YC) high-level ratings and low-level average ratings

Figure 4.4: Comparison of high-level ratings and low-level average ratings with error bars (a) RoboRaid (RR) (b) Young Conker (YC).

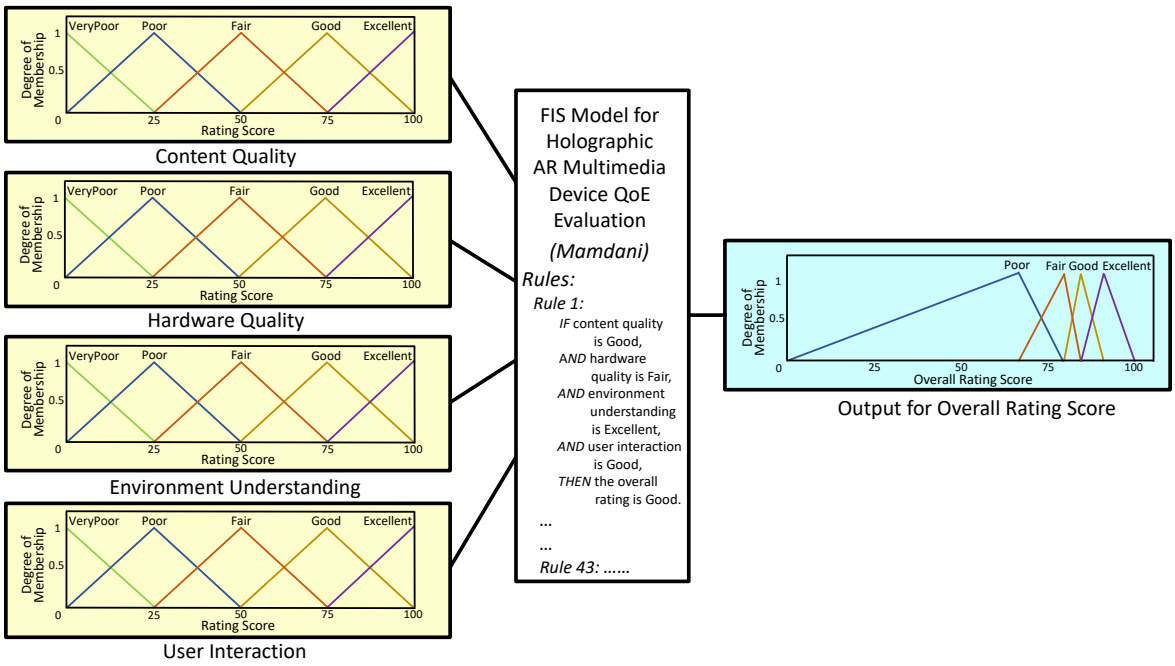


Figure 4.5: General architecture of the implemented Mamdani FIS model.

A. MFs for Input and Output

As previously introduced, each input variable is equally divided into five MFs with triangular shapes, to correspond with their five answer options (Figure 4.5 inputs). Since output variables are more dispersive, we start by applying the FCM clustering method, and then define its MFs.

FCM divides the overall rating scores into different clusters based on each value's degree of belonging to various groups. As we can see from Figure 4.6, all values are represented with the shape and color of its defined cluster (purple '$*$', green 'o', blue '$+$', and red '$\times$'), and the centers of all clusters are displayed with larger-size black symbols, with rating values of 90.80, 84.41, 79.58, and 66.58. These values are then used as the peak points for the MFs of the FIS outputs accordingly (Figure 4.5 output). With this information, given a value, we can decide which MFs it belongs to.

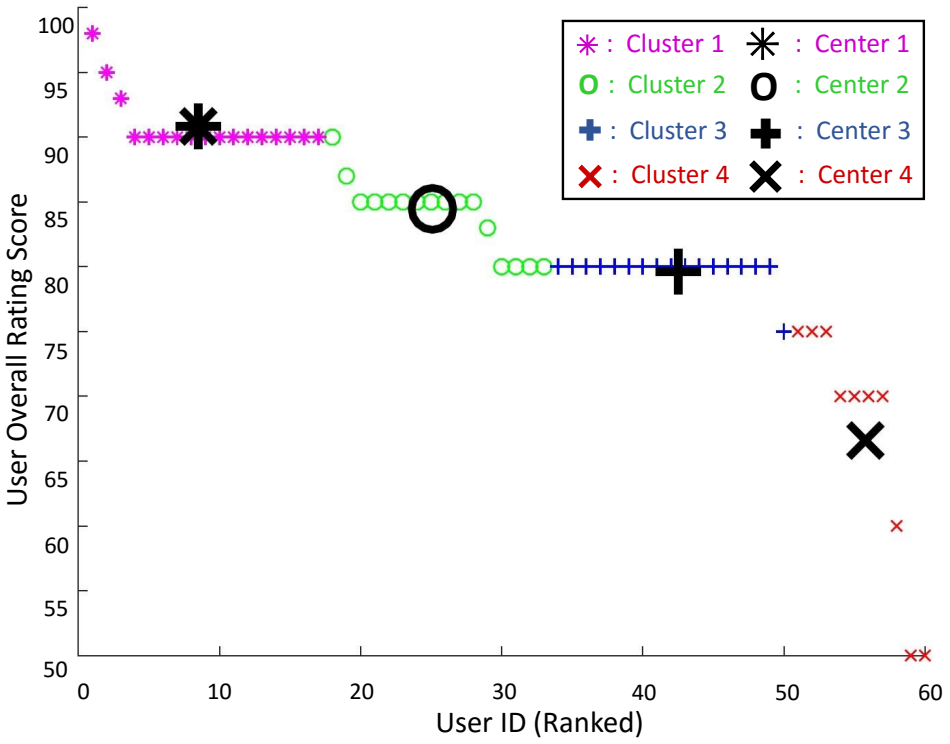


Figure 4.6: Fuzzy c-means (FCM) clustering results for a user's overall rating score.

B. Fuzzy Rules Derived

A fuzzy rule can be derived from each user data. For instance, if the rating for content quality is 75, hardware quality 50, environment understanding 100, user interaction 75, and the overall rating is 84, then we can derive this fuzzy rule as: *IF* content quality is *Good*, *AND* hardware quality is *Fair*, *AND* environment understanding is *Excellent*, *AND* user interaction is *Good*, *THEN* the overall rating is *Good*.

Since 75% of the user data (45 subjects) from the "RoboRaid" testing is used to derive the fuzzy rules, and the data from each questionnaire can be used to build one rule, a few rules may be repeated or conflicting. For a rule occurring more than once, we multiply the base weight value by this rule's appearance frequency to obtain its new weight value; for conflicting rules, we add them as separate fuzzy rules to balance the result. Based on this strategy, we built 43 fuzzy rules for our FIS model (Figure 4.5 rules).

C. View of the FIS Model

The FIS model can be represented with the overall rating score mapping from high-level parameters. Figure 4.7 is an example that shows two parameters (content quality and environment understanding) map with the overall rating score, when setting the other high-level parameters as average values (50).

4.1.5 Statistical Analysis

After building the FIS model, we considered the data of the remaining fifteen "RoboRaid" users and fifteen "Young Conker" users as the testing data. Each subject's data outputs have a ground truth overall user rating QoE_u from the user and an FIS-estimated-rating QoE_f from our model. Figure 4.8 shows the QoE_u and QoE_f pairs for "RoboRaid" and "Young Conker", respectively. From this figure, we can see that each pair of QoE_u and

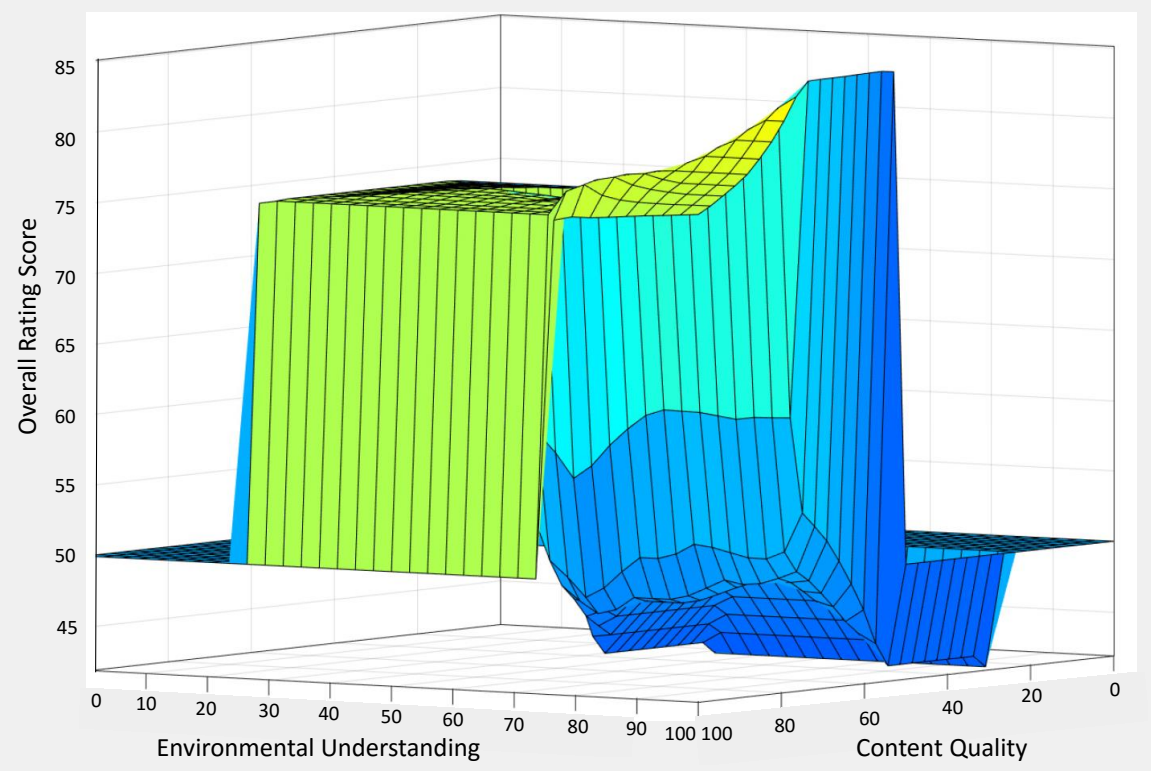


Figure 4.7: 3D surface view shows the mapping from content quality and environment understanding to overall rating score.

QoE_f has similar patterns.

Besides demonstrating the QoE figure, we also analyzed and listed key values of the testing results in Table 4.1. We first explored the descriptive statistics of both the "RoboRaid" and "Young Conker" applications, including the mean, 95% confidence interval, standard error, median, and standard deviation, of each series of data, to give a general overview. Subsequently, we compared each application's root-mean-square errors (RMSEs) of QoE_u and QoE_f, and finally we performed a paired-samples T-test to statistically analyze them.

A. Descriptive Statistics

Descriptive statistics for each individual overall user rating QoE_u and FIS-estimated rating QoE_f are shown in Table 4.1 for initial analysis. We observed that each pair of QoE_u and QoE_f for a given application are similar. For instance, the "RR" value of QoE_u has an average of 85.133 with a 95% confidence interval of 82.776 to 87.491, a standard error of the mean (SE_M) of 1.099, a median of 85.000, and a standard deviation of 4.257; the corresponding "RR" value of QoE_f has a mean of 85.853 with a 95% confidence interval of 83.450 to 88.257, an SE_M of 1.121, a median of 85.000, and a standard deviation of 4.340.

