Bernburg
Dessau
Köthen

Hochschule Anhalt
Anhalt University of Applied Sciences

emw
Fachbereich
Elektrotechnik, Maschinenbau
und Wirtschaftsingenieurwesen

Masterarbeit

zur Erlangung des akademischen Grades
Master of Sciences (M. Sc.)

Umer Ahmed

First Name Last Name

Biomedical Engineering, 2019, 4069367

Degree program, matriculation, matriculation number

Theme:

Teaching Tracking System: Developing a Low-Cost Multi-Sensor Tracking System for Virtual and Mixed Reality Education

Prof. Dr. Johannes Tümler
1. Examiner

Prof. Dr. Stefan Twieg
2. Examiner

16. 08. 2023
Submission on

DECLARATION OF AUTHORSHIP

I, Umer Ahmed, declare that this thesis, " **Teaching Tracking System: Developing a Low-Cost Multi-Sensor Tracking System for Virtual and Mixed Reality Education** " is my original work except where otherwise indicated. I am aware of the university’s plagiarism regulations, including those concerning disciplinary actions that may result from plagiarism. Any use of any author's works in any form is properly acknowledged at their point of use.

Sign: Umer

Date: 16-08-2023

ABSTRACT

In recent days several different types of tracking systems have been developed. In the context of a tracking system, tracking is a process to monitor and record the location or movement of an object over time. The tracking of an object can be whether in the real world or xR. xR comprises Mixed Reality, Augmented Reality, and Virtual Reality. As xR is an emerging field, there are multiple concepts enhance concepts that enhance the usability of xR systems, one of which is tracking. The xR industry is more often which is why students from engineering disciplines will get in touch with the new technologies therefore, students need to understand and learn it to improve productivity, efficiency, and safety by collecting and providing real-time data.

As the systems are not cheap to handle, reasonable work was done before in Hochschule Anhalt for developing a tracking system that is better and cost-effective. The existing system consists of an ultrasonic sensor attached to a moveable gondola which acted as a moving body. The ultrasonic sensor emits sound waves and waits for the sound to reflect, calculates the distance using the time of emission reception The sensors are connected to the Arduino UNO via I2C which is an essential protocol of Arduino to communicate with external devices such as sensors. Data is transferred to the “Raspberry Pi" for acquisition, calculation, and displaying in a graphical or numerical form. This system was incomplete concerning calibration and data handling. Multiple sensors should be used to get more efficiency and accuracy of moving objects, which leads to a better understanding of tracking.

A poorly implemented system can lead to negative consequences. For example, imprecise data can lead to flawed analysis and these systems can be expensive to maintain to enhance the precision of tracking data, the duration of data collection and to analyze high-quality data this project should be continued with the additional implementation of a camera and magnetic sensors, performing calibration and evaluating the tracking results.

Keywords: Tracking, Mixed Reality, Virtual Reality, Gondola, Ultrasonic sensor, Arduino UNO, Magnetic sensor, Raspberry Pi, Optical Sensor, ART

ACKNOWLEDGMENT

In the name of God, the most beneficial, the most compassionate. By His grace, I have completed my thesis work.

The thesis project has been supervised by my first supervisor Prof. Dr. Johannes Tümler, I am grateful to him for allowing me to learn and prove myself according to his expectations, and I am also thankful to the second supervisor Prof. Dr. Stefan Twieg for reviewing my thesis.

The work during this project has been inspiring and so engaging that it has let me discover certain properties of the subject I never imagined. This project is an accumulation of a year of intensive work and the study of the vast literature that took months.

I would also like to thank my family and friends for their support and patience. I am grateful that I enjoyed working together, supported, and learned a lot from each other during this project.

GLOSSARY

AR: Augmented Reality

CAVE: Automatic Virtual Environment

HMD: Head-Mounted display

HUD: Head-up display

MR: Mixed Reality

MRI: Magnetic Resonance Imaging

UEQ: User Experience Questionnaire

VR: Virtual Reality

XR: Extended Reality

ART: Advanced Realtime Tracking

OS: Optical Sensor

MS: Magnetic sensor

AS: Acoustic Sensor

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LISTINGS

1. MOTIVATION

In the evolving landscape of modern technology, there is a growing need for developing innovative tools that can enhance human life. As of today's computer, technology, a new human-computer interface has been created that tends to become increasingly closer to replicating natural human movements and interactions. To replicate a human movement or interaction it is important to understand the major aspects and consider the use of technology such as tracking. A basic example of replication of human movement is the computer mouse which we use in our daily life, as the person moves the mouse in a certain direction, we see that the cursor on the monitor screen also moves. This is just a basic idea of how challenging it would be to track the whole human body without a tracking system. The term 'Tracking' refers to a very crucial concept that is nevertheless an important technology used widely. Assume you have a key lost somewhere and there is a device that can help you find the specific location of the lost item. This device is called a tracker. It works out by sending signals or using some special sensors to keep track of the location or movement of the item. It can also be understood as a game called "hot and cold". If something or someone is near you then you are the "warmer" or "colder" alternatively. The warmer and colder is the close or far you Are from the item being tracked. One of many use cases of tracking for xR is defined in a paper from Burova, Mäkelä et.al. "Utilizing VR and Gaze Tracking to Develop Resolutions for Industrial Maintenance" concluded that how to introduce the approach of using simulation in Virtual Reality (VR) coupled with gaze tracking to enable resource-efficient redevelopment [1]. In the study, it was found theatreland with VR can be combined with gaze tracking for industrial maintenance. Mixed Reality, which is one of the latest technologies that emerged, uses tracking to provide an immersive experience a kind of real environment for the person who is actually in a virtual environment. Several different devices provide the experience of the virtual world and a good example is e.g., Oculus Quest, etc. These devices Are used for high-end applications and because of this, it is expensive.

The purpose of creating a low-cost tracking system is basically to help students understand the tracking concept and new technologies during AR/VR lectures. As many systems available in the market Are expensive and not easy to use that is why a low-cost tracking system is generally a cheap system to use which can be affordable for educational institutes and help students learn the basic concepts of the different possibilities of tracking something, and how it can be enhanced and maintained. It also would be beneficial to explain the errors and solutions to problems that can be emerged during the data collection and analysis. The low-cost tracking system will be using multiple sensors tracking over a calculated surface area and will determine the data generated during tracking.

1.1. SIGNIFICANCE OF THESIS

The major aspect of this approach is to help the students to understand technology and continuous development. It will help to improve the discipline of educational technology by providing continuous data and the pattern generated through it. This will also help students and researchers to gain valuable insights by determining the data collected from a tracking system and on the other hand, teachers to refine teaching methods and contribute to the base knowledge of education. The use of data analysis can be significantly helpful in contributing towards the understanding of the concepts of tracking. It also helps in designing the required training and simulations for disciplines like the Armed forces, medical engineering, clinical surgeries, etc. This training is essential because without this a person cannot be integrated directly to deal with human life or dangerous task which has the highest risks included. Therefore, using this system more advanced and accurate systems can be innovated and enhanced. Especially in the education field, students have the chance to simulate and understand the basics of tracking systems which is not possible to achieve using a high-cost system because of the level of technology used. Students can analyze the data in different ways and understand the sensor technology and software interface with hardware configuration. It is meant to be an easy-to-use and student-friendly system that can be used under basic protocols to find out the accuracy required to generate tracking systems for crucial applications.

1.2. STRUCTURE OF THESIS

The structure or the flow of this thesis commits to covering all the basic details, concepts, and understanding of how the project is designed. It starts with the introduction of tracking and its applications providing the reader with a very clear idea about the concept and understanding of its need in recent and future technologies. Further, an explained description of a low-cost Learning tracking system covers the next chapter including all the components, sensors, hardware, and software essentials along with the methods used for data collection, integration, and result generation. It follows with the literature review of related work and methodology for the data acquisition techniques and provides complete detail of how the project is designed. The idea is to give a brief explanation of how tracking works and how this project differs from the existing systems in every aspect. It is very important to understand the basic concepts along with the high-featured tracking systems to have a complete structure in mind of how technology helps human beings. The last sections of this report will include the results and discussions gathered after the data analysis and also be compared with the existing system to get the utmost information about the working. The conclusion explains the details of the key findings and discusses the potential use of this system in different applications.

