Lappeenranta–Lahti University of Technology LUT
School of Engineering Science
Computational Engineering and Technical Physics
Technomathematics

Karoliina Varso

IMPROVING THE ACCURACY OF TIME-OF-FLIGHT CAMERA -BASED FLOOR HEIGHT ESTIMATION IN MIXED REALITY HEAD-MOUNTED DISPLAYS

Master's Thesis

Examiners: Assoc. Prof. Tapio Helin
PhD. Antti Karjalainen

Supervisors: PhD. Antti Karjalainen
Thomas Carlsson
Assoc. Prof. Tapio Helin
Prof. Lasse Lensu

ABSTRACT

Lappeenranta–Lahti University of Technology LUT
School of Engineering Science
Computational Engineering and Technical Physics
Technomathematics

Karoliina Varso

IMPROVING THE ACCURACY OF TIME-OF-FLIGHT CAMERA -BASED FLOOR HEIGHT ESTIMATION IN MIXED REALITY HEAD-MOUNTED DISPLAYS

Master's Thesis

2021

58 pages, 36 figures, 2 tables.

Examiners: Assoc. Prof. Tapio Helin
PhD. Antti Karjalainen

Keywords: mixed reality, floor height estimation, time-of-flight, 3D point cloud

In recent years, the use of mixed reality headsets has gained a lot of popularity in several applications ranging from creative design work to implementation of complex training procedures. To ensure a pleasant user experience, it is important that the floor height of the headset is precisely defined. The objective of this thesis was to improve the floor height estimation algorithm of the Varjo XR-3 mixed reality headset. The floor height estimation of the headset is based on using a time-of-flight camera. Typical sources of error of time-of-flight cameras were first investigated by reviewing recent publications in the field. Then, attempts were made to correct the most significant systematic error sources found. For this purpose, two separate correction models were created. The performance of the developed correction models was experimentally estimated. After applying both of the corrections, the absolute maximum error between the calculated floor height estimates and the ground truth was greatly reduced. Based on the results the work was successful and the accuracy of the floor height estimation was significantly improved. Further research is required especially to make the algorithm work for different floor materials.

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT
School of Engineering Science
Laskennallinen tekniikka ja teknillinen fysiikka
Teknillinen matematiikka

Karoliina Varso

LENTOAIKAKAMERAAN PERUSTUVAN LATTIANKORKEUSARVIOINNIN TARKKUUDEN PARANNUS SEKOITETUN TODELLISUUDEN LASEISSA

Diplomityö

2021

58 sivua, 36 kuvaa, 2 taulukkoa.

Tarkastajat: Apul.prof. Tapio Helin
FT Antti Karjalainen

Hakusanat: sekoitettu todellisuus, lattiakorkeuden estimointi, lentoaika, 3D-pistepilvi

Viime vuosina sekoitetun todellisuuden lasien käyttö on kasvattanut suosiotaan useissa sovelluksissa, kuten esimerkiksi luovassa suunnittelutyössä sekä erilaisten monimutkaisten koulutusprosessien toteuttamisessa. Miellyttävän käyttökokemuksen varmistamiseksi on tärkeää, että sekoitetun todellisuuden lasien korkeus lattiasta on määritelty tarkasti. Tämän opinnäytetyön tavoitteena oli parantaa Varjo XR-3 -lasien lattiakorkeuden arviointialgoritmia. Lasien lattiakorkeuden arviointi perustuu lentoaikakameran käyttöön. Aluksi tutkittiin lentoaikakameroiden tyypillisiä virhelähteitä tarkastelemalla alan viimeaikaisia julkaisuja. Merkittävimmät löydetyt järjestelmälliset virhelähteet pyrittiin korjaamaan luomalla kaksi erillistä korjausmallia. Kehitettyjen korjausmallien suorituskykyä arvioitiin kokeellisesti. Molempien korjausten soveltamisen jälkeen absoluuttinen maksimivirhe laskettujen lattiakorkeusarvioiden sekä todellisen lattiakorkeuden välillä pieneni huomattavasti. Tulosten perusteella työ onnistui; lattiakorkeuden arvioinnin tarkkuus parani merkittävästi. Lisätutkimusta tarvitaan erityisesti, jotta algoritmi saataisiin toimimaan erilaisilla lattiamateriaaleilla.

PREFACE

I would first like to thank Antti, Tapio and Lasse for all the guidance from an academic perspective throughout the writing process. In addition, I'd like to warmly thank Varjo Technologies Oy for providing me this wonderful opportunity. It has been a pleasure to work with this interesting topic as my master's thesis. Thank you Thomas for all the encouraging and idea-rich support, Teemu for your contribution to the mechanical implementation, Yu-Jung for data acquisition and everyone else who has been part of this journey. It has been truly joyful to work with all the welcoming, talented, kind and considerate people of Varjo.

To my family: thank you for supporting me through all these years - you are the pillars of my life. To Henna, Teemu, and all of my friends, thank you for the amazing and memorable years at the university and all the wisdom that you have kindly shared with me.

Lastly, to my fiancé Matti and to my son Väinö: thank you for always brightening up my days. The years with you have been full of love, joy and unconditional support, and I would not have been able to do this on my own. I would like to dedicate this work to you.

Helsinki, June 21, 2021

Karoliina Varso

CONTENTS

LIST OF ABBREVIATIONS

2D	Two-Dimensional
3D	Three-Dimensional
APD	Avalanche Photodiode
AR	Augmented Reality
AV	Augmented Virtuality
CCD	Charge-Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CW	Continuous Wave
FOV	Field of View
HMD	Head-Mounted Display
IMU	Inertial Measurement Unit
IR	Infrared
LED	Light-Emitting Diode
MEMS	Micro-Electromechanical System
MPI	Multipath Interference
MR	Mixed Reality
NIR	Near-Infrared
RANSAC	Random Sample Consensus
SPADS	Single Photon Avalanche Diodes
ToF	Time-of-Flight
VR	Virtual Reality
XR	Extended Reality

1 INTRODUCTION

This chapter first introduces the reader to the topic of this thesis on a high level. After that, the research motive is introduced and objectives & delimitations for the work are stated.

1.1 Background

During the 1960's the world changed at a tremendous pace due to the many significant innovations and rapid advancements made in science and technology. The best-known achievements of the time are unquestionably related to human and robotic exploration of space, but the development of computer technology has perhaps made even more significant an impact on the modern society and the way people live and work today. Back then, a field of research called Virtual Reality (VR) emerged and began to grow interest, as the first known and documented Head-Mounted Display (HMD) (Fig. 1) was built by an American computer scientist, Ph.D. Ivan Sutherland [1]. The constructed device could be connected to a computer in order to create a visually perceptible pseudo-reality that the user of the device could see. The user was able to see rooms created with the computer, and even the perspective changed according to the movement of the head of the user. The first HMD was not very practical to use and the computer-generated graphics were only simple wireframe models of rooms [2], but the invention truly was one of a kind and in many ways ahead of its time.

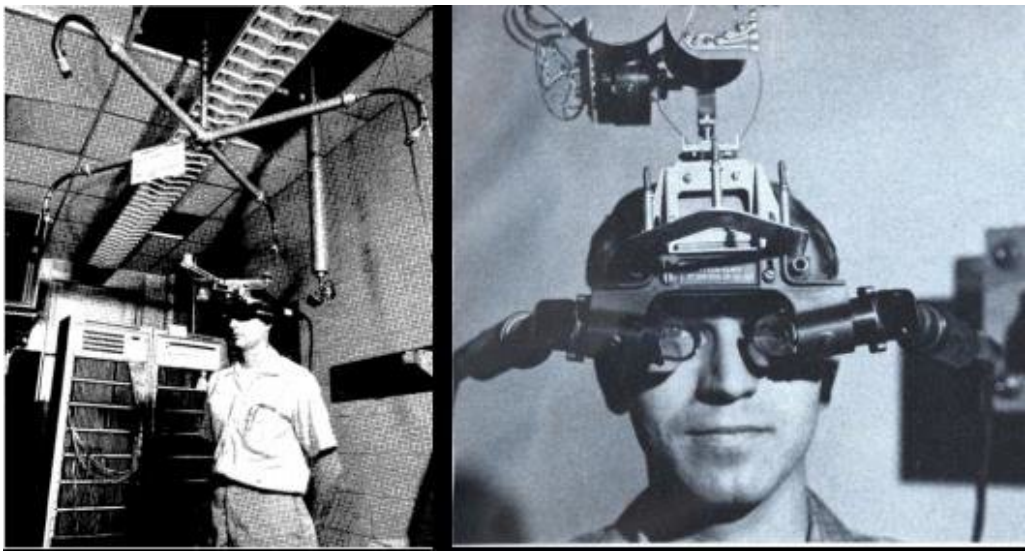

Figure 1. The first Head-Mounted Display built in the 1960's [1].

After the 1960's, a huge amount of both scientific and commercial research and development work has been done related to HMD systems and VR applications. The work done during the last few decades has also generated many additional and supplementary concepts alongside the VR. Most importantly, these include Mixed Reality (MR), Augmented Reality (AR) and Augmented Virtuality (AV).

In 1994, Milgram and Kishino presented a simplified definition for the difference between the concepts in the form of Figure 2 [3]. The main message of the figure is that MR should be thought of as the top concept for all the different technologies that combine virtual world and real world - regardless of the ratio in which they are mixed.

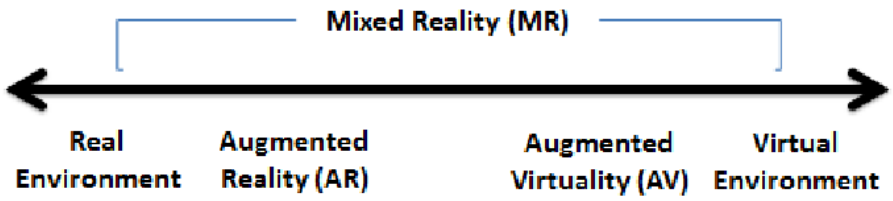

Figure 2. An illustration of the spectrum of virtuality showing the intermediate forms of fully real and fully virtual reality [3].

In MR, information about the surrounding reality is collected with various sensors to merge the virtual and real world together, which can significantly increase the feeling of immersion. Unlike in VR applications, selected parts of the physical world can be included in MR. In addition, users can typically interact with the augmented virtual parts. However, the concept of MR is not unambiguously defined, and in the absence of a common consensus it can either include or exclude some of the sub-categories setting in between the completely real and a completely virtual environment [4]. Lately, a new concept of Extended Reality (XR) has gained popularity, as it further clarifies the division. It has been proposed that XR could be used as the main category for AR, AV and MR [4].

In recent years, the development of XR devices has also increased user expectations towards the quality of virtual and real world merging. In order to meet these increasingly demanding expectations, XR devices have to be able to accurately track the user and his/her surrounding world in order to define how to successfully augment application-specific virtual content to the correct locations. For instance, it is very important to constantly track how high above the floor the eyes of the user are located, so that the augmented objects would not seem to float in midair or sink inside the floor. This is because the floor is one of the most often used planes for adding virtual objects onto. One way to track the

floor height is to utilize a Time-of-Flight (ToF) camera. ToF cameras are able to generate Three-Dimensional (3D) point clouds of their view. From the generated point cloud it is then possible to detect the floor plane and finally estimate the floor height.

Varjo Technologies Oy (hereafter referred to as Varjo), established in 2016, is a Finnish startup company producing extremely high-resolution industrial-grade VR and XR hardware for professional use. There is a wide range of both existing and potential applications, including training, simulation, design, engineering, medical and research. Practically the only limiting factor for possible use cases is imagination. One recent and particularly compelling user case example is Boeing, which chose to use the Varjo devices for training the astronauts of the Starliner program [5]. There are of course many other interesting but more down-to-earth examples of use cases as well, such as 3D sketching and operation of various kinds of interactive prototypes.

The purpose of this thesis is to first examine and then develop and improve the floor height estimation method used in the Varjo XR-3 HMD. In general, automatic floor-height detection improves user-centredness, as the users of the device are not required to manually calibrate the floor height through a cumbersome, frustrating and time consuming process. When successful, this work will significantly contribute to the development of automatic floor height estimation. It will also contribute to enabling the method to be utilized in the commercial product.

1.2 Objectives and delimitations

The aim of this thesis is to first study the different error sources of the ToF camera utilized in the Varjo XR-3 HMD. After determining the most significant error sources, possible correction methods and models are suggested, applied and evaluated. The research questions for this thesis are:

- What are typical error sources of a ToF camera?
- Can the ground truth be determined reliably enough to evaluate the improvement of the estimates?
- Can the floor height estimation be improved by modeling and correcting the most prominent error sources?
- How accurate is the estimation after the suggested corrections?
- If significant residual error is observed, what could be the cause of it?

This thesis is limited to cover only one floor material without any objects on the floor. Functionality of the developed floor height estimation method in motion will not be evaluated within the framework of this thesis. In addition, plane detection algorithms from the literature are not tested in practice due to the limited schedule.

1.3 Structure of the thesis

First, Chapter 1 introduces the reader to the topic of this thesis. Chapter 2 then describes the main components of the Varjo XR-3 HMD, used in this study. The chapter begins with a general description of the HMD and then gives a more detailed description of the components that are most essentially related to this thesis. Then, in Chapter 3, floor height estimation methods from the literature are first reviewed. After that, the currently used method for floor height estimation is introduced. Additions and changes to the current method are then proposed and described in detail. Chapter 4 introduces the methods used for collecting the experimental data. In addition, the methods utilized for determining the ground truth in the different test setups used are described. Finally, the acquired results are declared. In Chapter 5, the results of this thesis are discussed and the quality of the proposed floor height estimation method is evaluated. Propositions for future work are provided. Lastly, Chapter 6 concludes the study.

2 VARJO XR-3 HEAD-MOUNTED DISPLAY

Based on [6], this chapter first introduces the technical specifications of the Varjo XR-3 HMD that is used in the experimental part of this work. Then, selected subsystems most relevant to this thesis are described in more detail.

2.1 General description

Varjo XR-3 (hereafter referred to as the "selected HMD"), seen in Figure 3, provides ultrarealistic visual fidelity and, at the time of writing, the widest Field of View (FOV) of any commercially available XR headset. The device produces an extremely immersive mixed reality experience, wherein real and virtual elements blend together seamlessly due to continuous real-time depth determination.

Figure 3. Varjo XR-3 Head-Mounted display [6].

General technical specifications of the selected HMD are presented in Table 1. Noteworthy details include industry-first ToF-based depth tracking for detection of objects on the scene and boundaries of the surrounding environment, as well as rapid tracking of eye movement at a frequency of 200 Hz, allowing foveated rendering.

Table 1. Technical specifications of the Varjo XR-3 HMD. [6]

Display and Resolution	Full Frame Display with human-eye resolution 27° focus area of 1920 x 1920 px per eye. Peripheral area of 2880 x 2720 px per eye. Colors: 99% sRGB, 93% DCI-P3
Field of View	115°
Refresh Rate	90 Hz
Mixed Reality	Ultra-low latency Dual 12 MP video pass-through
XR Depth	ToF Camera + RGB, 0.4 to 5 m range
Hand Tracking	Ultraleap Gemini (V5)
Weight	594 g + headband 386 g
Dimensions	Width 200 mm, height 170 mm, length 300 mm
Positional Tracking	SteamVR™ 2.0 or 1.0 tracking system Varjo™ tracking with pass-through RGB cameras
Eye Tracking	200 Hz with sub-degree accuracy 1-dot calibration for foveated rendering
Connectivity	PC Connections: 2 x DisplayPort and 2 x USB-A 3.0+ 3.5 mm audio jack with microphone support
Comfort and Wearability	3-point precision fit headband Replaceable, easy-to-clean polyurethane face cushions Automatic interpupillary distance adjustment 59-71 mm

Eye tracking data can also be recorded for a wide range of uses, e.g. post-analysis of training situations or various research purposes. Other features include automatic determination of the interpupillary distance and a 90 Hz frame rate. Due to the three-point headband, the device is suitable for heads of all shapes and can also be used while wearing eyeglasses.

2.2 Inertial measurement unit

Inertial Measurement Unit (IMU) is a Micro-Electromechanical System (MEMS) sensor used to track the rotational and translational movement of the HMD. Rotational movement is defined as movement around the x, y, or z axes and translational movement is defined as movement that occurs in the direction of any of the axes.

IMU sensors typically consist of a three-axis accelerometer and a three-axis gyroscope, which track the directional and angular velocity along and around the x, y and z axes. Moreover, rotational movement is defined separately for each of the axes as pitch, yaw, and roll. The axes are illustrated in relation to the HMD in Figure 4.