B. Root-Mean-Square Error

To further evaluate the differences between these two outputs, we computed the root-mean-square error (RMSE), which measures the difference between values. In our case, it is applied to calculate the square root of the mean square errors between QoE_u and QoE_f

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left(QoE_u - QoE_f\right)^2}{n}} \tag{4.1}$$

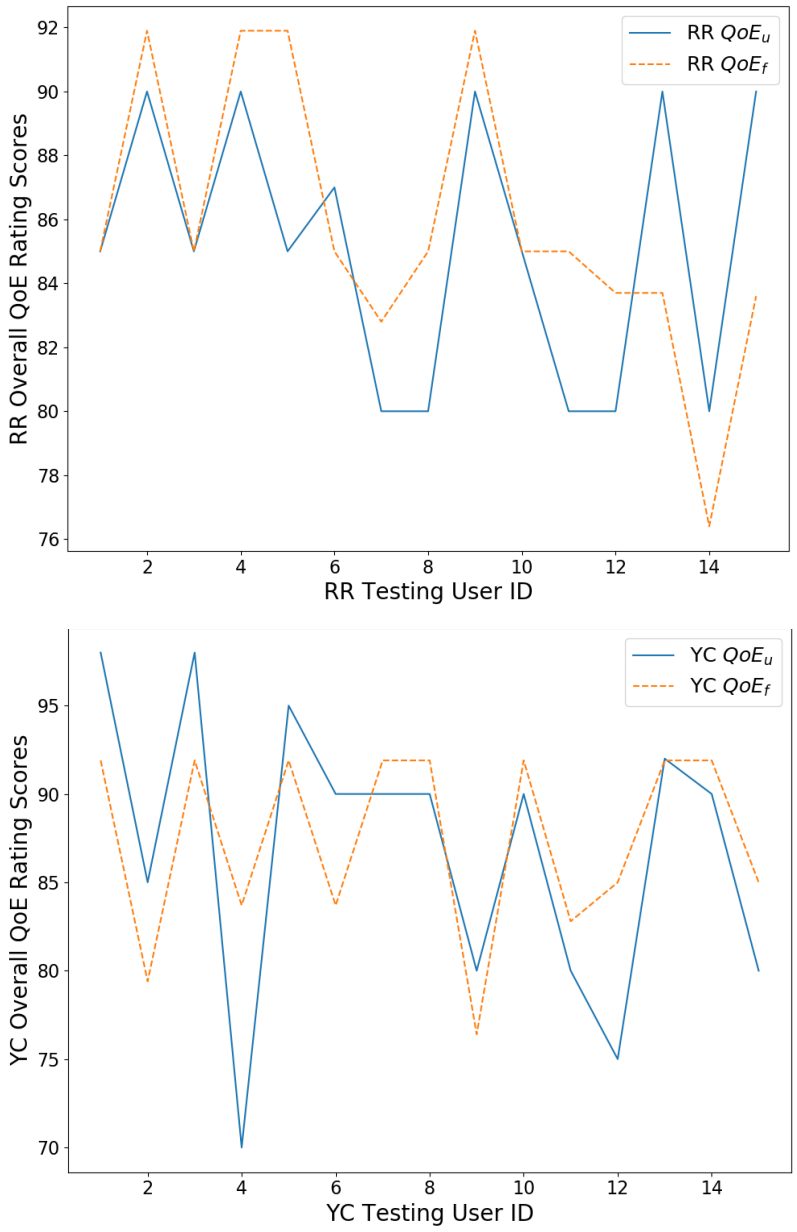


Figure 4.8: Overall QoE rating scores by users (QoE_u) and our FIS model (QoE_f) for "RoboRaid" (RR) and "Young Conker" (YC), respectively.

Table 4.1: Testing Results of the Overall User Rating (QoE_u) and the FIS-Estimated Rating (QoE_f) with Applications

Testing Application	QoE	Mean	95% Confidence Interval		Std. Error	Median	Std. Deviation	RMSE (%)	Paired T-Test P-Value
			Lower Bound	Upper Bound					
RoboRaid	QoE_u	85.133	82.776	87.491	1.099	85.000	4.257	3.895	0.493
	QoE_f	85.853	83.450	88.257	1.121	85.000	4.340		
Young Conker	QoE_u	86.867	82.282	91.452	2.138	90.000	8.280	5.791	0.728
	QoE_f	87.413	84.427	90.399	1.392	91.900	5.392		

After using this formula to analyze our results, we found that the RMSE for "RR" is 3.895, indicating that, on average, our FIS-estimated rating deviates from the ground-truth value of the user rating by 3.895 points on a scale of 100. Similarly, we obtained a RMSE of 5.791 on a scale of 100 for "YC" .

C. Paired-Samples T-Test

To statistically observe and determine if QoE_f is significantly different from QoE_u, we conducted paired-samples t-test for analysis.

The hypotheses can be expressed as

$$H_0 : \mu_1 = \mu_2 \tag{4.2}$$

$$H_1 : \mu_1 \neq \mu_2 \tag{4.3}$$

where H_0 represents the null hypothesis that the paired population means are equal, and H_1 means they are not equal. Then we calculate t and P values to check which hypothesis is correct. Following are the steps to process the data. Since t value can be obtained from the following equation

$$t = \frac{\overline{X}_1 - \overline{X}_2}{SE_D} \tag{4.4}$$

where $\overline{X}_1$ and $\overline{X}_1$ are the means of two samples separately, and SE_D is the standard error of the difference between these two samples, defined by

$$SE_D = \sqrt{(SE_{M1})^2 + (SE_{M2})^2 - 2r(SE_{M1})(SE_{M2})} \tag{4.5}$$

where SE_{M1} and SE_{M2} are the two groups' standard error of the means separately, and r is the correlation coefficient of them. For "RoboRaid" user study, r is 0.575. We can obtain SE_D is 1.023, and take it into Equation 4.4 to calculate t value, which is 0.704.

With the calculated t value and the degree of freedom (df=14, obtained from the number of testing values minus 1), we used the t distribution table to obtain the two-tailed p-values for "RR" and "YC" of 0.493 and 0.728, respectively, with both being

larger than the significance level α (0.05). Therefore, the null hypothesis H_0 holds, and there are no statistically significant differences between each pair of QoE_u and QoE_f, i.e., our FIS-estimated ratings are not significantly different from the ground truth overall user ratings, namely, our FIS model generates similar output results as the ones obtained from user ratings.

4.2 Correlation Between QoE and QoS Parameters

The general information regarding the experiments for the functional components evaluations of AR systems are as follows, and more details of each experiment are given later.

- All experiments were conducted in a closed 8 m×5 m room under ambient lighting conditions, controlled by means of lights with adjustable intensities and angles. The room also prevented echoes during the experiments because of its sound-absorbing walls.

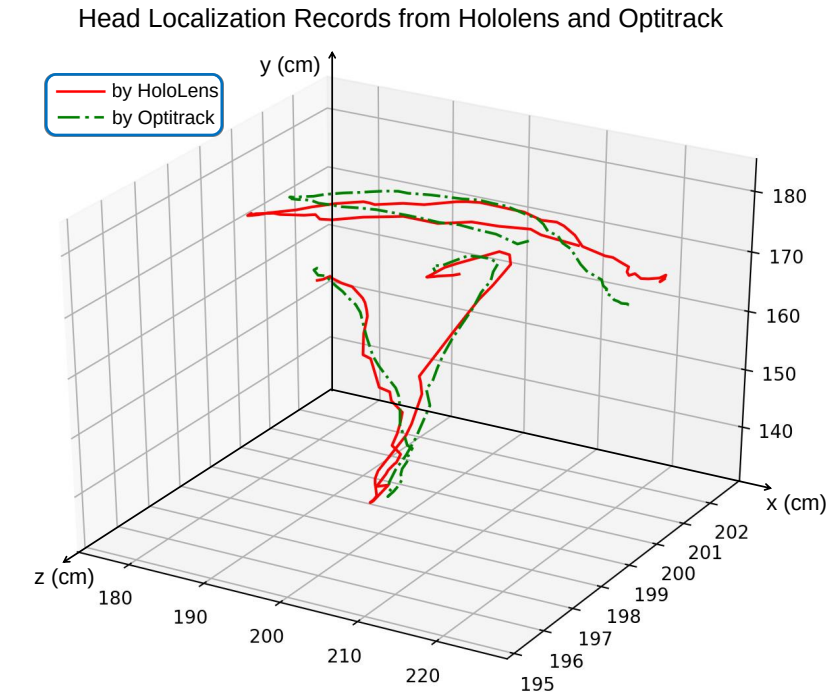
- The HoloLens applications used in our experiments were all developed with Unity and Visual Studio 2015 using the C# programming language for compatibility with HoloLens' Windows 10 OS.

- A total of twenty students and researchers from the University of Ottawa participated in our experiments. Their average age was 25.89, with a standard deviation of 6.11. They were either native or fluent English speakers. Each participant was given ten minutes to become familiar with HoloLens and then spent approximately twenty-five minutes performing all the experiments.

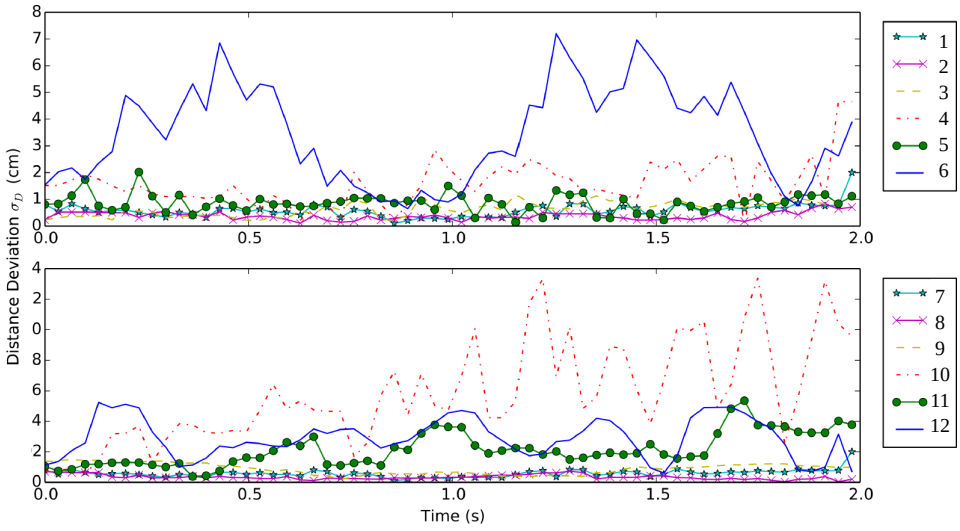
4.2.1 Head Localization Evaluation

We conducted several steps to evaluate the head tracking function. First, the OptiTrack system was calibrated to accurately record the ground truth for head localization. Second, the HoloLens and OptiTrack systems were both activated to record tracking data. Third, the participant, wearing the HoloLens unit with markers, was asked to perform several actions in the center of the circle of cameras in a random order, including squatting quickly (fast movement), tilting the body slightly (slow movement), looking at the corner of the room quickly (fast rotation) and swinging the head gently (slow rotation). A laboratory technician was trained to demonstrate these actions during the experiment, and the participant was asked to simultaneously imitate the demonstrated action and speed.

