Linköping University | Department of Computer Science
Bachelor thesis, 16 ECTS | Datateknik
2018 | LIU-IDA/LITH-EX-G--18/072--SE

# Impact of light on augmented reality

– Evaluating how different light conditions affect the performance of Microsoft HoloLens 3D applications

Effekten av ljus på augmented reality
– Utvärdering av hur olika ljusförhållanden påverkar funktionaliteten hos Microsoft HoloLens 3D-applikationer

**Lillemor Blom**

Examiner : Zebo Peng

External supervisor : Magnus Hammerin

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

# Abstract

Microsoft HoloLens is a headmounted augmented reality system providing users the ability to experience three-dimensional virtual content. This could be used in applications aimed at industry where users could use augmented reality to easily access information and receive instructions. For this to be suitable for industry, the system must be robust. One property of robustness was chosen for this thesis: system performance in conditions of different levels of light. A prototype implementing a use case for future industry was developed, as well as two additional smaller applications to facilitate experiments. Experiments were performed to investigate how different light levels affects the functionality in a 3D holographic application running on HoloLens and how the visibility of virtual content was affected in conditions with bright and heterogeneous backgrounds. The results showed that the functionality of the holographic application was not significantly affected except in very dark conditions, and that bright and messy backgrounds pose a problem to hologram visibility.

# Acknowledgments

I would like to thank my examiner Zebo Peng for his patience, my external supervisor Magnus Hammerin for help and support, Torbjörn and Per for sharing their wisdom on AR and otherwise, my test subjects Mårten, Gustav, and Alexander P, my (now) colleagues for being awesome in general, and last but not least my boyfriend Tomas for all his support.

# Contents

# List of Figures

# List of Tables

# Terminology and definitions

**Augmented reality (AR)** - a view of reality that is augmented with virtual objects, viewed using a device such as a tablet or a head mounted display.

**Field of view (FOV)** - in the context of augmented reality it is the area of the world that can be seen through the AR system, i.e. the part that can be augmented with virtual content.

**Head mounted display** - a device worn on the head, with displays positioned in front of the user's eyes. In combination with a computer it can be used to view reality, virtual reality or augmented reality.

**Hologram** - holographic or virtual content, something that is seen through the augmented reality system that is not a physical entity.

# 1 Introduction

Augmented reality is the concept of creating an experience where virtual elements are integrated in the real world. The virtual content augments the reality around the user. While the concept of augmented reality - AR for short - is not new, its popularity in both the general public and the scientific community may never have been higher [1]. A recent example of an augmented reality system which managed to gain the attention of the general public is the mobile phone application Pokemon Go [2]. There is extensive research made on how the use of augmented reality may be applied to many varying areas besides games, such as education [3] and health care [4]. There are different ways of implementing augmented reality, from apps running on smartphones to advanced systems developed specifically for the use of AR.

In this thesis a prototype application has been developed for wearable augmented reality device Microsoft HoloLens that implements a possible scenario of interaction between human and machine in an industrial context. That application and two smaller applications have then been used to examine how the performance of the HoloLens system is affected when used in different light conditions.

## 1.1 Background

Industry 4.0 is a term for a vision of how manufacturing industry can implement new technologies to reach new levels of efficiency and flexibility [5]. Systems and devices will be connected to each other and to the Internet in order to communicate and share data between themselves and with humans. Systems will be smart and automated and require as little human interaction as possible, and by being able to present large quantities of data to humans in an easily accessible manner, human decisions will be well-grounded and easy. One possibility of how this interaction between human and machine could be facilitated in such a scenario is to use augmented reality. A hypothetical scenario is a factory where a technician uses an AR system to be able to get information about a machine presented in his or her field of view just by looking at the machine, instead of having to interact with the machine via a conventional computer and screen.

## 1.2 Motivation

In 2016, Microsoft launched the first edition of the HoloLens wearable computer system [6]. It is a system that permits the creation of applications featuring advanced augmented reality, using well-known software development platforms and providing extensive documentation for developers. Due to the promising features of this system, it could be a suitable candidate for implementing software that meets the criteria of the scenario described above.

One question that can determine how useful augmented reality systems are in real-world use cases is how they function outside optimal conditions. Because of this, it is of interest how robust applications running on the HoloLens system are with regards to different environmental properties. A property which is likely to vary is lighting conditions in the surrounding environment. Lighting condition is a factor that is known to affect various aspects of performance and experience of augmented reality systems. Microsoft states in its documentation of HoloLens that some aspects of the performance of HoloLens can be affected by dark light conditions, but does not state exactly at what levels of light the problems may occur [7]. The displays used in HoloLens are based on a technology that may make it hard to see visual content in an environment with bright light [8].

## 1.3 Aim and research questions

The aim of the thesis is to answer the following questions:

1. How is the performance of a holographic application running on HoloLens affected by various light conditions?

2. Can different light conditions have a detrimental effect on user experience of HoloLens holographic applications?

A small prototype implementing some basic criteria of the scenario for future use was to be implemented. A set of properties that may be affected by various levels of ambient light was to be determined by the experiences gained when developing the prototype. Evaluation of these properties and the effects of various light conditions on user experience was to be evaluated using the implemented prototype.

## 1.4 Delimitations

The thesis does not compare the implemented solution to other augmented reality systems or to any non augmented reality solutions. The experiments are made under different levels of static light with as little dynamic change in light as possible. Dynamic changes in light are not part of the evaluation of the experiments.

# 2 Theory

This chapter presents general theory on the technical field of augmented reality, a description of the Microsoft HoloLens system and a short overview of some of the known effects of light on the performance of augmented reality systems.

## 2.1 Augmented reality

Augmented reality is the concept of creating an experience where virtual elements are added to the real world around the user. A commonly accepted definition of the term augmented reality is that an augmented reality system must implement the following three conditions [8], [9]:

1. It combines real and virtual elements
2. It is interactive in real time
3. It is registered in three dimensions

To fulfill the first condition the system typically has a display that can show both real and virtual elements. The second condition is fulfilled by the system having a computer that handles input and output in real time. The last condition is fulfilled by the system having tracking ability. The tracking ability is used to track the view and position of the AR system in the real world. This must be known for the system to be able to determine where and how to present virtual elements over the user's view of the real world [8]. Different technologies for displays and tracking are presented in later sections of this chapter.

A concept that is related to augmented reality is virtual reality. The difference between the two concepts is that in virtual reality the virtual environment completely replaces the real environment, whereas augmented reality presents a combination of real and virtual elements [8].

There are several fields where research is made exploring if and how augmented reality can be applied. Examples are for AR to be used in education [3], in experiencing art and culture [10], in the medical field [11], and in training and maintenance in industry [12].

As implied by the broad definition of an augmented reality system stated above, the implementation of AR systems can vary greatly. There are frameworks that permit the development of augmented reality applications that can be run on smartphones. There are several examples of head mounted devices specifically developed for AR and virtual reality. Augmented reality that can be experienced by a crowd simultaneously can be implemented by using external projectors that display virtual content by projecting it on real life objects.

### 2.1.1 Display types

A display used in an AR system must be able to show a combination of virtual and real elements. The technologies for achieving this can be divided into four categories: video-based, optical see-through, eye-multiplexed and projection on a physical surface [8].

For video-based displays and optical see-through the virtual content is superimposed on the view of the real environment, presenting a complete view where the virtual content is seemingly positioned in the real world. Video-based displays work as the display on a cell phone or on a TV. The real environment is filmed with one or several cameras, virtual content is added to the image of the real world and the combined image is presented as pixels on the display. Optical see-through differs from video-based displays in that only the virtual content has to be shown on the display. The real environment is seen through the transparent display just like looking at the world through a window or a pair of ordinary glasses.

Eye-multiplexed displays do not superimpose the virtual content on the view of the real world. The virtual content is presented such that the user must mentally add it the real environment. That virtual content can be presented in a smaller display that only covers part of a user's field of view.

Projection on a physical surface is where film projectors are used to project virtual content on real objects. An example of this can be to use several projectors to project different designs onto a physical model of a car.

### 2.1.2 Display configurations

There are several variations on where the display is positioned between the user's eyes and the world [8]. For a head-attached or head-mounted display (HMD) the display is positioned near the user's eyes similar to goggles or a pair of glasses. Hand-held or body-attached displays are held by the user, for example in the form of a smart phone or tablet.

A spatial display is more or less fixed in position, for example in the form of a semi-transparent mirror in front of the background in which the virtual content will be positioned. This configuration can be used to give many users access to the augmented reality view simultaneously.

As mentioned above, another form of spatial display configuration is where the virtual content is projected onto a real object, which means that the display is the real object itself.

### 2.1.3 Tracking technologies

To provide an experience where the user perceives that virtual objects exist in and interact with the real environment when seen through the AR system, the AR system must continuously determine where on the display the virtual content should be positioned.

The task of the tracking system is to determine the position and orientation, also known as the pose, of the AR system's view in relation to specific markers or natural properties [8], [13]. By tracking changes in the pose, the system can recalculate how to correctly display the virtual element as the user view moves. The tracking system must register how the user's view moves in three dimensions, left/right, closer to/farther away, and up/down. It must also register rotation in these three dimensions, i.e. the pitch, yaw and roll.

An example is a hologram of an apple that is intended to be shown in the AR view as if it is placed on a real table. As the user is moving towards the table with the apple, using a device with an AR display, the image of the virtual apple must be rendered increasingly larger in the AR view. If the users walks around the table watching the apple hologram, the image of the apple must rotate with the user's changing view.

### Vision-based tracking

Systems using vision based tracking use optical sensors for capturing data needed to determine the user's pose. Optical sensors can be divided into three categories, sensors that register infrared light, sensors that registers visible light, and 3D structure sensors [8].

Infrared sensors detect light in the infrared spectrum. The targets for the infrared sensors can be either passive or active. An active target emits its own infrared radiation. The passive target reflects back infrared radiation that is emitted by a source of infrared light, typically positioned near the camera.

Visible light sensors detect, as the name suggests, visible light. Two techniques using visible light sensors are fiducial tracking and natural feature tracking. Fiducial tracking is based on detecting fiducials, also known as markers, defined by Billinghurst et al. [8] as artificial landmarks that are added to the landscape to aid in registration and tracking. The fiducials can consist of a specific pattern printed on paper or formed by LEDs of different colors. Fiducial tracking is relatively simple and dependable since fiducials can be created to have a limited set of possible designs and to be easily discerned from naturally occurring features, which makes the image processing relatively simple. The downside is that it is not practical in all situations to add these markers to the environment. Natural feature tracking avoids the issue of having to add markers to the environment by using image processing to detect features of the existing environments to determine the position and orientation of the user's view. The downside is that the image processing is more complicated than for detection of fiducial markers.

3D-structure sensors can register information about three-dimensional objects by detecting depth information about the surrounding environment. Two techniques for detecting depth are structured light [14] and time-of-flight [15]. Structured light is used for example in the KinectFusion system [16]. The technique uses two cameras and at least one projector that projects a structured light pattern onto a physical scene. The two cameras then uses the structured light to find the pixel pairs between the two images, and that is used to calculate the depth map of that scene. Systems that uses time-of-flight sensors send out a light signal and detects how the returning signal has changed for a given set of points, which is then used to calculate the depth information of the scene.

### Inertial tracking

Inertial tracking is done through a combination of sensors such as accelerometers and gyroscopes, commonly known as an Inertial Measurement Unit (IMU) [8], [13]. These sensors can together determine the objects specific force and angular rate. Advantages of inertial tracking include not having range limitations, not being affected by competing optical, magnetic or other sources, as well as being able to operate where there is no line-of-sight. Disadvantages include accumulation of errors in position and orientation over time .

### GPS-tracking

This tracking technology uses the GPS satellite network to determine the position of the AR system. Since the technology uses satellites, the positioning is most reliable outdoors [8].

### Magnetic tracking

Magnetic tracking uses some source that emits a magnetic field and a sensor that can detect various properties of the detected magnetic field to determine the sensors position in relation to the transmitter. The tracking accuracy is affected by nearby objects emitting magnetic fields, such as electronic devices, and there is a strong limitation on the distance possible between the sensor and emitter [8].

#### Hybrid tracking

As the name suggests, systems that uses hybrid tracking combine sensors from different tracking techniques. This is typically a combination of vision-based tracking and for example inertial tracking or GPS tracking. It is a way of gaining the advantages of different tracking techniques while reducing the disadvantages, such as being able to correct the errors for the inertial tracker accumulated over time by using information from the vision based tracker [8].