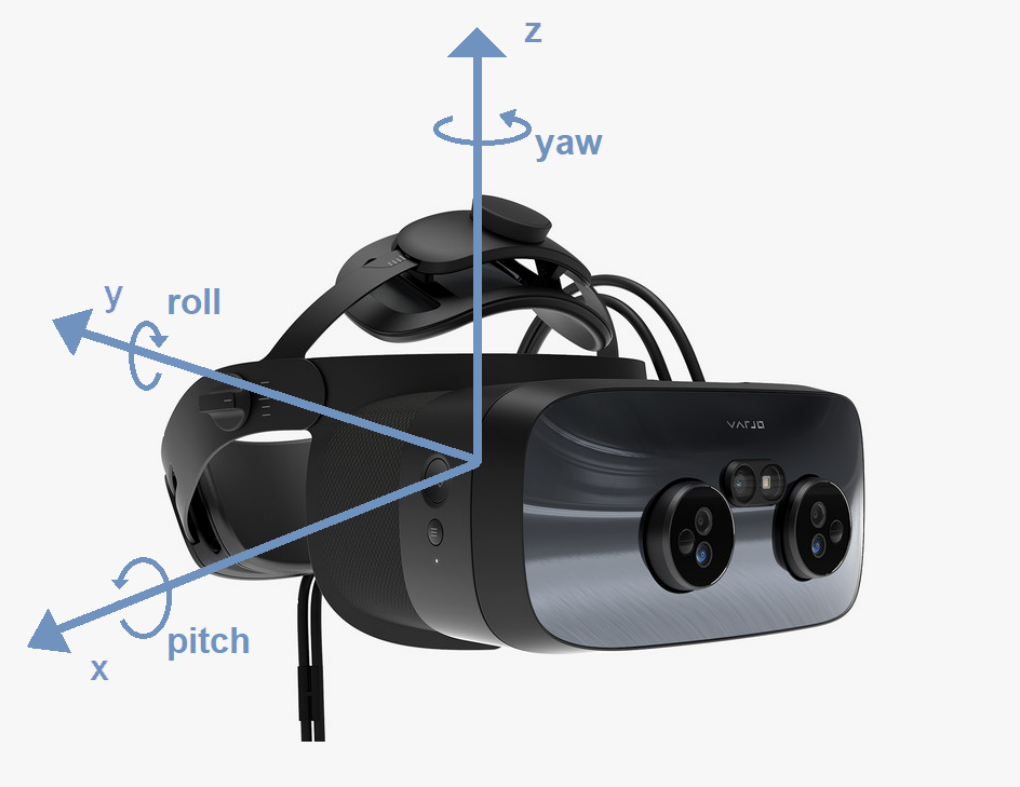

Figure 4. Coordinate system of a Head-Mounted display. Pitch, yaw and roll are the rotation directions relative to the axis.

By tracking the orientation of the HMD with the IMU, it is possible to keep track of the direction of the gravity vector, i.e. the information of in which direction the ground is located in relation to the HMD. The direction of the gravity vector is determined by combining information provided by the gyroscope and the accelerometer. Because the raw data is noisy and the position and rotation of the HMD are not measured directly, the extended Kalman filter is used in order to achieve more accurate estimates compared to direct use of the noisy measurements. In the context of floor height estimation, the gravity vector is used to rotate the ToF camera point cloud parallel to the real world coordinate system, regardless of the viewing angle of the device.

2.3 Time-of-flight camera

A ToF camera is a range imaging device that can be used to create Two-Dimensional (2D) images, where the depth is individually measured for each pixel. This kind of a 2D image is called a range image. The selected view is illuminated with modulated light and the distance between the ToF and each point from where the light reflects back can be calculated from the time it takes for the light to travel back and forth from the sensor. General working principle of a ToF camera is illustrated in Figure 5 [7]. For each pixel coordinate (x, y), the corresponding angle at which the light is reflected back to the sensor is also known [8]. Then, by combining the depth and the angle information, it is possible to construct a 3D point cloud from the detected scene.

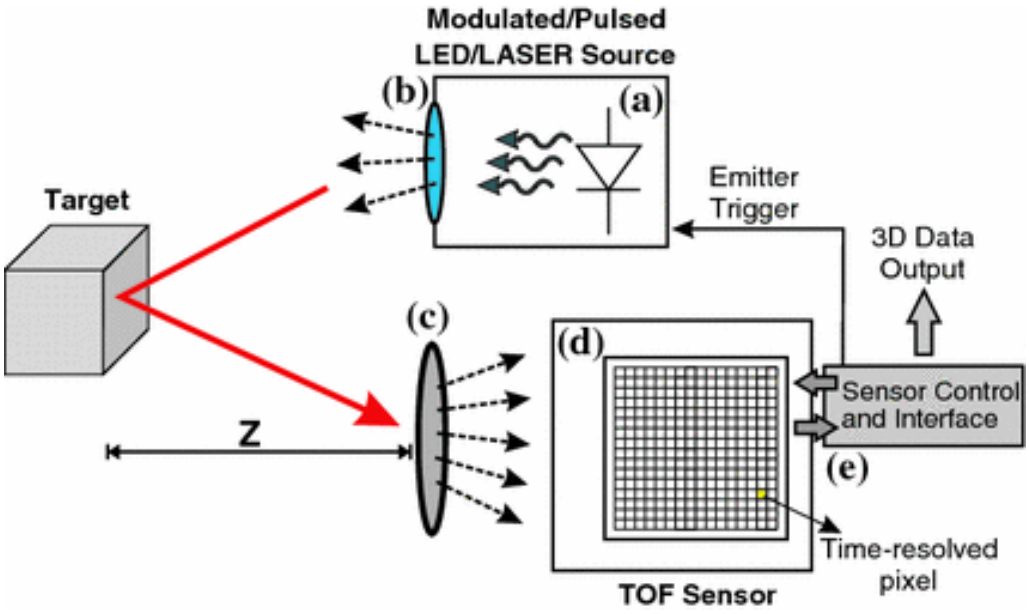

Figure 5. General working principle of a time-of-flight camera. The light source illuminates the target, from where the emitted light reflects back to the sensor. The distance between the target and the ToF camera can then be defined by measuring the roundtrip time of the light [7].

In order to illuminate the scene, ToF cameras use modulated light sources. Typically either pulsed modulation or Continuous Wave (CW) modulation technique is utilized to illuminate the scene [9]. Difference between these two methods is discussed in chapter 2.3.1. The light source used is typically either a laser diode or a Light-Emitting Diode (LED) due to their relatively low price and because both can be modulated rapidly, at frequencies up to 100 MHz [10]. Lasers and LEDs are also readily available at the wavelength range of interest for the application, which partly contributes to their widespread use. Both arrays and individual illumination units are used to produce Infrared (IR) [11] or Near-Infrared (NIR) [12] light. Using laser diode systems requires more safety attention

than LED-based systems [13], as high-intensity infrared light can damage eyes. IR lasers can be particularly dangerous, as IR light is invisible to the human eye. The wavelength utilized in this thesis is 850 nm, which is located at the NIR region of the electromagnetic spectrum.

Refractive optics, such as lenses, are often placed in front of the sensor, which helps in gathering and focusing the incoming light to the detective area of the sensor. Additionally, the optics used also determine the FOV of the ToF camera [14]. Figure 6 shows the FOV of the ToF camera used in this thesis.

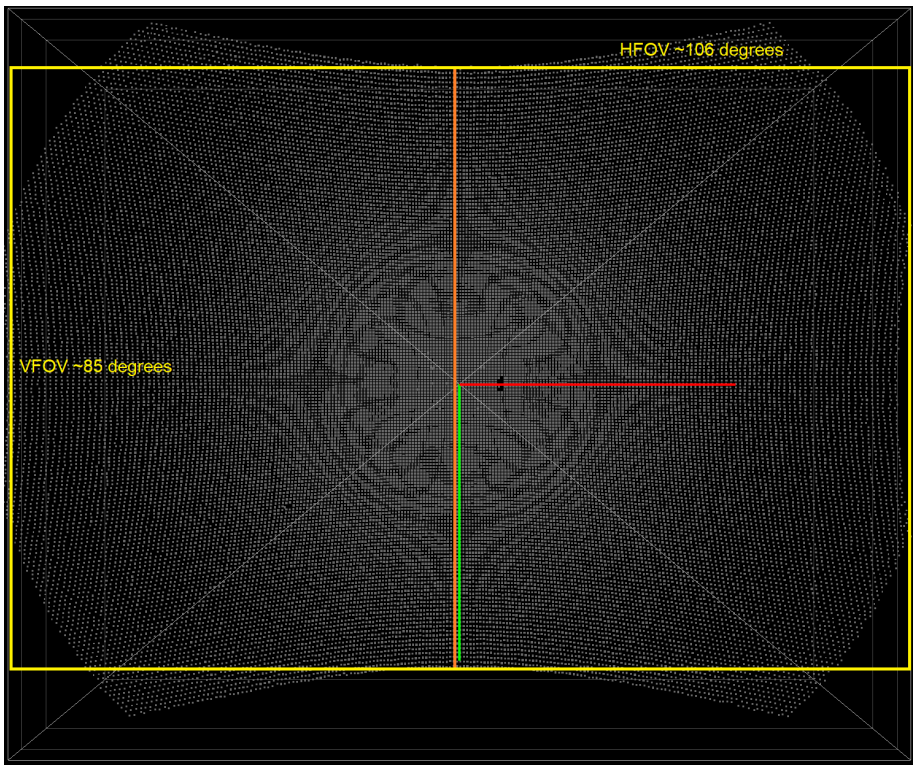

Figure 6. Vertical and horizontal field of view of the ToF camera. Since the lens of the ToF camera is round but the depth images produced by the camera are rectangular, the image is distorted. This distortion must be corrected. In this case, the final FOV corresponds to the range represented by the points.

For detecting and processing the reflected light, different sensor technologies, such as Avalanche Photodiode (APD) [15], Single Photon Avalanche Diodes (SPADS) [14], Complementary Metal Oxide Semiconductor (CMOS) or Charge-Coupled Device (CCD) [16, 17] are used. The selected HMD considered in this thesis utilizes a CW-modulated ToF camera with a CMOS sensor.

2.3.1 Infrared modulation

In the **pulsed modulation** method, the distance is defined by observing the time taken by the emitted light to return back to the sensor. The light is emitted in very short pulses, typically in the order of a few nanoseconds [14]. A timer is started when the light exits the source, and stopped at the moment when the light is detected by the sensor [7]. As the speed of light c is constant, the distance of the detected object from where the light reflected back can then be calculated as

$$d = \frac{\tau}{2}\, c, \tag{1}$$

where τ is the time it takes for the emitted light to return back to the sensor [18]. This method works well especially with long-distance measurements, up to 1500 m [7]. However, in order to measure distances in millimeter accuracy, the timer has to have an accuracy in the magnitude of picoseconds [19].

The **CW modulation** method is based on measuring the phase difference between the emitted and received light instead of directly measuring the time it takes for the light to return to the sensor. The working principle of CW modulation is illustrated in Figure 7 [11]. Modulation of the emitted light is created by alternating the current supplied to the light source with a desired waveform, typically either sine or square wave [14].

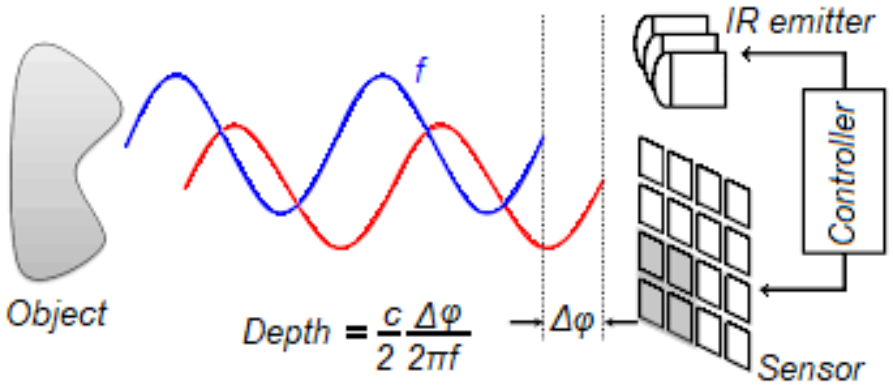

Figure 7. Principle of Continuous Wave modulation, where the phase difference is calculated between the emitted and detected light [11].

In more detail, four control signals with a phase difference of $90°$ from each other are used. Electric charges $Q_1, ..., Q_4$ are measured for each corresponding control signal $C_1, ..., C_4$ as seen in Figure 8 [11].

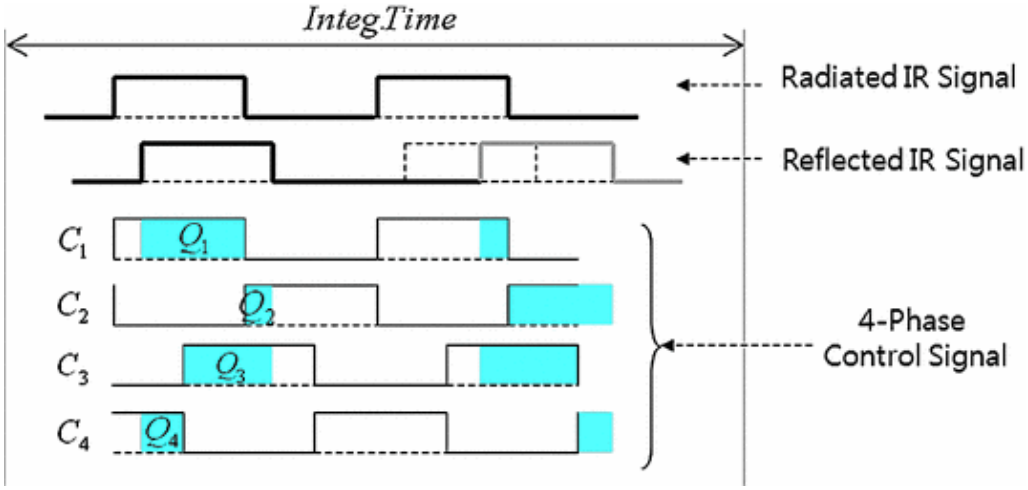

Figure 8. A visualization of the four separate control signals, $C_1, ..., C_4$, and the measured electric charges, $Q_1, ..., Q_4$ (highlighted with cyan), for calculating the phase difference between emitted and detected light [11].

Then, as described in [17, 20], the signal is demodulated using phase shift $\Delta\varphi$ for each pixel, so that

$$\Delta\varphi = \arctan\left(\frac{Q_4 - Q_2}{Q_1 - Q_3}\right). \tag{2}$$

Based on $\Delta\varphi$, depth of each pixel can then be calculated as

$$d = \frac{c}{2f}\frac{\Delta\varphi}{2\pi}, \tag{3}$$

where c is the speed of light and f is the modulation frequency of the emitted light [20].

2.3.2 Error sources and methods of correction

As no technical measurement device can be perfectly accurate, also ToF cameras have their own typical sources of error. These error sources can be roughly divided into two main categories: systematic errors and random errors. This division makes it possible to determine those errors that can be corrected or calibrated in a systematic manner, as well as those that cannot because of their stochastic nature.

The modulation of the emitted light is intended to be of specific waveform, usually sinu-

soidal [12]. However, the emitted light cannot be modulated to be perfectly sinusoidal. Instead, the modulated signal may have irregularities. Because of this modulation error, the depth measurement has a distance-dependent nonlinear offset. This offset, plotted as a function of distance, resembles sine wave and is hence called **circular error** or **wiggling error** [21]. Foix *et al.* [12] illustrated the distance offset with different integration times as shown in Figure 9. Changing the integration time will affect the depth offset. With longer integration times, there are more accumulated photons and therefore the depth estimation has more offset [22]. The HMD used in this thesis has a fixed integration time and therefore the effect of varying integration time is not considered.

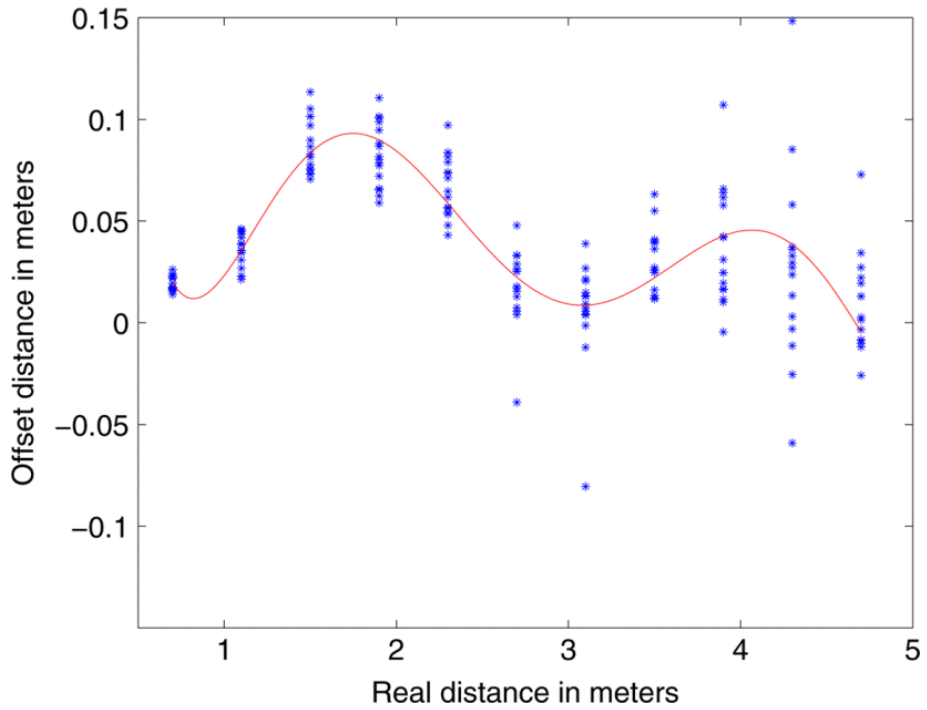

Figure 9. An example of typical depth-dependent circular error with different integration times. The blue dots represent the offset with different integration times and the red line is a fitted 6th degree polynomial function [12].