An example of tracking records from HoloLens and OptiTrack is shown in Figure 4.9(a), where the solid red lines represent the head localization records from HoloLens and the dotted green lines represent the records from OptiTrack. The distance deviations σ_D between the two records are shown in Figure 4.9(b). The upper subfigure shows the distance deviations σ_D caused by head movement, and the lower subfigure shows the distance deviations σ_D caused by head rotation. Since the distance deviations σ_D under each condition were all measured in a 3D coordinate system, groups of three lines (lines 1-3, 4-6, 7-9, and 10-12 in Figure 4.9(b)) are used to present the results in the x, y, and z dimensions for moving slowly, moving quickly, rotating slowly and rotating quickly, respectively. The average distance deviation values are 0.53, 1.63, 0.60, and 3.62 cm, with standard deviations of 0.03, 0.63, 0.02, and 0.47 cm, respectively. The record values from HoloLens and OptiTrack do not have significant differences ($F_{1,20} = 1.96$, $P = 0.52$). At high speed, it is difficult for HoloLens to correct the head localization results. This can be seen from the fact that the average distance deviation σ_D in the fast mode (both movement and rotation) is 2.63 cm, whereas it is 0.56 cm in the slow mode. The highest



(a)



(b)

Figure 4.9: Results of the head localization experiment. (a) Example of tracking records from HoloLens and OptiTrack. (b) Distance deviations between the two records.

distance deviation σ_D is 13.38 cm, caused by the head rotating quickly along the x axis.

4.2.2 3D Reconstruction Accuracy Comparison Results

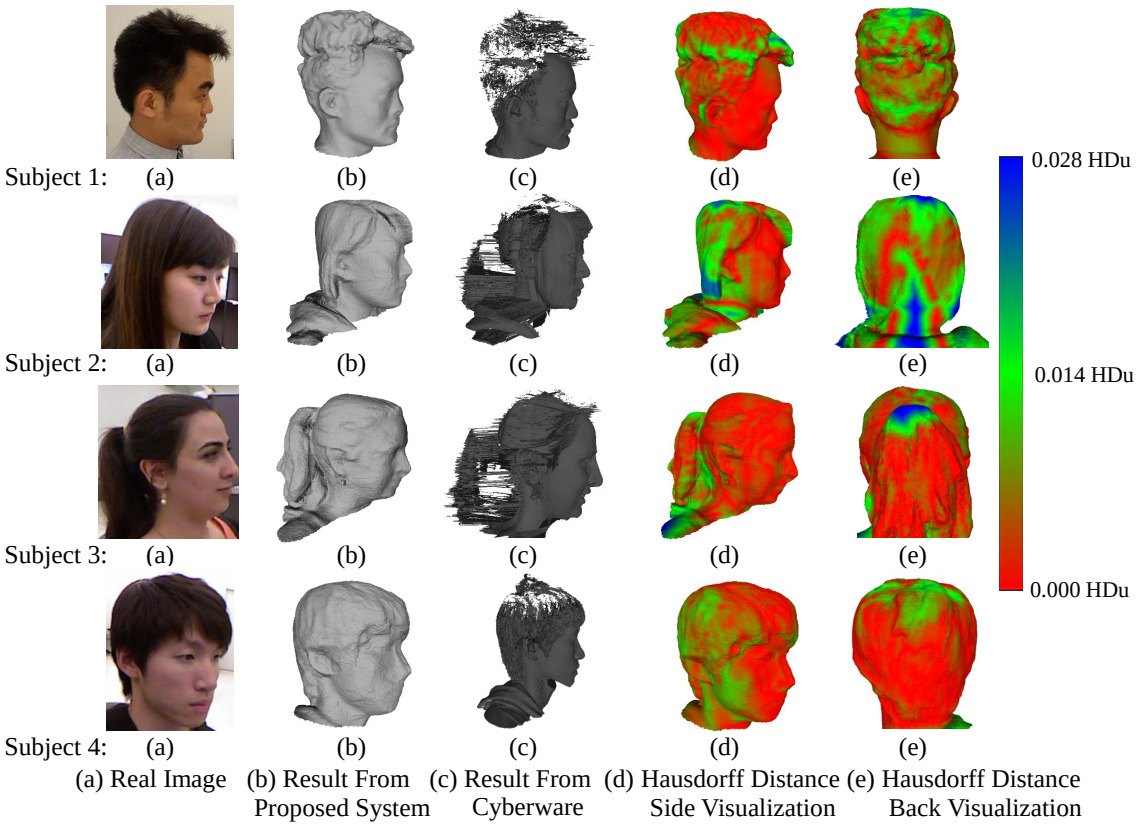


Figure 4.10: Male and female subjects' real images, scanned results from our proposed system and from Cyberware separately, and the Hausdorff distance visualization results.

To evaluate the accuracy of our proposed 3D scanning system, we compared our scanned models with the ones from commercial Cyberware laser scanning system on real-human subjects. We scanned real human subjects with both the proposed system and Cyberware, and then calculated the Hausdorff distance between the two acquired models.

We computed the geometric differences between the 3D model data acquired by different scanners. Our numerical evaluation is based on computing the approximate error between two triangular meshes representing the same surface or object ($M_1 \leftrightarrow M_2$), as introduced by Cignoni et. al. [146]. The approximation error is defined with the two-sided Hausdorff distance d_H [147], which is the maximum value from $M_1 \rightarrow M_2$ and $M_2 \rightarrow M_1$ in the Euclidean space, as follows:

$$\begin{aligned} & d_H\left(M_1, M_2\right) \\ & = \max\left\{ \sup_{m_1 \in M_1} \left(\inf_{m_2 \in M_2} d\left(m_1, m_2\right) \right) , \sup_{m_2 \in M_2} \left(\inf_{m_1 \in M_1} d\left(m_1, m_2\right) \right) \right\} \end{aligned}$$

where sup represents the supremum, inf is the infimum, $m = (x, y, z)$ is the 3D vertex points of the corresponding triangular mesh and d is the Euclidean distance between two points in Euclidean space E^3. To provide comparative results between the ground-truth model (M_{gt}) and the laser scanned model (M_f), as well as the Kinect v2 scanned mode (M_k), we took the full set of vertex points (68k) from the ground truth model and searched for the closest point on the scanned models to compute the error metrics based on the Hausdorff distance d_H. The unit used to represent the Hausdorff distance in MeshLab is Hausdorff Distance unit (HDu), represented as a color bar in Figure 4.10.

Detailed comparison results are shown in Figure 4.10, where we demonstrate the real images, the results from the proposed system, the results from Cyberware, and the Hausdorff distance visualization results of two male subjects and two female subjects. As we can see from Figure 4.10, Cyberware can reconstruct certain regions with more details than the proposed system, for example, human eyes, eyebrows, ears, nose, and mouth, mainly because of its closer capture distance, higher resolution, and more stable fixed rotation configuration. However, the Cyberware laser scanner does not work well with optically uncooperative materials, such as human hair. Thus, the back-view comparisons (Figure 4.10(e)) show a larger Hausdorff distance (green and blue areas in the visualization images) than other views, especially for women with long hair or ponytails. As a

result, our proposed system has superior hairstyle reconstruction capabilities compared with the Cyberware system. Based on the fact that most facial comparisons are in red (small Hausdorff distance range), and that our system has better hair-style reconstruction capabilities, a lower cost, and better mobility, the overall performance of our system can be considered to be comparable to that of Cyberware, an expensive commercial laser scanning system.

4.2.3 Computing Performance Evaluation Results

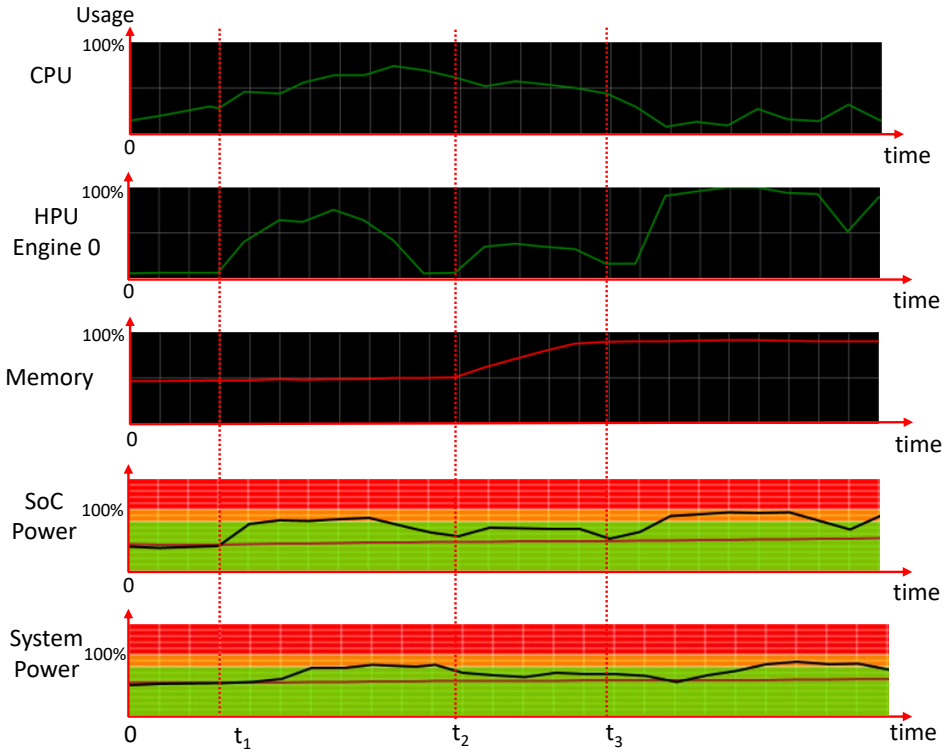


Figure 4.11: HoloLens computing performance tracking results

The computing performance tracking results of HoloLens, obtained while conducting designed interactions are shown in Figure 4.11. Before we started running our application (before t_1), the CPU usage was around 8%-20%, HPU engine #0 usage was around 5%,

memory usage was about 40%, namely, 800MB out of 2GB. Since time t_1, we started using gaze and gesture commands to open our application. Then, both CPU and HPU are heavily used, and memory usage starts to rise. After we completely open our application at t_3, the CPU usage drops, though about 10% is still used by our application; HPU usage varies based on the visualized data size in user's view, and occasionally has slight glitches due to heavy usage; the memory usage stays at 80% (1.6GB), which means that approximately 800MB is occupied by our application.

Table 4.2: HoloLens performance evaluation with different data sets

Data Name	City Model Size (MB)	Data Set Size (KB)	Memory (MB)	HPU (%)	CPU (%)	Frame Rate (fps)	SoC Power (%)	System Power (%)
No city model	0	0	880	6	8-30	58-60	37-40	50-53
Pure city model	75	0	1699	91-100	10-35	3.8-3.9	89-94	72-77
Places of interest	75	65	1708	93-100	10-35	3.8-4.1	90-97	72-82
Youth services	75	326	1715	93-100	10-35	3.8-4.2	90-97	75-81
Heritage inventory	75	1285	1747	93-100	10-35	3.8-4.2	90-97	75-82

Since HoloLens' performances are also influenced by data size, we conducted analyses with various data sets. The values are recorded when all of the augmented city contents are within HoloLens' field of view. Table 4.2 lists five situations: no city model (before opening our application), pure city model (75MB without city data sets), with city data

set "Places of interest" (65KB), "Youth services" (326KB), and "Heritage inventory" (1285KB), separately. The Toronto 3D city model is composed of fbx files, and data sets are in xml file format, as previously mentioned. From this table, we can see that memory usage rises when the city data size increases, namely, visualizing larger data sets requires more memory space. When viewing the whole augmented city model, HPU usage is more than 90%. Compared with visualizing a pure city model, it goes up a little when showing extra data set information on the model. SoC and system power consumptions mainly change with HPU usage in this condition. CPU usage rarely changes with different data sizes, and is actually just occupied around 3% by our application. The rendering performance, measured by frames per second (fps), also changes under different conditions. When there is no city model, whether the main menu is displayed or not, the frame rate is around 60fps; after loading a pure city model, it changes to 3.8-3.9fps, though it rises up to 25fps when looking at the city data type menu; with additional city data-sets, there are only slight differences (places of interest set is 3.8-4.1fps, while the other two sets are 3.8-4.2fps). The frame rate is not high, because once we select a city data to visualize, the HPU usage is high, and the augmented contents do not change a lot.