## 2.2 Microsoft HoloLens

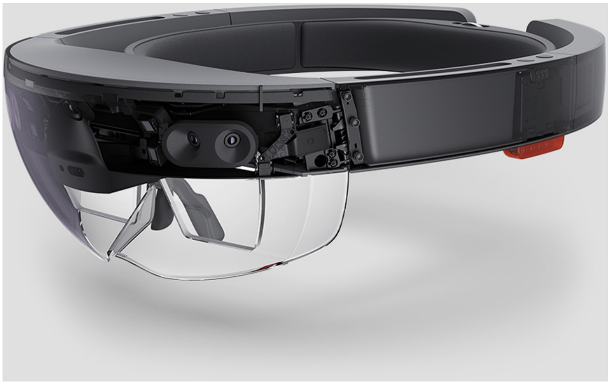


Figure 2.1: Microsoft HoloLens[17]

Microsoft HoloLens is in Microsoft's own words an untethered holographic computer [18]. HoloLens is a head-mounted augmented reality system, as shown in fig 2.1, that does not need an external computer, instead the computational power used is built into the device. It runs a version of Windows 10 as operating system. Applications developed for the Universal Windows Platform (UWP) can be run on the system, but cannot use holographic content. Microsoft provides an API for developing holographic applications in the game engine Unity [19].

### 2.2.1 Hardware

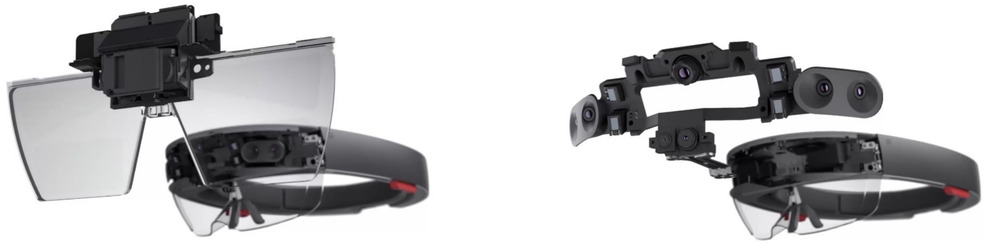
HoloLens has optical see-through lenses (see fig 2.2a). It has a number of sensors to obtain tracking information and information about the surrounding environment (see table 2.1 and fig 2.2b). What Microsoft calls environment understanding cameras are visible light cameras located to the right and left of the center camera, which are used to determine the fixed position of the user in space [20]. The depth camera emits infra-red light and uses time-of-flight sensors to gather depth information for the environment [21].

The device has built-in speakers and an audio jack for headphones. The only buttons on the device are the power button, and three buttons used to, among other things, adjust brightness and volume and take screenshots. It can communicate through Wifi, Bluetooth or MicroUSB cable.

It is possible to use Bluetooth keyboards, gamepads and some other devices that uses Bluetooth. There is also a specially developed hand-held device called a clicker that can be used for scrolling and clicking in HoloLens applications [22].

| Property | Definition |
|---|---|
| Optics | See-through holographic lenses |
| Sensors | 1 IMU<br>4 environment understanding cameras<br>1 depth camera<br>1 2MP photo/HD video camera<br>4 microphones<br>1 ambient light sensor |
| Processors | Intel 32 bit processor<br>Custom-built Microsoft Holographic processor |
| Memory | 64GB Flash<br>2GB RAM |
| Weight | 579 g |
| Battery life | 2-3 hours of active use<br>Up to 2 weeks standby |
| Connectivity | Wifi 802.11ac<br>Bluetooth 4.1 LE<br>MicroUSB 2.0 |

Table 2.1: Hardware specifications of Microsoft HoloLens[18].



(a) Displays

(b) Sensors

Figure 2.2: HoloLens hardware details. a) The see-through displays [23]. b) The positions of the sensors [24]. The depth camera is located at the center, above the main camera. The two environment understanding cameras are placed one each at the left and right of the center camera.

### 2.2.2 Optics

HoloLens uses a variant of optical see-through display technology called waveguides with surface relief gratings (see fig 2.2a) [21]. This is different from some see-through optics technologies where an image of the virtual content is projected onto a semi-transparent mirror and reflected into the user's eyes while the background is also visible. Instead, the image is transferred into the waveguides and transmitted through them by total internal reflection, where the grates bend the transmitted light so that it exits the waveguides towards the users eyes at the intended position [25]–[27].

The maximum supported resolution for HoloLens applications is 720p (1268x720 pixels) [28]. The technical documentation suggests that applications show graphics at 60 frames per second for an optimal user experience [29]. Microsoft provides values for two optical properties they have named the holographic resolution and holographic density, which are 2.3 million total light points and 2.5 thousand light points per radian, respectively. An explanation of the relations between common resolution, holographic resolution, and holographic density could not be found in the official documentation. Other sources explained that common resolution and holographic resolution and density are different measurements that both relate to the quality of the holograms [30]. Higher holographic density is said to lead to brighter and sharper holograms [31]. The exact reason why conventional resolution is not a good measurement could not be found by the thesis author.

### 2.2.3 Spatial mapping

Microsoft HoloLens has the capability to create a 3D-model of the surrounding space, a process Microsoft calls spatial mapping [32]. A visualization of the spatial map with the polygons highlighted can be seen in fig 2.3. The spatial map is used to guide where to place virtual elements so that they seem to interact with the real environment, e.g. a virtual apple placed on a table or a virtual ball bouncing against a wall. This also permits the virtual content to be occluded by real objects, e.g. virtual content can be hidden behind real objects. The spatial map can be saved for later use on the same HoloLens device or shared with others. When using the spatial mapping functionality in a 3D HoloLens app, the map is updated continuously as more sensor data is received to create a more detailed and updated model of the surrounding environment.

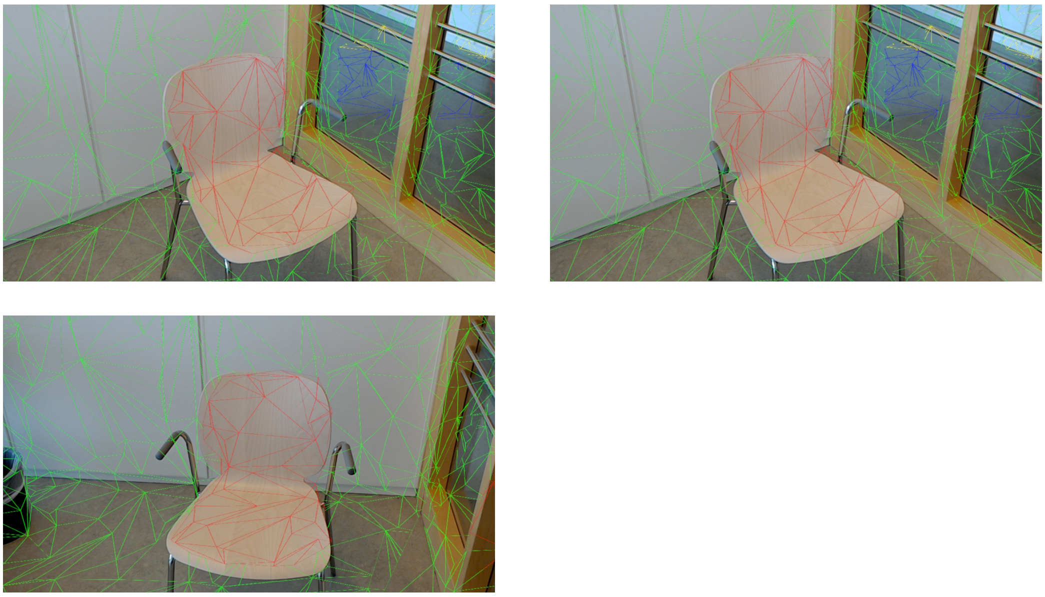


Figure 2.3: Screenshots of visualization of the spatial map created by HoloLens. The application used for the spatial map experiments described in section 3.5.2 were used when taking these pictures. Different distances are indicated by different colors in the mesh.

### 2.2.4 Spatial persistence

Virtual objects in a HoloLens app are able to retain their position in space between application sessions [32]. A hologram that is positioned on a chair will have that position the next time the application is launched. The Holographic API provides the ability to create a spatial anchor, a point in space that virtual content can be positioned in relation to. Spatial anchors can be saved to the HoloLens secondary memory and retrieved the next time the app is started. The caveat is that the system must be able to recognize the surroundings, which mean that there are limits to how much the environment can change for this to work. No documentation on the exact method used by the HoloLens to determine the position of a spatial anchor from the available sensor data could be found.

### 2.2.5 Methods for input and output

The modes used for input and output on Microsoft HoloLens differ in some aspects when compared to a laptop, a mobile phone or a tablet.

#### Gaze tracking

A form of input that is used to target objects in HoloLens is gaze tracking [33] (not to be confused with the augmented reality term tracking explained earlier in this chapter). This corresponds to using a computer mouse to be able to target different objects in a computer system with a graphical user interface. The gaze determines where the user is looking at the world by registering the position and orientation of the user's head. It is the position of the head and not the direction of the eyes that determines where the gaze is aimed, which means that to move the gaze the user has to move his or her head.

#### Gesture input

HoloLens has built in support to recognize a number of hand gestures that can be used for user input [34]. HoloLens can determine the position and some characteristics of the right and the left hand of the user while it is positioned in an area in front of the HoloLens. These basic gestures that HoloLens can recognize permit the user to, among other things, click, hold and scroll in a 2D application, and move or activate some part of a hologram. Combined with the gaze tracking, gestures can be used to interact with holograms and 2D-applications in ways that in some aspects are consistent with how a user would interact with conventional device, such as a tablet or mobile with touchscreen. Developers can also use the ability to detect the position and characteristics of the hand and the basic gestures to combine into custom modes of interaction with an app.

#### Voice input

Another form of user input is by using voice commands [35]. Detection of some standard phrases are implemented in the system, such as "select", which can be used instead of air-tapping or using the clicker. It is also possible to make selections by speaking the text displayed on buttons and menus. Many of the features of the Windows 10 digital assistant Cortana are also available for HoloLens [36].

There is extensive support for using custom voice commands when developing applications for HoloLens. Creating custom voice commands can be implemented as simply as by providing a string of the new command to an element of the HoloLens API that recognizes keywords [37]. Only English pronunciation of voice commands is supported so far.

#### Spatial sound

In addition to visual output, Hololens features the ability to play spatial sound from holographic content. Spatial sound is the mechanism of playing sound in a way that simulates the sound coming from different locations in space [38].

### 2.2.6 Operating system and holographic shell

Microsoft HoloLens runs Windows 10 as operating system, with almost all functionality available as on laptops and desktops running Windows 10. The holographic shell is the name for what can be seen as the HoloLens alternative to the desktop when running Windows 10 on a conventional computer, i.e. the starting point from which applications can be started and where application windows can be placed. Apps based on the Universal Windows Platform (UWP) [39] that are executed on HoloLens are placed as 2D windows along physical walls or objects in the holographic shell. When starting a third party 3D application, the surrounding virtual space is taken over by the content of that application and the content placed in the holographic shell is not visible.

### 2.2.7 Developing software for HoloLens

Developing software can be divided into two separate fields, developing 2D-applications without holographic content and developing 3D-applications with holographic content [40]. 2D-applications are developed on the Universal Windows Platform (UWP) that permits the application to be run on Windows mobile phones, Windows tablets and laptops and desktops that run Windows 10. The development can be done in many different languages, e.g. C#, C++, or javascript. Since this thesis concerns holographic applications, this subject of developing 2D applications will not be further expanded on.

#### Holographic development

Microsoft has developed their own Holographic API that third party developers are free to use. Development of applications using the Holographic API can be done with the Unity game engine [41] or by using DirectX directly. Unity is a development tool for developing 2D- or 3D-games and other applications with 2D and 3D graphical content.

To simplify the process of learning to develop for HoloLens with the Holographic API, Microsoft has published several tutorials explaining different crucial concepts to guide beginner developers on various aspects of development of holographic applications [42]. Microsoft has also released the open source projects HoloToolkit [43] and HoloToolkit Unity [44] that are collections of scripts, prefabricated Unity components and other resources to help with development of HoloLens applications.

## 2.3 Issues of augmented reality and different light conditions

There are various known ways that different light conditions may affect the augmented reality experience.

The modelling of the 3D space by HoloLens is accomplished by a combination of Time-of-flight IR cameras and visible light cameras. Time-of-flight IR cameras must handle the noise from ambient light containing light of the same wavelengths as that being used by the system [15].

Microsoft states that HoloLens may encounter problems with tracking in dark environments, but does not go into any detail on the light levels where those problems may arise [7].

The display technology used in the AR system will also affect how the user experiences the virtual content in different light conditions [8]. One benefit that video-based augmented reality displays have over optical see-through displays is that image processing can be performed on the captured video of the real environment. This means that there are more options of handling issues of inconsistent lighting between the real world and virtual elements when using video-based displays than when using optical see-through. Optical see-through displays are limited by the fact that they can not manipulate the image of the real world, and the maximum light intensity of the virtual content must compete with the light in the surrounding environment. Color saturation is diminished in the presence of too much light in the surrounding real environment [25]. Microsoft recommends that applications are tested in the environments in which they will be used to ensure that visibility and color are experienced as intended [45].