There are multiple studies about detecting and correcting the circular error of ToF camera depth values. Kahlmann *et al.* proposed using a lookup table for correcting individual depth values with different integration times, as shown in Figure 10 [15]. As the error is sinusoidal by shape, fitting a b-spline [23] or polynomial function [24] to the measured data is a sufficient way to capture the error. It should be noted though, that when a polynomial function is fitted to the data, it is possible for the model to cause undesirable behavior outside of the fitting range [12]. These approaches all need multiple measurements to be performed at varied distances in order to get sufficiently comprehensive fitting data for the correction model. An accurate distance meter is also needed to determine the true

distance, which is used to define the offset of the erroneous distance.

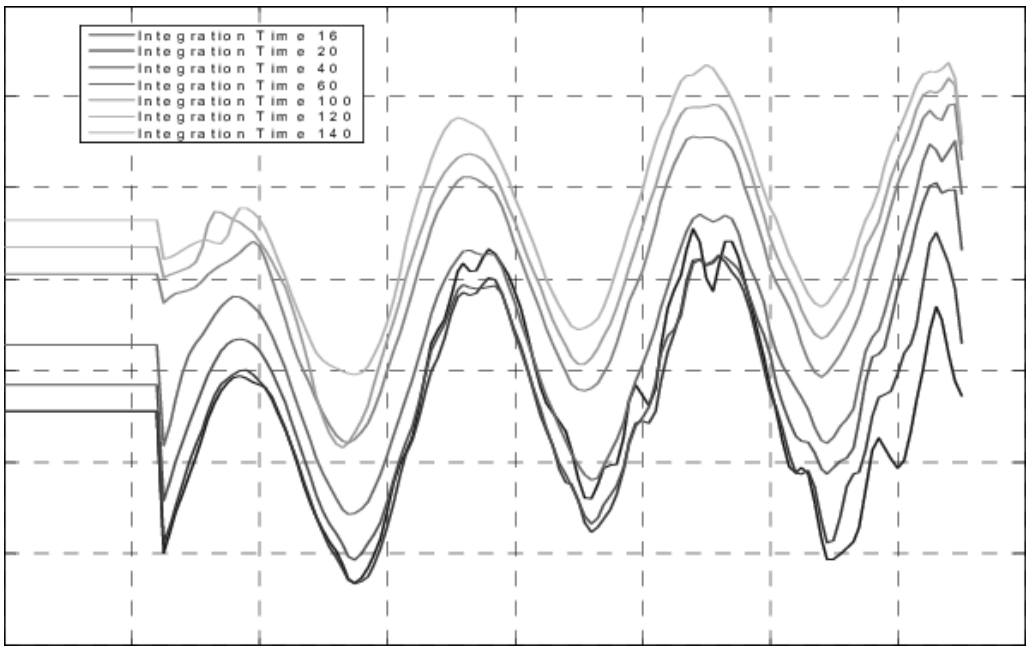

Figure 10. A lookup table for different integration times, proposed by Kahlman *et al*. [15]

Lindner and Kolb proposed that instead of making a global correction model with fitted b-splines, additional pixel information could improve the overall accuracy of the correction [25]. Their study covered both global calibration and additional pixel-wise calibration. The result was that after adding the pixel-wise information, the corrected data was not significantly improved compared to a global correction based model. Also Schiller *et al*. have investigated a method for pixel-wise correction [26]. Their algorithm was simpler and is based on only the measured depth d and pixel position (x, y) as follows:

$$d^* = p_0 + (1.0 + p_1)d + p_2x + p_3y + p_4d^2 + p_5d^3, \tag{4}$$

where d^* is the corrected depth and p_i, $i = 0, ..., 5$ are estimated parameters.

Amplitude-related errors are caused by variation in the intensity of the reflected signal. As the measured depth also depends on the intensity of the detected light, better results are achieved with higher intensities. The amplitude error is larger near the edges of the detector array, as there the intensity of the incoming light is weaker. This is because the light source is not perfectly point-like but, instead, has a non-uniform angular intensity distribution. Due to this, as the distance from the center of the detector gets larger, the points detected are misinterpreted to be further away from the sensor than they actually

are. In addition to that, the depth measurement can also be erroneous in the center of the detector in case the detected object is too close to the sensor and the integration time is too long. In such case, the center-area pixels are oversaturated which causes the determined depth values to be significantly distorted. An illustration of the weaker illumination close to the edges of the amplitude distribution can be seen in Figure 11 [12].

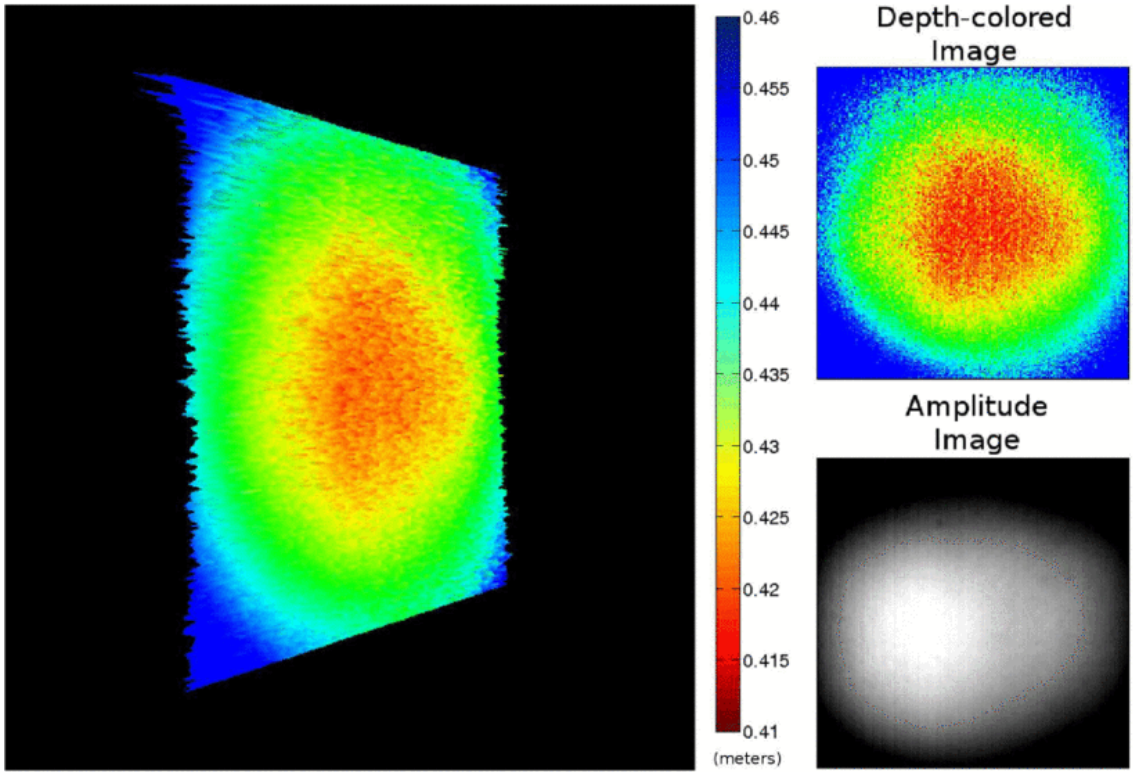

Figure 11. The amplitude image shows that the illumination of the detector is weaker at the edges which causes the depth estimation to be overly large. The image has been taken from a flat wall, at a distance of 0.43 m [12].

Not as many readily available solutions were found for dealing with the amplitude related error as for correcting the circular error. The method used most often is filtering out the pixels with too low an amplitude [27]. However, filtering the pixels using hard-coded limits can lead to most of the data being filtered out and the whole dataset being useless. This is because setting the proper threshold for the amplitude can be challenging, as the amplitude values are highly related to the distance of the detected object. The threshold should hence vary based on the detected distance. Therefore, this method is not useful in the case of this thesis. Guðmundsson *et al.* proposed that the offset caused by the amplitude error is highly correlated with inverse amplitude value [28]. In Figure 12 it is shown that linear regression of the inverse amplitude values and depth measurements are highly correlated. This information can be used to make corrections to the depth values.

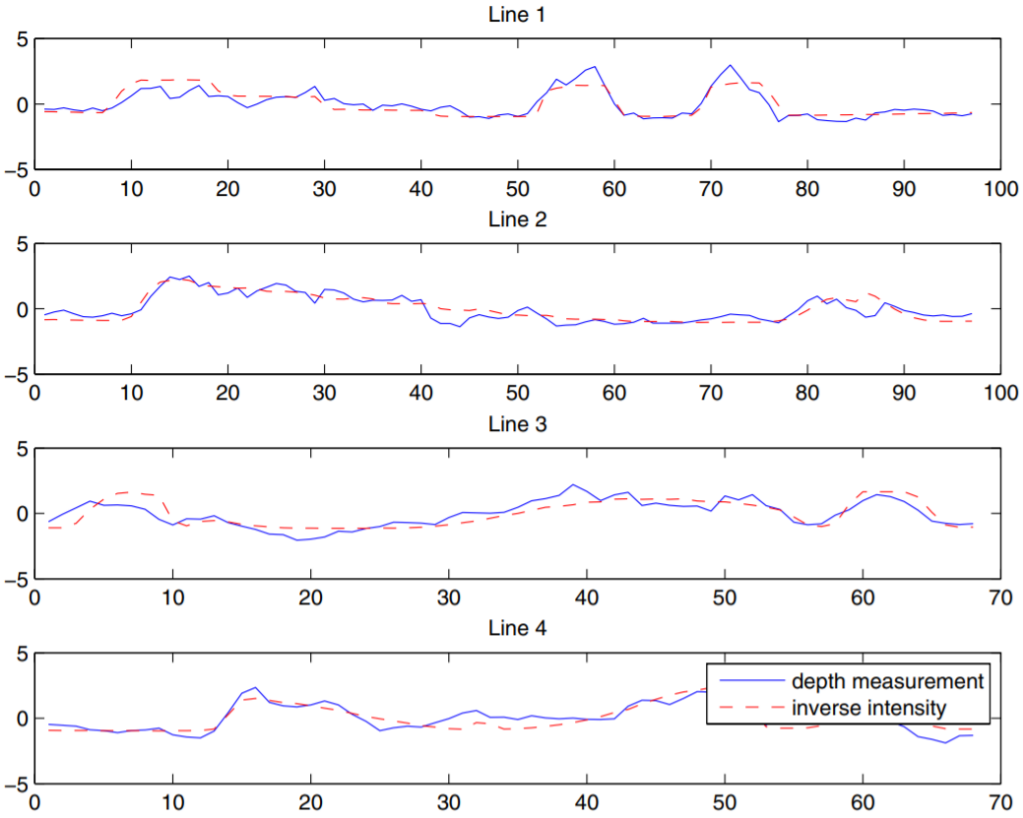

Figure 12. Correlation between depth offset and inverse of amplitude, introduced by Guðmundsson *et al.* in [28]. Both values are standardized in order to compare the values. The standardization is performed as follows: the mean value of all the samples is first subtracted from each of the values, and after that, each value is divided by the standard deviation of all the samples. Lines 1 to 4 represent different areas of a grayscale test pattern. All horizontal axes are represented in pixel coordinates.

Material and color effect to the distance measurement is caused by the same effect as in the amplitude-related errors. As different materials and colors absorb and reflect light differently, the distance estimation can vary based on the material of the detected object [29]. As highly reflective materials reflect more light, the measured amplitude is higher. Respectively, for materials with weak reflectivity near the wavelength of interest, the amplitude of the detected light is lower. Hence, even though the distance of two objects made of different materials would be exactly the same, their distance would be misinterpreted to be different depending on the material of the object. If the observed scene would be full of different materials with different reflectivities, the determined depth values would be incorrect as the depth map would contain areas of discontinuity [11].

Temperature-based errors are common with semiconductor devices. As semiconducting materials have temperature-dependent properties, these errors are mainly caused by fluctuations of temperature in the surrounding environment. Additionally, these errors can be caused by internal self-heating of the device, depending on what kind of a detector

or detector array is used, and how it is electrically connected [15]. The selected HMD is calibrated and intended to be used only in room temperature, so this error source is not relevant considering the scope of this thesis.

Lens distortion is an inevitable problem related to refractive optics used with various kinds of cameras, and ToF cameras are not an exception. It is important that the images captured with **ToF** cameras correspond to the real world as accurately as possible. To further clarify by an example, straight lines in the real world should also be interpreted as straight in the captured images. This is not fully accomplished with cameras, as the optics used for improving the efficiency of light-collecting are not perfect. The lenses used do not guide the incoming light perfectly to the surface of the sensor, and therefore the captured image contains distortions. Most commonly this radial distortion is divided into three categories. In barrel distortion, the lines are bent outward from the center, whereas in pincushion distortion the lines are bent inward from the center (see Figure 13 [30]). The third type of lens distortion, mustache distortion, is a combination of the two above. It creates a pattern resembling a mustache to the image.

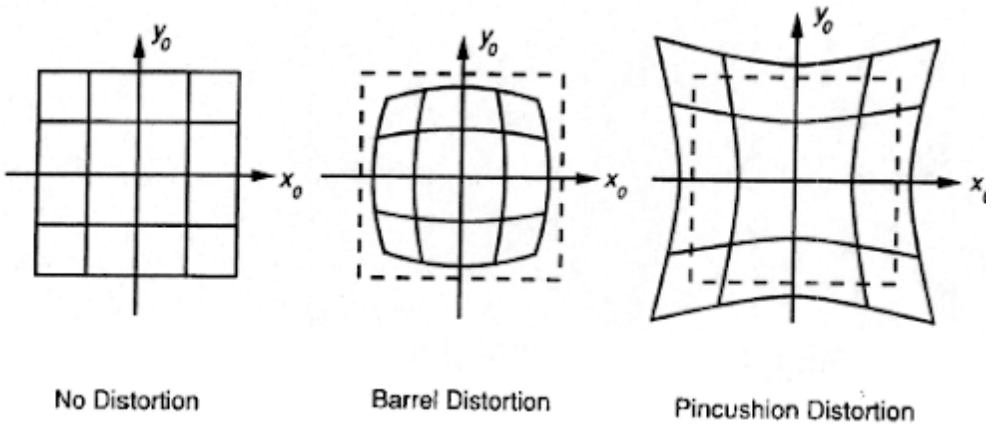

Figure 13. An illustration showing the effect of barrel and pincushion lens distortion [30].

Usually, the amount of distortion is measured using a pattern resembling a checkerboard [21]. A flat plane with colored squares, usually black and white, is placed in front of the camera and an image is acquired. With this kind of image where the features are well-known, it is possible to define the distortion coefficients for each pixel. This error is individual for each device so calibration correction has to be determined separately for each device. This process, known as lens calibration, has been already done for the HMD used in this thesis, and therefore the topic of lens distortion error is not covered further. It is also worth noting that extensive research has already been done on the subject, so it would not be particularly useful to further address the topic in the context of this thesis.

Multipath Interference (MPI) is an error which occurs when light reflects multiple times between separate objects before returning back to the sensor, as shown in Figure 14 [31]. Such an error would be extremely hard, if not impossible to predict, as it is dependent on the arrangement of the objects within the scene [32] but also on the continuously changing point of view of the HMD. The MPI error often occurs near corners - or more generally, in locations where non-parallel surfaces such as the floor, ceiling, walls, stairs and the like connect [33]. As the MPI error is practically non-predictable, correcting it is limited out of the scope of this thesis.

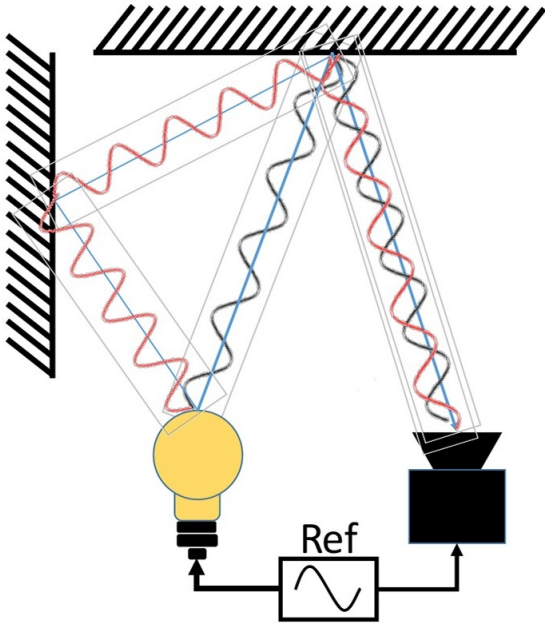

Figure 14. An illustration of how light can be reflected multiple times in the scene before returning back to the sensor [31].

Light scattering, in contrast to MPI, is caused by multiple reflections occurring within the structure of the sensor itself. A small portion of the incoming rays may reflect back and forth between the lens and the detector, causing the detected depth at a pixel to be smaller as it is [12]. This is mainly a problem when objects are located very close to the sensor [11].

When either the ToF camera or an object is moved around the scene, the phase shift between the emitted and detected light changes. This causes an error in the depth calculation and results in a **motion blur** effect in the acquired depth image [34]. Of the studies found on the subject, at least Lee has proposed a method for compensating the motion blur effect in [34]. As the ToF camera or any objects on the scene were not moved in the experiments done as a part of this thesis, this error is not considered further.