2. INTRODUCTION

2.1. OVERVIEW

As in this era of industrialization, many new technologies have emerged. One of the currently developing technology is the xR. “Extended reality (XR) is an umbrella term for any technology that alters reality by adding digital elements to the physical or real-world environment to any extent and includes but is not limited to, Augmented Reality (AR), Mixed Reality (MR) and Virtual Reality (VR)” [2]. All these technologies under xR have aided multiple industries and domains, especially gaming, training and education. XR industry has provided a wide scope of looking at the real world from another dimension and perceiving the things which can be only possible in the human mind till today. AR/VR Are the common technologies used under the xR umbrella, which need to be learned by future engineers. For the learning of these technologies, certain concepts should be understood.

The technology sector is experiencing a leading shift in various fields. One important transformation is digitalization due to which industries are revolutionized including the education sector. “Digitalization is the use of digital technologies to change a business model and provide new revenue and value-producing opportunities” [3]. It helps AR/VR technologies to be used widely in multiple industries. With digitalization, a wide impact is also on the education industry which is now more attractive to students using xR. These technologies hold a special impact on transforming traditional Learning methods into digital Learning methods. Enhanced visualization, hand on experience, interactivity, and collaboration facilitate interactive Learning. xR allows students to visualize complicated design models which they can experience in a risk-free environment. Multiple application domains, like navigation, ubiquitous computing, robotics, biomechanics, biomedical engineering, and education require knowledge of real-time monitoring of the position and orientation of objects with some frame of reference. For the mentioned industries and system ‘Tracking’ is the common method used. ‘Tracking’ is the act or process of following something in real-time [4]. In context with students, it is an important concept to learn and understand, therefore, a teaching Learning tracking system with multiple tracking systems was modified from an existing tracking system previously build at Hochschule Anhalt (details in 3.3 section XR in Education Industry). With the easy-to-use availability and cost-effectiveness of the system, it is helpful for the student to get the details of how tracking can be done with different sensors.

2.1. EXTENDED REALITY

2.1.1. XR IN GENERAL

Computer technology and human-machine interactions helped us to combine the real and virtual worlds. Virtual is a term that refers to the meaning of something not real, it can be seen but cannot be touched. This concept of Virtuality enabled the concept of extended reality or xR as the name suggests, anything which is extended or is far from reality. As early as 1838, Sir Charles Wheatstone outlined the basis of stereopsis, based on which the first stereoscopes were developed. With the help of these devices, 3D images can be made with pair of images, giving the user an illusion of depth [5].

The term xR acts as a reservoir of representative of VR and also Mixed Reality. The line between reality and the world gets vanished as you immerse yourself in a world visually or haptically. This is possible using computer technology and wearables.

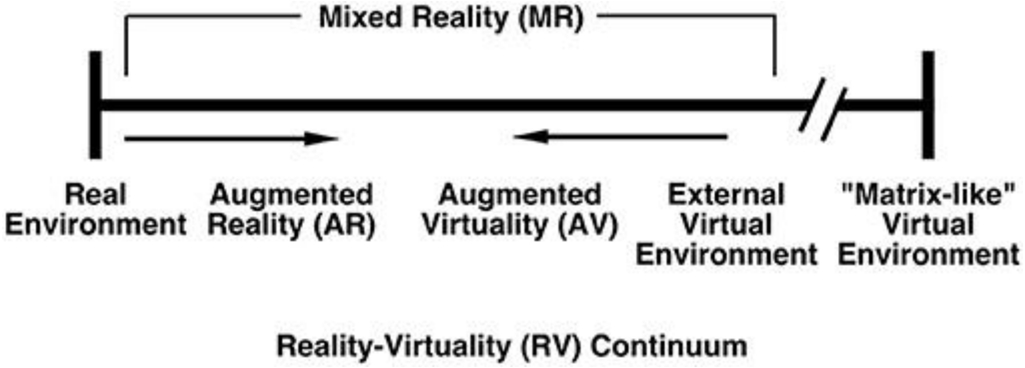


Figure 2.1: Reality-virtuality continuum infographic with examples: real environment, Augmented Reality, augmented virtuality and Virtual Reality [6].

Milgram's Reality-Virtuality Experiment is depicted in Figure 2.1. Continuum is a line that connects Augmented Reality and Augmented Virtuality, with Augmented Reality being closer to reality and Augmented Virtuality being more like the actual world [7]. Virtual reality, according to Burdea, is a complicated end-user computer interface that involves real-time simulations and interactions via several sensory channels. These sensory modes are visual, aural, tactile, olfactory, and gustatory. [8].

The definition of AR was proposed by Azuma in a 1997 survey paper as “Augmented Reality (AR) is a variation of Virtual Environments (VE), or Virtual Reality as it is more commonly called”. Because a user is entirely immersed inside a synthetic environment employing VE technology, the user cannot see the real world. In contrast, it allows the user to observe the real environment while having virtual items superimposed or composited with it [9]. It can be considered that AR supplements reality rather than replacing it as it is in Virtual Reality. It is clearer that Virtual Reality is interactive and immersive but there is also a feature of VR that is not commonly known and this is imagination. As Burdea mentioned in his book ‘Virtual Reality Technology’ “Virtual Reality is, therefore, an integrated trio of immersion-interaction-imagination. The imagination part of VR refers also to the mind’s capacity to perceive nonexistent things” [10].

The term Augmented Virtuality is discussed earlier. This term referred to using real-world objects in a virtual environment. As in a virtual workspace, the computer contents such as the monitor and keyboard reflect the original position which allows you to work seamlessly. Virtual items can be placed on or composited with the actual surroundings in a compositing environment to provide consumers with a more realistic and intuitive sensory experience [11]. Common examples to understand Augmented Reality is the Pokémon Go and Snapchat. This application gives the user an experience of how can virtual or digital objects are imposed on real environments. Hence, providing a detailed description of utilizing AG in multiple industries. In Augmented Reality, the user is not isolated from the real-world virtual objects that can be superimposed in the real world to enhance the experience of Virtual Reality. The concept of Virtual Reality will be discussed in detail in the next chapter.

In Mixed Reality (MR), digital and real objects exist side by side and can interact with each other in real time. MR is the latest in immersive technology and is sometimes referred to as hybrid reality [12]. The other concept is Mixed Reality. In context with Augmented Reality, Mixed Reality is neither completely real nor virtual. It can be assumed a mixture of Augmented Reality and virtuality doesn’t span the complete virtual environment.

2.1.1. CHARACTERISTICS OF EXTENDED REALITY

The sensory organs of human beings such as sight, hearing, touch, smell, and taste play a vital role in sensing any kind of stimuli. As to research, around 80% of human sensory impressions are collected through our eyes [13]. Back in 1955, filmmaker Morton Heilig built a prototype and in 1962 introduced Sensorama. It was a mechanical device that displayed stereoscopic 3D images in a wide-angle view and incorporated body tilting, stereo sound, wind, and Aromas. Heilig who is every so often called the Father of Virtual Reality, directed short films for the device to demonstrate including a bicycle ride over Brooklyn [14]. It is not easy to consider reality until the user has seen or smelled, therefore in 1959, Aroma Rama technology blew scent through the air conditioning system of a theater showing a movie called Behind the Great Wall. In the same year, an attempt was made with Smell-o-Vision technology, propelling scents into the viewers from ventilators placed under the theater seats. It was used only once in a movie named Scent of Mystery [15]. xR provides a completely immersive environment for users to convince them that what they are looking at or feeling is genuine and not artificial. The more immersive platform, the more interactive it becomes.