## 2.4 Measuring light conditions

When looking at use of an AR system outside of an office environment in various industrial environments, light conditions can be expected to vary considerably. Ideally, an AR-system for industrial use should work as intended in light conditions ranging from dim lighting or darkness, to bright sunshine.

The metric that was used to quantify light conditions is illuminance. Illuminance is a measurement of how much light is falling onto a surface. The light can come from either artificial or natural sources, such as overhead light fixtures and daylight through a window. The SI derived units - e.g. derived from units of measurement defined by the International System of Units [46] - to measure illuminance is lux or lumens per square metre [47].

# 3 Implementation of software

This chapter will present the software created for this thesis. A larger prototype application was developed as a simple implementation of the proposed scenario, to help determine how the experiments would be designed and to then be used in some of those experiments. Two small additional applications were also developed, since separating functionality into several applications instead of integrating it all into the larger prototype would simplify the design of the experiments.

## 3.1 Prototype specification

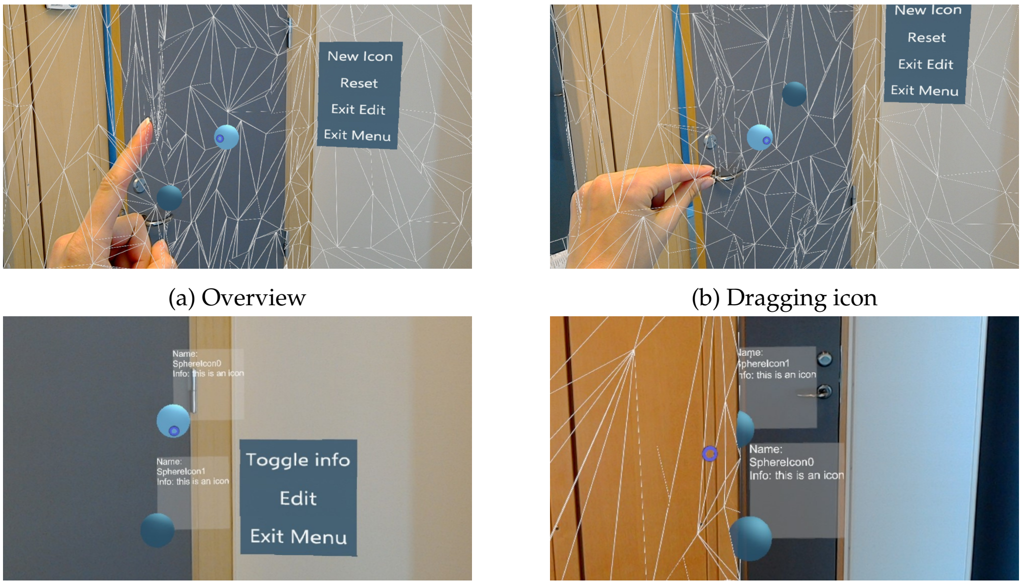
The aim of the prototype was to implement a simple demo that showcases some fundamental concepts of developing holographic applications for HoloLens and as far as possible implement the proposed scenario. The nature of what the prototype should implement was decided through discussion of the scenario described in the introduction with the external supervisor. Through those discussions, a small set of characteristics that the prototype should fulfill was determined:

- The application is a 3D augmented reality application
- It must be possible to display information particular to an object or location in the real environment
- The information must be reachable through augmented reality interaction, such as looking at an object
- The point in space that relates to a specific set of information must be persistent, i.e. the application must remember between executions the location in space that is connected to reaching information about a specific object

## 3.2 Development platform and frameworks

The prototype was developed using the game engine Unity version 5.5.2p1 Personal and Visual Studio Community 2015 Version 14.0.25431.01, with scripts written in C#. The choice was mainly made due to this setup being recommended by Microsoft. Unity has a user friendly graphical user interface and extensive support through official and unofficial channels, which makes it easier as a beginner to get started with developing holographic applications than any other alternatives.

Microsoft provided HoloToolKit and HoloToolKit-Unity [44] (later renamed MixedRealityToolkit [43] and MixedRealityToolkit-Unity), two open source projects that contain resources that can be used by HoloLens developers. HoloToolKit-Unity was of main interest since the prototype was developed using Unity. The resources it contains are scripts, shaders, materials and prefabricated Unity components as well as complete examples of various functionalities. The resources were used for the development of the prototype in various ways, in some cases the resources could be used as they were, in other cases used in modified form or used to learn how the system works. Several tutorials provided by Microsoft were also used as learning tools, and some components from the code examples were used in modified or unmodified form [42].



(a) Overview

(b) Dragging icon

(c) Information flags

(d) Icons occluded

Figure 3.1: Screenshots from the implemented prototype.

## 3.3 Design choices

There were several choices to be made: how would the user interact with the application to retrieve the desired information, and how should the information be presented to the user?

How the user would choose what information they are interested in can roughly be divided into two modes: either there is a virtual object that exists to provide a target for interaction, or the information is retrieved by looking at the real object of interest without an obvious virtual object for interaction.

The former alternative was chosen, since it was simpler and would provide the virtual content suitable for use in the light condition experiments. It was decided that a simple three-dimensional sphere would function as that interaction object (see fig 3.1). From here on, such spheres are named icons.

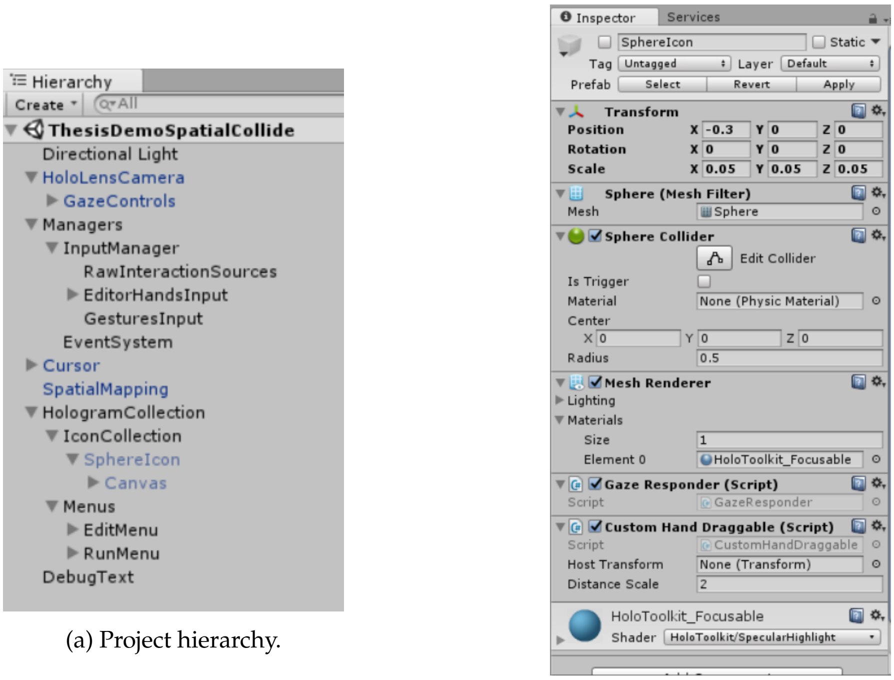
The other choice was how to present the information. Either the information could be presented next to the real object or the icon, or it could be presented as headlocked information that follows the users view. The first alternative was chosen since Microsoft design guidelines recommends not using headlocked content for HoloLens applications, as well as being more suitable for use in the upcoming experiments.

The application was designed to have two modes of operation, an editing mode where icons can be created and placed at the desired position in space, and a mode where the user retrieves information from the icons by looking at them.

An interaction is made up of two parts, how to focus on the object to interact with, and then choosing how to interact with it. Gaze, i.e. a marker directed by the user's head movement, was chosen to direct focus. It is the choice recommended by Microsoft. To move an icon to its desired position in space, the choice was made to use the gaze to focus on the icon, and then move it by making a holding gesture (see fig 3.1b). An alternative method for moving an icon was also available, where after having selected the icon to be moved by using an air tap hand gesture the icon's position would be directed by where the user's gaze hits the spatial map. After a brief test that alternative was discarded, since icons would not necessarily be positioned near a real object and since it made it hard to make small and accurate adjustments in position.

## 3.4 Implementation

A Unity project generally consists of a set of basic components, a light source, a camera, managers that handle events, and 2D or 3D graphical elements that will be rendered in the application view. All components can have scripts and various other elements connected to them that decide their behavior, such as scripts handling what happens when an object comes into focus or is selected. Fig 3.2 shows the hierarchy of project components for the prototype and an example of the various scripts and other elements that can be connected to a component, in this case to the component SphereIcon. As the figure shows, this project consists of many components with connected elements, too many to cover in this report. Only the main components that are specific to development of HoloLens applications and to this prototype will be covered. Documentation on development of general Unity applications is available on the official Unity website [41].



```
▼ ThesisDemoSpatialCollide
    Directional Light
  ▼ HoloLensCamera
    ▶ GazeControls
  ▼ Managers
    ▼ InputManager
        RawInteractionSources
      ▶ EditorHandsInput
        GesturesInput
      EventSystem
  ▶ Cursor
    SpatialMapping
  ▼ HologramCollection
    ▼ IconCollection
      ▼ SphereIcon
        ▶ Canvas
    ▼ Menus
      ▶ EditMenu
      ▶ RunMenu
    DebugText
```

(a) Project hierarchy.

```
□ SphereIcon                     □ Static
Tag  Untagged        Layer  Default
Prefab  Select  Revert  Apply

▼ Transform
Position     X -0.3   Y 0      Z 0
Rotation     X 0      Y 0      Z 0
Scale        X 0.05   Y 0.05   Z 0.05

▼ Sphere (Mesh Filter)
Mesh             Sphere

▼ ☑ Sphere Collider
                 Edit Collider
Is Trigger       □
Material         None (Physic Material)
Center
  X 0            Y 0            Z 0
Radius           0.5

▼ ☑ Mesh Renderer
▶ Lighting
▼ Materials
    Size         1
    Element 0    HoloToolkit_Focusable

▼ ☑ Gaze Responder (Script)
Script           GazeResponder

▼ ☑ Custom Hand Draggable (Script)
Script           CustomHandDraggable
Host Transform   None (Transform)
Distance Scale   2

HoloToolkit_Focusable
Shader  HoloToolkit/SpecularHighlight
```

(b) Elements connected to component SphereIcon.

Figure 3.2: Hierarchy of project components (a) and example of scripts and other elements connected to a component (b).

### 3.4.1 Interaction

The GazeManager script avaliable in the HoloToolKit [44] manages all that is related to the gaze, such as determining where it hits and which virtual object the gaze is aimed at. It is a standard component in many HoloLens applications and could be used without modification.

The ability to drag an object by performing a holding gesture is handled by a script based on the HandDraggable script available in the HoloToolkit. Some alterations and additions were made to handle how the object behaves when dragged. Handling of spatial anchors and interaction of the dragged objects with the mesh created by spatial mapping of the surrounding environment were also implemented.

The ability to use voice commands can be implemented by using standard components from the HoloToolkit, and was added to the prototype to facilitate application interaction during experiments.

### 3.4.2 Spatial persistence

When the icons are positioned in space they must retain that position even between application sessions and when the HoloLens has been powered off and turned on again, regardless of the time that has passed between these events. For HoloLens, this is achieved by attaching world anchors to some or all virtual objects, defining the objects' spatial coordinates in relation to those world anchors and storing them in the world anchor store. This was done by implementing a world anchor store that stores the world anchors locally on the device with simple strings as identifiers. World anchors can then be retrieved and used to give virtual objects the position in space which they had when the application was closed.

The WorldAnchorManager script available in the HoloToolkit was used largely without modification. Examples of usage of the world anchors was found in various examples in the Holographic Academy code tutorials and on developer forums online.

### 3.4.3 Spatial mapping

Mapping of the physical objects in the surround environment was needed since the icons should not be possible to place inside of physical objects. This functionality was avaliable in the HoloToolkit as scripts. Rudimentary collision handling was implemented using raycasting against the retrieved spatial map. Raycasting is the act of shooting, or casting, an invisible ray from a point of origin in a determined direction to detect if that ray intersects with another virtual object. In this case this was used to determine if collision with the spatial map had occurred in the direction of movement when dragging an object. The use of raycasting in collision detection is a tried technique in Unity development, and examples of use were found in both Unity documentation and in the Holographic Academy tutorials [48].

Occlusion is the concept of rendering virtual objects so that they are seemingly hidden behind physical objects, as can be seen in fig 3.1d. This is optional, but was chosen to further encourage the illusion that the virtual icons were real objects. This was implemented using components from the HoloToolkit, with guidance from various examples in the Academy tutorials.