3 FLOOR HEIGHT ESTIMATION

In this chapter, methods for plane detection and floor height calculation are first reviewed from the literature. Then, the algorithm currently used for estimating the floor height of the selected HMD is introduced. Lastly, the proposed improvements for the current algorithm are described in detail.

Floor height estimation can be divided into three main steps. Before any calculations, corrections should be done to the acquired raw data in order to minimize the impact of the errors described in section 2.3.2. After implementing the corrections, the floor has to be detected from the corrected and filtered point cloud data. The floor height can then be calculated based on the points that correspond the floor.

3.1 Methods from the literature

Plane detection from point clouds is a very popular problem in current point cloud research. Most common method for simple feature-detecting is the Hough transform, first proposed by Paul Hough in 1962 [35]. Since then, the method has been updated by Duda and Hart into the form it is most widely known nowadays [36]. Based on the work of Duda and Hart, the Hough transform has been extended to detect 3D planes in Hough space [37]. The Hough transform plane detection is based on first declaring the Hough space (θ, ϕ, ρ) and each point of the space relate to one of the planes in $\mathbb{R}^3$. Then, each plane in the Hough space can be defined as:

$$\rho = x\cos(\theta)\sin(\phi) + y\sin(\phi)\sin(\theta) + z\cos(\phi), \tag{5}$$

where ρ is the distance to the origin, θ is the angle of the xy-plane's normal vector, ϕ is the angle between the xy-plane and the normal vector parallel to the z-axis, and (x, y, z) is a point in the original point cloud. First for each point (x, y, z), all the planes (θ, ϕ, ρ) are detected that satisfy the equation (5), and the planes including most points within the point cloud are selected. [38, 39] The Hough transform is very time-consuming, so for time-critical applications, Probabilistic Hough transform - utilizing a random subset of points instead of the whole point cloud - could be used [40]. Other examples of modified methods include Adaptive Probabilistic Hough transform [41] and Randomized Hough transform [42], as well as decreasing the effect of random noise with Progressive Probabilistic Hough transform [43].

Another popular method for plane detection is the Random Sample Consensus (RANSAC) [44]. In this method, the plane is defined by taking three random points and fitting a plane function to them. Then, all points that belong to the defined plane within a chosen threshold are selected. These selected inlier points construct the voting number of the plane. The algorithm loop is stopped after no significant improvement in defining the plane is seen. In addition to the abovementioned Hough transform, also RANSAC has been modified in order to improve the accuracy and reduce the computational time. Problems such as finding a proper threshold for the points to be included to the plane can be reduced by making the decision based on maximizing the likelihood of the points to belong to the plane, instead of just maximizing the number of points included in the plane. This is called the maximum likelihood estimation sample consensus [45].

If the floor is detected as a plane, the floor height can be directly defined from the plane by using the best fit. In case the floor is detected without directly fitting a plane, some data filtering can be applied to the point cloud in order to increase the accuracy of the floor height estimation. Various methods have been proposed for point cloud filtering, including but not limited to growing neural network [46], moving least squares [47] and bilateral filter [48]. These methods are usually used with cases where multiple objects are situated on the observed scene. In such cases, the filtering should be efficient and exact, so that no important information is removed. In case of floor detection, if the points constituting the floor are detected properly and the errors due to the ToF camera are reduced to the extent possible, the remaining point cloud should form a plane with only minor residual noise. As detecting distance of a uniform surface is simpler than detecting objects of different shapes, it could even be efficient and accurate enough to estimate the floor height by simply taking a mean or median value of the error-corrected point cloud.

3.2 The current method

The current floor height estimation algorithm in the selected HMD is based on detecting peaks in the measured point height distribution. As the floor is a horizontal plane with a fixed height, the measured height distribution typically peaks close to the actual height of the floor [49]. It is assumed that the HMD is always located above the floor; hence the points corresponding to the floor are vertically negative. It is also assumed that the eyes of the user are located at least 0.5 m above the floor. The point cloud is first filtered by removing all the points having a height value z larger than -0.5 m. This way the size of the point cloud is reduced and the algorithm can calculate the final floor height estimation faster than if the entire point cloud would be used as an input. These filtered values are

divided into bins, each having a width of 5 cm. From these bins, the algorithm detects peaks and votes for which of them most likely represent the floor. In case the bins next to the detected floor peak include at least 25% of the amount of points that the floor peak does, the adjacent values are also included to the points considered to construct the floor. An example of a point cloud height distribution can be seen in Figure 15. The bars are color-coded to visualize the different steps of the estimation process.

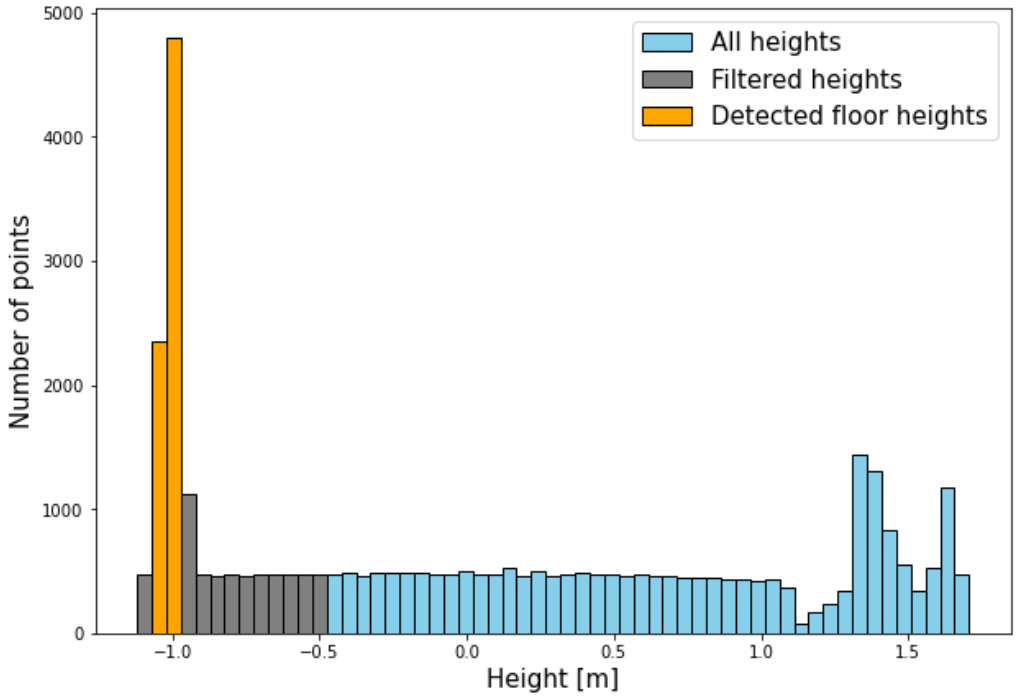

Figure 15. An example showing the floor height distribution divided into bins with a 5 cm width (blue). These values are first filtered to include only the points with an altitude less than -0.5 m (gray). The floor bins are detected from the filtered set of points (orange).

The final estimate is determined by calculating the median of the height values from the points classified as the floor. The current floor height estimation algorithm is presented in Algorithm 1.

Algorithm 1: Current floor height estimation method

Data: **z** = point cloud height values < -0.5 m

Result: estimate = floor height estimate

bins $\leftarrow$ Divide **z** into bins with width of 0.05 m;

allPeaks $\leftarrow$ [ ];

Find the local maxima bins, i.e. the peaks, and store their index;

for *each* $b \in$ *bins* **do**
    if b *is a local maximum* **then**
        allPeaks $\leftarrow$ index of b;
    end
end

Find the index of the bin that is most likely the one that includes floor points;

if *size(allPeaks)* $:= 1$ **then**
    floorIndex $\leftarrow$ allPeaks;
else
    for *each* $p \in$ *allPeaks* **do**
        vote $\leftarrow$ probability of bin *p* to include the floor points;
    end
    floorIndex $\leftarrow$ from allPeaks, select the most voted one;
end

Check if the adjacent bins contain floor points;
$\text{diff}_1 \leftarrow$ size(bin[floorIndex-1]) - 0.25·size(bin[floorIndex]);
$\text{diff}_2 \leftarrow$ size(bin[floorIndex+1]) - 0.25·size(bin[floorIndex]);

if $diff_1 > 0$ *or* $diff_2 > 0$ **then**
    if $diff_1 > diff_2$ **then**
        significantPoints $\leftarrow$ bin[floorIndex] and bin[floorIndex-1];
    else
        significantPoints $\leftarrow$ bin[floorIndex] and bin[floorIndex+1];
    end
else
    significantPoints $\leftarrow$ bin[floorIndex]
end

estimate $\leftarrow$ median(significantPoints);

The selected bin width for the current algorithm is based on the observed standard deviation of the measured depth values, which is estimated to be approximately 25 mm. The bin width is then selected to be two times the standard deviation. Furthermore, the reason for including the adjacent bins to the detected floor data is based on the division of the data into bins. For example, if the true floor height would be 1.30 m and the bin division would be made so that one bin includes points with height ranging from 1.25 m to 1.30 m and the next bin includes points with height ranging between 1.30 m and 1.35 m, then the actual floor points would be divided into two different bins because of the standard deviation. For this reason, it is checked if either of the bins adjacent to the selected main bin include part of the floor points. In practice this is done by comparing the bin sizes. If either of the adjacent bins includes more than the selected amount of points, it can be assumed that the floor points are divided into two bins, in which case both of them are selected to represent the actual floor points. In addition, the amount of points that represent the floor is large. Therefore, median of the values is calculated in order to detect the desired floor height value out of the large sample. The median value is also more robust against outliers than for example the mean value, as all the selected points may not be from the floor.

3.3 Proposed method

The proposed method introduces additional corrections to the point cloud data. The process of error-correcting is divided into two different models. The first correction model reduces the circular error, which is caused by irregularities in the modulated light. The second model corrects the distance error caused by the detector being unevenly illuminated due to the angular intensity distribution of the light source. Both of the errors are described in more detail in section 2.3.2.

For applying the corrections to the point cloud, the distance $|\boldsymbol{p}|_2$ of each point $\boldsymbol{p} = (x, y, z)$ is first calculated as

$$|\boldsymbol{p}|_2 = \sqrt{x^2 + y^2 + z^2}\,. \tag{6}$$

The direction of each point $\boldsymbol{p}$ is defined using unit vector $\boldsymbol{u}$ so that

$$\boldsymbol{u} = \frac{\boldsymbol{p}}{|\boldsymbol{p}|_2} \in \mathbb{S}^2\,, \tag{7}$$

where the $\mathbb{S}^2 \in \mathbb{R}^3$ is the unit sphere.

The distance offset is corrected by first using the circular error correction model f_{circ}. Then, the second correction model, namely amplitude correction f_{amp}, is applied to the data. The corrected distance $d(\boldsymbol{p}, a)$ can then be calculated as

$$d(\boldsymbol{p}, a) = |\boldsymbol{p}|_2 - f_{\text{circ}}(|\boldsymbol{p}|_2) - f_{\text{amp}}(a) \ , \tag{8}$$

where $|\boldsymbol{p}|_2$ is the measured distance and a is the measured amplitude for point $\boldsymbol{p}$. Each point can then be corrected using equations (7) and (8), assuming that the direction $\boldsymbol{u}$ of the corrected point $\boldsymbol{p}_{\text{corr}}$ is same as the direction of the original point $\boldsymbol{p}$. This yields:

$$\begin{aligned} \boldsymbol{p}_{\text{corr}} &= d(\boldsymbol{p}, a) \cdot \boldsymbol{u} \\ \boldsymbol{p}_{\text{corr}} &= \boldsymbol{p} - [f_{\text{circ}}(|\boldsymbol{p}|_2) + f_{\text{amp}}(a)] \cdot \boldsymbol{u} \ . \end{aligned} \tag{9}$$

3.3.1 Circular error correction model

The circular error is dependent on the measurement distance. Because of this, multiple measurements at different distances have to be performed in order to correct the offset created by the error. As stated in Section 2.1, the optimal detecting range of the ToF camera in the selected HMD is 0.4 m - 5 m. For this reason, measurements for defining the offset model were done at distances of 0.2 m to 4.1 m with steps of 0.05 m. The true distance for each case was measured using a laser distance meter. For each measurement, an area of 0.3 radians from the lens center was extracted and used to define the measured distance. These same measurements were done similarly for three individual devices, resulting in multiple repetitive measurements for each of the distances. The dataset used for defining the correction model was $(d_{\text{i}}, d_{\text{i}}^{\text{off}})_{\text{i}=1}^{N_1}$, where $N_1 = 834940$. The final offset model was defined by fitting a spline $f_{\text{circ}} : \mathbb{R}_+ \to \mathbb{R}$ to these measurements and is shown in Figure 16 as a function of distance. The fitting was done by using cubic splines and 20 knots. More details for fitting splines can be found, for example, from [50]. In addition, marginalized standard deviation σ of the model was calculated as

$$\sigma(d) = \sqrt{\frac{\sum\limits_{|d-d_{\text{i}}|<\epsilon} (d_{\text{i}}^{\text{off}} - f_{\text{circ}}(d))^2}{N(d) - 1}} \ , \tag{10}$$

where d_{i} is the measured distance, $d_{\text{i}}^{\text{off}}$ is the offset of measured distance, $f_{\text{circ}}(d)$ is the function value, d is the true distance, $N(d) = \#\{i \leq N : |d - d_{\text{i}}| < \epsilon\}$ is the number of

measurements and ϵ is the width used for each standard deviation calculation.

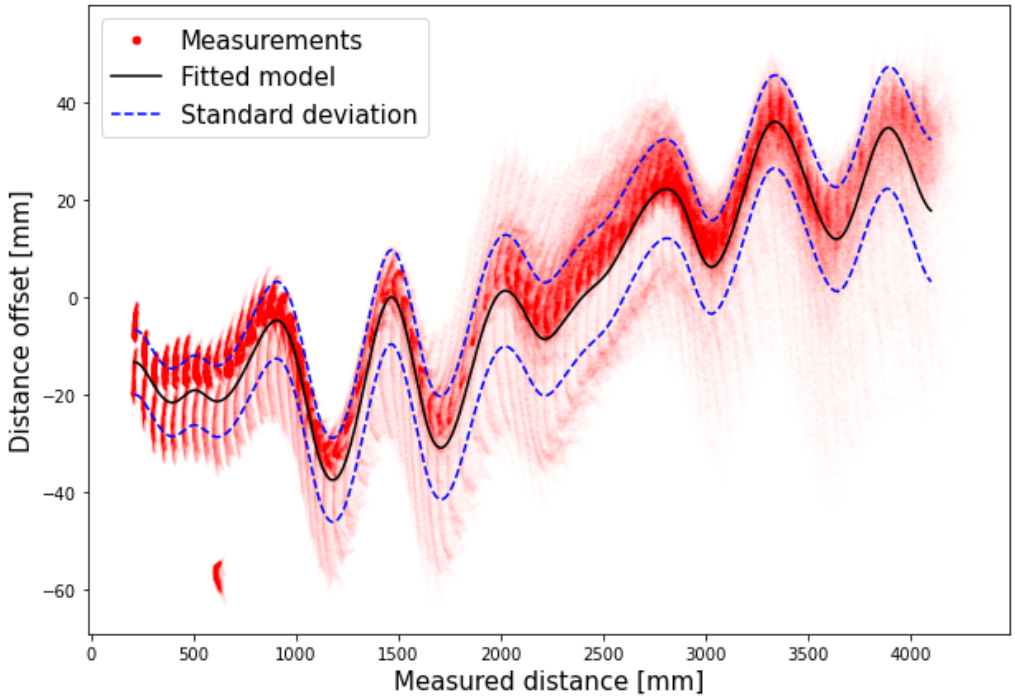

Figure 16. Median sensor offset as a function of distance. The median offset was determined by repeating the measurements using three individual devices. Marginalized standard deviation was calculated for each measurement distance and is shown in the figure as blue dashed lines.

As expected, the resulting correction model still includes the wiggling effect. The effect of the circular error can be reduced using Equation (8). This correction model is hereafter referred to as the Circular correction.

3.3.2 Amplitude error correction model

The amplitude error is assumed to be independent of the circular error, and therefore the effect of the circular error is first removed from the data. Since the material of the measured surface can affect the amplitude of the detected light, the floor material used in defining the correction model for the amplitude error is the same as in the dataset used to evaluate the performance of the model. After correcting the circular error from the data, the relation between the amplitude and the remaining distance error can be evaluated. Guðundson *et al.* proposed that the inverse of amplitude correlates with the distance offset [28]. By plotting the remaining error against the inverse of amplitude, the correlation can be determined. This effect is visualized in Figure 17.