The concept of identifying between reality and artificial is a human perception. Human differentiates reality from virtuality over several factors and sensing. The world of Virtual Reality itself describes the main phenomenon, it's a reality that is immersed in a virtual environment. A virtual environment refers to a domain where the user can perceive, sense, and interact with objects not present in physical space. Imagine a human flying over a virtual Berlin by flapping his wings or a person visualizing the solar system remaining in a physical space.

On the other hand, rerefers to a special ability to see, hear or understand something which we know as perception. "Perception is the top-down way our brains organize and interpret information and put it into context" [16].

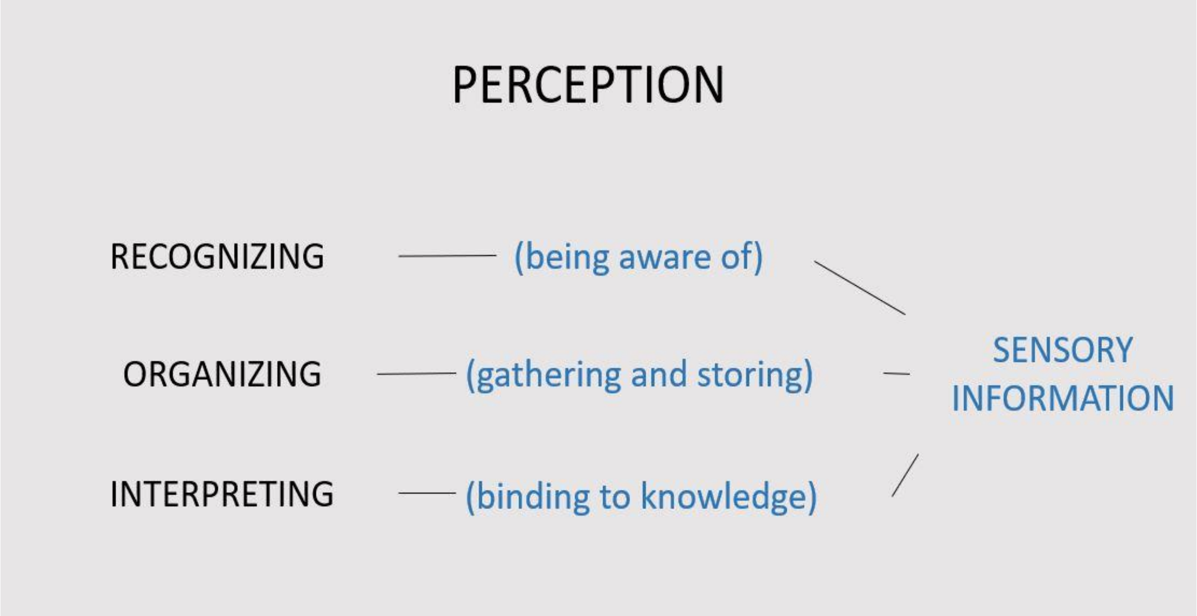


Figure 2.2: Parts of Perception [17].

As shown in Figure 2.2. Perception consists of 3 multiple processes, namely recognition, organizing, and interpreting. Recognition happens for everything we are aware of such as lights and colors, storing this information is organizing and interpretation is the process to understand the stimuli that affect us. All these processes end up as sensory information. The ability of XR to enhance and stimulate the user's senses arises percentual management namely visual, tactile, auditory, and vestibular [18]. Through this manipulation of perception, user experience is advanced by creating real-world sensations in the human world and targeting specific neural networks which are associated with sensorimotor, especially for effective potential medical training. Therefore, the thing or object users use and look at in remakes them perceive that it is originally placed at some place in our real world. It provides real-time information and interactivity.

One of the explanations of Virtual Reality given by Seth, A. et al is VR technology associates multiple human-computer interfaces to provide sensations like visual, haptic, and auditory. This enables users to become immersed in a computer-generated scene and interact using natural human motions. The ultimate goal is to provide an “invisible interface” that makes users interact with the virtual world the same as they do with the real world [19].

Virtual Reality unlike other xR technologies is not mixed with reality. It supervises others and provides a completely immersive environment for the user to interact, manipulate, and sense. It uses the immersion and presence abilities, immersion is the ability of a computer and presence is the ability of humans, these phenomena provide the sensation of ‘to be there’.

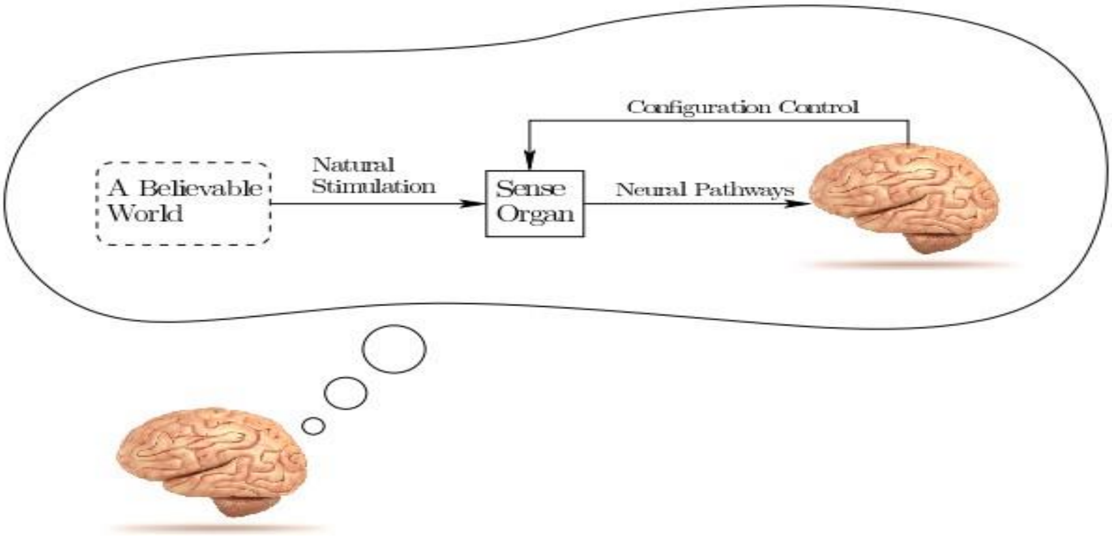


Figure 2.3: Fooling brain to believe VR [20].

The Figure 2.3. above illustrates how the sensation organs work to believe that the virtual environment is the real physical environment. The computer generates a display using environmental characteristics which seems to be real because the human brain is capable of understanding what the natural environment looks like. The environment generated is sensed by the brain using sensory organs and hence the user is made to believe that the environment in his space is real and not virtual. Visual, audio, haptic, and vestibular are the big multiplier of immersion.

In Virtual Reality, the term DoF (Degree of Freedom) is commonly used. As mentioned by, Sherman and Craig in ‘Understanding Virtual Reality’ “A degree of freedom (Dof) is a particular way in which a body may move in space” [21]. This number 3DoF or 6DoF shows how many axes can be tracked. 3DoF refers to 3 dimensions normally experienced in 360 videos because you are not able to change the distance. The user can only move in a 3D environment but for making use of a 3D environment a 6DoF is required. It allows the user to track physical movement which enables virtual movement. These axes are mainly rotational axes and three translational axes.

ELEVATING

Elevation is where a person **moves up or down** (i.e. when bending down or standing up)

STRAFING

Strafe is where a person **moves left or right** (i.e. when sidestepping)

SURGING

Surge is where a person **moves forwards or backwards**(i.e. when walking)

ROLLING

Rolling is where the head **pivots side to side** (i.e. when peeking around a corner)

PITCHING

Pitch is where the head **tilts along a vertical axis** (i.e when looking up or down)

YAWING

Yaw is where the head **swivels along a horizontal axis** (i.e when looking left or right)

Figure 2.4: 6DoF "Degree of Freedom" [22].