### 3.4.4 Information presentation

The information was decided to be presented as text in panes that pop up adjacent to the icons (see figs 3.1c, 3.1d). This was decided mainly because of its relative simplicity to implement and its suitability for the upcoming light condition experiments. This functionality could be created after having consulted the Unity documentation.

## 3.5 Additional applications

Two small applications were created to facilitate performing experiments, instead of extending the prototype further.

### 3.5.1 Spatial map viewer

This small holographic application was created in Unity with the only purpose of showing the spatial map created by HoloLens.

The application was based on the example from a tutorial on holographic programming issued by Microsoft [49]. The only components used in the Unity project were a main camera, a directional light and a spatial mapping asset. The spatial mapping asset creates a spatial map, and visualizes it by highlighting the spatial map mesh with colors that depend on the distance from the camera to the created mesh (see fig 2.3).

### 3.5.2 Visibility check

The application written for the visibility experiment was written in C# as a Universal Windows Platform app. It consists of an application window with a word written in different color combinations, as can be seen in fig 3.3. The word is randomly selected from a list of 27 words of approximately the same length. By pressing a button a new word is displayed, until five words have been displayed and the test is over.

The colors were chosen to give many possible combinations of colors that were either common in software color schemes, such as white and dark backgrounds with white or dark text, and high visibility colors such as red and blue. The light gray was used since it is recommended by Microsoft to be used instead of white, as white can be seen as too bright or intense [45]. The white was chosen to test the maximum brightness. The dark slate gray was chosen as a dark background, since solid black will be transparent in HoloLens applications [45]. The more unusual color combinations, such as red background and blue text, was chosen to also test if color that conventionally wouldn't be used in software used on non see-through displays would be useful for HoloLens see-through display.

| Color name | ARGB value |
|---|---|
| White | #FFFFFFFF |
| Light gray | #FFEBEBEB |
| Dark slate gray | #FF2F4F4F |
| Red | #FFFF0000 |
| Blue | #FF0000FF |

Table 3.1: Colors used in visibility application, with their corresponding ARGB hexadecimal values.

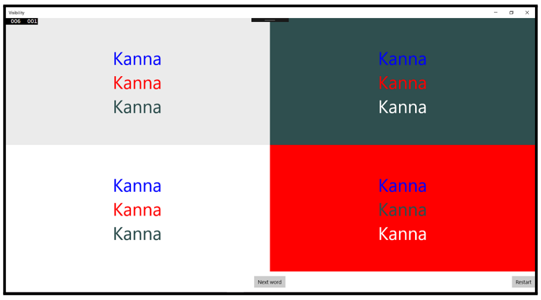


Figure 3.3: Screenshot of application written for the visibility experiments.

# 4 Experimental work

This chapter presents the different experiments made, using the prototype application and the two additional applications that were developed for this thesis.

## 4.1 Properties to investigate

The following aspects of the functionality of the Microsoft HoloLens system that could be affected by different light conditions were determined during the development of the prototype:

- the ability to create a spatial map that is as accurate as that created in favorable conditions
- the system recognizing gestures
- holograms maintaining spatial persistence
- holograms staying stationary
- the visibility of virtual content

## 4.2 Experiments

It was decided that the experiments would be designed to investigate the differences in application behavior in environments with different static levels of light, after the system had stabilized to the current light level. Such experiments would be relatively easy to design and evaluate, and that would be a natural starting point to investigation in this topic. A follow-up on that could have been to investigate how the system behave during dynamic variations of the light level, for example how the system would behave when going from indoor lighting to daylight outside, if time would have permitted.

### 4.2.1 Spatial mapping

No suitable methods for evaluating the quality of the spatial map using quantitative measurements could be found. Therefore, an experiment was designed where the spatial map of a set of physical objects created under different static levels of ambient light would be observed and qualitatively compared to the spatial map created of the same set of objects during reference lighting conditions. The spatial mapping application which only reveals the created spatial mesh was used for this experiment.

How well the spatial map correlated to the physical objects was observed first in light conditions that corresponds to normal office lighting conditions, with light coming both from artificial light sources as well as daylight through windows. This reference was chosen to be the normal light conditions during daytime in the offices where the author was working, which was within the recommended limits for lighting of an office as stated by Arbetsmiljöverket [50]. "Normal" in this context would perhaps better be seen as "not extreme", as what is normal office lighting is somewhat subjective and could vary with season and time of day. The light levels were then regulated in steps to lower and higher levels. The experiment was also performed outdoors in natural daylight.

Two cardboard boxes with the dimensions 30 x 30 x 10 cm and 32 x 10 x 10 cm were placed on the floor against a wall in a room with no windows or on the ground on a patio outside. The height of the HoloLens was 110 cm above the floor and 150 cm from the wall where the boxes where placed. The application was started, the user (the author) took one step to the right, one step to the left while still looking at the boxes, and then waited 10 s before the quality of the spatial map was noted. One of the boxes was then moved approximately 10 cm towards the other box before repeating the above procedure before again noting the shape of the resulting spatial map. The position of the boxes for both configurations was marked on a large piece of paper to be able to replicate the pattern for all the measured light levels. The reason for moving the box was to force the spatial map to change according to the change in the real environment, to make sure that the system did not depend on old data in the case that the light conditions were too poor to register new data. It was noted if the spatial map was noticeably less accurate compared to the spatial map created under normal office lighting conditions, or if it could not be at all created by the system.

The light level of the room was controlled using artificial light sources from the ceiling lamps or a dimmable floor lamp for the experiments performed indoors, and for the experiments outdoors during daytime the natural light from the sun was registered. The illuminance was measured at the cardboard box using the Physics Toolbox Light Meter luxmeter app [51] running on a Sony Z5 Compact smartphone [52]. Two measurements of the illuminance was made for each light level, one at the corner of one of the boxes with the sensor aimed in the direction of the incoming light and one 0.85 cm above the floor at 1 m from the back edge of the boxes. The latter measurement was an attempt at following the method for measuring the ambient light level in an office as defined by Arbetsmiljöverket, which defines it as the mean value of the light measured at 0.85 m above the floor in a room [50].

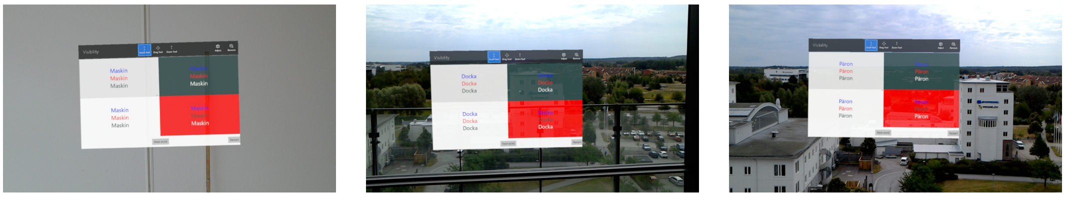
### 4.2.2 Behaviour of holograms and hand gestures

To evaluate the behaviour of holograms and recognition of hand gestures a similar setup as above was used when running the prototype application.

The application was started and two icons were created and placed at positions close to a wall, at approximately 130 cm height from the floor. The positions of the icons were outlined with a marker on a large piece of paper fixed to the wall behind the holograms to be able to recreate the positions. One of the icons was moved to a new position using a dragging hand gesture. The new position was also outlined on the paper on the wall.

What was noted at each light level was if the holograms remained positioned correctly and if they were stationary without trembling or moving. At each light level, the application was closed and reopened. Any deviation from expected behaviour was noted when dragging the icon to the second position using the holding hand gesture. Any deviation from expected behaviour in the recognition and use of hand gestures was also noted. The icons were observed from a height of 160 cm above the floor and 150 cm from the wall.

### 4.2.3 Visibility of virtual content



(a) Setup A (b) Setup B (c) Setup C

Figure 4.1: Views of the different setups of the visibility experiments.

The visibility of virtual content was evaluated by performing a small user survey with three participants. The virtual application window, showing five randomly selected words shown in different color combinations, was positioned 1.5 m from the participant with the upper edge of the application window at 1.6 m height. The size of the virtual text corresponded to 2 cm in height for an uppercase letter "C" where the virtual text was positioned.

The participant was asked to read the word. It was noted if it was correct or not by live streaming the HoloLens camera to a browser where the visibility of holographic content is better than when wearing the HoloLens. The participants were asked which color combination/combinations they thought were most legible, which combination/combinations they thought had poor or bad legibility, and if they had other comments on the experience in terms of visibility and legibility.

This procedure was performed with three different light condition setups for all three participants (see fig 4.1). Setup A: in an office during daytime, with the application window placed against a white wall. Setup B: in an office with the application window placed against a window overlooking forest, buildings and sky. Setup C: outside on the balcony of the building, with the application placed in front of the participant, with buildings and sky as background.

# 5 Results

This chapter presents the results of the performed experiments.

## 5.1 Illuminance measurements

The results from the illuminance measurements performed using the Physics Toolbox Light Meter luxmeter app are presented as intervals, due to fluctuations of the resulting values when performing the measurements. The biggest cause seemed to be small changes in tilt of the smartphone in relation to the direction of the light. Outdoors the values fluctuated considerably more due to changing light levels caused by clouds passing the sun on a windy day. The values for the outdoors measurements also fluctuated more due to the fact that the light levels were higher and thus small changes in tilt resulted in large differences in measured light. Indoors the values where more consistent but lower. Because of these differences, the intervals are of different size for the different light levels.

## 5.2 Spatial mapping

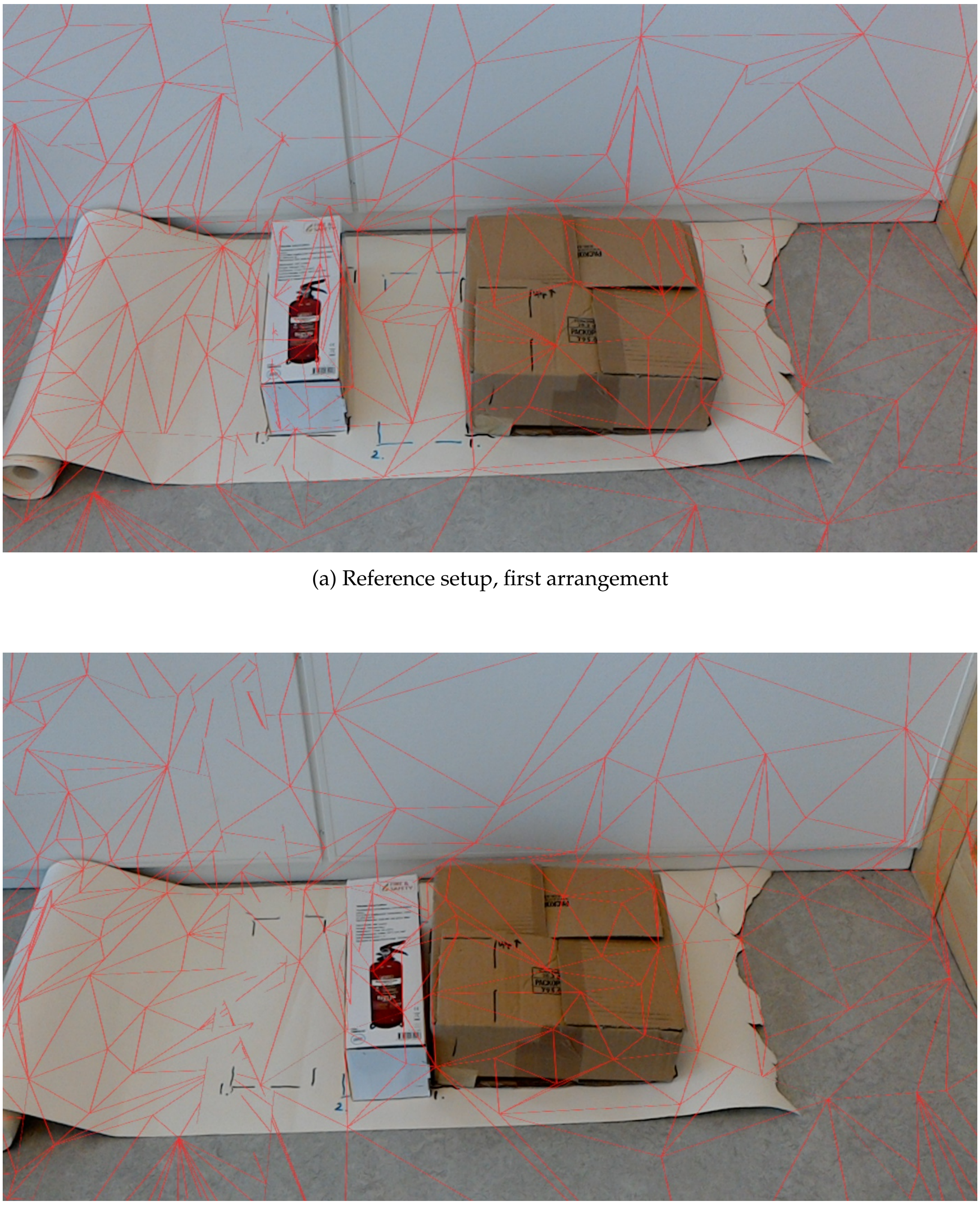
The results from the experiments that determine the quality of the created spatial map are presented in table 5.1 and fig 5.1. Since no spatial map was created by the system at the lowest measured light level, no images are presented for that measurement.