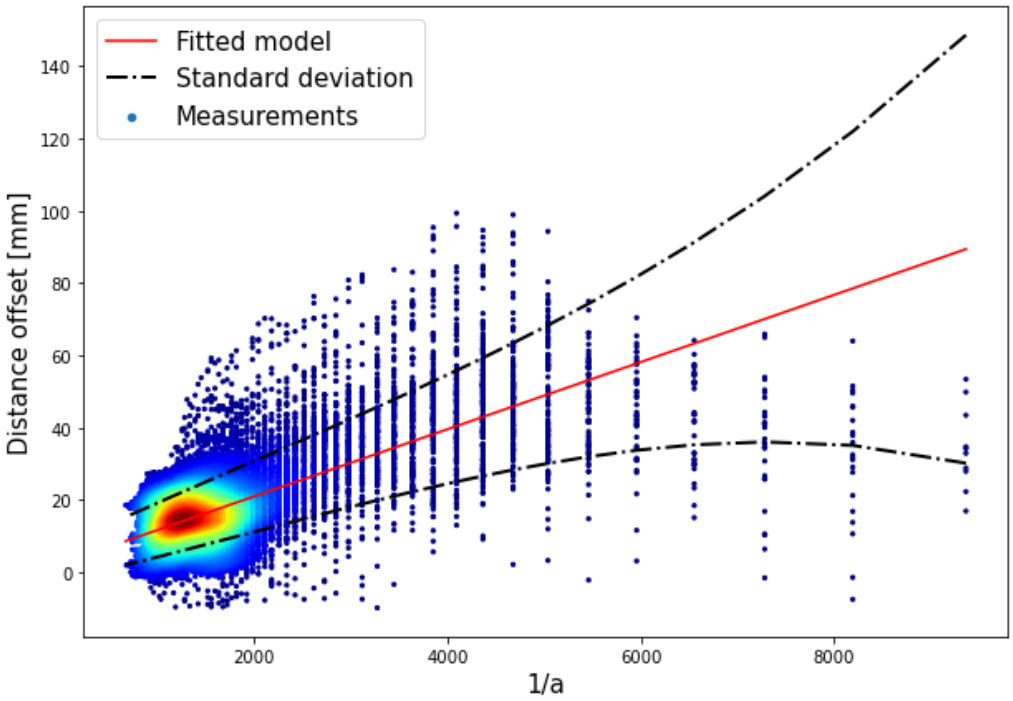

Figure 17. Correlation between the distance offset and the inverse of amplitude after removing the circular error. In order to better see how the points are distributed, the relative density of the points is colored. The fitted regression line (red) shows that the values are correlated. Marginalized standard deviation was calculated for each measured amplitude value and this is shown in the figure as black dash-dotted lines.

The gathered dataset $(d_{\mathrm{i}}^{\mathrm{off}}, a_{\mathrm{i}})_{\mathrm{i}=1}^{N_2}$, where $N_2 = 18607$, is depicted in Figure 17. The function f_{amp} for correcting the offset distance, dependent on the amplitude, can then be defined as

$$f_{\mathrm{amp}}(a) = d_0 + \frac{k}{a}\,, \tag{11}$$

where a is the arbitrary amplitude value of a pixel. Parameters d_0 and k are defined by minimizing the squared error E between the values from the function $f_{\mathrm{amp}}(a)$ and the measurement distance offset d_{off}, as

$$E(d_0, k) = \sum_{\mathrm{i}=1}^{N_2} |f_{\mathrm{amp}}(a_{\mathrm{i}}) - d_{\mathrm{i}}^{\mathrm{off}}|^2\,. \tag{12}$$

The new point coordinates are then defined using equations (8) and (9). Also, the standard deviation for the model is calculated analogously to (10) around f_{amp}. This correction model is hereafter referred to as the Amplitude correction.

4 EXPERIMENTS

In this chapter, the experimental setups used for data acquisition are introduced. Selection criteria for the data used for evaluating the performance of the proposed correction methods are described. The selected dataset is then reviewed in more detail. Lastly, results of the experiments are presented.

4.1 Test setups

Experimental data was collected in two different ways in order to create variation of floor materials and the ground truth sources. The first test setup was based on a custom-built mechanical jig with wheels, and the second test setup on a computer-controlled robotic arm manufactured by ABB.

In order to enable reliable evaluation of the proposed floor height estimation method, it is very important to acquire highly accurate reference ground truth values against which to compare the performed estimations. The ground truth values were measured in three different ways of which the most accurate was chosen to be used as the reference. The first method for measuring the ground truth was based on the use of SteamVR lighthouses. Different amounts of lighthouses were placed inside the room in which the HMD was used. The SteamVR lighthouses recorded positional and rotational information of the HMD while it was used. The Second method of ground truth determination was based on measuring the vertical location of the ToF camera using a measuring tape. In the third and last method the ground truth was obtained by recording the internally determined positional and rotational information of the ABB robotic arm.

The ToF camera's actual height from the floor does not directly correspond to the heights recorded using the SteamVR and the ABB robot, as they record the information in relation to a separately defined tracking point. The difference of the ToF camera location and the tracking point location is illustrated in Figure 18.

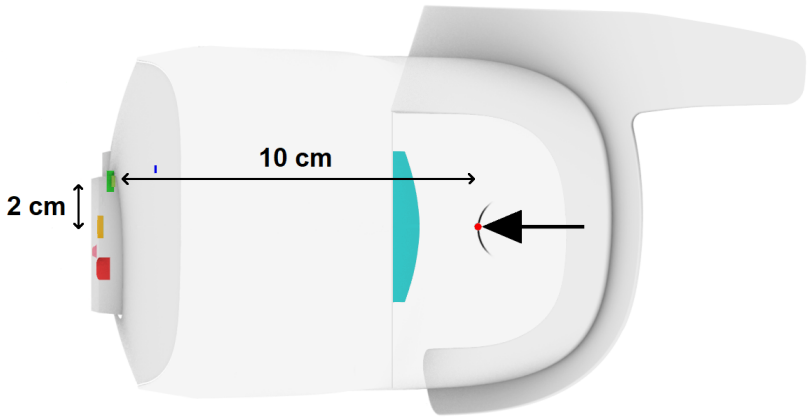

Figure 18. The relation between the locations of the ToF camera (green dot on the front panel) and the tracking point (indicated by the single-headed arrow). The tracking point is located approximately in between the eyes of the user and the ToF camera is located 10 cm in front of and 2 cm above it.

All values of the resulted datasets were recorded in relation to the tracking point. The manually measured ToF camera heights were also translated to correspond to the tracking point height. The manually measured heights were only used with the datasets recorded with the jig setup - therefore the roll angle does not need to be taken into consideration.

When the pitch angle of the HMD is $0°$, the ToF camera height h_{ToF} is measured. The tracking point is located below the sensor, as shown in Figure 19. The tracking point height from the floor at the $0°$ angle can be calculated as

$$h_1 = h_{\text{ToF}} - d_{\text{TT}}, \tag{13}$$

where d_{TT} is the height difference between the ToF camera and the tracking point.

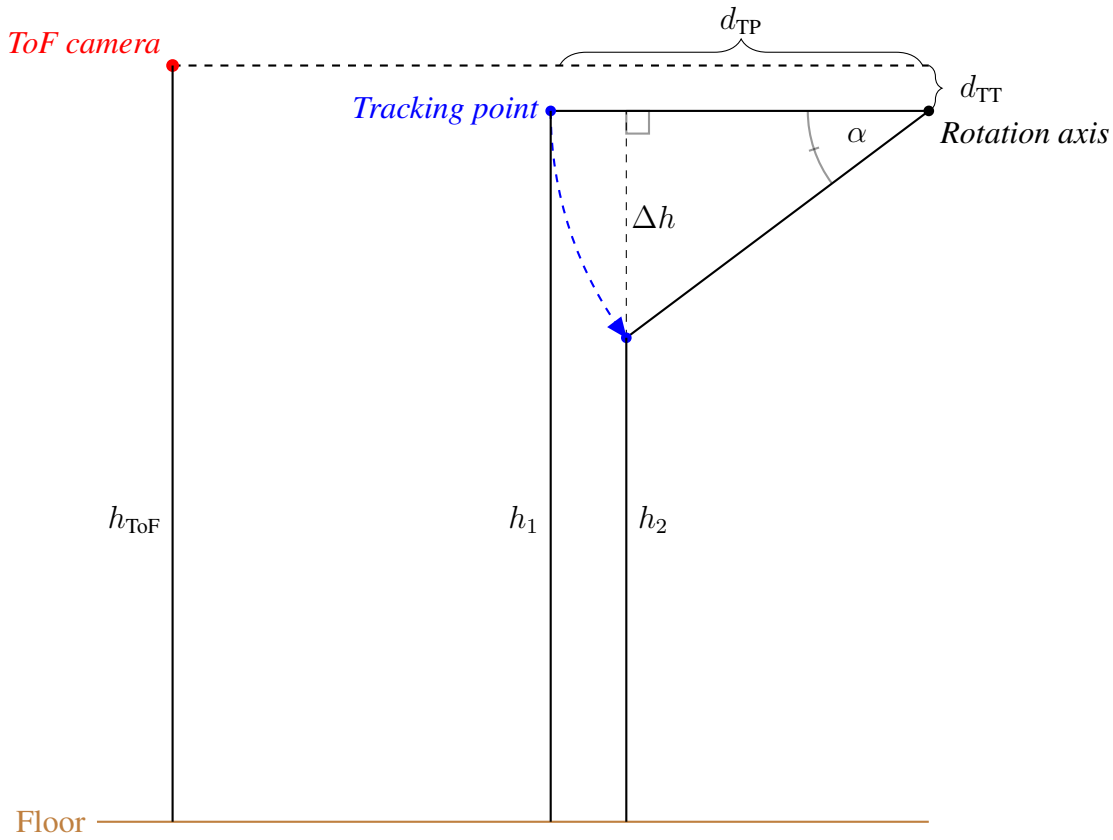

Figure 19. An illustration showing the locations and the vertical relation of the ToF camera and the tracking point, as well as their distance from the floor. The effect of rotation angle α on the height of the tracking point is also displayed.

By changing the angle from $0°$ to α, the tracking point height in Figure 19 changes. The change Δh can be calculated as

$$\Delta h = \sin(\alpha) \cdot d_{\text{TP}} \, , \tag{14}$$

where d_{TP} is the distance from the tracking point to the center of the rotation axis and α is the angular change. Then, with equations (13) and (14), the height of the tracking point at any angle α can be calculated as

$$h_2 = h_1 - \Delta h \, . \tag{15}$$

4.1.1 Portable jig

A portable mechanical jig was built for recording data with the HMD. Figure 20 represents the built jig and demonstrates that the angle and height of the HMD can be adjusted. Multiple datasets were collected using pitch angles of 0, -30, -60 and -90 degrees, at heights of 1 and 1.5 m. The zero-degree angle is defined as the angle where the HMD is parallel in relation to the floor, whereas the angle of -90 degrees is defined as the angle where the ToF camera is aligned perpendicular to the floor.

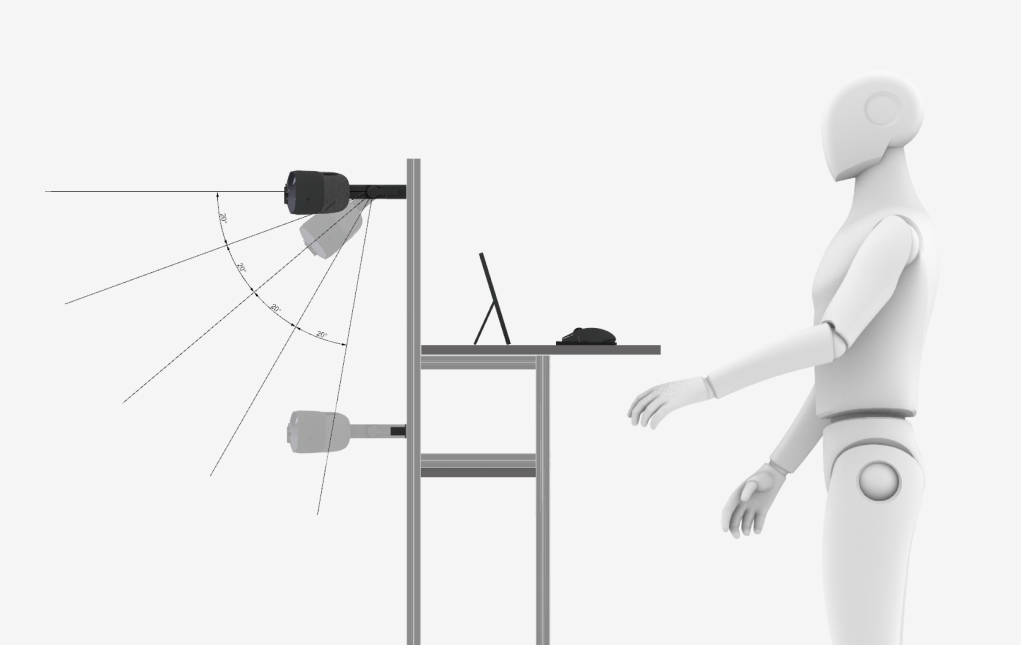

Figure 20. A schematic picturing the jig being used for positioning the HMD into desired known angles and heights.

The jig could be transferred to different locations, which enabled recording data at multiple rooms with different kind of floor and wall materials. The locations used for gathering the experimental data were:

1. Atrium; white, highly reflective, concrete floor.
2. Office; gray, weakly reflective, fitted carpet.
3. Green room; green fabric covering both the floor and the walls.

4.1.2 ABB robotic arm

The second data collection campaign was performed using an ABB robotic arm located in the robot room of the Varjo headquarters. The robot room has a black concrete floor and black walls. The HMD was attached to the end of the robotic arm, enabling moving the HMD into different heights and pitch & roll angles. This made it possible to create more variation to the orientation of the headset. The robot room, including the black surfaces and the robotic arm, can be seen in Figure 21. In the case of the robotic arm measurements, the heights used were 1.5 m and 1.1 m, and the pitch angles were 0, -30, -45, -60 and -90 degrees. Roll angles of -90, -45, 0 and 90 degrees were used, respectively. The robotic arm was held immobile during the recordings.

Figure 21. The selected HMD attached to a robotic arm manufactured by ABB. In addition, the black-painted surfaces of the robot room can be seen in the background.

The third data collection campaign was carried out using the robotic arm again, but this

time so that the black floor of the robot room was covered using a green fabric, as seen in Figure 22. The fabric was of same material and colour as in the green room recordings with the jig, and therefore the reflectivity of the material had already been found to be sufficient.

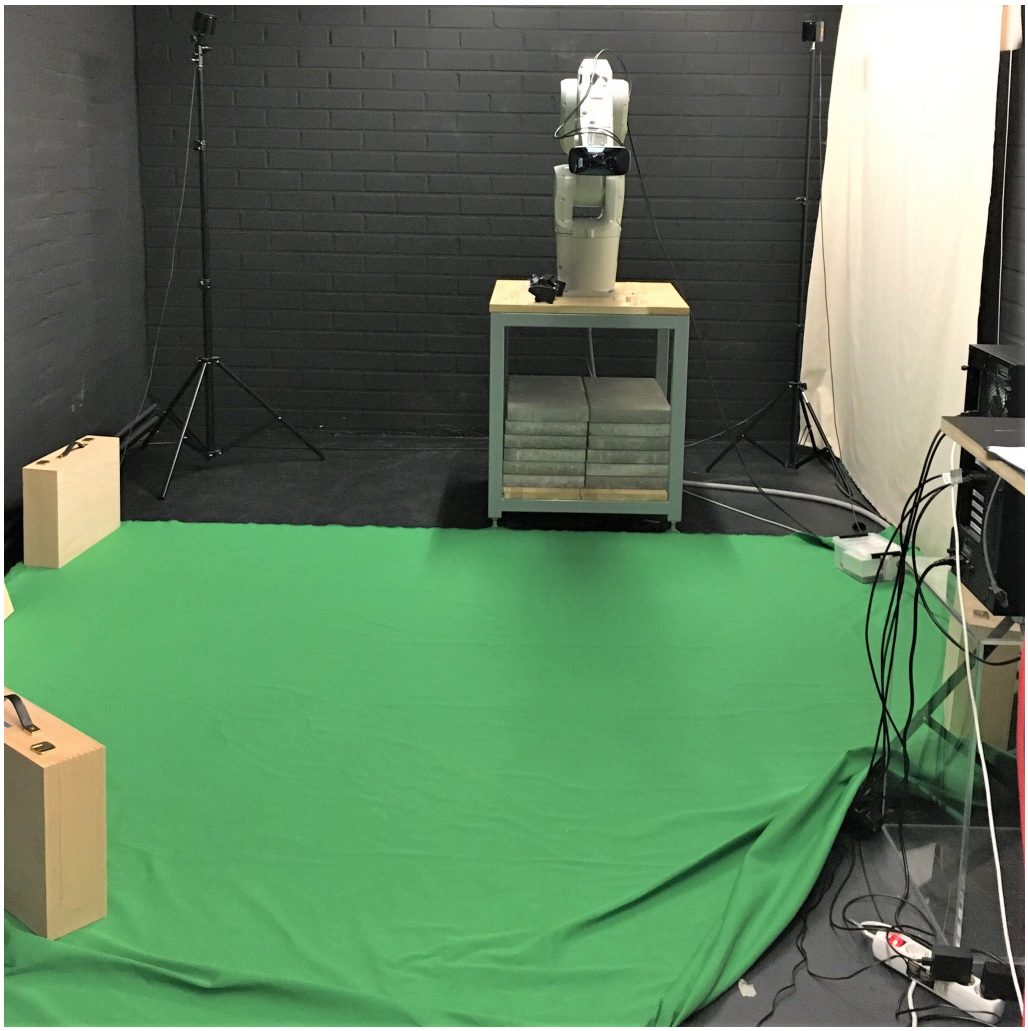

Figure 22. The third data collection setup in the robot room. A green fabric was placed on the floor.

The third recordings were made at heights of 1.25 m to 1.55 m at intervals of 5 cm, using pitch angles of -30, -60, and -90 degrees and roll angles of -45, 0, and 90 degrees. The robotic arm was again held immobile during each of the recordings.