4.2.4 Real Environment Reconstruction

During the real environment reconstruction evaluation experiment, *first*, the participants were asked to walk around the room to reconstruct a 3D model of the room under bright lighting conditions. *Second*, the experimental data acquired by HoloLens were recorded. *Third*, the participants were asked to place the red circle of the gaze-tracking application on the locations indicated, to obtain the dimensions of the objects' reconstructed models under bright and dark lighting conditions, or through a glass pane placed at a certain distance. For example, the participants were asked to place the red circle on the nine markers on the flat surface to obtain the distance between the reconstructed markers.

Similarly, the participants were asked to place the red circle on the two vertices of the box, corresponding to each side of the box lying at the convex or concave angle, to calculate the length of the box. To avoid interference between different influencing factors, the sequence of the experiment varied for each participant.

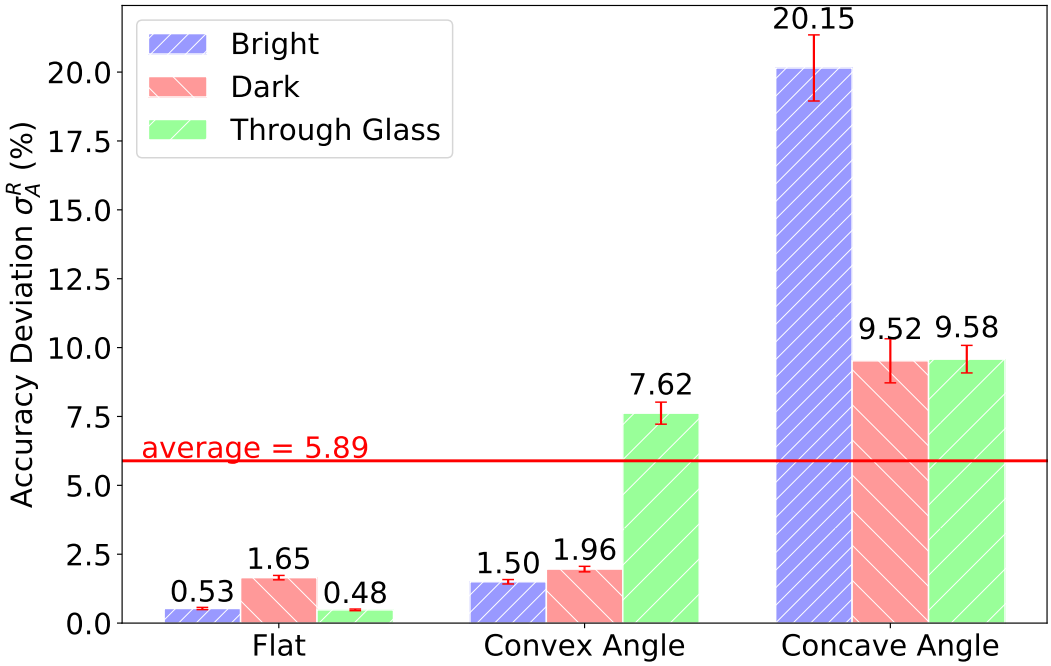


Figure 4.12: The reconstruction accuracy σ_A^R for real environment reconstruction.

The reconstruction accuracy σ_A^R was evaluated based on the difference between the real environment and its reconstructed model:

$$\sigma_A^R = \sum_{i \subset N} \left(\frac{|L - l_i|}{L \cdot N} \right) \tag{4.6}$$

where N represents the number of measurements, L is the length of one edge of the real object, and l_i is the corresponding edge length measured in the i-th measurement by HoloLens. R is used to denote that this reconstruction accuracy refers to the real environment reconstruction performance. The values of σ_A^R are greater than or equal to 0, where 0 means that the real object and its corresponding holographic model are

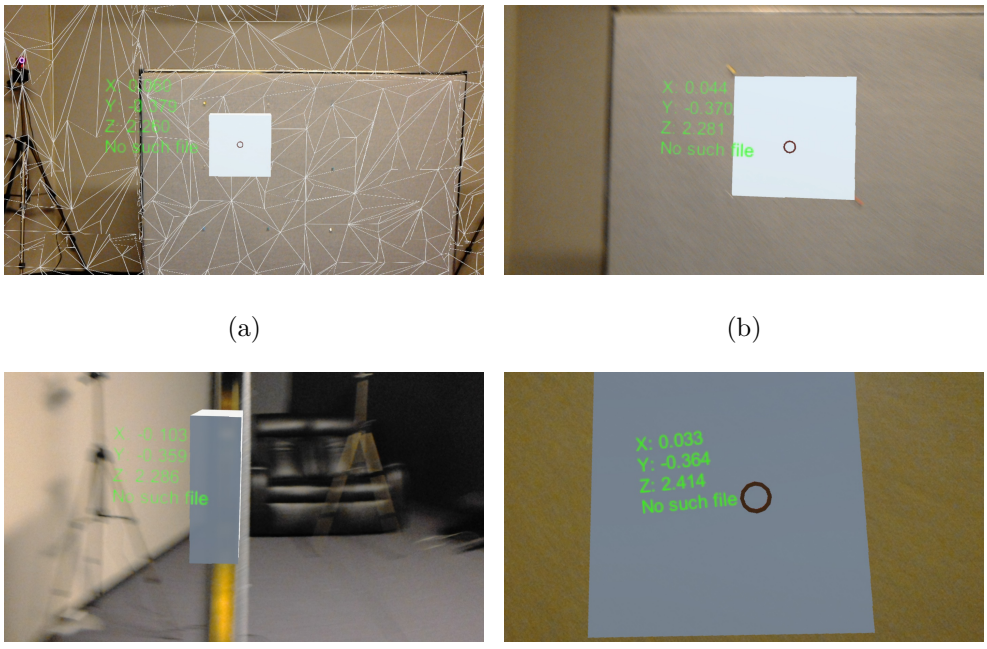
identical and higher σ_A^R values indicate an increasing difference between the object and the model.

The reconstruction accuracy σ_A^R for the tested objects are shown in Figure 4.12, where the error bars represent the standard deviation for each condition, the blue and red bars represent the object size deviations captured under the bright and dark lighting conditions, respectively, and the green bar represents the reconstruction accuracy σ_A^R for an object behind glass. The average σ_A^R is 5.89%, and its standard deviation is 6.18%. The lowest σ_A^R value is 0.53%, which is obtained when the object's surface is flat and it is under bright light, whereas the highest σ_A^R value is 20.15%, which is obtained when the object is a box at a convex angle. This indicates that better environmental reconstruction results can usually be obtained with flat surfaces under bright lighting conditions compared with the reconstruction of uneven surfaces (convex or concave) in dark environments.

4.2.5 Spatial Mapping

For our spatial mapping evaluation experiment, the participants were first asked to look around to scan the target surface. Then they were asked to record the positions of the markers on the target surface and attach the holographic box to those positions from different distances (0.5 m to 3.5 m, with a step size of 0.5 m). Once the holographic box had been mapped onto the surface, as the third step, the positions of the holographic box were recorded from three views to calculate the gap or overlap between it and the target surface. The experimental results are shown in Figure 4.13(a)-(d), presenting the front, side and back views, respectively, of the box attachment results achieved through spatial mapping. The red circle is the cursor on which the user could focus, and the green text shows the position information of the cursor.

The accuracy deviations σ_A^S of the spatial mapping results are shown in Table 4.3.



(a) (b)

(c) (d)

Figure 4.13: Procedure for and results of the spatial mapping experiment. (a) The spatial mapping process. (b) Front view of the box attachment results. (c) Side view of the results. (d) Back view of the results.

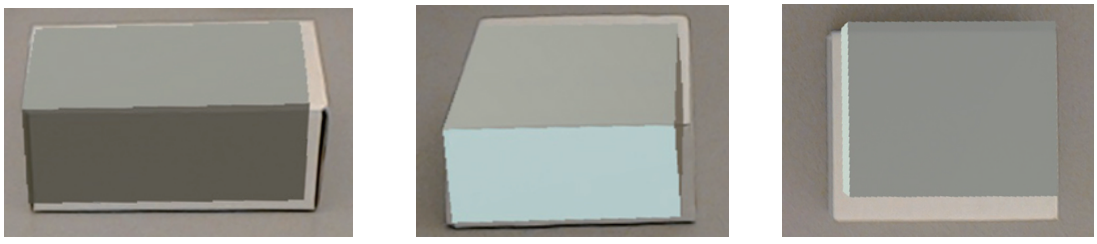
The average σ_A^S value is 73.8%, with a standard deviation of only 2.70%. Since the distance is calculated between the hologram and the target surface, the results depend only on the spatial mapping algorithm and are not influenced by possible reconstruction problems. Although the deviations for different distances are only slightly different, the best spatial mapping results are achieved at 1.5 m and 2.5 m, which may be beneficial for high-accuracy tasks such as mapping mechanical components.

Table 4.3: Accuracy deviations for spatial mapping, σ_A^S

Distance (m)	0.5	1	1.5	2	2.5	3	3.5
Accuracy Deviation σ_A^S (%)	76	72	71	77	70	74	77

4.2.6 Hologram Visualization

In our hologram visualization experiment, the participants were first asked to move the visualized hologram to overlap with the real box, as exactly as possible, and then to take three photographs of the overlapped boxes from the front, side and top views. These three photographs were used to evaluate the visual deviations of the visualized hologram.



(a) (b) (c)

Figure 4.14: Three views of the overlapping effect between a real object and a corresponding hologram. (a) Front view. (b) Side view. (c) Top view.

An example of the results acquired from all three views is shown in Figure 4.14, where the gray box is the visualized hologram and the white box is the real box used

for testing. It is shown that the visualized hologram exhibits good overlap with the real box. The hologram has a slight shift in the top view, which is due to the head posture approximation error of HoloLens when the user moves his/her head quickly. The accuracy deviations of the length, width and height collected throughout the entire experiment are shown in Figure 4.15, where the error bars represent the standard deviation for each condition. The average σ_A^V value is 6.64%, with a standard deviation of 3.29%, which proves that holograms can be visualized precisely using HoloLens. The average difference in terms of the Euler distance is 1.25 cm, with a standard deviation of 0.25 cm. To human eye, the hologram is essentially the same size as the real object.