The spatial map was redrawn about once a second. The spatial map in the images presented in fig 5.1 did not differ to any large extent from how the spatial map fluctuated at the time the picture was taken.

The highest value for illuminance that was detected in the experiments was 30’000 lux. This seems to be the limit for the sensor in the camera.

| Setup name | Max light near boxes(lx) | Ambient light(lx) | Spatial map quality |
|---|---|---|---|
| *Ref* | *570-590** | *580-600** | *Point of reference** |
| A | 30000 | 14500-16200 | No significant change |
| B | 22200-24100 | 19300-21200 | No significant change |
| C | 34-40 | 72-85 | No significant change |
| D | 5-10 | 16-27 | No significant change |
| E | 2-3 | 1-3 | No significant change |
| F | 0 | 1-2 | No significant change |
| G | 0 | 0-1 | Could not be created |

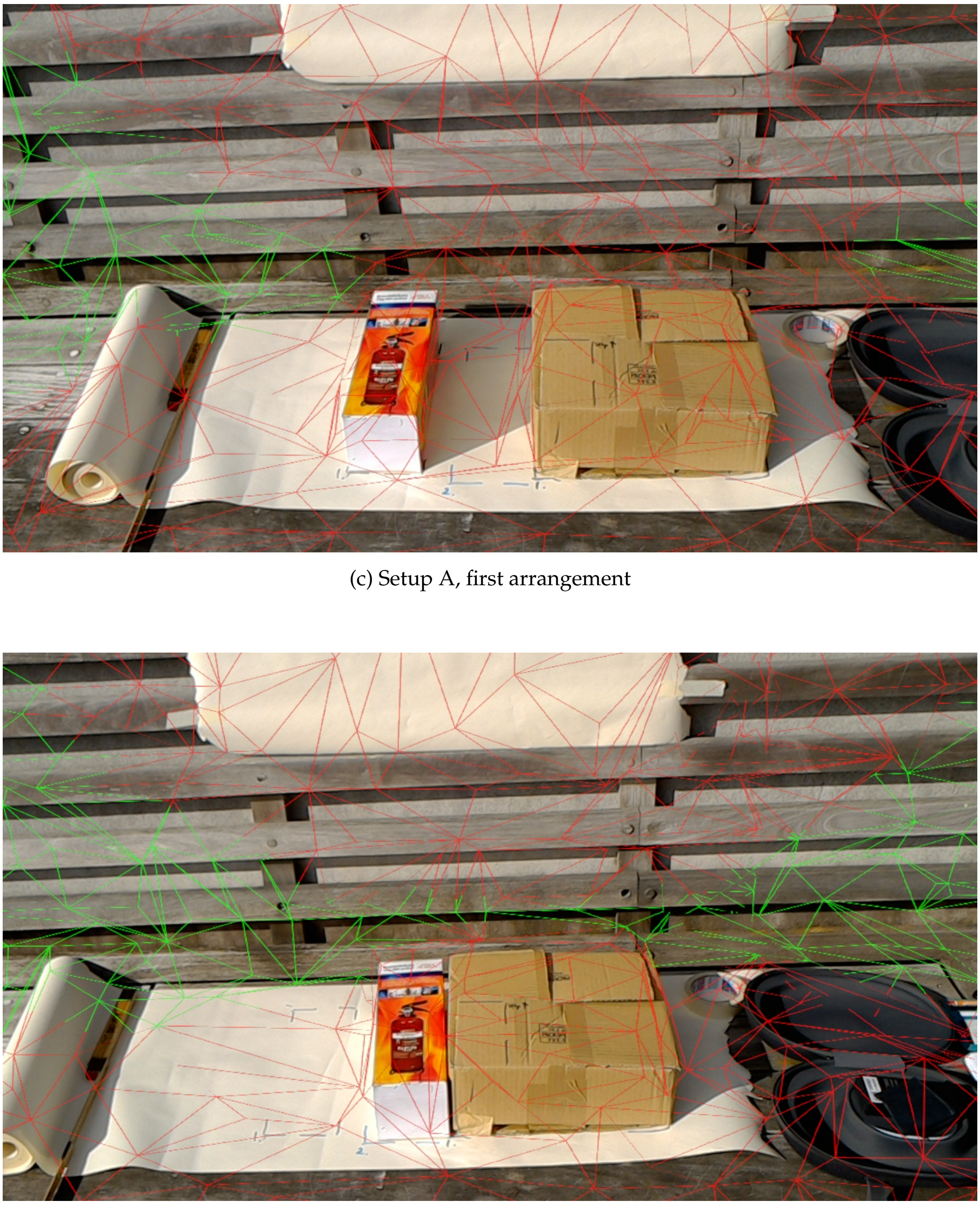
Table 5.1: Results of experiments observing the created spatial map during different light conditions. Max light near boxes: the value of the illuminance measured at one of the boxes, with the sensor facing the direction of the incoming light. Ambient light: the light at 0.85 m from the floor, 1 m in front of the boxes, with the sensor facing straight up. *) indicates normal office light conditions, used as a point of reference.



(a) Reference setup, first arrangement

(b) Reference setup, second arrangement

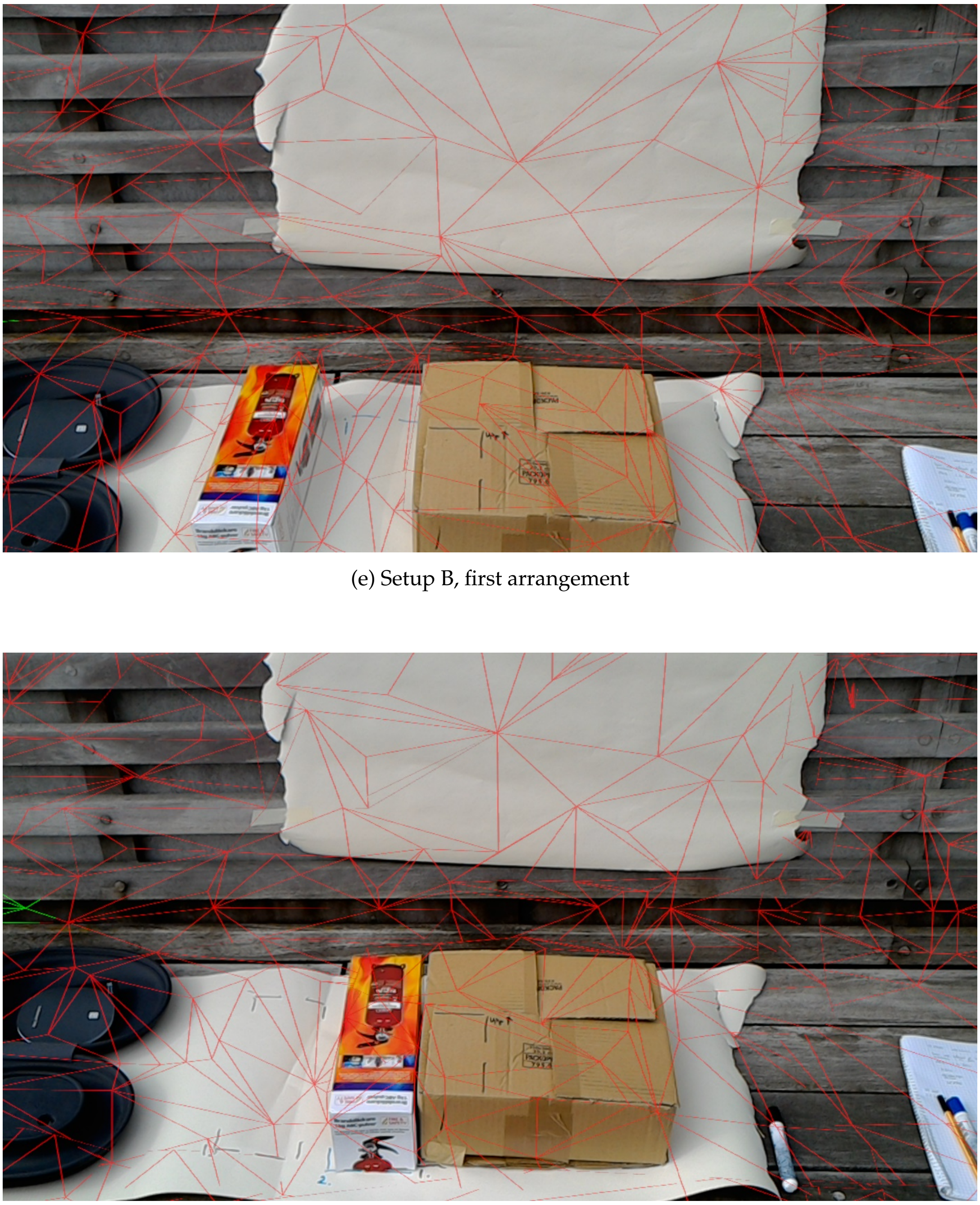
Figure 5.1: Images from experiments observing the created spatial map during different light conditions. See table 5.1 for the measured light levels for each setup.



(c) Setup A, first arrangement

(d) Setup A, second arrangement

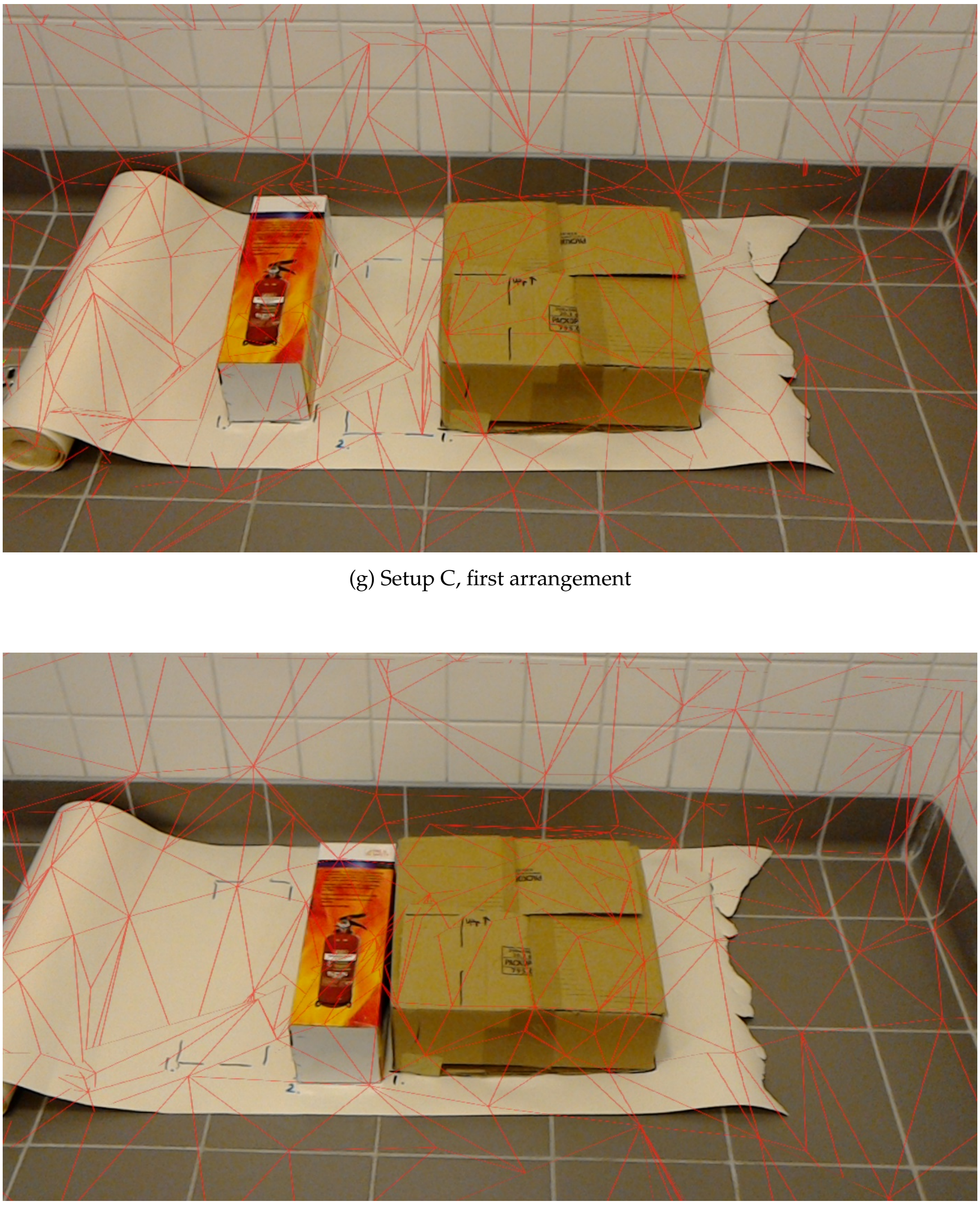
Figure 5.1: Images from experiments observing the created spatial map during different light conditions, continued.