4.1.3 Selection of data for algorithm evaluation

The accuracy of the SteamVR ground truth turned out to be insufficient for its intended use as the ground truth recorded using the SteamVR was not as accurate and precise as initially assumed. The manually measured ground truth values had to be discarded for the same reason. The variation of the ground truth values recorded at a single reference height are presented in Figure 23. The variation of the ground truth values is approximately 10 cm.

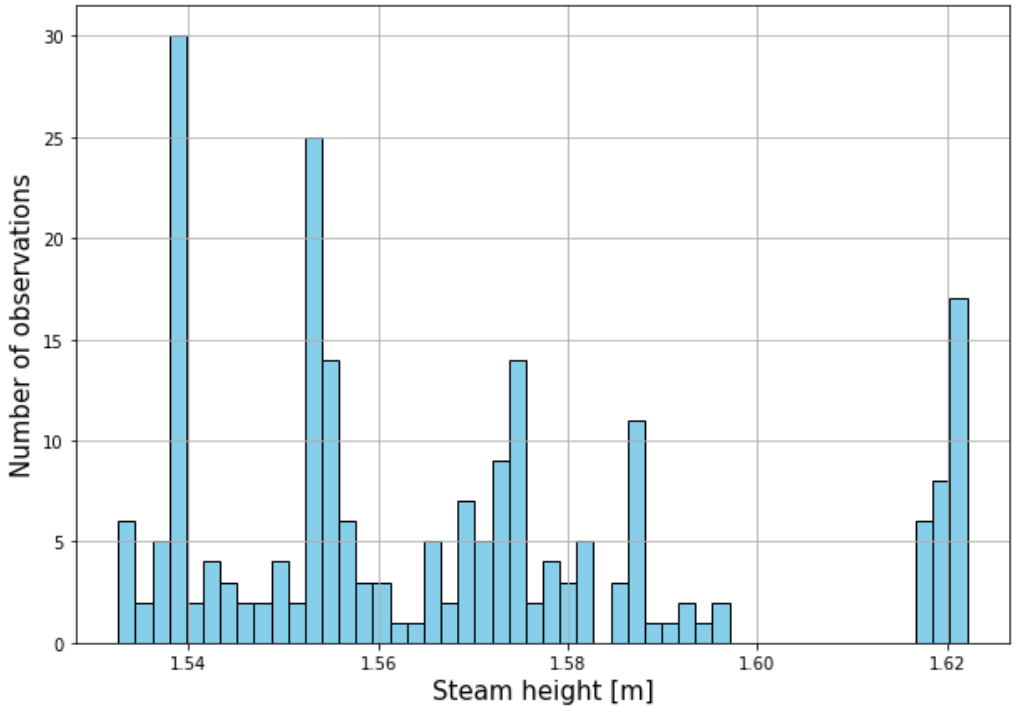

Figure 23. Exemplary ground truth values recorded using the SteamVR. All of the datasets had the same reference height, yet the heights provided by the SteamVR were significantly inconsistent.

One advantage of recording data using the robotic arm as a mechanical support for the HMD was, that two different sources could be used for determining the ground truth. The internal coordinate tracking system of the robotic arm was used in addition to the SteamVR. The robotic arm is placed on top of a table, which was also defined as the origin point for the height of the robot. The table height was added to the internally defined height value of the robot, so that the resulted height values were the same as the tracking point height from the floor. These ground truth values at a selected height are presented in Figure 24. There were not as many datasets with the same height and angle settings as there were with jig, but the three different recordings had ground truth values less than 5 mm difference from each other.

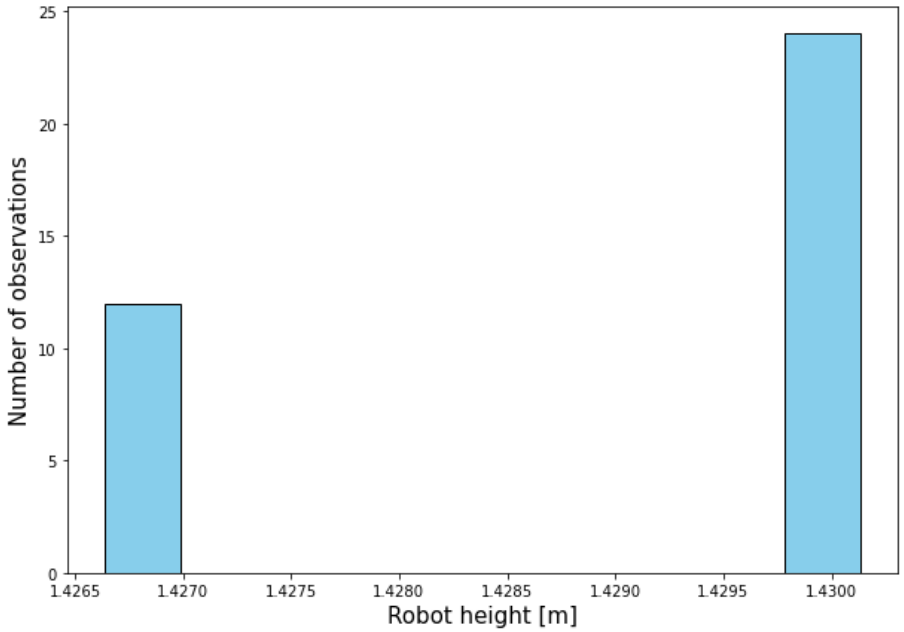

Figure 24. Ground truth values from the ABB robotic arm. All three different recordings had the same reference height and the resulted ground truth values had less than 5 mm variation.

The datasets recorded with the jig included three different floor materials with different reflectivities. Therefore, the height estimations were not the same between the recorded datasets. The difference between the performed floor height estimations was approximately 2.5 cm. The effect of floor materials to the floor height estimate can be seen in Figure 25.

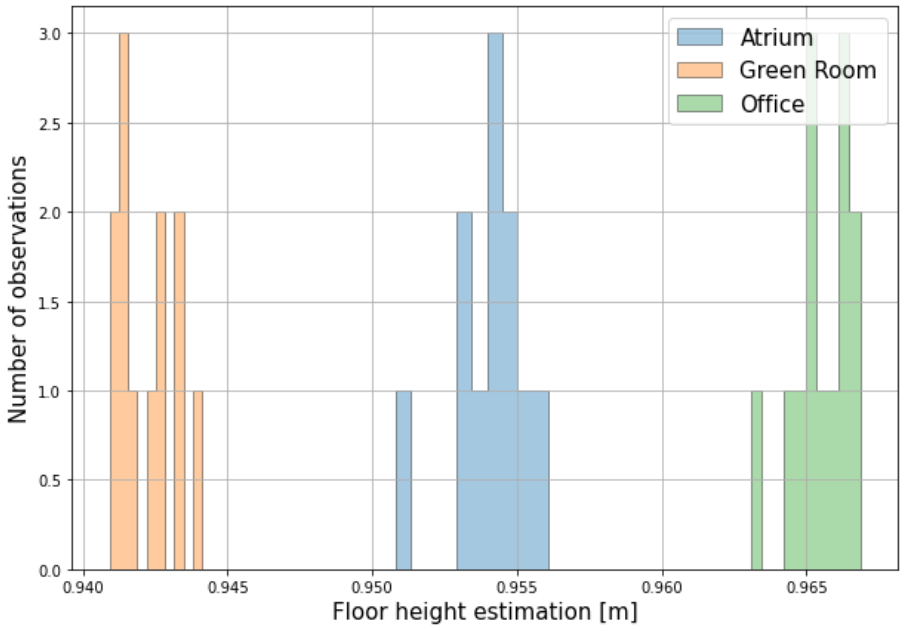

Figure 25. Floor height estimations for different floor surface materials. All three datasets had the same ground truth of 0.948 m. The estimates significantly vary depending on the floor material.

The reflectivity of the floor material in the robot room turned out to be too low for acquiring feasible data. The low reflectivity resulted in the measurement data being very noisy. The effect of the low reflectivity to the data is shown in Figure 26.

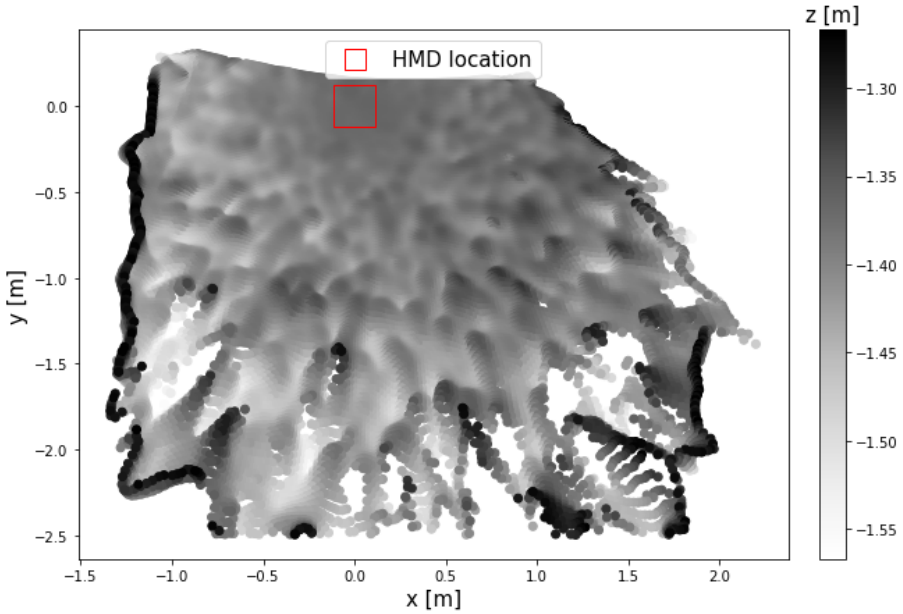

Figure 26. Floor point cloud data from the robot room with a matte black floor. The HMD is located approximately at the origin of the plot. The height values, presented as a grayscale gradient, are corrupted by excessive noise.

The datasets recorded with the green fabric covering the floor of the robot room were significantly less noisy, as shown in Figure 27.

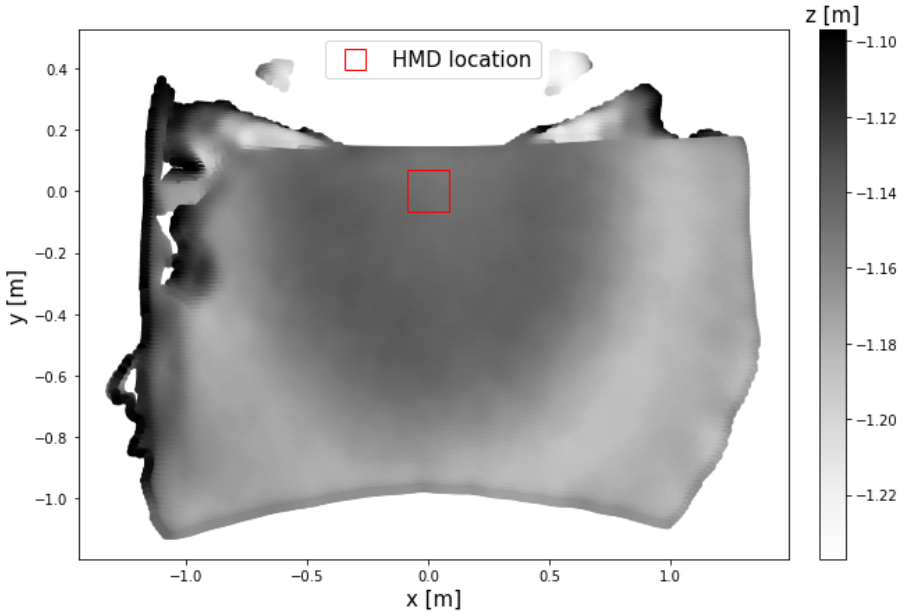

Figure 27. Floor point cloud data from the robot room with the green fabric covering the floor. The HMD is again located approximately at the origin of the plot. The height values vary less compared to the height values in Figure 26.

For the reasons mentioned above, the first and second datasets had to be discarded. The dataset selected to be used for evaluating the performance of the floor height estimation algorithm was the third one, as it was of good quality regarding both the acquired, data and the ground truth.

4.2 An overview of the selected data

The final dataset was recorded in the robot room using the ABB robotic arm as the mechanical support, and green fabric as the floor material. All the other floor materials were excluded to eliminate the effect of varying characteristics of different materials. In addition, the SteamVR lighthouses were not used for determining the ground truth.

A total of 63 datasets were recorded. All data recordings lasted 15 seconds, resulting in a total of 12 usable data frames. Each frame includes a 3D (x, y, z) point cloud that represents the observed scene. An example point cloud at 1.3 m, 90° pitch and 0° roll is shown in Figure 28. The point cloud includes points representing the floor as well as points representing the table, on top which the robotic arm is located. The HMD is directed perpendicular to the floor and thus none of the walls nor the ceiling is within the FOV of the ToF camera.

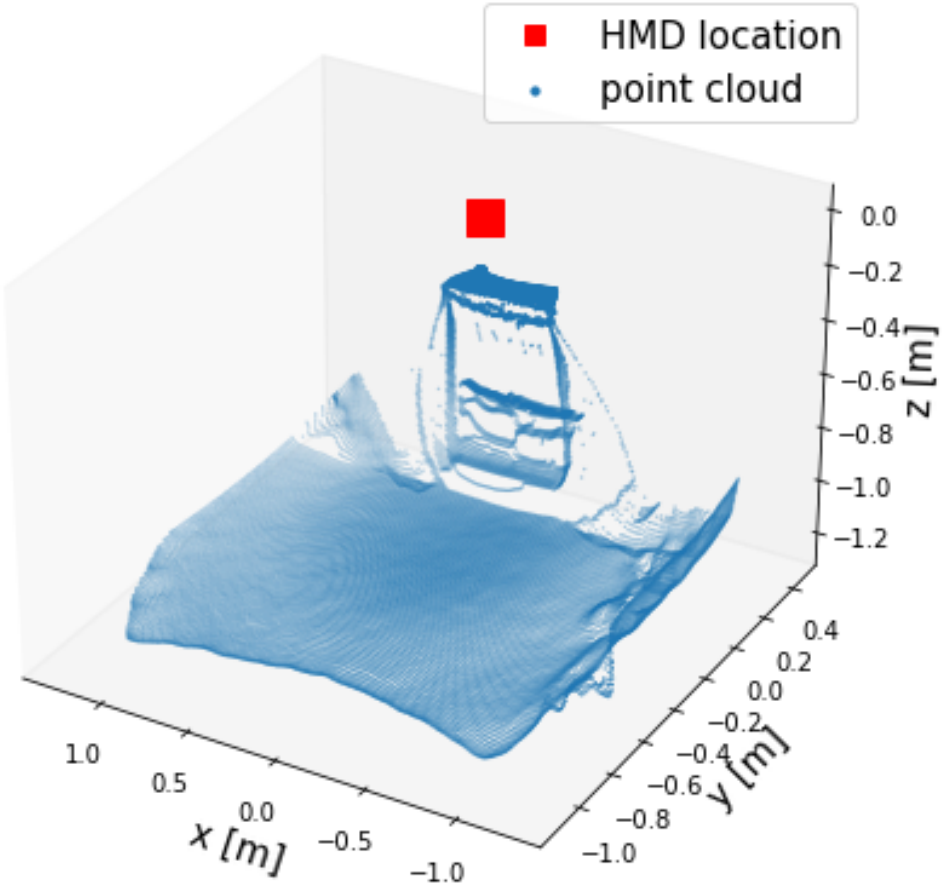

Figure 28. An example of a point cloud acquired using the ABB robotic arm for supporting the HMD at a height of 1.3 m, a pitch angle of -90° and a roll angle of 0°. The HMD is located approximately at the origin of the figure.

In addition to the 3D coordinates, each point of the point cloud also includes depth and amplitude measurements obtained from the ToF camera. Each recording also includes relative robot coordinate data, where the z-value corresponds to the vertical distance from the table to the tracking point of the HMD. However, the desired ground truth value should be the vertical distance from the floor to the tracking point - this is achieved by adding the distance between the floor and the surface of the table to the robot's z-value.

4.3 Results

The raw data of the ToF camera is transformed into 3D point clouds, where x and y values represent the floor plane and the z-value corresponds to the floor height. The detected floor can then be pictured on an xy-plane where the z-value, i.e. the height of the points, is presented as a color gradient. With this kind of a plot it is possible to demonstrate how

evenly flat the floor is according to the data. Figure 29 shows an example of the floor data before making any corrections to the raw data. The floor data is not smooth, but instead has differences depending on the distance from the xy-origin, where the ToF camera is located. The effect caused by the circular error is clearly visible as rings of different colors around the location of the HMD.