In Figure 2.4. 6DoF, that is 6 Degrees of Freedom is shown. In these DoF, the three rational axes can measure the movement of the head. These movements are rolling, the movement of the head from side to side or from left to right, or vice versa. Pitching, when looking straight and moving the head in the vertical axis away from the neck in the Z axis. Yawning, is the Y-axis movement of the head in which the chin is slowly moved towards the right shoulder. In the other three axes, Strafing, the movement in the X-axis pr sideways in a straight line. Surging is the Y-axis movement when a person moves front and back in a straight line. The last is the Elevating, Z-axis movement of the body vertically. So, in 6DoF tracking of all 6 axes are possible, which is important for a virtual system until and unless it is not a 360-degree video.

This all includes a very important concept called tracking. Without tracking the interaction of humans is not possible, therefore no immersion is created and the experience is just like watching a pre/implemented display. Tracking makes it happens, using hand or head or even full body tracking the person is immersed in the environment which creates the feasibility of using VR applications in several domains. There are several technologies present in the market which are developed using the concept of Virtual Reality.

VR HEADSETS: aims to increase immersion. It is widely used in the 3D modeling or gaming industry. It has been successful in training programs or creating educational stuff [23], [24], [25].

Google Cardboard: Google Cardboard is a low-priced hand-held device that gives a Virtual Reality (VR) understanding using any smartphone running on a Cardboard-enabled application [26]. The cardboard viewer functions like a VR Google using the smartphone. The cardboard viewer allows one to turn a smartphone into a VR device. Virtual Reality is a keen technology for providing new reforms, motivating students to learn to track and make innovations. It helps simulate dangerous equipment or perform training that can be risky if performed in a physical space.

All these abilities and tasks are possible through spatial computing. The term was coined by the researcher Simon Greenwold in 2003 at MIT Spatial computing is human interaction with a machine in which the machine retains and manipulates referents to real objects and spaces. It is considered an essential module for making the machines completer partners in the work and play [27]. This is a type of computing where the user dives into the world of computer technology. This merges real, virtual, and user interaction to use applications immersive. Powerful computer devices provide real-time rendering for high-quality applications so the user experiences a smooth interacted environment. The field of xR is evolving rapidly with new technologies, it provides new possibilities for innovation and further integration of xR in daily life.

2.2. TRACKING AND SYSTEM

A system is a group of components that participate together to achieve a specific goal or to do any function [28]. A system can be understood as a collection of different devices or elements that are working together to perform one or more tasks. It can involve hardware or software or both of them to achieve a specific goal. On the other hand, tracking may be explained as recording the specific movements of objects or people [29]. Tracking is the process of monitoring or recording the location, movement, or position of an object sometimes tracking systems are designed to monitor or locate or identify a thing that is not under the scope of the naked eye. These can take on various forms including optical, magnetic, acoustic, or mechanical systems [30]. It is useful to collect the data from the device whether mobile or not moveable working at a specific location. Other relevant parameters include such as positioning, shape, or speed of movement can be of interest in various applications. As Steinbring, Jannik et al. mentioned in their paper that knowledge about tracking a complete human body is very important for a large number of domains including research, computer graphics, animation or robotic applications, examination for rehabilitation, and human intention understanding [31]. Today’s world is focusing on tracking the parameters of the human body because it is the major contributor to the field of computer graphics, animations, and even robotics. To gain knowledge about whole-body tracking various technologies are used along with tracking techniques to directly examine the reconstructed movement or to use it for producing humanoid robots while others only track certain parts of the body.

	ACCURACY	SCALABILITY	RANGE	DATA TRANSFER RATES	SUMMARY
BARCODES	accurate to scanning location	very scalable within limits of manual scanning	very short / needs line-of-sight	N/A	effective for identification but not asset location
RFID	varies widely; accurate to last-scan only	very scalable	up to 600 m	N/A	effective for object detection but not true localization
WISER UWB	~1 m - 10 cm	very scalable	up to 100 m	up to 27Mbps	effective for localized, precise asset tracking
BLE	~2 - 3 m	limited	up to 100 m	up to 2Mbps	effective for localized zone-based tracking
WIFI	~3 m	very limited	up to 50 m	up to 1Gbps	effective for zone-based tracking at a small scale
LPWAN	~50 - 800 m	very scalable	up to several km	variable	many effective tracking options depending on region
CELLULAR	~30 - 100 m	scalable outdoors	~30 km	100Mbps+	effective for tracking large / high-value assets outdoors
GPS	~7 - 10 m	scalable outdoors	global access within satellite line-of-sight	N/A	effective for tracking large assets outdoors

Table 2.1: A comprehensive guide to asset tracking technologies [32].

The Table 2.1. explains a comprehensive guide to asset tracking technologies. Tracking systems have a wide range of applications including vehicle tracking, asset tracking, personnel tracking, environmental tracking, and inventory tracking. Vehicle tracking is used for route tracking or delivery tracking or maybe fleet tracking. For inventory management or supply chain optimization, asset tracking is used. Personnel tracking has applications related to employee monitoring or workflow optimization. For weather monitoring and research, environmental tracking is used. Inventory tracking aids in inventory management or supply chain industries. These kinds of tracking used multiple different technologies such as barcodes, RFID, WISER UWB, BLE, WIFI, LPWAN, CELLULAR, and GPS.

Barcodes are also symbols that can be scanned electronically using a laser or image-based technology [33]. They are inexpensive and extremely lightweight. They can be used for massive numbers of assets and are more affordable than RFID. Barcodes require manual scanning and it is short-ranged. It provides unique identification but nor provide timing or location data. Radio Frequency Identification (RFID) is a wireless system that consists of two parts, namely tags and readers [34]. RFID systems work via electromagnetic readers and tags, these tags contain data that is read by readers. This can be costly depending on the range of frequency used, low, high, or ultrahigh frequencies. This inexpensive, low power consumption, and wide range of RFID tags make tracking simple. UWB tracking is a technique that has a range of 30 to 100 meters. The tags in UWB devices are powered by a battery, indicating that the gadget is active [35]. It requires cabling for data and time management but they are considered more expensive than RFID. UBW signal strength also has low emission limits, for example, the limit is -51. dBm above 10.6 GHz, means it is not easy to implement [36]. Unlike classic Bluetooth, BLE is deliberately designed for technology with lower energy consumption. Applications with high power requirements can communicate with BLE devices [37]. It is a technology applied in many indoor positioning systems. It has low power with accuracy between one to three meters but latency can be an issue in many applications.

Wi-Fi is a wireless networking technology that allows devices such as computers, mobile devices, and other devices to connect to the Internet [38]. It uses Wi-Fi networks to track the location of certain devices. It can provide real-time location but also bring security risks. This technology operates between 60 to 100 meters.

Low-power WAN (LPWAN) is also a wireless wide-area network technology that provides connectivity to battery-powered devices with lower bandwidth over long distances [39]. It is a wireless wise area network designed for low-bandwidth battery-powered devices. This provides long-range wireless connectivity and can be used for long-distance tracking. Along with it, another tracking technology is Cellular positioning, which utilizes a cellular network for tracking. It is also long range but it needs a third-party cellular network to operate. The most commonly used tracking technology is GPS, GPS is so universal that it has become an exclusive technology for Global Navigational Satellite Systems (GNNS). GPS provides global visibility of tracked objects with a stable system. It uses a receiver in the ground to get signals and calculates accordingly its distance from the satellite. It is a high-power consumption technology, with an accuracy between 3 to 10 meters.

As discussed above, tracking is a technology that comprises certain components to get versatile data sometimes. For multiple application purposes, medical or humanoid different techniques are used by using some essential components. These components are sensors, specialized data acquisition systems, communication protocols, data processing and analysis techniques and visualization will be discussed in detail in further chapters.

2.3. MULTI-SENSOR TRACKING TECHNOLOGY

Multisensory tracking is a system that employs multiple sensors to collect data, which is then analyzed and interpreted to generate accurate and efficient results. As R. Mobus and U. Kolbe in their paper mentioned the algorithms and techniques for single and multi-sensor combination of infrared and radar data. The results show that target detection distances increase when radar data is combined with infrared data [40]. Unlike their previous system that relied merely on acoustic sensor tracking, this new approach uses a mixture of sensors to capture data from various scopes. By combining multiple trackers and collecting diverse sets of data, an increase in multiple parameters and derive comprehensive results can be obtained.