(e) Setup B, first arrangement

(f) Setup B, second arrangement

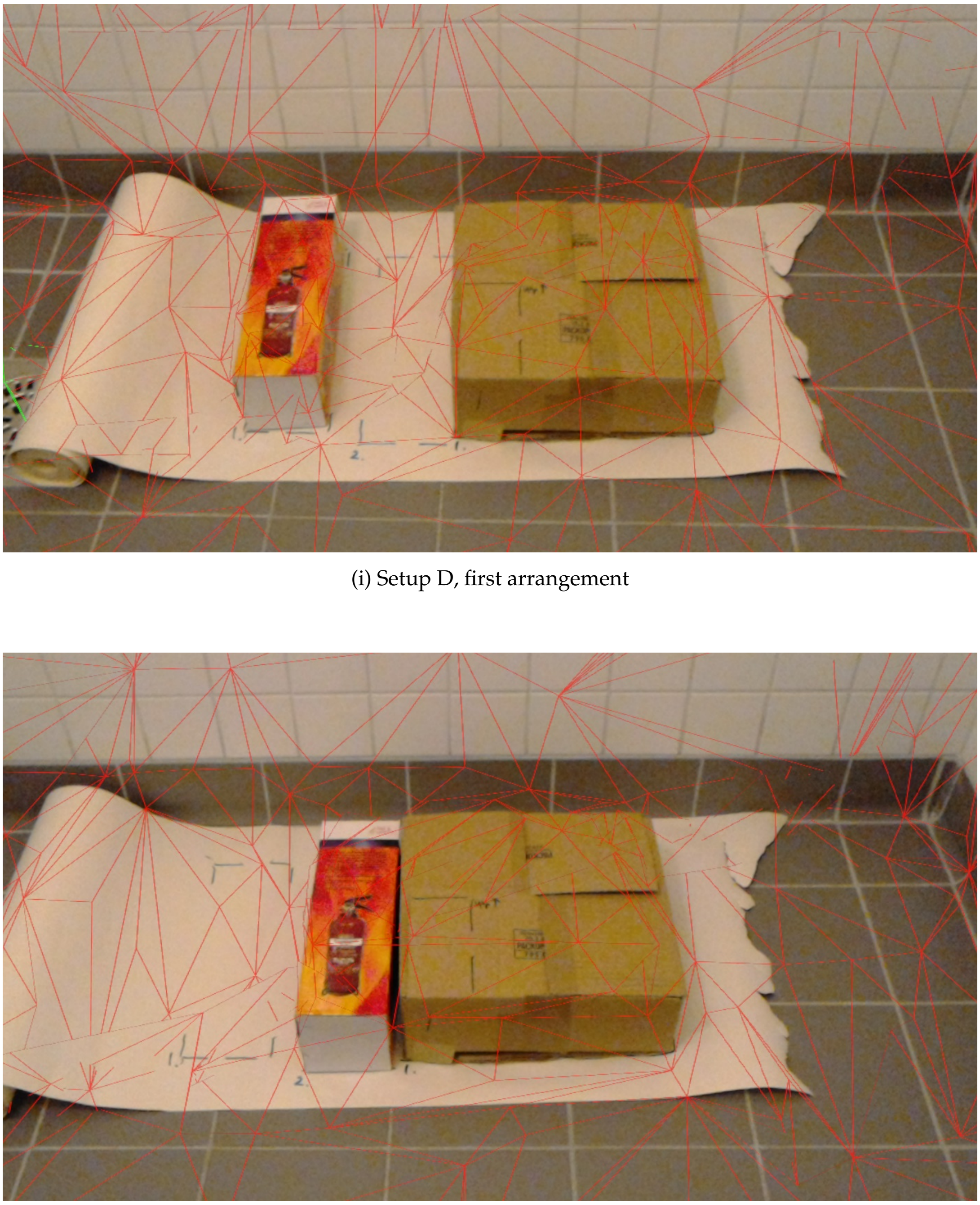
Figure 5.1: Images from experiments observing the created spatial map during different light conditions, continued.



(g) Setup C, first arrangement

(h) Setup C, second arrangement

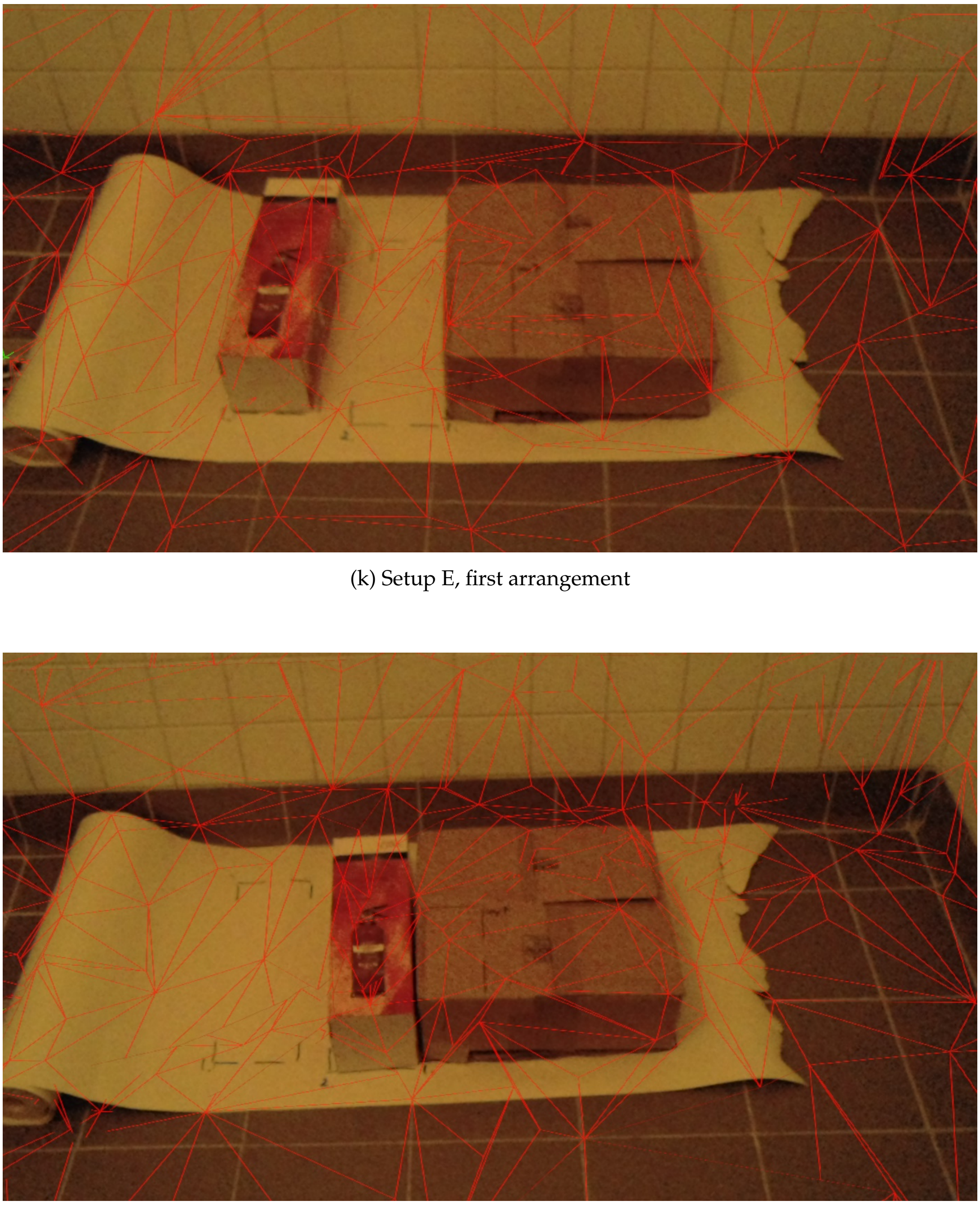
Figure 5.1: Images from experiments observing the created spatial map during different light conditions, continued.



(i) Setup D, first arrangement

(j) Setup D, second arrangement

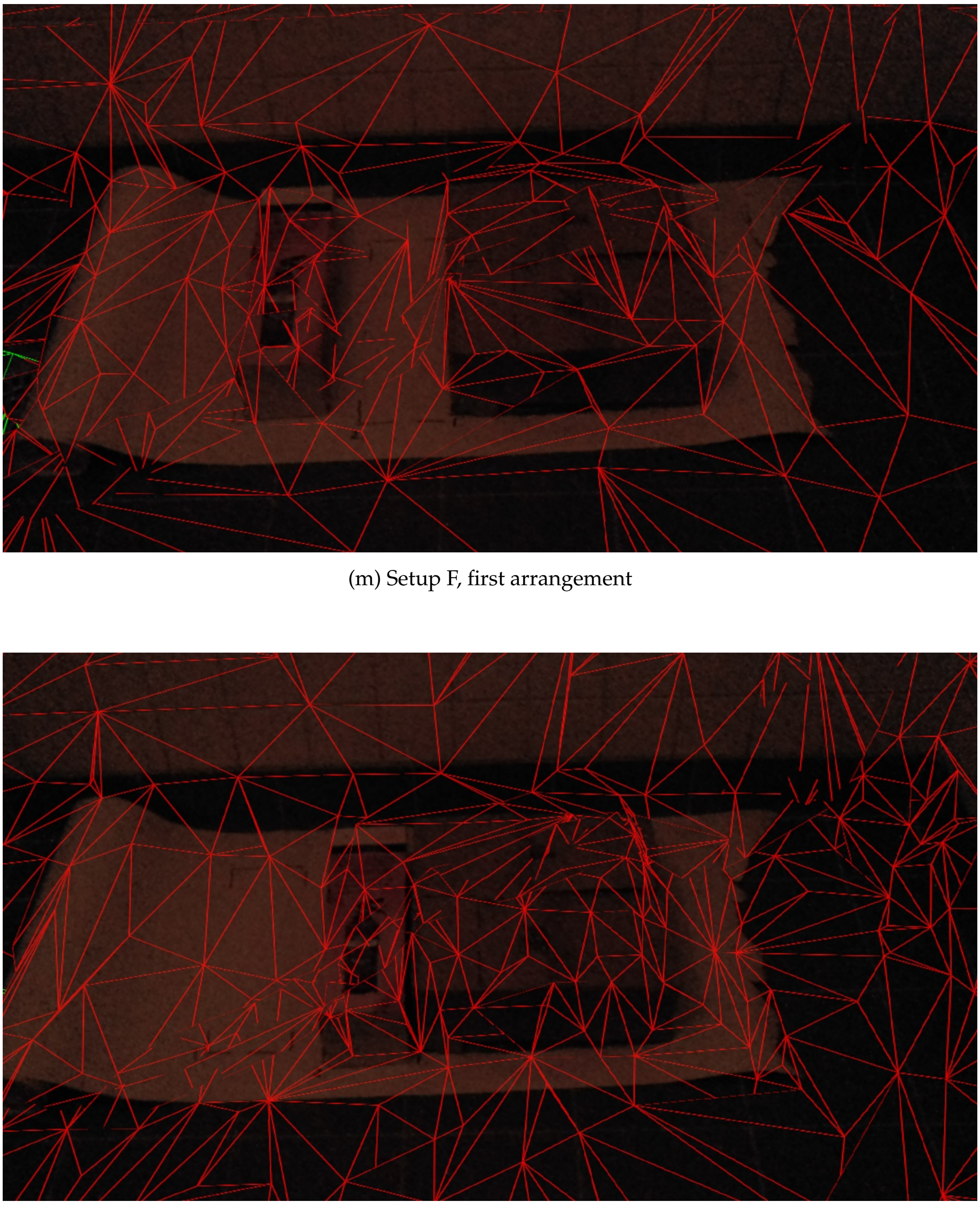
Figure 5.1: Images from experiments observing the created spatial map during different light conditions, continued.



(k) Setup E, first arrangement

(l) Setup E, second arrangement

Figure 5.1: Images from experiments observing the created spatial map during different light conditions, continued.



(m) Setup F, first arrangement

(n) Setup F, second arrangement

Figure 5.1: Images from experiments observing the created spatial map during different light conditions, continued.

## 5.3 Behavior of holograms and hand gestures

The results from the experiments that investigated how hologram behavior and recognition of hand gestures by the system were affected by different light levels are presented in tab 5.2.