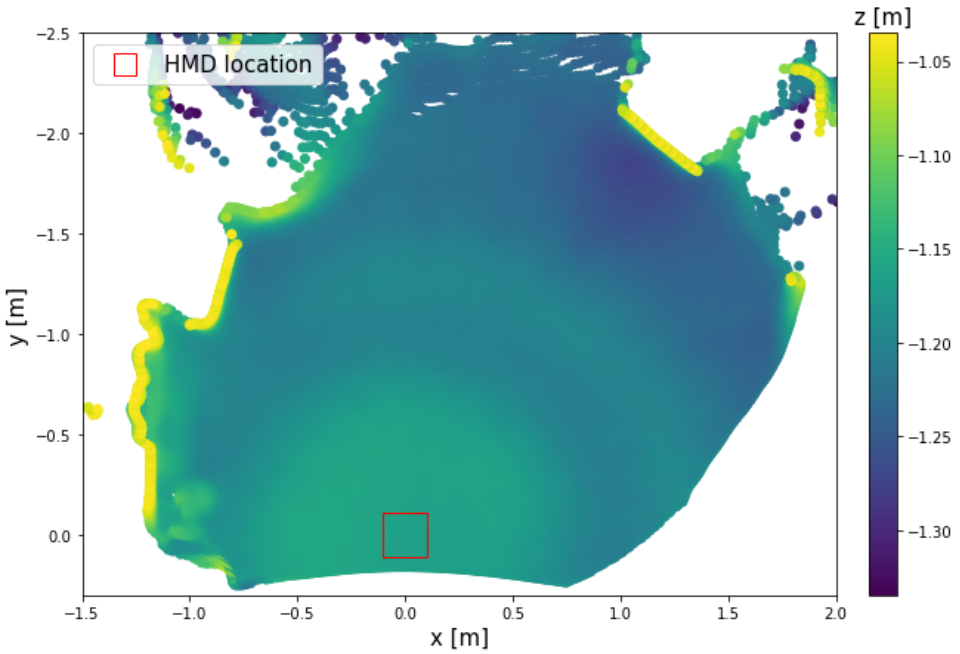

Figure 29. A scatter plot of the floor heights presented on an xy-plane. The ground truth, 1.185 m, is located at the center of the color gradient scale. The HMD is located approximately at the origin of the plot. The roll angle is 0° and the pitch angle is -60°. The effect of the circular error is seen as ring-like patterns on the raw floor data.

After applying the Circular correction to the raw data, the ring-like pattern fade away, as seen in Figure 30. Even though there is no visible error pattern in the floor plane plot anymore, the values between the center and the edges are still inconsistent. It should also be noted that after the Circular correction, the floor height is shifted further away from the ground truth.

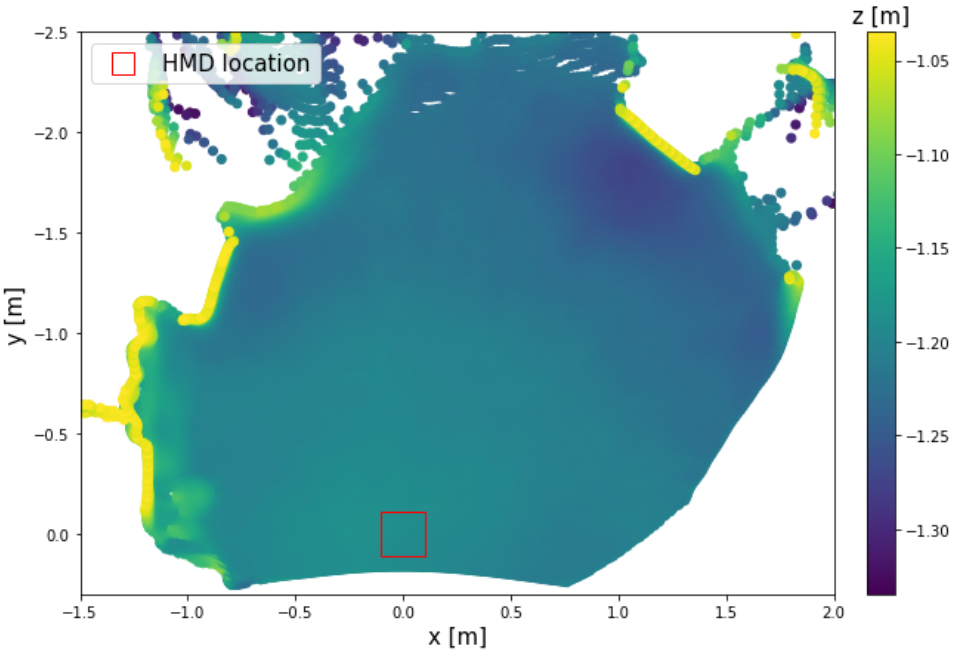

Figure 30. The floor data of Figure 29 after applying the Circular correction. The effect of the circular error is clearly reduced and the floor appears to be smoother, as it should be. Still some minor difference can be seen between the origin and the areas close to the edges of the plot.

Then, after applying the Amplitude correction, the height difference between the floor height data and the ground truth is further reduced. The floor data is much smoother and the height values are very close to the ground truth over an area larger than before, as can be seen in Figure 31.

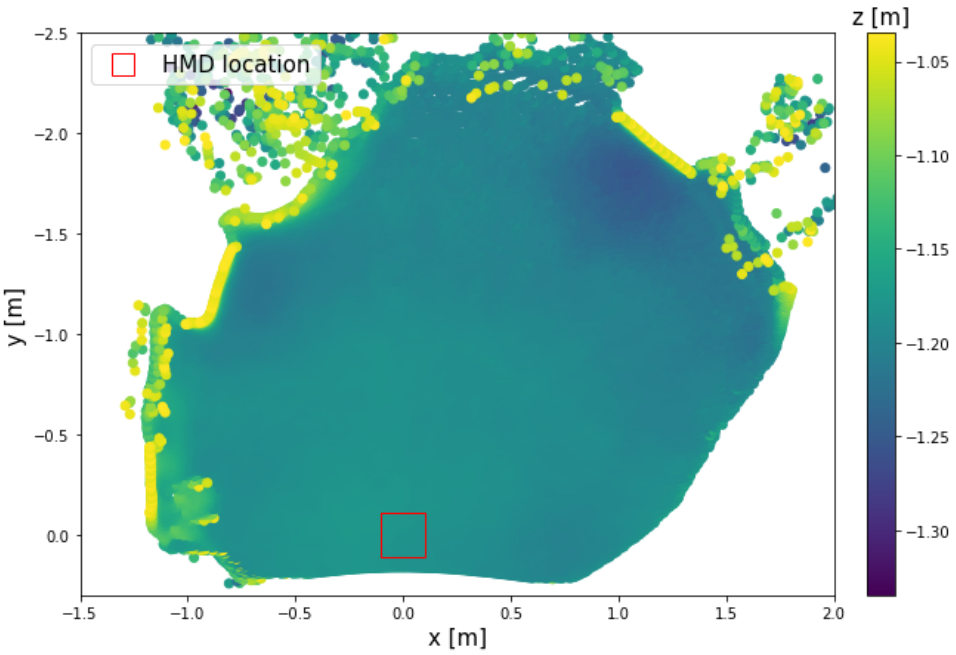

Figure 31. The floor data of Figure 29 after applying both the Circular and the Amplitude correction. Height difference between the center and the areas closer to the edges is clearly reduced. Overall, the floor height data is very close to the ground truth.

In addition to viewing the floor data as a whole, two straight line-like sections were extracted from the floor data, as demonstrated in Figure 32.

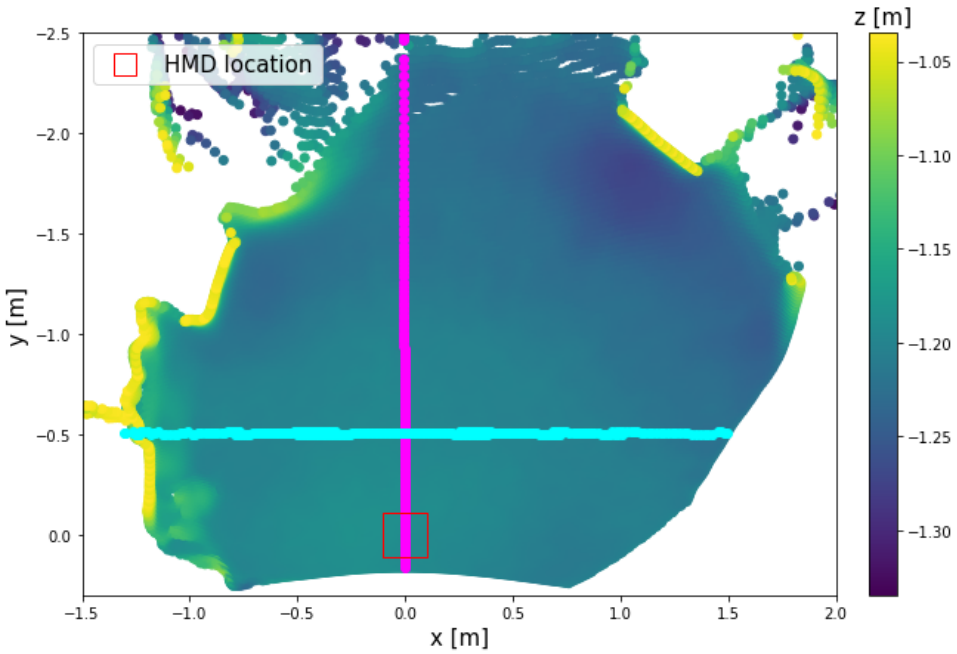

Figure 32. Two perpendicular line profiles, highlighted with magenta and cyan, were extracted from the floor data. The extracted data is used in figures 33 (magenta) and 34 (cyan) to demonstrate how the offset of the floor height changes after each of the applied corrections.

By plotting the height offset of the extracted data points as a function of their y-dimension value, it is possible to concurrently compare the effect of each correction. Such a comparison is provided in Figure 33.

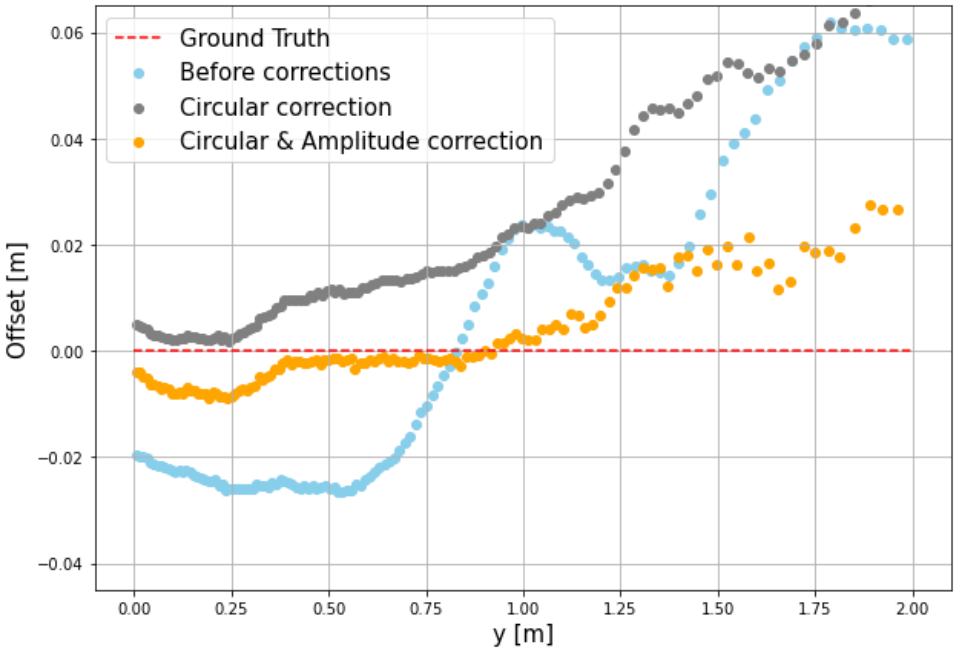

Figure 33. Offset of the floor height before and after each correction. The presented data is extracted parallel to the y-axis (highlighted with color magenta in Figure 32).

The same comparison was repeated in the direction of the x-axis. As seen in Figure 34, after applying the Circular correction, the data has more offset on the positive side of the x-axis. After applying the Amplitude correction, the difference between the negative and positive side of the x-axis is reduced.

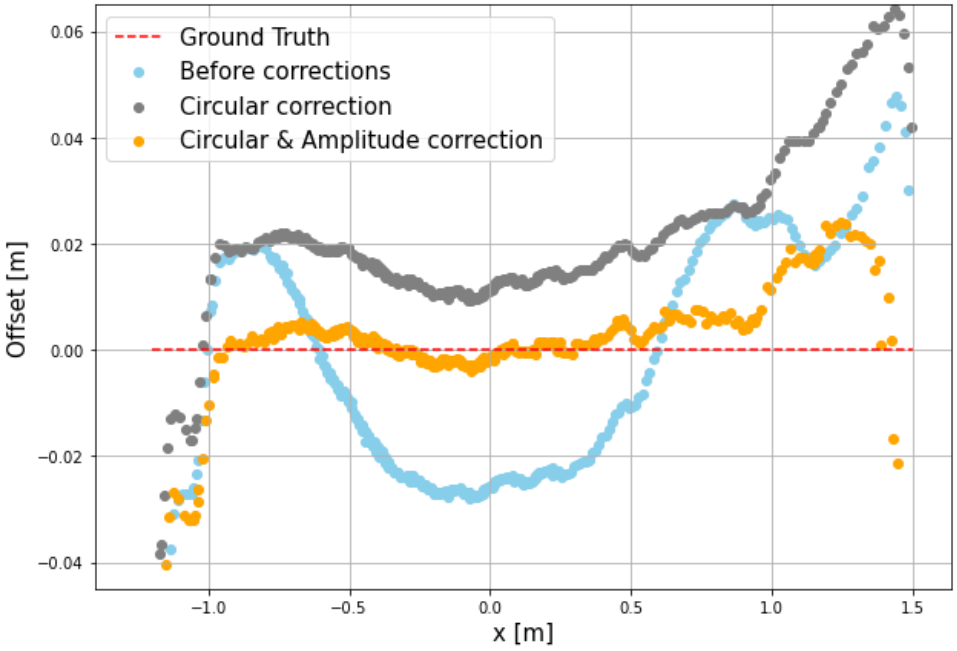

Figure 34. Offset of the floor height before and after each correction. The presented data is extracted parallel to the x-axis (highlighted with color cyan in Figure 32).

Additionally, the proportion of corrected points with a height value within ± 10 mm of the ground truth was examined. The amount of points within the chosen threshold is shown in Figure 35.

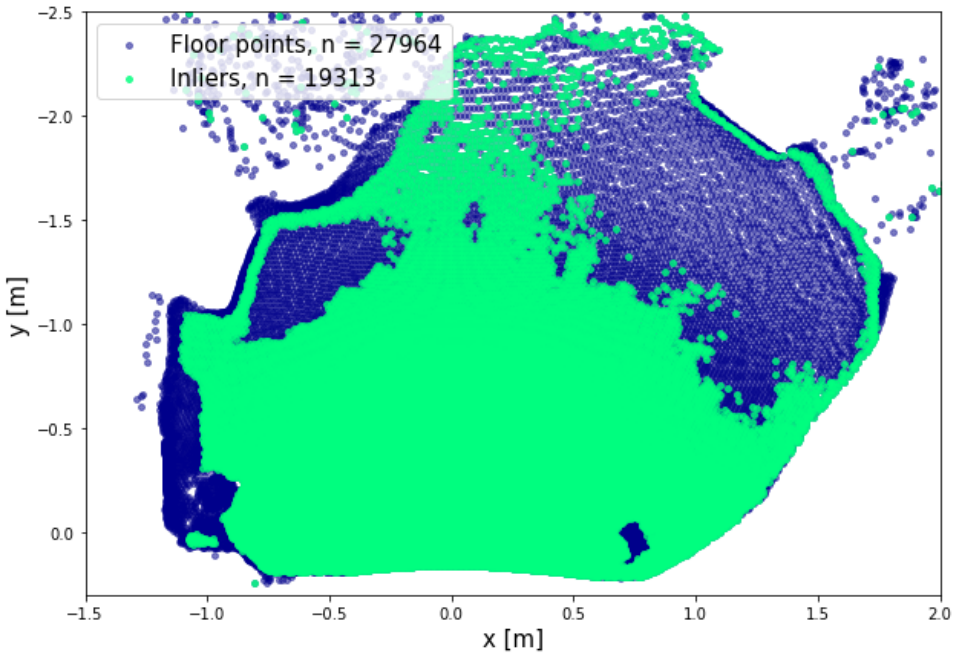

Figure 35. The points within the selected threshold of 10 mm from the ground truth are presented in green.

The Circular and Amplitude corrections were applied to all of the data frames of every dataset, and a floor height estimate was calculated before and after each correction, resulting in 644 estimations. These three estimates were compared to the ground truth value of the ABB robot, and the absolute error of each estimate was calculated. The absolute errors are shown as histogram distributions in Figure 36.