Burdea in ‘Virtual Reality Technology’ defines mechanical trackers as “A mechanical tracker consisting of serial or parallel kinematic structures composed of links interconnected using sensorized joints” [41]. Mechanical sensors have multiple advantages over other technologies such as they are simple, accuracy is constant, and are immune to magnetic fields that may occur. On the other hand, these trackers have limited range or movement, if linked together their weight increases which causes problems if it has to be supported by the user. Whatever 3D measurement technique is used, it should not cause trouble with the user’s freedom of motion during tracking. In the need for this requirement, noncontact 3D measurement techniques have replaced traditional mechanical trackers. To facilitate this teaching tracking system, different sensors were considered: magnetic, ultrasonic, and optical. Each sensor offers unique capabilities and characteristics for data collection. By harnessing the strengths of these sensors, we can enhance our tracking capabilities and obtain valuable insights across different domains.

To effectively track an object, three key pieces of information are crucial: visual distance, precise location, and distance from the origin. Through the combination of optical, distance, and location sensors, we achieve a comprehensive multisensory technology. The combination of optical, distance, and location sensors offers numerous advantages, primarily by providing diverse readings that enhance our understanding of the sensors' methodologies and tracking systems. Students can leverage these three sensors in various ways and combinations to enhance their Learning and comprehension. They can explore and exploit all the distinct features of each sensor, drawing connections between them and utilizing them collectively. This opens up multiple Learning possibilities for students. The integration of multiple sensors enhances the system's efficiency by improving accuracy and reducing the likelihood of errors. Any potential errors originating from one sensor can be compensated for by the other sensors in the combination. This also helps to understand the comparison and difference of each sensor tracking technology. This multisensory tracking approach is crucial for educational purposes and contributes to overall system efficiency. Moreover, this Learning system is designed to be cost-effective. A comparison of sensors is shown below in the table.

Sensors	**MPU9250**	**Pi Camara**	**HRS04**
Type	Inertial Measurement Unit	Camara	Ultrasonic Distance Sensor
Interface	I2C, SPI	CSI (Camera Serial Interface)	GPIO
Measurement Range	Accelerometer: ±16g	N/A	2 cm - 400 cm
	Gyroscope: ±2000°/s		
	Magnetometer: ±4800µT		
Resolution	Accelerometer: 16-bit	N/A	0.3 cm
	Gyroscope: 16-bit		
	Magnetometer: 16-bit		
Image Sensor Type	N/A	CMOS	N/A
Image Resolution	N/A	Up to 12MP	N/A
Field of View (FOV)	N/A	Varies	N/A
Frame Rate	N/A	Varies	N/A
Accuracy	N/A	N/A	±3 mm
Power Supply Voltage	2.4V - 3.6V	5V	5V
Dimensions	3.0 mm x 3.0 mm x 1.0 mm	Varies	Varies
Weight	0.03 g	Varies	Varies

Table 2.2 : An Overview of the specification of sensors used [42], [43], [44].

In Table 2.2. every sensor has different specifications such as **Interface,** I2C is a bus interface connection protocol connected to serial communication devices [45]. It is a bi-directional line for data communication. SPI (Serial Peripheral Interface) is also an interface often used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards [46]. I2C uses 2 wires whereas SPI uses 4 wires for communication. The Raspberry Pi has a Mobile Industry Processor Interface (MIPI) Camera Serial Interface Type 2 (CSI-2), which enables the linking of a small camera to the main Broadcom BCM2835 processor. In Picamera CSI (Camera serial interface) is a port that allows connection between two devices.
Measurement range, the measuring range is the range of measured values for a measurement, in which distinct, agreed, or certain error limits are not exceeded. the measurement range for the magnetic and acoustic sensor is mentioned in Table 2.2. Pi camera has a focal length (Details in 2.3.3. Optical Sensor) of 50cm. It is important to understand that the Pi camera is designed for general-purpose recording rather than distance measurement. Measurement in context with distance is not a specific specification for a Pi camera.

Resolution is the lowest possible distinct metric difference that a sensor can sense or detect [47]. The magnetometer in MPU 9250 has a dynamic measurement range and high resolution with low current consumption. Whereas for Pi camera **Image resolution,** the sensor itself has a native resolution of 5 megapixels and has a fixed focus lens onboard.

Field of view (FOV) is an open, perceptible area that a person can see with the naked eye or with an optical device. For optical devices, the FOV is the largest area that the device can capture [48].

Frame rate, on the other hand, measures how fast frames can appear over time (in seconds) [49]. For both cases, it Varies in the Pi camera due to multiple versions available.

Calibration is a key parameter for any system to run without causing any systematic error. Systematic error is any error that is caused by the system itself due to any failure. To avoid these errors and ensure the durability of system calibration is an essential step to start with. This involves fine-tuning and adjustments for the sensors to counter any bias which can affect the data generated. The calibration process ensures that the data gathered is aligned with the actual physical parameters.

2.3.1. ACOUSTIC SENSOR

“An ultrasound tracker is a noncontact position movement measurement device that uses an ultrasonic signal produced by a stationary transmitter to determine the real-time position of a moving receiver element” [50]. An acoustic sensor/ultrasonic sensor is a module or device which utilizes ultrasonic waves to measure the distance at some distance by emitting and receiving these waves. Ultrasonic waves are inaudible sound waves exceeding the hearing limit of 20kHz. As shown in Figure 2.5, the waves generated travel at a distance, and concerning the attached extra acoustic sensors which act as a receiver, a distance can be measured by converting that wave or echo into an electrical signal. The distance measured is proportional to the width of the echo received and with this measurement, it is possible to identify the exact location of the sensor in the moveable area.

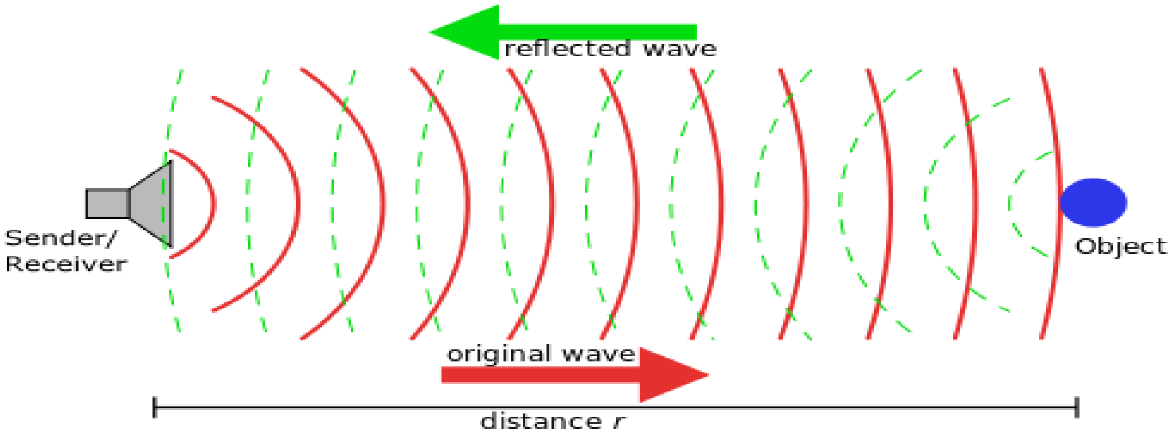


Figure 2.5: Measuring distance with echolocation concept [51].

Figure 2.6: HC-SR04 [52].

There might be some questions Arises that why this sensor was chosen for this low-cost Learning tracking system.

The acoustic sensor/ultrasonic sensor HC-SR04 is suitable for measuring distances and distances to other objects. To do this, it is needed to send an ultrasound signal and receive its echo. To measure the distance to the target, the propagation time of the signal must be measured and the distance calculated from it [53]. When it came to choosing a good ultrasonic sensor, several goals needed to be achieved to use this sensor in the follow-up system**.**

Cost-effective: The HC-SR04 sensor is widely available and relatively inexpensive compared to other acoustic sensor options. This affordability makes it accessible for a wide range of applications and budgets.