No apparent change in behavior was observed except for the darkest experiment setup when the HoloLens system paused the application showing an error message about not being able to create the spatial map.

| Max light near icons(lx) | Ambient light(lx) | Holds position | Does not jitter | Dragging works | Gestures recognized |
|---|---|---|---|---|---|
| *580-610** | *590-610** | *Yes** | *Yes** | *Yes** | *Yes** |
| 30’000 | 14500-16200 | Yes | Yes | Yes | Yes |
| 22200-24100 | 19300-21200 | Yes | Yes | Yes | Yes |
| 34-40 | 72-85 | Yes | Yes | Yes | Yes |
| 5-10 | 16-27 | Yes | Yes | Yes | Yes |
| 2-3 | 1-3 | Yes | Yes | Yes | Yes |
| 0 | 1-2 | Yes | Yes | Yes | Yes |
| 0 | 0-1 | - | - | - | - |

Table 5.2: Results of functionality experiments of the prototype during different light conditions. *) indicates normal office light conditions, used as a point of reference.

## 5.4 Visibility of virtual content

Results from the experiments investigating the effect of different light conditions on visibility of virtual content can be seen in tab 5.3. When several combinations are listed as clearest or most unclear combination, the participant could not decide which of them were significantly more or less clear than the other.

Other than the answers to the specific questions posed, a number of remarks from the participants were noted. When the virtual text was placed on the test setups B and C (see section 4.2.3), i.e. with buildings and sky as the physical background as opposed to a white wall, all participants remarked that the dark gray virtual background made it hard or almost impossible to see the text, especially in setup C with the highest level of surrounding light. One participant noted that text on a dark gray virtual background drowns in the real background, especially if that real background is messy. Messy in this context is interpreted as containing many high contrast and detailed objects, which in this case were light colored buildings, dark asphalt, green trees and sky. One participant remarked that the sky and the light colored building made it hard or very hard to see the virtual content, especially with the dark gray application background. Two of the participants noted that the red background also made it hard or uncomfortable to view the text in setups B and C.

| Participant | Test setup | Backlight (lx) | Incident light towards virtual content(lx) | Correct words(%) | Clear color combination(s) | Unclear color combination(s) |
|---|---|---|---|---|---|---|
| 1 | A | 570-600 | 580-610 | 100 | B: dark gray + T: white, red | none |
| | B | 2900-3300 | 590-620 | 100 | B: white + T: all | B: dark gray + T: all |
| | C | 22500-24500 | 2300-2700 | 100 | B: white + T: all,<br>B: light gray + T: all | B: dark gray + T: all,<br>B: red + T: all |
| 2 | A | 580-600 | 590-610 | 100 | B: white + T: all | B: red + T:all |
| | B | 3400-3900 | 580-610 | 100 | B: white + T: all | B: dark gray + T: all,<br>B: red + T: all |
| | C | 22400-25000 | 1800-2000 | 100 | B: white + T: all,<br>B: light gray + T: all | B: dark gray + T: all,<br>B: red + T: all |
| 3 | A | 580-600 | 580-610 | 100 | B: red + T: blue,<br>B: white + T: all | B: dark gray + T: red, white |
| | B | 3500-3900 | 680-710 | 100 | B: white + T: all | B: dark gray + T: blue,green,<br>B: red + T: blue,green |
| | C | 19700-22000 | 2300-2700 | 100 | B: white + T: all,<br>B: light gray + T: all | B: red + T: all,<br>B: dark gray + T: all |

Table 5.3: Results of experiments on virtual content visibility during different light conditions. B is short for application background, T is short for text. "B: dark gray + T: white, red" would mean both the color combinations dark gray background and white text, and dark gray background and red text.

# 6 Analysis and discussion

This chapter covers the validity of the methods and how to interpret the results, as well as other observations and propositions on how to continue the work.

## 6.1 Experimental method

There are several aspects of the experimental methods used that can affect how the results are to be interpreted.

The measurements of illuminance were made by using an app on a smartphone. Tests on how well apps on smartphones compare to professional light meters in terms of accuracy have shown that they are not very reliable [53]. Due to the lack of access to a professional light meter, an app for a smartphone was the best option available. The app used was chosen since it had instructions on what it was intended to measure and how to perform measurements, something many apps lacked. The exact deviation or margin or error for the used app is not known. Nevertheless, the measurements followed the change in light conditions, i.e. turning down the light always resulted in a lower illuminance value, and vice versa. Due to the uncertain accuracy of the light meter app, the exact values for illuminance should be seen as rough approximations. They were also used as a tool to be able to make different experiments under roughly the same light conditions, to be able to make internal comparisons if needed.

A study of the available literature did not result in finding an existing suitable method of how to study the quality of a spatial map. The method used was therefore designed by the thesis author. Since the quality of the spatial map was evaluated by only one person by subjectively evaluating if the quality had changed, the results were filtered through the author's biases and preconceived notions. These factors can affect the reliability and replicability of the results. The experiments were also limited in scope. Spatial maps of larger scenes where real objects were moved around more would maybe have revealed differences in spatial map quality that were not revealed in the performed experiments. Experiments measuring the time taken for the spatial map to be corrected after changes in the real environment could have given interesting information.

As mentioned in the experimental work chapter, the next step could have been experiments on application behavior during a change in light. Differences in application behavior might be so small that they could not be detected in the experiments performed for this thesis, but may have been more noticeable when making continuous observation during dynamic changes in light.

One aspect that was considered was if specifics of how the prototype is implemented may have affected the performance of the prototype, and by extension also the experiments. An example could have been if poorly optimized code took a larger part of available CPU or memory than necessary, resulting in poor behaviour due to limited resources instead of due to the light conditions. Since the results of the experiments were not unexpected after having used applications developed by Microsoft or third party developers and were mostly positive, implementation details are not thought to have adversely affected the experiments to any substantial degree.

The environments for the visibility tests were chosen as suitable from the experiences of the author from using the HoloLens system, but a number of other situations could have been chosen. The colors of the visibility test app were chosen for several reasons. White background and dark gray, blue or green text or dark gray background and white text were chosen as being representative for common software color schemes. The light gray background was chosen due to Microsoft recommending it as substitute for white, as white can be seen as too intense [45]. The red background was chosen as a representative of a color that stands out from the colors in the surrounding environments, and that draws the eye. The other combinations, such as dark gray background and red text, or red background and blue text, were used to test the effect on unconventional color combinations. The experiments could also have been performed in more extreme conditions, such as brighter daylight or in environments with more reflective or detailed background.

The aim of the visibility tests was to test the overall difference in perceived visibility in different light conditions, and if the color combinations that worked best or worst would change between different experiment setups. No tests of visibility in poor light conditions were made, as it was deemed obvious that virtual content shown by using its own light source will be more visible than objects or text that does not have their own light source. The number of participants in the visibility test was very limited. This was in part due to the fact that the tests ideally had to be made in similar light conditions which were changing with the weather, and limitation in how much time was available for the author and the participants. Due to the small sample size, the results are preferably seen as suggestions of possible issues with visibility of virtual content in different light conditions.

## 6.2 Results

This topic discusses how the results should be interpreted and what they indicate.

### 6.2.1 Spatial map and hologram behaviour

The quality of the created spatial maps by the HoloLens system did not seem to be affected to any large degree by strong sunshine and limited light levels, until the light levels were very poor. Even in the poorest light conditions where the HoloLens system could create a spatial map, the difference in quality of the spatial map was so small or non-existent as to not being noticed in this experiment.

What the results do indicate is that no large differences in the registered spatial map could be seen for small scenes. They also show that the spatial map can be created even in light conditions that are much too dark for most activities without using an additional light source, such as trying to read text in a book or on a printed sign. It is shown that there is a limit on how dark the environments can be for the system to work, i.e. it does in some aspect rely on visible light. It is also shown that the system works in very bright light without seeming to be affected by light noise.

### 6.2.2 Visibility

The visibility experiments were conducted specifically to investigate if higher levels of ambient light or backlight would affect how visible virtual content was perceived.

The experiments were very limited, with a small set of participants, but some conclusions can still be made from those results. Not all participants agreed on what color combinations were most clear in normal office conditions without backlight. All participants agreed that white was the best background in conditions with bright ambient light. All participants remarked that the background had a large impact on the visibility of the virtual content when in conditions of strong light and backlight, especially when the virtual background was relatively dark.

The experiences from the participants in that experiment suggests that light, bright and messy backgrounds could become a problem when viewing virtual content with HoloLens, and that care would have to be taken when designing holographic applications for a user experience that would be acceptable outside of optimal conditions.

Something that was not investigated is how the holograms would be perceived in poor lighting conditions. It is obvious that virtual content will be more visible since it provides its own light source. Microsoft recommends not to use completely white as it can come off as too intense, but in this case it was perceived as the best against a strongly lit or heterogeneous background. This can indicate that different color schemes will be necessary for apps that are intended to be used in conditions with large disparities in light levels. Preferably the light sensor on the HoloLens could be used to automatically adjust the color scheme. To change color schemes depending on light levels to provide a pleasant user experience would be a task for the app developer, but due to the hardware limitations of the see-through displays some environments would probably pose problems that would be hard to solve merely through application design.

### 6.2.3 Other observations

When using HoloLens during development of the app, and when testing other functionality with existing apps and holograms, some additional observations were made.

Holograms and other virtual objects that are supposed to remain stationary can sometimes change position by a significant amount, either by moving small distances, changing orientation, or sometimes seeming to have disappeared altogether but then being discovered in another room or halfway through a wall. The cause for this was not always evident. Sometimes, it seems that this happens if there is a large change in the environment where the hologram is placed, such as moving some larger object. This can also happen if a hologram is placed before the HoloLens has had time to map the environment properly, and as the spatial map undergoes a significant change the hologram is displaced. When using applications with holograms that were supposed to be spatially persistent for several hours per day, these disruptions were observed on a daily basis on several occasions.

It is very hard to see holograms that are not very bright, are relatively small, are very detailed or contain many colours when outside in bright sunshine, especially against a white or light background. This was partly covered by the visibility experiments. Experiences the author had when viewing more complex virtual content made it evident that light conditions would affect such content even more than the high-contrast virtual content with large monochromatic surfaces that was used in the visibility experiments.

No large effects on application behaviour were noticed when running the applications in the presence of reflective surfaces, such as mirrors, polished tile, or glass windows. Due to limited variance in the environments where the application was used, it can not be ruled out that application behaviour can be affected by environments with other types or larger quantities of reflective surfaces.

## 6.3 Suitability for the given scenario

All of the determined criteria that the application had to fulfill were possible to implement thanks to the hardware and software functionality of HoloLens, using the recommended development framework Unity 3D. By using Unity even a beginner in 3D application development could relatively easily produce a simple application for demonstration purposes, thanks to the large and active community of Unity developers and the resources Microsoft provides to get developers started in developing for Microsoft. If development of commercial applications was to be done, the age of the Unity framework and its relative popularity in mainly game development means that there are experienced developers that could take on the task. The prototypes developed and the experiments that were performed, show that the HoloLens system is capable of creating an user experience that is both intuitive and impressive, if not always reliable.

There are a number of disadvantages, where the disadvantage that seems the most detrimental to use in industrial setting is the relative instability of the spatial persistence of the virtual content. The virtual content could not be relied upon to keep its position when using the application for relatively short amounts of time, in environments that were not extreme either in light conditions, size, or movement of objects and people. This would not be acceptable in applications aimed at industrial use.

Another disadvantage is that the visibility of the virtual content would be dependent of not having bright light sources in the environment, and especially when being outdoors where the weather conditions would have a large effect on the application experience. On the other hand, HoloLens applications could mitigate poor visibility in relatively dark environments when compared to for example printed instructions, since its virtual content provides its own light.

Another question is how to develop the applications for use in different environments by different users. In the prototype, a very limited use was demonstrated where the icons could be created by the user and placed in the surrounding environment. Those icons were then used as points for interaction to show panels of simple text. It would have to be possible to share spatial anchors and other information such as the text from the information panels with other HoloLens devices, something that in some respects is already possible [54], [55]. But what if the application should have more advanced functionality, such as a sequence of instructions where arrows or other symbols should point at different objects at certain times in the sequence? That would demand that a lot of labor intensive configuration work would have to be done on site to position all the virtual content correctly. If objects in the surrounding environment moved, the positions of the virtual content would have to be corrected. One way of mitigating the problem of virtual content having to be repositioned when objects in the environment moved, would be to use fiducial markers to calibrate the position of the virtual content, which in some ways seems to defeat the purpose of the marker-less system that is marketed as an important feature of HoloLens.

# 7 Conclusion

The purpose of this thesis was to investigate robustness of applications running on the HoloLens system. To narrow the scope of the thesis, the effect on application behavior by different levels of light in the surrounding environment was chosen as the subject for investigation.