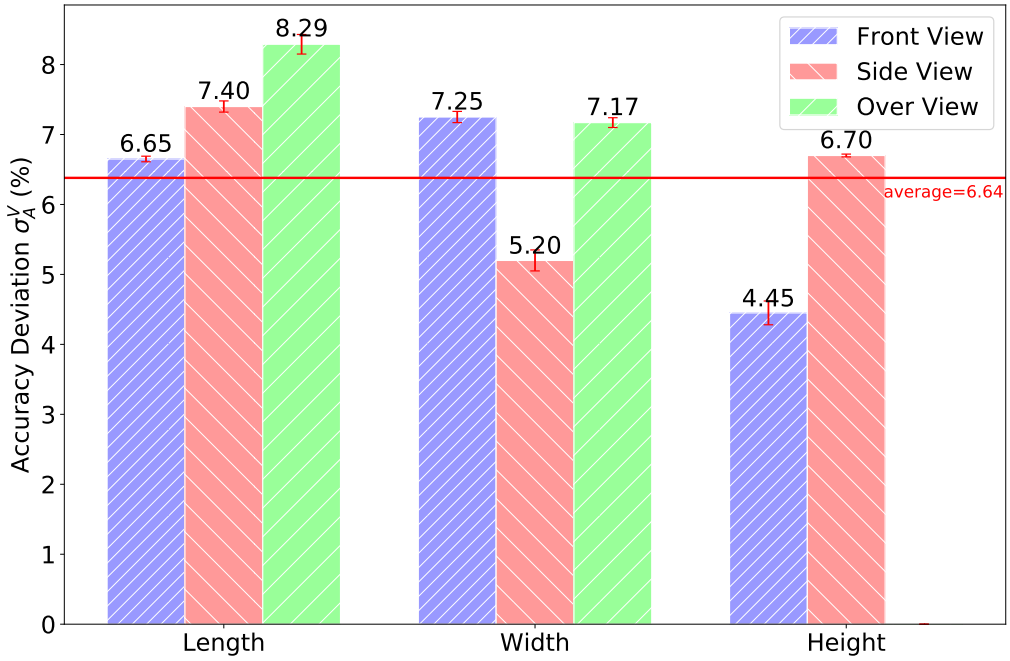


Figure 4.15: The accuracy deviations σ_A^V for hologram visualization. The size of the visualized box was 0.25 m×0.2 m×0.1 m.

4.2.7 Speech Recognition

In speech recognition experiment, the participants were first asked to practice speaking the eight user-defined commands and the eight system-defined commands. Each command was required to be identified by HoloLens at least five times. The participants were then asked to speak each command ten times in a random order. The number of recognized commands was counted to evaluate the speech recognition capability of HoloLens.

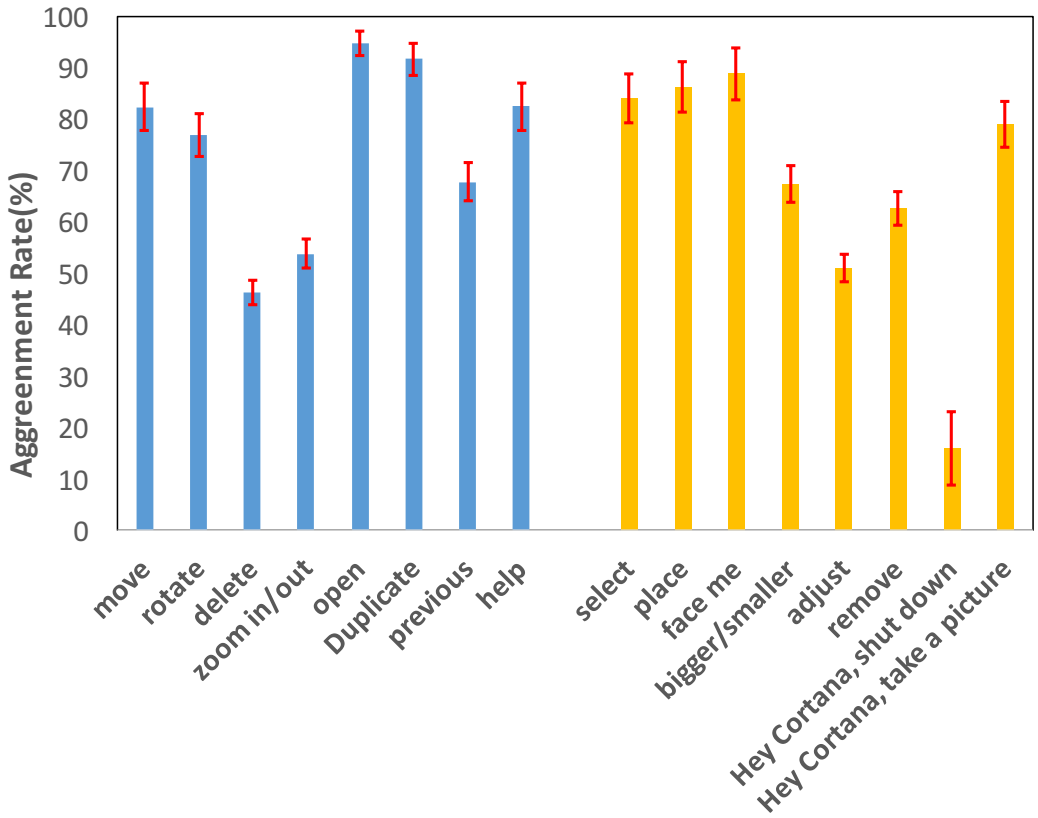


Figure 4.16: The agreement rates A_r for speech recognition, where the blue and yellow bars represent the agreement rates A_r for the user-defined commands and system-defined commands, respectively.

All participants had been living in an English-speaking environment for more than

2 years and could speak English fluently. The native languages of the participants were English (6), Chinese (7), Arabic (3), French (2) and Hindi (1).

To analyze the speech recognition capability of HoloLens, we computed the agreement rates A_r for the selected commands. The agreement rate A_r represents the level of consensus among the participants for a specific referent r and is defined as

$$A_r = \sum_{P_i \subset P_r} \left(\frac{|P_i|}{|P_r|} \right)^2 \tag{4.7}$$

where P_r is the set of operation commands for referent r and P_i is a subset of P_r. The value of A_r ranges from $|P_r|^{-1}$ to 1, where $|P_r|^{-1}$ indicates no agreement and 1 indicates perfect agreement.

The agreement rates for the selected referents are shown in Figure 4.16, where the blue and yellow bars represent the agreement rates A_r for the user-defined commands and system-defined commands, respectively, and the error bars represent the standard deviation for each condition. The average agreement rates A_r for the user-defined commands and the system-defined commands are 74.47% and 66.87%, respectively, with standard deviations of 16.21% and 22.77%, respectively. A system malfunction occurred during the testing process: the system-defined command "Hey Cortana, shut down" could be correctly recognized but could not call the relevant event. In addition, during the testing process, it was difficult for HoloLens to recognize phrases when a participant spoke two words separated by only a very short pause or no pause at all.

Chapter 5

Discussion

5.1 QoE Evaluation Discussion

During our user study, participants reported several advantages and disadvantages of the semi-transparent AR system HoloLens, which influenced their experience. To validate that our designed questionnaire could accurately reflect a user's experience, we compared user feedback with the questionnaire results to discuss their correlations.

First, most participants enjoyed the movement freedom offered by HoloLens, due to its capability to process all data itself without the requirement to connect with external computing machines through cables (average score 3.92 out of 5). However, as a completely integrated head-mounted-display (HMD) device, the weight of all the components is around 579g, and it is entirely carried by the user's head. Therefore, its heavy weight degrades a user's comfortable level while wearing the device, and causes fatigue after a while. Because of the weight, after a test of approximately 15 minutes, there was always marks on the forehead of users. Thus, the average score for device comfort level is just 3.21. As a result, the corresponding high-level parameter, namely, hardware quality, earned a score of 3.63.

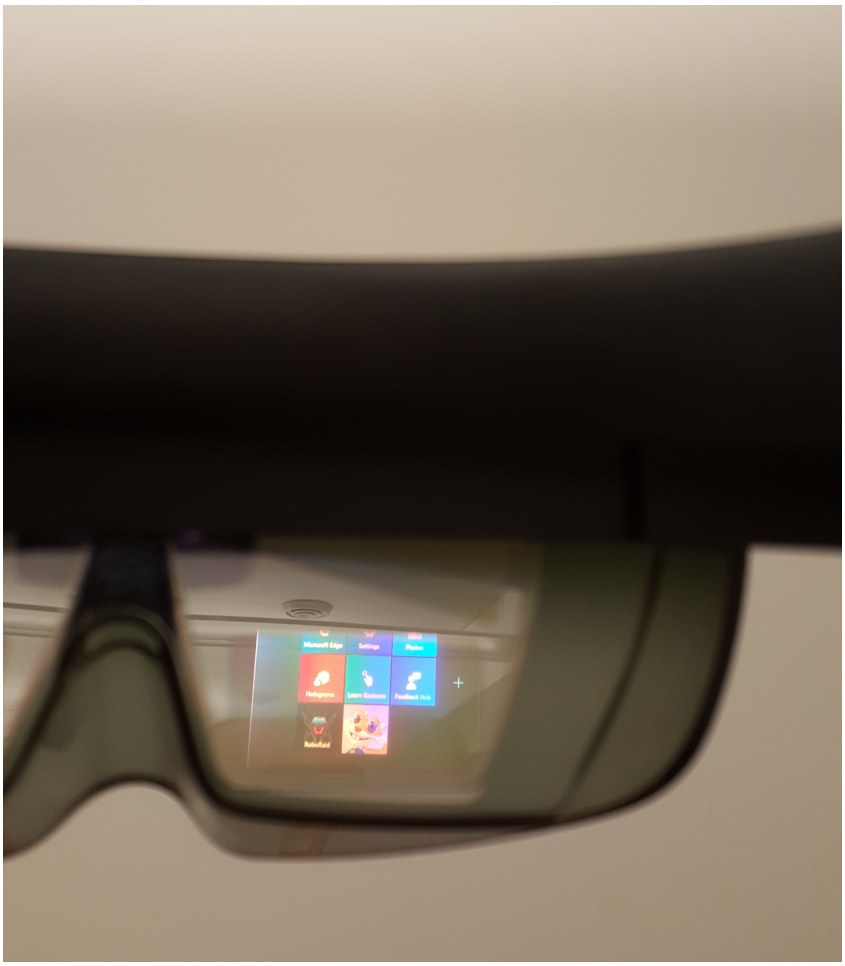


Figure 5.1: Limited visual display region of HoloLens.

Second, though HoloLens can vividly display augmented 3D contents upon the real surrounding environment with its advanced holographic projecting technology, several participants complained about its limited visual regions, which affect the visual quality score (average 3.83). This is because the holographic projectors are embedded in fixed locations, and can only project contents on certain areas, as shown in Figure 5.1. Therefore, before starting the test, users usually needed to carefully adjust the position of HoloLens to make sure their eyes were able to see most of the projected contents. During the test, because of the heavy weight of HoloLens, it occasionally slides down during strenuous movements, and made users' viewing angle change, which at times caused users to hold it with one hand while playing. This is another factor that affected the user's AR experience.