Non-contact measurement: The HC-SR04 sensor operates based on the principle of ultrasound, which enables non-contact measurement. This means it can measure distances or detect objects without physical contact, making it suitable for various applications where contact may be impractical or undesirable.

Versatility: The HC-SR04 sensor can be used in different environments and scenarios due to its versatility. It can measure distances accurately, making it useful for applications such as obstacle detection, object tracking, level sensing, and even as a basic proximity sensor.

Easy integration: The HC-SR04 sensor is designed to be relatively straightforward to use and integrate into various systems. It typically comes with easy-to-understand documentation, libraries, and example codes that facilitate its implementation, even for those with limited technical expertise.

Wide operating range: The HC-SR04 sensor offers a relatively wide operating range, typically up to several meters. This range makes it suitable for applications that require both short-range and medium-range measurements.

Low power consumption: Compared to other acoustic sensor options, the HC-SR04 sensor consumes relatively low power 5V and 15mA, making it suitable for battery-operated or energy-conscious applications.

Proven reliability: The HC-SR04 sensor has been widely used in many projects and applications, such as MIYBot, a simple robot board that uses trinkets Pro, HC-SR04, Pololu line sensors and, motor rotation servos [54].

Another Arduino radar project used the HC-SR04. Another project that uses this sensor is the Arduino Ultrasonic Distance Meter with HC-SR04 OLED Display [55]. All these projects demonstrate the use of this sensor in different fields [56].

2.3.2.MAGNETIC SENSOR

“A magnetic tracker is a noncontact position measurement device that uses a magnetic field produced by a stationary transmitter to determine the real-time position of a moving receiver element” [57]. A magnetic sensor is a device that measures the magnetic field. It is widely used in modern industry to measure parameters such as current, direction, or position with the induced magnetic intensity. The sensor can be used to track the location and movement of objects. When a magnetic sensor attaches to an object and the object moves, the sensor can detect changes in the surrounding magnetic field. By analyzing the sensor, these changes can be used to determine the position of the object and sometimes infer its movement. A magnetic sensor ideally converts magnetic field measurements into current or voltage that can be measured with a simple ammeter or voltmeter. It can be designed to measure the magnetic field in three dimensions. They are used to detect magnetic field strength and geomagnetism caused by current or magnet. The output of the magnetic sensors is used to control the direction, position, rotation, angle, distance, speed, location, and presence of electric current. Magnetic sensors can detect the linear and rotational motion of small iron objects without an external high-resolution power supply. They can measure speeds up to 6,00,000 rpm [58].

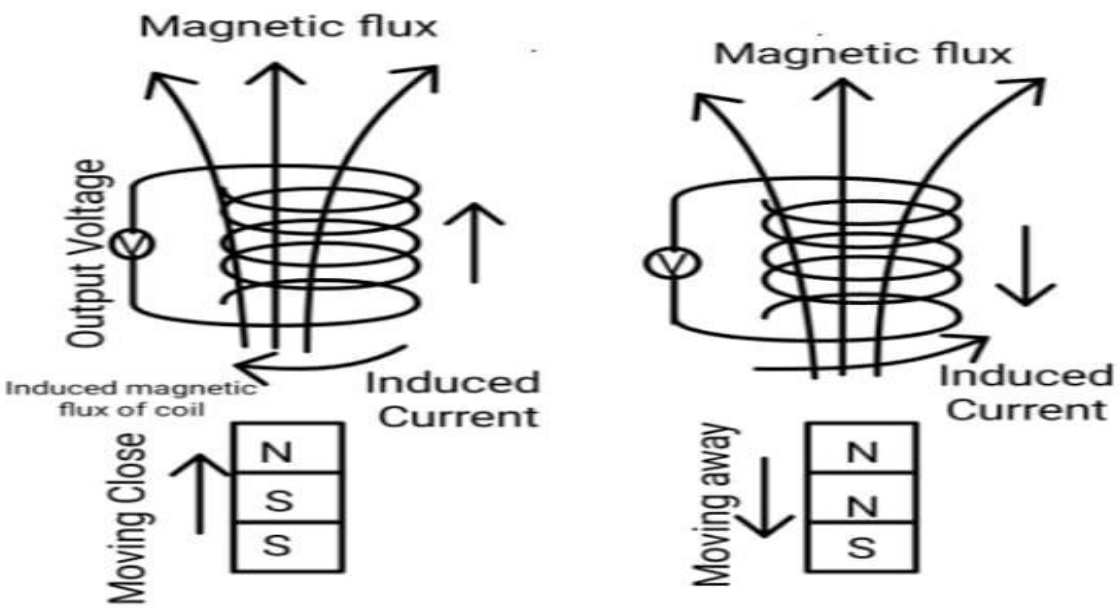


Figure 2.7: Working principle of the magnetic sensor [59].

As is shown in Figure 2.7, a magnetic sensor works with the Earth's magnetic field. When an iron object is brought near a coil or a wire wound around a permanent magnet, it causes a change in the magnetic flux passing through the coil. As a result, a voltage is induced at the terminals of the coil. To comprehend how a magnetic sensor functions, think about the coiled magnetic sensor, where the coil is employed as a basic sensor element. However, only a little amount of functionality is offered by the coil. To build a highly sensitive magnetic sensor, it is paired with coils made of other magnetic materials. In a variety of magnetic sensor applications, coils such as fluxgate sensors, search coils, resolvers, rotation sensors, and angle sensors are employed.

Although the coil itself cannot directly measure magnetic fields, it may identify alterations or oscillations in the magnetic field. The coil's magnetic flux density will rise as the magnet is positioned closer to it. As a result, an induced current and electromotive force are produced. There are multiple types of magnetic sensors available based on their technology or element used:

Magneto resistive kind Magnetic Sensors: These sensors are designed to measure changes in electrical resistance based primarily on an applied field. These effects were discovered by William Thomson in 1856, Albert Feet and Peter Grunberg in 1988, and Terunobu Miyazaki in 1995 [60].

Hall Effect Magnetic Sensors: This type of sensor converts a magnetic field into an electrical signal by way of applying a voltage across a cord positioned perpendicular to the magnetic device.

The precept behind this sensor is the Hall impact, which changed into found via Edwin H. Hall in 1879 [61].

Fluxgate or Coiled kind Magnetic Sensor: These sensors degree the distinction in magnetic discipline on the ends of a vertical rod and display the facts on a grid. They make use of a coiled configuration to achieve correct measurements.

Magnetic Induction type Sensor: This sensor includes a coil surrounding a ferromagnetic center, which famous changes in permeability due to the Earth's magnetic field. The versions in permeability are used to degree the magnetic discipline.

Over Hauser Magnetic Sensors: Also called nuclear precession type sensors, those sensors make use of a liquid that carries electron-rich hydrogen. The aggregate is exposed to an RF (radio frequency) signal to facilitate the dimension of magnetic fields.

Precession Proton Magnetic Sensors: These sensors appoint liquids inclusive of methanol and kerosene, that have a high density of hydrogen atoms. The precession of those protons is used to come across and measure magnetic fields.

SQUID Magnetic Sensors: SQUID stands for Superconducting Quantum Interference Devices. These distinctly sensitive sensors are capable of measuring magnetic fields within the femto Tesla variety. They discover packages in medical and neuroscientific fields, as well as in numerous gadgets that rely upon magnetic field measurements.

Optically Pumped Magnetic Sensor: This kind of sensor utilizes mild of a specific wavelength to polarize an alkaline fuel. The polarization of the gasoline is used to hit upon and measure magnetic fields.

The magnetic sensor used in this monitoring device is the MPU 9250 module, which is a highly capable sensor module ready with a three-axis accelerometer, gyroscope, and magnetometer. The magnetometer within the module makes use of corridor sensor generation to measure the magnetic subject in 3 dimensions.