In this thesis, a set of simple applications for HoloLens were developed and used in experiments to investigate two research questions: how is the performance of a holographic application running on HoloLens affected by various light conditions, and can different light conditions have a detrimental effect on the user experience of HoloLens holographic applications? Addressing the first question, the thesis demonstrated that light levels do affect the ability of a HoloLens application to determine its surroundings only in very dark environments, to the point that the application became unusable, but that overall application behavior is not affected for light levels ranging from bright sunshine to very dim. Addressing the second research question, it is shown that bright light equivalent to backlight from an open window during daytime may interfere with visibility of virtual content.

During use of the prototype in development and use of other HoloLens applications, it was observed that the applications would sometime lose track of the environment which would lead to holograms floating, jittering, being displaced or seemingly disappear. This could happen if major rearrangements of objects in the surrounding happened, but sometimes no apparent cause could be determined.

The conclusions from these results are that due to the effect of bright light on the visibility of virtual content and the tendency for virtual content to lose their position during and between application execution, the system may not be reliable enough for use in industry in its current form.

Future work on this topic should primarily be aimed at why the system loses track of the surroundings which leads to displacement of virtual content, and what could be done to mitigate that problem. Another direction of inquiry would be to investigate how to design virtual content in terms of shape, contrast and color to best handle varying light conditions, especially bright light.

# Bibliography

[1] D. Weinswig, “Virtual And Augmented Reality Become Realistic Revenue Generators”, *Forbes*, Oct. 26, 2016. [Online]. Available: https://www.forbes.com/sites/deborahweinswig/2016/10/26/virtual-and-augmented-reality-become-realistic-revenue-generators/#560064916fc5 (visited on 01/04/2018).

[2] N. Wingfield and M. Isaac, “Pokémon Go Brings Augmented Reality to a Mass Audience - The New York Times”, *The New York Times*, Jul. 11, 2016. [Online]. Available: https://www.nytimes.com/2016/07/12/technology/pokemon-go-brings-augmented-reality-to-a-mass-audience.html (visited on 01/04/2018).

[3] Á. Di Serio, M. B. Ibáñez, and C. D. Kloos, “Impact of an augmented reality system on students’ motivation for a visual art course”, *Computers & Education*, vol. 68, pp. 586–596, 2013.

[4] D. Gasques Rodrigues, A. Jain, S. R. Rick, L. Shangley, P. Suresh, and N. Weibel, “Exploring mixed reality in specialized surgical environments”, in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 2017, pp. 2591–2598.

[5] B. Marr. (Sep. 2, 2018). "What is Industry 4.0? Here’s A Super Easy Explanation For Anyone", Forbes Media LLC, [Online]. Available: https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/ (visited on 10/06/2018).

[6] F. Lardinois. (Feb. 29, 2016). "Microsoft Launches Pre-Orders For $3,000 Hololens Development Kit, Will Start Shipping March 30", TechCrunch, Oath Tech Network, [Online]. Available: https://techcrunch.com/2016/02/29/microsoft-open-pre-orders-for-hololens-development-kit-will-start-shipping-march-30/ (visited on 12/10/2018).

[7] A. Turner, M. Zeller, E. Cowley, and B. Bray. (n.d.). "Coordinate systems", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/coordinate_systems#Handling_tracking_errors (visited on 08/07/2017).

[8] M. Billinghurst, A. Clark, and G. Lee, *A survey of augmented reality.* Ser. Foundations and trends in human-computer interaction: 8:2-3. Hanover, Massachusetts : Now Publishers, 2015, ISBN: 9781601989208.

[9] R. T. Azuma, “A survey of augmented reality”, *Presence: Teleoperators and virtual environments*, vol. 6, no. 4, pp. 355–385, 1997.

[10] J. Amores and J. Lanier, “Holoart: Painting with holograms in mixed reality”, in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, ACM, 2017, pp. 421–424.

[11] N. Navab, T. Blum, L. Wang, A. Okur, and T. Wendler, “First deployments of augmented reality in operating rooms”, *Computer*, vol. 45, no. 7, pp. 48–55, 2012.

[12] E. R. Eiriksson, D. B. Pedersen, J. R. Frisvad, L. Skovmand, V. Heun, P. Maes, and H. Aanæs, “Augmented reality interfaces for additive manufacturing”, in *Scandinavian Conference on Image Analysis*, Springer, 2017, pp. 515–525.

[13] J. Peddie, *Augmented reality: Where we will all live*. Springer, 2017.

[14] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light”, in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, IEEE, vol. 1, 2003, pp. I–I.

[15] S. B. Gokturk, H. Yalcin, and C. Bamji, “A time-of-flight depth sensor-system description, issues and solutions”, in *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW’04. Conference on*, IEEE, 2004, pp. 35–35.

[16] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “Kinectfusion: Real-time dense surface mapping and tracking”, in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, IEEE, 2011, pp. 127–136.

[17] (n.d.). "See-through". Used with permission from Microsoft, Microsoft, [Online]. Available: `https://az835927.vo.msecnd.net/sites/mixed-reality/Resources/images/See-through.jpg` (visited on 08/03/2017).

[18] M. Zeller and B. Bray. (n.d.). "Hololens hardware details", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details` (visited on 03/29/2017).

[19] (n.d.). "Microsoft Hololens", Microsoft, [Online]. Available: `https://www.microsoft.com/microsoft-hololens/en-us` (visited on 03/29/2017).

[20] S. Colaner, “What’s Inside Microsoft’s Hololens And How It Works”, *Tom’s Hardware*, Aug. 23, 2016. [Online]. Available: `http://www.tomshardware.com/news/microsoft-hololens-components-hpu-28nm,32546.html` (visited on 04/25/2017).

[21] B. C. Kress and W. J. Cummings, “11-1: Invited paper: Towards the ultimate mixed reality experience: Hololens display architecture choices”, in *SID Symposium Digest of Technical Papers*, Wiley Online Library, vol. 48, 2017, pp. 127–131.

[22] M. Zeller and B. Bray. (n.d.). "Hardware accessories", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/hardware_accessories` (visited on 04/25/2017).

[23] (n.d.). "Displays". Used with permission from Microsoft, Microsoft, [Online]. Available: `https://az835927.vo.msecnd.net/sites/mixed-reality/Resources/images/Displays.jpg` (visited on 08/03/2017).

[24] (n.d.). "Sensor_bar". Used with permission from Microsoft, Microsoft, [Online]. Available: `https://az835927.vo.msecnd.net/sites/mixed-reality/Resources/images/Sensor%5C_bar.jpg` (visited on 08/03/2017).

[25] A. A. Cameron, “Optical waveguide technology and its application in head mounted displays”, in *Proc. SPIE*, vol. 8383, 2012, 83830E.

[26] A. Yip. (Jan. 9, 2017). "How Do Augmented Reality Displays Work?", Forbes Media LLC, [Online]. Available: `https://www.forbes.com/sites/quora/2017/01/09/how-do-augmented-reality-displays-work/` (visited on 08/07/2017).

[27] C. Chinnok, “Waveguide-based Displays Maturing for Augmented Reality Applications”, *Display Daily*, vol. 22, 6 2015. [Online]. Available: `https://www.displaydaily.com/free-content/sponsored-articles/waveguide-based-displays-maturing-for-augmented-reality-applications` (visited on 10/06/2018).

[28] A. Turner, M. Zeller, E. Cowley, and B. Bray. (n.d.). "Rendering", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/rendering` (visited on 04/25/2017).

[29] A. Turner, M. Zeller, B. Bray, and E. Cowley. (n.d.). "Hologram stability", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/hologram_stability` (visited on 04/25/2017).

[30] (2016). "What’s a light point?", [Online]. Available: `https://forums.hololens.com/discussion/92/whats-a-light-point` (visited on 04/25/2017).

[31] M. Hachman, “We found 7 critical Hololens details that Microsoft hid inside its developer docs”, *PCWorld*, Mar. 2, 2016. [Online]. Available: `http://www.pcworld.com/article/3039822/consumer-electronics/we-found-7-critical-hololens-details-that-microsoft-hid-inside-its-developer-docs.html` (visited on 04/25/2017).

[32] M. Zeller, K. Baker, and B. Bray. (n.d.). "Spatial mapping", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_mapping` (visited on 08/03/2017).

[33] A. Turner, M. Zeller, rwinj, and B. Bray. (n.d.). "Gaze", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/gaze` (visited on 03/29/2017).

[34] rwinj, M. Zeller, and B. Bray. (n.d.). "Gestures", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/gestures` (visited on 03/29/2017).

[35] hak0n, M. Zeller, and B. Bray. (n.d.). "Voice input", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/voice_input` (visited on 03/29/2017).

[36] (n.d.). "Cortana on Hololens", Microsoft, [Online]. Available: `https://support.microsoft.com/en-us/help/12630/hololens-cortana-on-hololens` (visited on 04/25/2017).

[37] A. Turner, Y. Park, lk711, M. Zeller, K. Baker, and B. Bray. (n.d.). "Voice input in Unity", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/voice_input_in_unity` (visited on 04/25/2017).

[38] hak0n, M. Zeller, K. Baker, and B. Bray. (n.d.). "Spatial sound", Microsoft, [Online]. Available: `https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_sound` (visited on 03/29/2017).

[39] T. Whitney, M. Satran, M. Jacobs, devfables, D. Das, and S. Weston. (n.d.). "What’s a Universal Windows Platform (UWP) app?", Microsoft, [Online]. Available: `https://docs.microsoft.com/en-us/windows/uwp/get-started/whats-a-uwp` (visited on 04/13/2017).

[40] M. Zeller, N. Schonning, and B. Bray. (n.d.). "Development overview", Microsoft, [Online]. Available: `https://docs.microsoft.com/en-us/windows/mixed-reality/development-overview` (visited on 12/10/2018).

[41] (n.d.). "Unity3D", Unity Technologies, [Online]. Available: https://unity3d.com/ (visited on 04/13/2017).

[42] (n.d.). "Academy: Code, tutorials, and lessons", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/academy (visited on 09/02/2017).

[43] (n.d.). "Github - Microsoft/MixedRealityToolkit-Unity: MixedRealityToolkit-Unity uses code from the base MixedRealityToolkit repository and makes it easier to consume in Unity", [Online]. Available: https://github.com/Microsoft/MixedRealityToolkit-Unity (visited on 08/15/2017).

[44] (n.d.). "GitHub - Microsoft/HoloToolkit-Unity", [Online]. Available: https://github.com/Microsoft/HoloToolkit-Unity (visited on 05/05/2017).

[45] M. Vitazko, M. Zeller, rwinj, and B. Bray. (n.d.). "Color design", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/color_design (visited on 08/08/2017).

[46] (n.d.). "The International System of Units (SI)", BIPM, [Online]. Available: https://www.bipm.org/en/measurement-units/ (visited on 10/23/2018).

[47] (n.d.). "Illuminance", [Online]. Available: https://en.wikipedia.org/wiki/Illuminance (visited on 08/02/2017).

[48] Kurtis, M. Zeller, E. Cowley, N. Wadhwa, and B. Bray. (n.d.). "Holograms 230", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_230 (visited on 03/29/2018).

[49] Kurtis, M. Zeller, E. Cowley, N. Wadhwa, and B. Bray. (n.d.). "Holograms 101", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_101 (visited on 08/14/2017).

[50] (n.d.). "Belysning på kontor", Arbetsmiljöverket, [Online]. Available: https://www.av.se/inomhusmiljo/ljus-och-belysning/belysning-pa-kontor/ (visited on 08/03/2017).

[51] (n.d.). "Vieyra Software | Physics Toolbox Light Meter", Vieyra Software, [Online]. Available: https://www.vieyrasoftware.net/physics-toolbox-light-meter (visited on 08/08/2017).

[52] (n.d.). "Xperia$^{TM}$ Z5 Compact – Sony's compact smartphone - Sony Mobile", Sony Mobile Communications Inc, [Online]. Available: https://www.sonymobile.com/global-en/products/phones/xperia-z5-compact/ (visited on 08/08/2017).

[53] (Feb. 23, 2016). "Luxmeter App versus measuring device: Are smartphones suitable for measuring illuminance?", DIAL, [Online]. Available: https://www.dial.de/en/blog/article/luxmeter-app-versus-measuring-deviceare-smartphones-suitable-for-measuring-illuminance/ (visited on 05/05/2017).

[54] A. Turner, M. Zeller, E. Cowley, and B. Bray. (n.d.). "Coordinate systems - Spatial anchor sharing", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/coordinate_systems#spatial_anchor_sharing (visited on 01/02/2018).

[55] Kurtis, M. Zeller, K. Burns, E. Cowley, N. Wadhwa, and B. Bray. (n.d.). "Holograms 240", Microsoft, [Online]. Available: https://developer.microsoft.com/en-us/windows/mixed-reality/holograms_240 (visited on 01/02/2018).