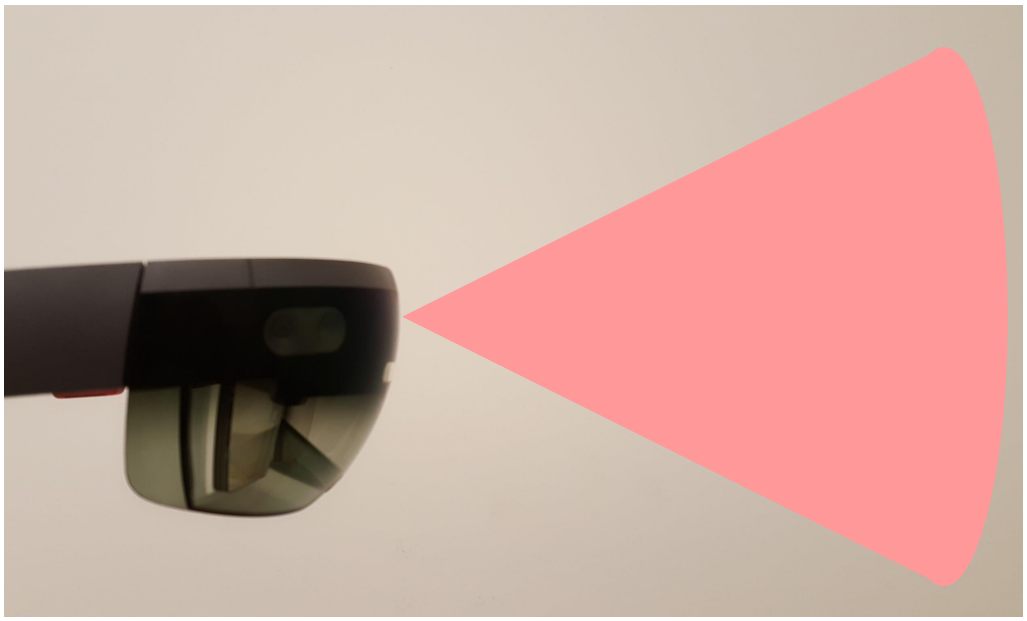


Figure 5.2: Gesture detection region (cone shaped area).

Third, since gesture commands are detected and recognized with a small-size depth camera, embedded inside the front of HoloLens, it can only recognize a few gestures within a small region, as the cone-shape shown in Figure 5.2. Therefore, the air-tap

gesture, instead of more natural gestures (e.g. finger gun gesture), is defined as the shooting action in the "RoboRaid" application, and just received a score of 3.16. On the other hand, using body movements to dodge attacks provides participants a natural interaction experience, users can simply move their body to realize this interaction. Therefore, the average score for this part was 4.24.

These facts also prove that the parameters in our proposed QoE taxonomy for holographic AR multimedia device evaluation can reflect participants' actual AR experience, and would be helpful for the design of these devices and their application with specific aspects.

5.2 Discussion about QoE and QoS Evaluation

As an inter-disciplinary research field, the area of semi-transparent AR systems consists of works from many other related areas. By dividing a semi-transparent AR system into several main functional components, including head localization, 3D content generation or reconstruction, computing performance, real environment reconstruction, spatial mapping, hologram visualization, and user interaction, we designed and conducted experiments for each of them to technically evaluate and measure the performance of each sub-system.

In the previous chapter, we have proven that user ratings for each high-level QoE parameter are similar to the average rating of its related low-level parameters. Since low-level QoE parameters connect more closely with QoS parameters, it is possible to find the relationships between them, and then finally map a QoE model with a QoS model. We explored several functional components of semi-transparent AR systems, but more work needs to be done in the future to discover the entire mappings.

Through testing these seven principal components of semi-transparent AR systems, we also explored their relationships with QoE parameters in our proposed taxonomy, to

understand the reasons behind the rating scores obtained for certain QoE parameters. Because semi-transparent AR systems contain several components from various research fields, our experiments did not cover all the small functional components. We selected one or more aspects from each component as the example to evaluate the sub-system. Therefore, there are also some small aspects of each principal component that can be further investigated. For example, besides speech recognition, the user interaction component can also be tested with gesture recognition for different hand types and body movement interactions. If all QoS technical aspects can be explored and matched with related QoE parameters, it would greatly benefit the design of semi-transparent AR systems, since QoS parameters are more intuitive for manufactures and developers and may help them improve the product, while QoE parameters are more suitable to describe a user's experience.

In addition, since each functional component may affect several QoE parameters, and each QoE parameter may also be influenced by several functional components, there exists weighting values used to describe the influence between them. So far, the exact relations between QoE and QoS for semi-transparent AR systems have not been fully discovered, which remains to be the future work.

The current version of HoloLens is only developing edition, it is a prototype of an independent head-mounted semi-transparent AR system. In our experiments, its overall QoE rating has obtained an average of 82.853 out of 100 points. With the improvements of CPU, HPU, memory size, device material, and other aspects, the next version of HoloLens, or other similar AR systems, will surely further improve user's AR QoE with faster and more accurate interactions, a lighter weight, a broader field-of-view, and possibly additional features.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Augmented Reality (AR) technology enables users to see a combination of real and virtual contents in real-time through augmenting virtual holograms on the real environment. Besides displaying visual contents, semi-transparent AR system also combines multiple types of sensors and components to improve users' interactions with these contents. Therefore, AR systems are able to provide a multimedia experience to users.

In this thesis, we proposed a Quality-of-Experience (QoE) taxonomy for evaluating semi-transparent AR systems, through referencing QoE taxonomy of other related fields as well as the feedback from our user study. We also designed a QoE evaluation study to understand the relationships of QoE parameters, and selected high-level QoE parameters as inputs to build a Fuzzy-Inference-System (FIS) model.

To prove that the parameters in our proposed QoE framework can accurately reflect users' experiences with a holographic AR device, we estimate users' general experience by analyzing their ratings of the corresponding parameters. In other words, we created a model that takes the parameter ratings as inputs and generates the estimated general

experience score as the output. We performed a user study, where we asked participants to complete a questionnaire regarding all parameters in the framework (except haptic feedback, odor and taste) and to provide a general rating score (a value between 0 and 100) for their AR experience. Subsequently, we used a part of the data to develop a FIS model and test it with the remaining data. Since the general scores estimated by the FIS model were similar to the general rating scores obtained from the users (the ground truth), both our proposed framework and the FIS model are validated.

In addition, since semi-transparent AR systems consist of multiple functional components, we also designed several experiments to explore these components separately. Through conducting specific testing for head localization, 3D content generation or reconstruction, computing performance, real environment reconstruction, spatial mapping, hologram visualization, and user interaction, we technically evaluated and measured the performance of each sub-system of the Microsoft HoloLens AR system.

By comparing and analyzing the user ratings for the QoE evaluation as well as the measurements for the corresponding sub-functional components, we discovered several relationships revealing the connections between QoE and QoS parameters. For example, by evaluating the computing power of HoloLens (QoS parameter), we found that it is occasionally slow to respond to user interactions, because of its limited computing power as an independent HMD device, which explains why user ratings for the corresponding QoE parameter "system response" is not high.

Through our work, we conducted a comprehensive evaluation of semi-transparent AR systems, which is of great value to AR developers and users, allowing them to be aware of the capabilities and limitations of AR systems. Future developments and improvements of similar devices can also use our work as a reference to increase efficiency.

6.2 Future Work

Since a semi-transparent AR system is a combination of multiple sub-systems, we plan to integrate more aspects, such as haptic feedback, digital odor, and taste, to explore their influence on users' QoE. We believe that these additions would improve the user experience. Measuring and evaluating these multimodal attributes for semi-transparent AR multimedia devices would be of great value to developers and designers, to help them improve their products and provide users a richer multimedia experience, including audio, video, haptics, and more.

We have already explored the relationships between several QoE parameters and QoS technical parameters. However, since one QoE parameter may depend on several QoS parameters, and one QoS parameter could affect multiple QoE ratings, more experiments and analyses need to be designed and conducted to discover more detailed connections between them. With complete mappings between QoE and QoS evaluations, we could freely adjust the desired parameters when developing or improving applications and projects. In addition, a model can also be built to estimate the influence of certain parameters on the general experience.

Computing hardware is developing at an unprecedented pace, creating great opportunities in the field. With more powerful computing capabilities, semi-transparent AR systems would be able to provide users a better experience. Therefore, an important part of future work is to develop and improve the computing components within semi-transparent AR systems, especially the CPU and HPU specifically designed for AR systems.

With improved computing power, machine-learning and deep-learning methods can also be utilized to learn from each user's interaction history, in order to identify and predict specific scenarios and user preferences, and provide a faster and more accurate response. With more and more user data, the design of semi-transparent AR systems

will better fit each user.

In addition, since our current work mainly focuses on a single user, exploring AR technology for teamwork will also be valuable. Evaluating the experience of using multiple AR multimedia devices simultaneously could guide developers and users to improve efficiency under collaborative conditions. For example, multiple AR users, located in different cities, could use a collaborative AR system to plan the structure of the "same" virtual city. One important aspect of multi-user AR system is finding the trade-off to balance the different preferences among users. For instance, the data synchronization frequency of the system may not be adjustable for each user, therefore the frequency that can obtain the best QoE ratings in general could be selected, instead of the one that only suits a curtain number of users.

Semi-transparent AR systems can be widely used for various occasions and purposes. Therefore, an adaptive QoE model, which can be easily adjusted to suit different types of AR applications, would be very important for this field.

Appendix A

Guideline to Perform Experiments with Semi-Transparent AR Device

Through conducting experiments to evaluate semi-transparent AR device and applications, we summarize the experience and lessons learned to provide a guideline for future AR user study and application development.

Experimental Environment Setup

Since semi-transparent AR device mainly works with indoor environment, the experiments should be conducted in a room with normal light conditions. Being too bright or too dark will affect the performance of optical see-through glasses. The room size should also be large enough to allow users move around to interact with holograms freely.

Due to the nature of AR application, it requires interaction with real surrounding environment, such as wall, floor, desk, chairs, and so on. Therefore, a room with certain amount of stuffs would better evaluate AR applications.

Subjects Selection

To conduct comprehensive evaluation for AR application, subjects should be selected with various backgrounds, such as different gender, age, eduction level, AR experience, game experience, relationship with the developer / experimenter, and so on.

As many AR applications need users to move around or rotate head / body to interact with holograms, proper instructions and waiver before the experiment may be helpful.

AR Device Setup

For head-mounted AR device, adjusting the device position for users is very important. The display region of the device should fit the users' eyes to enable a wide and complete AR view.

Device tightness also plays a significant role: being too tight may cause user headache, while being too loose will cause the device to slide down. Users' front hair should be placed outside the device, since it makes the device more easily to slide down.

Device portal is a useful tool to mirror the view of the user and display it on a computer. Thus, developer / experimenter can be aware of the user's views real-time, to give instructions and hints based on the situation.

Basic interactions, such as calling main menu, selecting icon, moving with holograms, speaking voice commands, can be demonstrated to users step by step first. After users get familiar with each interaction, they can practice more complex interactions with the applications.

Questionnaire Design

The initial questionnaire may miss some aspects needed to evaluate the AR application. Therefore, after the experiment with each user, a discussion with them about their AR experience and feedback for the questionnaire is crucial. If we realize certain key parameters are missing in the questionnaire, then we need to either find a way to make up the missing ones, or even discard all previous experiments data if necessary.

Other Tips

Based on our experiments, we also obtained some other tips:

- Avoid making user moving or rotating too rapidly, since AR device may have localization errors in these situations.
- Experiments can be conducted with both intense and relaxed applications to enable users focus on different aspects.
- Holograms may not interact with the real environment properly all the time, so testing with various conditions before experiments would be helpful.

References

[1] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77:1321–1329, 1994.

[2] Cyberware whole body color 3D scanner bundle. http://cyberware.com/products/scanners/wbx.html, accessed Aug. 30, 2018.

[3] Creatform MetraSCAN 210. http://www.creaform3d.com/en, accessed Aug. 30, 2018.

[4] LMI HDI Advance R3x. http://www.creaform3d.com/en, accessed Aug. 30, 2018.

[5] Microsoft Kinect. http://www.microsoft.com/en-us/kinectforwindows/, accessed Aug. 30, 2018.

[6] MESA-Imaging SwissRanger 4500. http://www.mesa-imaging.ch/swissranger4500.php, accessed Aug. 30, 2018.