2.3.3. OPTICAL SENSOR

Optical tracking comprises the use of cameras to detect indicators static in an unbending arrangement that is followed in a frame in real-time [62]. When a person looks at something using a binocular, it can easily determine where and how far the thing is, in the same way, cameras help to determine approximately how far is the thing placed. This type of tracking in which a distance of a certain object can be determined can be possible using an optical sensor.

Optical tracking adjusts object location by tracking the position of active or passive infrared tags attached to the object in real-time [63]. It includes the use of cameras, sensors, and specialized software programs to determine the location, motion, and distance of the object placed concerning the origin.

The idea behind optical tracking is to verily understand and analyze the data gathered using an optical sensor about the object being tracked. The result generated after the data analysis will certainly help to understand how the object was moving and about the motion. Optical tracking, as shown in Figure 2.8, is a technology that allows continuous localization of the position and rotation of objects, with the aid of an optical camera.

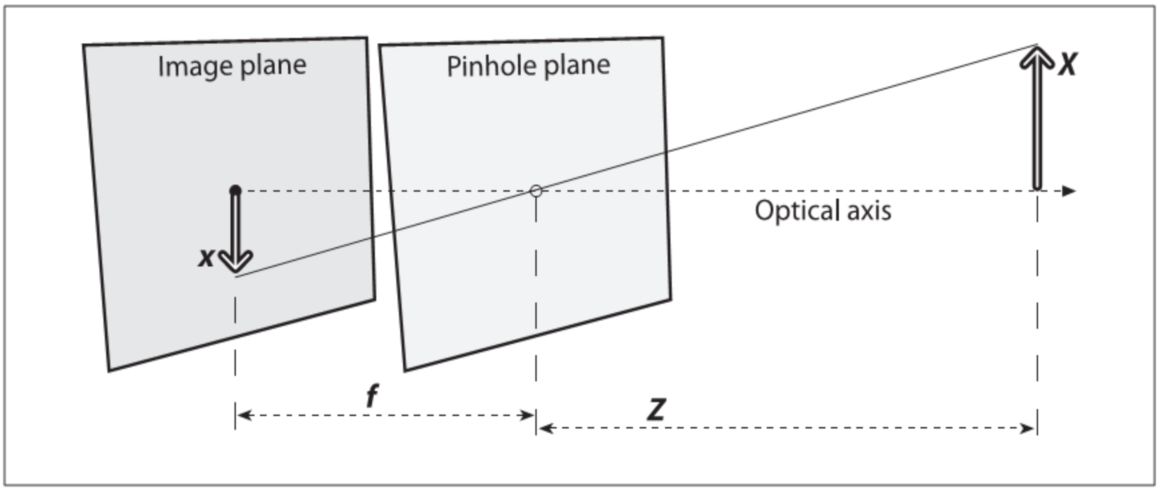


Figure 2.8: Pin hole camera model [64].

Figure 2.8, shows the basics of how imaging works. The pinhole plane model is referred to as an infinitesimal hole that acts as an aperture. The hole passes only the incoming rays which pass through it and block otherwise referred to as O in Figure. The rays which are passed through the aperture are projected on a plane behind called the Image plane. In optics, the image plane is the plane that contains the projected image of the object and is outside the back focal plane [65]. This is the place where the image is formed, the distance from the object to the aperture is Z and the distance from the aperture to the plane is f respectively. That f is called the focal length. The focal length is the distance measured in millimeters between the optical shutter of the lens and the camera sensor, where the light data is recorded [66]. It determines how strongly it focuses the light. A perpendicular line out of the image plane is the optical axis.

The camera for optical sensing shown in Figure 2.9 is used in teaching tracking systems as a Raspberry Picamera or simply a Picamera. This camera communicates with Raspberry Pi (will be discussed in section 4.3.4 Optical Sensor Tracking and Calibration) using serial interface protocol. The normal application related to the Picamera is image processing and recognition, machine Learning, or surveillance projects. The camera module offers a 5MP fixed and focus camera that supports 1080p30, 720p60, and VGA90 resolutions. It is accessible via the MMAL and V4L APIs [67]. Small size, low price, and good quality make it feasible and compatible with the systems. The camera is simply attached to Raspberry Pi and supported by a vast range of library OpenCV. OpenCV is a large open-source computer vision library that allows one to perform image processing and computer vision task.

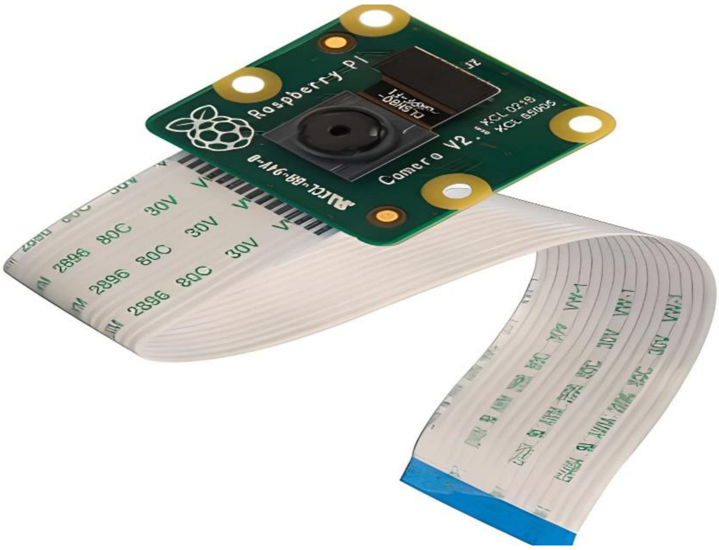


Figure 2.9: Pi camera [68].

The important thing for camera tracking is to find correspondence points between the real environment and their 2d image projections. In-camera tracking, it is important to find correspondence points between the real environment and their 2D image projections. This concept is known as pose estimation, which is very important in computer vision applications. To optimize the position of a person using computer vision and machines, the name of the local estimation is Learning. It predicts where body points are located by processing images provided by a neural network [69]. Although it is a very difficult step to estimate the current pose, one of the approaches makes it easier. This approach is the use of binary square markers, they provide the 4 corners to obtain a camera pose. This marker is usually known as Arco Markers shown in Figure 2.10. “An ArUco marker is a synthetic square marker composed of a wide black border and an inner binary matrix that determines its identifier (id)” [70]. The black border allows it to be quickly identified in the image, and the binary coding allows its detection and error detection and correction techniques. The size of the token determines the size of the internal table. For example, a character size of 4x4 consists of 16 bits [71].

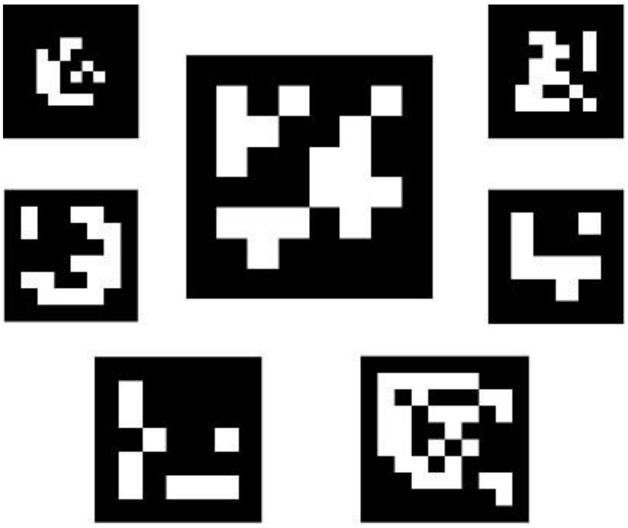


Figure 2.10: Example of ArUco markers [72].