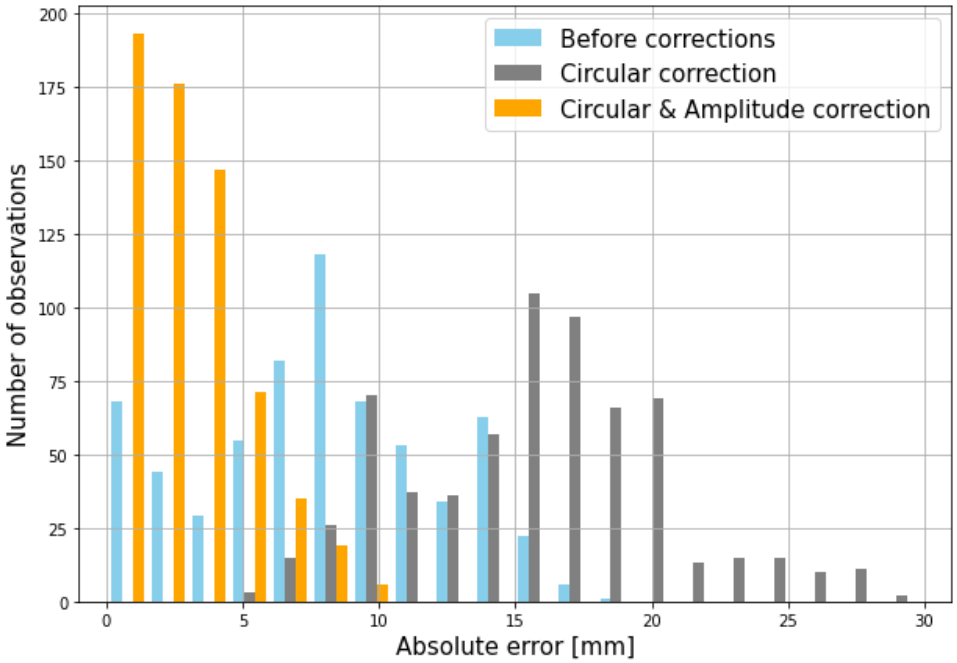

Figure 36. Absolute errors in height estimation before and after the different corrections. A total of 644 estimations were performed.

In addition, the minimum and maximum error and the standard deviation of the distributions was calculated. The results can be seen in Table 2.

Table 2. The lower & upper limit and the standard deviation of the absolute estimation error (in millimeters). The absolute estimation error is significantly reduced after applying both corrections. A total of 644 estimations were performed.

	Before corrections	**After the Circular correction**	**After the Circular and the Amplitude correction**
MIN [mm]	0.037	5.832	0.024
MAX [mm]	18.473	29.917	9.407
SD [mm]	4.358	4.712	2.102

5 DISCUSSION

In this chapter, the results presented in Chapter 4.3 are further analyzed. In addition, propositions for further research are given.

5.1 Current study

The two correction models proposed in this thesis address the two most significant, hitherto uncorrected, sources of systematic error degrading the ToF camera-based depth measurements of the Varjo XR-3 HMD. Other major errors related to the ToF camera of the selected HMD, such as the lens distortion and errors based on temperature variation had already been corrected through calibration. The two proposed correction models, namely the Circular correction and the Amplitude correction, were successfully applied to the raw data acquired with the ToF camera of the HMD. The dataset used in this thesis included only one floor material in order to eliminate the effect of different materials and their unequal reflectivity. Also, only a single HMD, including a single ToF camera, was used for testing the performance of the proposed algorithm. The reference ground truth was acquired using the ABB robotic arm. Ground truth values acquired with the SteamVR were discarded as their accuracy was worse. The used ground truth contained only a small amount of variation; less than 5 mm between the datasets. The slight variation detected is likely the result of the robotic arm being at different position or translation, as the location of the robotic arm is not perfectly accurate. However, this ground truth value was considered to be accurate enough in order for it to be used for evaluating the quality of the developed floor height estimation algorithm.

The Circular correction successfully eliminated the circular error from the raw data. However, the floor height estimates calculated based on the Circular-corrected data were found to be larger than the floor height estimates calculated based on the raw data. It was found out, that the height offset of the raw data is typically distributed on both sides of the ground truth, but after the Circular correction is applied, all the values shift above the ground truth. As the floor height estimate is defined by calculating the median of the selected altitude values, the estimates tend to be larger after the Circular correction has been applied.

Both the remaining error and the offset resulting from the Circular correction were significantly reduced after applying the Amplitude correction to the data. After both of the corrections, the height values converged closer to the ground truth. The floor height estimates were significantly improved comparing to the estimates made after only the Cir-

cular correction had been applied to the data, but also comparing to the estimates based on the uncorrected raw data. The estimates calculated based on the raw data resulted in an absolute maximum error of 18.5 mm, while for the estimates calculated after applying the Circular correction the absolute maximum error was higher; close to 30 mm. After applying both of the corrections, the absolute maximum error was reduced to less than 9.5 mm, which is approximately 50% of the original absolute maximum error. Hence, by correcting the selected errors with the proposed correction methods, more accurate floor height estimates were achieved.

It should be noted, that the floor height data also contained areas where the floor height was overestimated, even after the corrections had been applied to the data. These areas became visible after inspecting how large portion of the corrected floor data is within an error interval of ± 10 mm of the ground truth. The areas with an overestimated floor height are not within the selected threshold. One possible explanation for this kind of an error is the MPI, described in Chapter 2.3.2. This could be due to the various kind of boxes that were used to straighten the green fabric that was spread on the floor of the robot room. These boxes could have created a possibility for the emitted light to be reflected from the floor to the sides of the boxes (or vice versa) and then back to the sensor, causing local incorrect depth measurements. The amount of points believed to be corrupted by the MPI error was relatively small compared to the total amount of points used to estimate the floor height. Therefore, it can be assumed that the total effect of this particular error to the floor height estimate is also relatively small. Furthermore, the inliers within the selected threshold covered a large area of the floor points used, which further indicates that the used corrections made a desired effect to a large amount of the data points within each point cloud.

The floor data did not converge perfectly to the ground truth values, but instead included residual errors and noise. The reason behind the residual error can be that the error correction models were formed from different measurements and applied by fitting the model to measured data. It is hence likely that the models are not ideal for all of the datasets used.

5.2 Future work

This thesis solely focused on one floor material. In future, the implementation of the proposed methods could be extended to cover different kinds of surface materials. For example, the Amplitude correction model could be generalized so that it could be used anywhere, regardless of the floor material. Alternatively, individual models could be created for some of the most common floor materials, and information from the visible

wavelength range video cameras of the HMD could be used for determining the floor material type. Various machine learning algorithms could then be considered for utilizing the information acquired from the cameras to identify the specific floor material. This way it could be possible to automatically select the most suitable correction model based on the detected material. The model can also be expanded to detect distances from walls and ceilings, as well as other planar surfaces such as tables.

The Amplitude correction proposed in this thesis was defined using fewer samples than the Circular correction. Therefore, the regression line fitted to the measurements could likely be improved by making more measurements at different distances. Also, defining a different amplitude correction model for a selection of different distances could improve the accuracy of the floor height estimation.

The two largest sources of error excluded from the scope of this study were motion blur and MPI. The effect of MPI was visible in the results of this study, but the areas affected by the error were so small that the error was irrelevant to the outcome. However, both motion blur and MPI should be studied more closely, as in actual use the device is typically in continuous motion and quite likely in a space including a large number of obstacles and walls nearby. In small spaces, MPI can affect a larger portion of the floor, degrading the accuracy of the floor height estimate. To address the error caused by motion blur, an additional correction model could be implemented using data acquired while moving the HMD in a space whose exact dimensions are known. As for the MPI, attempts could be made to reduce the error caused by the effect, for example, by using the Sparse Reflections Analysis or L1 optimization.

Additionally, it might make sense to consider reviewing and, if needed, redesigning the floor height estimation algorithm, as the current algorithm is specifically designed to work for noisy, uncorrected point cloud data. Hence, by changing or improving the floor height estimation algorithm, the accuracy of the estimates could potentially be further improved. Also, the current algorithm uses only one data frame to calculate the estimate. Combining information from multiple data frames would make it possible to filter out the lowest quality data, which could result in more accurate estimates.

The estimates were evaluated by comparing them to the ground truth values acquired from the ABB robotic arm. However, the measured ground truth certainly includes errors and does not fully correspond to the actual floor height. For this reason, use of higher quality instruments for ground truth determination should be considered. The actual height could be measured e.g. with a laser distance meter or some other highly accurate distance measurement device. The success of the floor height estimation could also be evaluated from a user experience perspective in a staged yet realistic use case, by testing how the

perceived error of the floor height estimate corresponds to the actual error of the estimate at different settings. After all, a positive user experience is of paramount importance, and even a device producing a fairly inaccurate floor height estimate can produce a strong sense of immersion, for the floor height is only one of the several factors that make up the overall experience.

6 CONCLUSION

The objective of this thesis was to improve the floor height estimation algorithm used in the Varjo XR-3 HMD. The objective was pursued by first studying the different error sources related to ToF cameras, and then developing two separate correction models for correcting the two most significant systematic sources of error found. The designed correction models were experimentally evaluated.

The general specifications of the HMD used in this thesis were introduced. Additionally, the most essential subsystems from the point of view of the performed study were discussed in more detail, and their principles of operation were presented. The different kind of error sources associated with ToF cameras were particularly carefully addressed, as it was assumed that correcting them could most significantly improve the performance of the floor height estimation algorithm.

Several test setups utilizing different means of mechanical support and ground truth determination were constructed for experimentally evaluating the performance of the improved algorithm. Out of the different setups built, the one working best was chosen. The setup chosen was the one that utilized the ABB robotic arm both as the mechanical support for the HMD and the source of ground truth.

The performance of the two correction models created was defined by applying them to experimentally acquired raw point cloud data produced by the integrated ToF camera of the HMD. According to the results, the unfavorable effect of the selected error sources was successfully minimized and the accuracy of the floor height estimation algorithm was improved approximately 50%.

REFERENCES

[1] Ivan Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, pages 757–764. Association for Computing Machinery, 1968.

[2] Frank Steinicke. The science and fiction of the ultimate display. In *Being Really Virtual: Immersive Natives and the Future of Virtual Reality*, pages 19–32. Springer, 2016.

[3] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, E77-D(12):1321–1329, 1994.

[4] Christopher Andrews, Michael Southworth, Jennifer Silva, and Jonathan Silva. Extended reality in medical practice. *Current Treatment Options in Cardiovascular Medicine*, 21(4):18, 2019.

[5] Varjo Technologies Oy. Varjo & boeing: A new era in astronaut training using virtual reality. https://varjo.com/boeing-starliner/, 2021. [Online; accessed June, 3, 2021].

[6] Varjo Technologies Oy. Varjo XR-3 Product Page. https://varjo.com/products/xr-3/, 2021. [Online; accessed May, 29, 2021].

[7] Dario Piatti, Fabio Remondino, and David Stoppa. State-of-the-art of TOF range-imaging sensors. In *TOF Range-Imaging Cameras*, pages 1–9. Springer, 2013.

[8] Melexis NV. Time-of-flight basics. [Application Note], 2017.

[9] Filiberto Chiabrando, Roberto Chiabrando, Dario Piatti, and Fulvio Rinaudo. Sensors for 3D imaging: Metric evaluation and calibration of a CCD/CMOS time-of-flight camera. *Sensors*, 9(12):10080–10096, 2009.

[10] Bernhard Buttgen and Peter Se. Robust optical time-of-flight range imaging based on smart pixel structures. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 55(6):1512–1525, 2008.

[11] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Horaud. *Time of Flight Cameras: Principles, Methods, and Applications*. Springer, 2012.

[12] Sergi Foix, Guillem Alenya, and Carme Torras. Lock-in time-of-flight (ToF) cameras: A survey. *IEEE Sensors Journal*, 11(9):1917–1926, 2011.

[13] Thorsten Ringbeck. A 3D time of flight camera for object detection. In *Optical 3-D Measurement Techniques*, ETH Zürich, Germany, 2007.

[14] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménier. An overview of depth cameras and range scanners based on time-of-flight technologies. *Machine Vision and Applications*, 27(7):1005–1020, 2016.

[15] Timo Kahlmann, Fabio Remondino, and Hilmar Ingensand. Calibration for increased accuracy of the range imaging camera SwissRanger. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVI Part 5, pages 136 – 141. ISPRS, 2005.

[16] Andreas Kolb, Erhardt Barth, Reinhard Koch, and Rasmus Larsen. Time-of-flight sensors in computer graphics. In *Eurographics 2009 - State of the Art Reports*, pages 119–134. Eurographics Association, 2009.

[17] Stefan May, David Droeschel, Dirk Holz, Stefan Fuchs, Ezio Malis, Andreas Nüchter, and Joachim Hertzberg. Three-dimensional mapping with time-of-flight cameras. *Journal of Field Robotics*, 26(11–12):934–965, 2009.

[18] Andrea Corti, Silvio Giancola, Giacomo Mainetti, and Remo Sala. A metrological characterization of the Kinect V2 time-of-flight camera. *Robotics and Autonomous Systems*, 75(Part B):584–594, 2016.

[19] Larry Li of Texas Instruments Inc. SLOA190B: Time-of-flight camera - an introduction. [Technical White Paper], 2014.

[20] Sobers Francis, Sreenatha Anavatti, Matthew Garratt, and Hyunbgo Shim. A ToF-camera as a 3D vision sensor for autonomous mobile robotics. *International Journal of Advanced Robotic Systems*, 12(11):156, 2015.

[21] Marvin Lindner, Ingo Schiller, Andreas Kolb, and Reinhard Koch. Time-of-flight sensor calibration for accurate range sensing. *Computer Vision and Image Understanding*, 114(12):1318–1328, 2010.

[22] Jiejie Zhu, Liang Wang, Ruigang Yang, James Davis, and Zhigeng Pan. Reliability fusion of time-of-flight depth and stereo geometry for high quality depth maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1400–1414, 2011.

[23] Marvin Lindner, Andreas Kolb, and Thorsten Ringbeck. New insights into the calibration of tof-sensors. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–5. IEEE, 2008.

[24] Stefan Fuchs and Gerd Hirzinger. Extrinsic and depth calibration of tof-cameras. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2008.

[25] Marvin Lindner and Andreas Kolb. Lateral and depth calibration of PMD-distance sensors. In *International Symposium on Visual Computing*, volume 2006, pages 524–533, 2006.

[26] Ingo Schiller, Christian Beder, and Reinhard Koch. Calibration of a PMD-camera using a planar calibration pattern together with a multi-camera setup. In *Proceedings of ISPRS conference*, 2008.

[27] Stefan Fuchs and Stefan May. Calibration and registration for precise surface reconstruction with time-of-flight cameras. *International Journal of Intelligent Systems Technologies and Applications*, 5(3):274–284, 2008.

[28] Sigurjón Guðmundsson, Henrik Aanæs, and Rasmus Larsen. Environmental effects on measurement uncertainties of time-of-flight cameras. In *International Symposium on Signals, Circuits and Systems*, pages 1–4. IEEE, 2007.

[29] Yu He and Shengyong Chen. Recent advances in 3D data acquisition and processing by time-of-flight camera. *IEEE Access*, 7:12495–12510, 2019.

[30] Junhee Park, Seong-Chan Byun, and Byung-Uk Lee. Lens distortion correction using ideal image coordinates. *IEEE Transactions on Consumer Electronics*, 55(3):987–991, 2009.

[31] Micha Feigin, Refael Whyte, Ayush Bhandari, Adrian Dorington, and Ramesh Raskar. Modeling "wiggling" as a multi-path interference problem in AMCW ToF imaging. *Optics Express*, 23(15):19213–19225, 2015.

[32] Stefan Fuchs. Multipath interference compensation in time-of-flight camera images. In *2010 20th International Conference on Pattern Recognition*, pages 3583–3586. IEEE, 2010.

[33] David Jiménez, Daniel Pizarro, Manuel Mazo, and Sira Palazuelos. Modeling and correction of multipath interference in time of flight cameras. *Image and Vision Computing*, 32(1):1–13, 2014.

[34] Seungkyu Lee. Time-of-flight depth camera motion blur detection and deblurring. *IEEE Signal Processing Letters*, 21(6):663–666, 2014.

[35] US3069654A. Method and means for recognizing complex patterns. Paul Hough, United States. US17715A, 25.3.1960. Issued 18.12.1962. 6 pp.

[36] Richard Duda and Peter Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commununications of the ACM*, 15(1):11–15, 1972.

[37] Frederico Limberger and Manuel Oliveira. Real-time detection of planar regions in unorganized point clouds. *Pattern Recognition*, 48(6):2043–2053, 2015.

[38] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nuchter. The 3D Hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):13, 2011.

[39] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nüchter. A data structure for the 3D Hough transform for plane detection. *IFAC Proceedings Volumes*, 43(16):49–54, 2010.

[40] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[41] A. Yla-Jaaski and N. Kiryati. Adaptive termination of voting in the probabilistic circular hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):911–915, 1994.

[42] Lei Xu and Erkki Oja. Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities. *CVGIP: Image Understanding*, 57(2):131–154, 1993.

[43] Jiri Matas, Charles Galambos, and J. Kittler. Progressive probabilistic Hough transform. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Fort Collins, CO, USA, 1998. IEEE.

[44] Martin Fischler and Robert Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[45] Philip Torr and Andrew Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, 2000.

[46] Sergio Orts-Escolano, Vicente Morell, José García-Rodríguez, and Miguel Cazorla. Point cloud data filtering and downsampling using growing neural gas. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2013.

[47] Zhang Yingjie and Ge Liling. Improved moving least squares algorithm for directed projecting onto point clouds. *Measurement*, 44(10):2008–2019, 2011.

[48] Julie Digne and Carlo de Franchis. The bilateral filter for point clouds. *Image Processing On Line*, 7:278 – 287, 2017.

[49] Wei Song, Liying Liu, Yifei Tian, Guodong Sun, Simon Fong, and Kyungeun Cho. A 3D localisation method in indoor environments for virtual reality applications. *Human-centric Computing and Information Sciences*, 7(1):39, 2017.

[50] Ling Jing and Li Sun. Fitting b-spline curves by least squares support vector machines. In *2005 International Conference on Neural Networks and Brain*, volume 2, pages 905–909. IEEE, 2005.