[7] X-View X5000. http://www.xviewct.com/industrial-ct-systems/x-view-computed-tomography/m5000-series, accessed Aug. 30, 2018.

[8] Toshiba TOSCANER 20000AV. http://www.toshiba-itc.com/cat/en/prod01.html, accessed Aug. 30, 2018.

[9] A. Nathan and S. Gao. Interactive displays: The next omnipresent technology [point of view]. *Proceedings of the IEEE*, 104(8):1503–1507, 2016.

[10] J. Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42:73–93, 1992.

[11] W. Wu, A. Arefin, R. Rivas, et al. Quality of experience in distributed interactive multimedia environments: toward a theoretical framework. In *proceedings of the 17th ACM International Conference on Multimedia*, pages 481–490, 2009.

[12] A. Hamam, A. El Saddik, and J. AljaAm. A quality of experience model for haptic virtual environments. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 10:28:1–28:23, 2014.

[13] A. El Saddik. Digital twins: A multimedia perspective. *IEEE Multimedia*, 25, 2018.

[14] I. Sutherland. A head-mounted three dimensional display. In *proceedings of the Joint Computer Conference*, pages 757–764, 1968.

[15] W. Krueger, T. Gionfriddo, and K. Hinrichsen. Videoplace an artificial reality. In *ACM SIGCHI Bulletin*, volume 16, pages 35–40, 1985.

[16] J. Carmigniani, B. Furht, M. Anisetti, et al. Augmented reality technologies, systems and applications. *Multimedia Tools and Applications*, 51:341–377, 2011.

[17] L. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *proceedings of IEEE Virtual Reality Annual International Symposium*, pages 76–82, 1993.

[18] K. Pagano, A. Haddad, and T. Crosby. Virtual reality-making good on the promise of immersive learning: The effectiveness of in-person training, with the logistical and cost-effective benefits of computer-based systems. *IEEE Consumer Electronics Magazine*, 6:45–47, 2017.

[19] M. Billinghurst, A. Clark, and G. Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8:73–272, 2015.

[20] L. Zhang, H. Dong, and A. El Saddik. From 3D sensing to printing: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 12:27:1–27:23, 2015.

[21] K. Ikeuchi. Modeling from reality. In *Proceedings of the 3rd IEEE International Conference on 3-D Digital Imaging and Modeling*, pages 117–124, 2001.

[22] F. Remondino and S. El-Hakim. Image-based 3D modelling: A review. *The Photogrammetric Record*, 21:269–291, 2006.

[23] S. Zhu and J. Gao. 3D modeling and rendering based on uncalibrated single view image in undergraduate final design. In *Proceedings of International Conference on Computer Science and Information Processing*, pages 1336–1340, 2012.

[24] P. Rauschnabel and A. Rossmann. An adoption framework for mobile augmented reality games: The case of pokémon go. *Computers in Human Behavior*, 76:276–286, 2017.

[25] Y. Ro, A. Brem, and P. Rauschnabel. Augmented reality smart glasses: Definition, concepts and impact on firm value creation. In *Augmented Reality and Virtual Reality*, pages 169–181. Springer, 2018.

[26] Z. Lv, L. Feng, H. Li, and S. Feng. Hand-free motion interaction on Google Glass. In *proceedings of SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, page 21, 2014.

[27] P. Rauschnabel, D. Hein, J. He, Y.Ro, and S. Rawashdeh. Fashion or technology? A fashnology perspective on the perception and adoption of augmented reality smart glasses. *i-com*, 15:179–194, 2016.

[28] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, et al. Holoportation: Virtual 3D teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 741–754, 2016.

[29] M. Kalantari and P. Rauschnabel. Exploring the early adopters of augmented reality smart glasses: The case of Microsoft Hololens. In *Augmented Reality and Virtual Reality*, pages 229–245. 2018.

[30] R. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6:355–385, 1997.

[31] R. McNaney, J. Vines, D. Roggen, et al. Exploring the acceptability of google glass as an everyday assistive device for people with parkinson's. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, pages 2551–2554, 2014.

[32] O. Matei, I. Vlad, R. Heb, et al. Comparison of various epson smart glasses in terms of real functionality and capabilities. *Carpathian Journal of Electrical Engineering*, 10:31–38, 2016.

[33] R. Jain. Quality of experience. *IEEE MultiMedia*, 11:96–95, 2004.

[34] S. Moller P. Le Callet and A. Perkis. Qualinet white paper on definitions of quality of experience. *European Network on Quality of Experience in Multimedia Systems and Services*, 3:1–19, 2012.

[35] F. Zhou, L. Duh, and M. Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202, 2008.

[36] H. Ling. Augmented reality in reality. *IEEE MultiMedia*, 24:10–15, 2017.

[37] I. Goradia, J. Doshi, and L. Kurup. A review paper on oculus rift & project morpheus. *International Journal of Current Engineering and Technology*, 4:3196–3200, 2014.

[38] R. Vassallo, C. Chen A. Rankin, and M. Peters. Hologram stability evaluation for Microsoft Hololens. In *proceedings of the SPIE Medical Imaging*, volume 10136, pages 14:1–14:6, 2017.

[39] D. Van Krevelen and R. Poelman. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, 9:1–20, 2010.

[40] Y. Liu, X. Qin, S. Xu, E. Nakamae, and Q. Peng. Light source estimation of outdoor scenes for mixed reality. *The Visual Computer*, 25:637–646, 2009.

[41] A. Takagi, S. Yamazaki, Y. Saito, and N. Taniguchi. Development of a stereo video see-through HMD for AR systems. In *proceedings of the IEEE and ACM International Symposium on Augmented Reality*, pages 68–77, 2000.

[42] F. Biocca and J. Rolland. Virtual eyes can rearrange your body: Adaptation to visual displacement in see-through, head-mounted displays. *Presence*, 7:262–277, 1998.

[43] M. McCartney. Margaret mccartney: Game on for Pokémon Go. *BMJ Journal*, 354:4306, 2016.

[44] Z. Lv, L. Feng, H. Li, and S. Feng. Hand-free motion interaction on google glass. In *Proceedings of SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*, page 21, 2014.

[45] S. Blessenohl, C. Morrison, A. Criminisi, and J. Shotton. Improving indoor mobility of the visually impaired with depth-based spatial sound. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 26–34, 2015.

[46] W. Choi, J. Cho, and I. Shin. Toward the holographic reconstruction of sound fields using smart sound devices. *IEEE Multimedia*, 23:64–74, 2016.

[47] L. Yang, L. Zhang, H. Dong, et al. Evaluating and improving the depth accuracy of kinect for windows v2. *IEEE Sensors Journal*, 15:4275–4285, 2015.

[48] M. Levoy, K. Pulli, B. Kari, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 131–144, 2000.

[49] L. Barazzetti, M. Scaioni, and F. Remondino. Orientation and 3D modelling from markerless terrestrial images: combining accuracy with automation. *The Photogrammetric Record*, 25:356–381, 2010.

[50] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3D shape scanning with a time-of-flight camera. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1173–1180, 2010.

[51] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 559–568, 2011.

[52] A. Newcombe, S. Izadi, O. Hilliges, et al. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[53] M. Rooker, A. Angerer, J. Capco, C. Heindl, A. Olarra, E. Fuentes, C. Wögerer, and A. Pichler. Flexible grasping of electronic consumer goods. In *Robotics in Smart Manufacturing*, pages 158–169. Springer Berlin Heidelberg, 2013.

[54] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3D full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics*, 18:643–650, 2012.

[55] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proceedings of Experimental Robotics*, pages 477–491, 2014.

[56] L. Clemente, A. Davison, I. Reid, J. Neira, and J. Tardós. Mapping large loops with a single hand-held camera. In *Proceedings of the Robotics: Science and Systems Conference*, 2007.

[57] A. Chatterjee, S. Jain, and V. Govindu. A pipeline for building 3D models using depth cameras. In *Proceedings of the 8th Indian Conference on Computer Vision, Graphics and Image Processing*, pages 38:1–38.8, 2012.

[58] G. Sansoni, M. Trebeschi, and F. Docchio. State-of-the-art and applications of 3d imaging sensors in industry, cultural heritage, medicine, and criminal investigation. *Sensors*, 9:568–601, 2009.

[59] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1362–1376, 2010.

[60] J. Rau and P. Yeh. A semi-automatic image-based close range 3D modeling pipeline using a multi-camera configuration. *Sensors*, 12:11271–11293, 2012.

[61] J. Wang, C. Zhang, W. Zhu, Z. Zhang, Z. Xiong, and P. Chou. 3D scene reconstruction by multiple structured-light based commodity depth cameras. In *Proceedings of*

IEEE International Conference on Acoustics, Speech and Signal Processing, pages 5429–5432, 2012.

[62] Y. Kang and Y. Ho. High-quality multi-view depth generation using multiple color and depth cameras. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1405–1410, 2010.

[63] E. Ciaccio, C. Tennyson, G. Bhagat, S. Lewis, and P. Green. Use of shape-from-shading to estimate three-dimensional architecture in the small intestinal lumen of celiac and control patients. *Computer Methods and Programs in Biomedicine*, 111:676–684, 2013.

[64] M. Adm and A. Said. Interactive image-based 3D modeling. In *Proceedings of International Conference on Computer & Information Science*, pages 1000–1005, 2012.

[65] N. Patel and M. Zaveri. 3D model reconstruction and animation from single view face image. In *Proceedings of International Conference on Audio, Language and Image Processing*, pages 674–682, 2012.

[66] L. Zhang, J. Saboune, and A. El Saddik. Transforming a regular screen into a touch screen using a single webcam. *Journal of Display Technology*, 10:647–659, 2014.

[67] W. Wang and J. Zhao. Hiding depth information in compressed 2d image/video using reversible watermarking. *Multimedia Tools and Applications*, 75:4285–4303, 2016.

[68] L. Zhang, H. Dong, and A. El Saddik. A multisensory datafusion-based 3D plank-coaching system. In *Digital Media Industry & Academic Forum*, pages 154–157, 2016.

[69] K. Žbontar, M. Mihelj, B. Podobnik, F. Povše, and M. Munih. Dynamic symmetrical pattern projection based laser triangulation sensor for precise surface position measurement of various material types. *Applied Optics*, 52:2750–2760, 2013.

[70] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Transactions on Graphics*, 22:587–594, 2003.

[71] Y. Zhou, K. Liu, J. Gao, K. Barner, and F. Kiamilev. High-speed structured light scanning system and 3D gestural point cloud recognition. In *Proceedings of IEEE Conference on Information Sciences and Systems*, pages 1–6, 2013.

[72] J. Sturm, E. Bylow, F. Kahl, and D. Cremers. CopyMe3D: Scanning and printing persons in 3D. In *Pattern Recognition*, pages 405–414. Springer Berlin Heidelberg, 2013.

[73] K. Liu, Y. Wang, D. Lau, Qi. Hao, and L. Hassebrook. Dual-frequency pattern scheme for high-speed 3-D shape measurement. *Optics Express*, 18:5229–5244, 2010.

[74] J. Salvi, S. Fernandez, and and X. Llado T. Pribanic. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43:2666–2680, 2010.

[75] K. Liu, Y. Wang, D. Lau, Q. Hao, and L. Hassebrook. Gamma model and its analysis for phase measuring profilometry. *Journal of the Optical Society of America A*, 27:553–562, 2010.

[76] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 129–134, 2011.

[77] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. Algorithms for 3D shape scanning with a depth camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1039–1050, 2013.

[78] U. Wijenayake, S. Baek, and S. Park. An error correcting 3D scanning technique using dual pseudorandom arrays. In *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 517–523, 2012.

[79] J. Smisek, M. Jancosek, and T. Pajdla. 3D with kinect. In *Consumer Depth Cameras for Computer Vision*, pages 3–25. Springer London, 2013.

[80] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. In *Proceedings of Eurographics (State-of-the-Art Report)*, pages 119–134, 2009.

[81] R. Lange and P. Seitz. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37:390–397, 2001.

[82] B. Gokturk, H. Yalcin, and C. Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *Proceedings of Computer Vision and Pattern Recognition Workshop*, pages 35–35, 2004.

[83] A. Kirmani, A. Benedetti, and P. Chou. Spumic: Simultaneous phase unwrapping and multipath interference cancellation in time-of-flight cameras using spectral methods. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 1–6, 2013.

[84] D. Anderson, H. Herman, and A. Kelly. Experimental characterization of commercial flash ladar devices. In *Proceedings of International Conference of Sensing and Technology*, pages 17–23, 2005.

[85] M. Hansard, S. Lee, O. Choi, and R. Horaud. *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer Science & Business Media, 2012.

[86] T. Möller, H. Kraft, J. Frey, M. Albrecht, and R. Lange. Robust 3D measurement with pmd sensors. In *Proceedings of the 1st Range Imaging Research Day*, page 8, 2005.

[87] J. Kruth, M. Bartscher, S. Carmignato, R. Schmitt, and and A. Weckenmann L. De Chiffre. Computed tomography for dimensional metrology. *Manufacturing Technology*, 60:821–842, 2011.

[88] L. De Chiffre, S. Carmignato, J. Kruth, R. Schmitt, and A. Weckenmann. Industrial applications of computed tomography. *Manufacturing Technology*, 63:655–677, 2014.

[89] R. Ketcham and W. Carlson. Acquisition, optimization and interpretation of x-ray computed tomographic imagery: Applications to the geosciences. *Computers & Geosciences*, 27:381–400, 2001.

[90] C. Hu, M. Meng, P. Liu, and X. Wang. Visual gesture recognition for human-machine interface of robot teleoperation. In *proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1560–1565, 2003.

[91] S. Gao, J. Lai, C. Micou, and A. Nathan. Reduction of common mode noise and global multivalued offset in touch screen systems by correlated double sampling. *Journal of Display Technology*, 12:639–645, 2016.

[92] L. Zhang. Development of a haptic video chat system. Master's thesis, University of Ottawa, 2012.

[93] Z. Zhang and S. Ying. Visual screen: Transforming an ordinary screen into a touch screen. In *Proceedings of IAPR Workshop on Machine Vision Applications*, pages 215–218, 2000.

[94] K. Cheng and M. Takatsuka. Initial evaluation of a bare-hand interaction technique for large displays using a webcam. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 291–296, 2009.

[95] X. Wang, X. Zhang, and G. Dai. Tracking of deformable human hand in real time as continuous input for gesture-based interaction. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, pages 235–242, 2007.

[96] A. Anagnostopoulos and A. Pnevmatikakis. A realtime mixed reality system for seamless interaction between real and virtual objects. In *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, pages 199–204, 2008.

[97] W. Wang and J. Zhao. Robust image chroma-keying: a quadmap approach based on global sampling and local affinity. *IEEE Transactions on Broadcasting*, 61(3):356–366, 2015.

[98] L. Yin, W. Wang, and J. Zhao. Real-time video chroma keying: a parallel approach based on local texture and global colour distribution. *IET Image Processing*, 10:638–645, 2016.

[99] S. Gao, D. McLean, J. Lai, C. Micou, and A. Nathan. Reduction of noise spikes in touch screen systems by low pass spatial filtering. *Journal of Display Technology*, 12:957–963, 2016.

[100] S. Kang, M. Nam, and P. Rhee. Color based hand and finger detection technology for user interaction. In *Proceedings of the International Conference on Convergence and Hybrid Information Technology*, pages 229 –236, 2008.

[101] L. Zhang, H. Dong, and A. El Saddik. Technical evaluation of hololens for multimedia: A first look. *IEEE Multimedia*, 25:8–18, 2018.

[102] H. Li, C. Miao, L. Zhang, W. Tian, and J. Wang. Research on key technologies of ecg monitoring system based on internet of things. *Application Research of Computers*, 27:4600–4603, 2010.

[103] F. James and R. Gurram. Controlling user interfaces with contextual voice commands, 2014. US Patent 8788271B2.

[104] J. Jacobsen, C. Parkinson, and S. Pombo. Remote control of host application using motion and voice commands, 2016. US Patent 9,235,262.

[105] E. Marchand, H. Uchiyama, and F. Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22:2633–2651, 2016.

[106] G. Simon, A. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality*, pages 120–128, 2000.

[107] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19(2):4–10, 2012.

[108] R. Furlan. The future of augmented reality: Hololens - Microsoft's AR headset shines despite rough edges. *IEEE Spectrum*, 53:21, 2016.

[109] Inside Microsoft's HoloLens. https://www.theverge.com/2016/4/6/11376442/microsoft-hololens-holograms-parts-teardown-photos-hands-on, accessed Aug. 30, 2018.

[110] Microsoft's Hololens secret sauce: A 28nm customized 24-core DSP engine built by TSMC. https://www.theregister.co.uk/2016/08/22/micro soft_hololens_hpu/, accessed Aug. 30, 2018.

[111] L. Avila and M. Bailey. Augment your reality. *IEEE Computer Graphics and Applications*, 36(1):6–7, 2016.

[112] A. Lu, J. Huang, S. Zhang, et al. Towards mobile immersive analysis: A study of applications. In *Proceedings of IEEE Immersive Analytics Workshop*, 2016.

[113] L. Zhang, S. Chen, H. Dong, and A. El Saddik. Visualizing Toronto city data with HoloLens: Using augmented reality for a city model. *IEEE Consumer Electronics Magazine*, 7:73–80, 2018.

[114] S. Mohanty, U. Choppali, and E. Kougianos. Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5:60–70, 2016.

[115] H. Dong, G. Singh, A. Attri, and A. El Saddik. Open data-set of seven canadian cities. *IEEE Access*, 5:529–543, 2017.

[116] Z. Lv, X. Li, B. Zhang, et al. Managing big city information based on WebVRGIS. *IEEE Access*, 4:407–415, 2016.

[117] L. Zhang, B. Han, H. Dong, and A. El Saddik. Development of an automatic 3D human head scanning-printing system. *Multimedia Tools and Applications*, 76:4381–4403, 2017.

[118] S. Chen and K. Nahrstedt. An overview of quality of service routing for next-generation high-speed networks: Problems and solutions. *IEEE Network*, 12:64–79, 1998.

[119] A. Gupta and R. Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015.

[120] S. Islam, P. Liu, and A. El Saddik. Nonlinear control for teleoperation systems with time varying delay. *Nonlinear Dynamics*, 76:931–954, 2014.

[121] K. Banerjee and G. Nitsche. Quality of service for WLAN and bluetooth combinations, 2015. US Patent 8929259B2.

[122] P. Agrawal, S. Baba, and F. Vakil. System and method for quality of service management in mobile wireless networks, 2014. US Patent 8631104B2.

[123] W. Bul'ajoul, A. James, and M. Pannu. Improving network intrusion detection system performance through quality of service configuration and parallel technology. *Journal of Computer and System Sciences*, 81:981–999, 2015.

[124] S. Mohanty, U. Choppali, and E. Kougianos. Everything you wanted to know about smart cities: The internet of things is the backbone. *IEEE Consumer Electronics Magazine*, 5:60–70, 2016.

[125] C. Hua and P. Liu. Teleoperation over the internet with/without velocity signal. *IEEE Transactions on Instrumentation and Measurement*, 60:4–13, 2011.

[126] O. Gaddour, A. Koubâa, and M. Abid. Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL. *Ad Hoc Networks*, 33:233–256, 2015.

[127] K. Chen, Y. Chang, H. Hsu, D. Chen, C. Huang, and C. Hsu. On the quality of service of cloud gaming systems. *IEEE Transactions on Multimedia*, 16:480–495, 2014.

[128] A. Abdelmaboud, D. Jawawi, I. Ghani, A. Elsafi, and B. Kitchenham. Quality of service approaches in cloud computing: A systematic mapping study. *Journal of Systems and Software*, 101:159–179, 2015.

[129] D. You and K. Chung. Quality of service-aware dynamic voltage and frequency scaling for embedded gpus. *IEEE Computer Architecture Letters*, 14:66–69, 2015.

[130] D. Sahin, V. Gungor, T Kocak, and G. Tuna. Quality-of-service differentiation in single-path and multi-path routing for wireless sensor network-based smart grid applications. *Ad Hoc Networks*, 22:43–60, 2014.

[131] C. Ge, N. Wang, G. Foster, and M. Wilson. Toward QoE-assured 4K video-on-demand delivery through mobile edge virtualization with adaptive prefetching. *IEEE Transactions on Multimedia*, 19:2222–2237, 2017.

[132] K. Mok, W. Chan, and K. Chang. Measuring the quality of experience of http video streaming. In *proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management and Workshops*, pages 485–492, 2011.

[133] A. Bentaleb, C. Begen, R. Zimmermann, and S. Harous. SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming. *IEEE Transactions on Multimedia*, 19:2136–2151, 2017.

[134] C. Keighrey, R. Flynn, S. Murray, and N. Murray. A QoE evaluation of immersive augmented and virtual reality speech language assessment applications. In *proceedings of the 9th International Conference on Quality of Multimedia Experience*, pages 1–6, 2017.

[135] G. Evans, J. Miller, M. Pena, et al. Evaluating the Microsoft Hololens through an augmented reality assembly application. In *proceedings of the SPIE Degraded Environments: Sensing, Processing, and Display*, volume 10197, pages V:1–16, 2017.

[136] L. Zhang, H. Dong, and A. El Saddik. Towards a qoe model to evaluate holographic augmented reality devices: A hololens case study. *IEEE MultiMedia*, 2019.

[137] L. Zhang, J. Saboune, and A. El Saddik. Development of a haptic video chat system. *Multimedia Tools and Applications*, 74:5489–5512, 2015.

[138] A. El Saddik, M. Orozco, M. Eid, and J. Cha. *Haptics technologies: Bringing touch to multimedia*. Springer Science & Business Media, 2011.

[139] F. Arafsha, L. Zhang, H. Dong, and A. El Saddik. Contactless haptic feedback: state of the art. In *proceedings of the IEEE International Symposium on Haptic, Audio and Visual Environments and Games*, pages 1–6, 2015.

[140] A. Hamam and A. El Saddik. User force profile of repetitive haptic tasks inducing fatigue. In *proceedings of the 7th International Workshop on Quality of Multimedia Experience*, pages 1–6, 2015.

[141] K. Chang. *Geographic information system*. Wiley Online Library, 2006.

[142] A. Nathan and S. Gao. Interactive displays: The next omnipresent technology [point of view]. *Proceedings of the IEEE*, 104:1503–1507, 2016.

[143] O. Wobbrock, R. Morris, and D. Wilson. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1083–1092, 2009.

[144] J. Silva and A. El Saddik. Exertion interfaces for computer videogames using smartphones as input controllers. *Multimedia Systems*, 19:289–302, 2013.

[145] R. Steinmetz and C. Engler. *Human perception of media synchronization*. European Networking Center, 1993.

[146] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. In *Computer Graphics Forum*, volume 17, pages 167–174, 1998.

[147] R. Rockafellar and R. Wets. *Variational analysis*, volume 317. 2009.