The question that arises here is why ArUco markers are chosen for pose estimation? In response, here are some key features in contrast with QR code:

Aspects	ArUco Markers	Other Fiducial Markers (QR, AptilTag etc)	Why ArUco Markers Are Better?
Marker Design	Square Or Rectangular	Different Shapes	ArUco markers have a unique, easily identifiable shape that aids in robust detection.
Encoding	Binary or QR-like	Multiple Encoding Schemes	ArUco's binary encoding identifies errors and there are more correction capabilities including enhancing reliability.
Marker Size	Variable	Fixed or Variable	ArUco markers can be adapted to different scales, offering flexible to choose.
Dictionary Variability	Multiple Dictionary	Limited Variability	ArUco supports diverse dictionaries, growing the number of unique markers available.
Detektion Algorithem	Corner Detection	Edge Detection	ArUco's corner detection is more robust against distinctions in lighting and perspective.
Computational Efficiency	Generally Efficient	Varies	ArUco's corner-based detection requires less computation compared to other complex designs.
Open Source	Yes	Varies	ArUco's open-source nature encourages community support, extensions, and continuous improvement.
Integration	Widely Supported	Custom Implementations	ArUco libraries are readily available at multiple platforms.
Robustness	Good	Varies	ArUco's encoding and corner-based detection contribute to its robustness in challenging conditions.
Tracking Applications	Commonly Used	Varies	ArUco markers are well-suited for real-time tracking.

Table 2.3: Comparison between ArUco marker and other fiducial markers.

Distinctive Shape: ArUco markers have a unique shape (square or rectangular) with clear corners. This shape makes it identifiable which aids in accurate detection even in complex environments.

Binary encoding: ArUco markers enable error detection and correction leading to more reliable pose estimation, specifically when there is noise.

Adaptive Marker Size: ArUco markers are flexible for size estimation. It allows flexibility in choosing sizes based on the requirement of the application.

Multiple Dictionaries: It also offers support for different marker dictionaries, enabling a variety of unique markers.

Robust Detection Algorithm: ArUco corner detection is more robust than others. It is better towards changes in lighting, and angle of viewing, compared to some edge-based methods used by other fiducial markers.

Computational Efficiency: It is the amount of time or memory required for the computer to do the evaluation or calculation [73]. l Its corner-based detection algorithm is computationally more efficient in comparison to others.

According to a multi-demand evaluation benchmark study, ARTag and ArUco are computationally more efficient, but ArUco proves to be better in terms of measurement accuracy. [74].

2.3.4. ARDUINO UNO

On the other hand, a very useful microcontroller board in this tracking system is Arduino UNO shown in Figure 2.11. The Arduino UNO is a microcontroller based on ATmega328. It has in total 14 I/O digital pins and 6 analog pins making it compatible to support the microcontroller. It is a plug-and-play module, the connection can be easily made via USB, and the operating system for Arduino UNO is also called Arduino software which provides the complete programming platform. In Arduino UNO, ‘UNO’ is an Italian word that means one, as this was version 1.0.

Figure 2.11 : Arduino UNO [75].

The UNO has several different communication channels that allow it to communicate with other computers or microcontrollers. It supports UART and I2C protocols which are serial communication [76]. The module can be powered using a USB port or with an external power source.

Arduino UNO is often used by beginners in electronic systems because of its simple protocols. The Arduino UNO board was considered the most used board by beginners in their projects [77]. This board can be connected to other systems via a USB port which also supplies power to the boards.

The software used to write and compile the board is called Arduino IDE (Integrated Development Environment). It operates on 5V and has a current rating of 40mA maximum. Arduino UNO has a crystal oscillator of 16Mhz which acts as its operating frequency. It has around 14 digital pins and 6 analog pins allowing it to plug and play with various modules at once. As communications are important, 3 types of protocols are available including I2C, SPI, and Serial interface (Details in Multi Sensor Technology). It also can interface with other Arduino boards, microcontrollers, or computers. Arduino UNO comes with a wide range of applications including embedded systems, digital electronics, medical instruments, and more. In a possible application, Hankin Tao used Arduino UNO for building custom real-time sensors for Virtual Reality applications as a master thesis. Tao used the MPU-6050 sensor for exploring to connect multiple sensor configurations for VR applications to be able to customize motion sensors to specific applications. A total of three experiments were conducted to validate the implementation of real-time simulation in a virtual environment, and the results show that this work can provide accurate real-time motion tracking for VR applications. [78].

2.3.5. RASPBERRY PI

If we take a look at the history of computers, we would know that first-generation computers were built massively big with small processing power. Over time, compact and small-size versions of computers were available in the market and now the computers are transformed into more compact sizes such as to fit in one's pocket. One of the leading technologies in computer development is a small, low-power-consuming, and easily available product Raspberry Pi.

Raspberry Pi, shown in Figure 2.12. is a small single-chip card-size computer that runs on a very low power hence providing slow but good processing if compared to a traditional computer. By connecting external hardware such as a keyboard or mouse it works as a mini-computer. It is used mainly for applications in robotics, IoT, and image processing. Raspberry Pi is defined as a credit card-sized mini-computer that can be used with any input and output hardware such as a monitor, TV, mouse, or keyboard - settings can be effectively customized to become a fully developed computer. low price [79].

Figure 2.12: Raspberry Pi Board [80].

Raspberry computers are compatible with an Operating system like Linux, it is also possible to install third-party versions like Ubuntu, RISC OS, etc. The Raspberry Pi foundation provides Debian-based Raspbian OS, officially available which is free to use. It includes GUI tools for programming, gaming, or browsing. There are multiple versions of Raspberry Pi available in the market which multiple different features.

Raspberry Pi is a small, affordable single-board computer to promote basic computer science education. The size is not bigger than a credit card, making it portable. It has GPIO (General Purpose Input Output) pins that can be used to connect external sensors or other electronic components for various tasks. It is available with 1GB, 2GB, or 1GB of LPDDR4 SDRAM. Low-Power Double Data Rate (LPDDR), also known as LPDDR SDRAM, is a synchronous dynamic random-access memory that consumes less power and is aimed at laptops and devices such as mobile phones [81]. It consists of 2 USB ports, 40 pins GPIO header, 2 Micro HDMI, 4K Video MIPI, DSI display port, 1 MISI CSI camera port, 4 pin stereo audio and composite video, H.265 decode (4kp60), H.264 decode (1080p60), H.264 encode (1080p30), OpenGL ES 1.1, 2.0, 3.0 graphics Micro SD for OS & data storage5V/3A DC via USB type C Connector,5V DC via GPIO, PoE (Power over ethernet) capable via optional HAT extension. This specification makes it better in performance and operates like a traditional computer.

3. LITERATURE REVIEW

3.1. LITERATURE RESEARCH METHOD

This chapter describes the details of literature research methods used for the thesis to collect appropriate academic Articles and tracks associated with teaching tracking systems in various domains. It outlines the systematic technique used to identify and examine the literature, ensuring the presence of numerous perspectives.

The literature is searched in a complex mixture of electronic databases, educational journals, convention claims, and relevant online resources. Keywords and search terms including “teaching systems”, “tracking systems” and related versions have been used to search Articles relevant to the research topic.

Search Terms	Search Engines	Total Results	Relevant Results
"tracking system development", "sensor integration in tracking", "Arduino and Raspberry Pi in tracking", "calibration techniques for sensors", "GUI design for tracking systems", "case studies in tracking technology", "limitations of tracking systems", "magnetic tracking", "acoustic tracking", "optical tracking", "ArUco marker tracking", "multi sensor tracking", "sensor fusion", "mechanical tracking", "tracking in xR", "VR headsets tracking", "AR headsets tracking", "Mixed Reality tracking", "tracking technology challenges"	Google Scholar, IEEE Xplore, ACM Digital Library, ScienceDirect, PubMed, ResearchGate	364	247

Table 3.1: Literature Research Method.

To ensure the presence of appropriate and applicable literature, precise selection standards had been applied. Initially, Articles have been screened mainly based on their titles and abstracts to assess their relevance to the study’s topic. Later, the full-text Articles had been assessed for their theoretical and practical contributions, methodological accuracy, and affiliation with the research aims. Peer-reviewed Articles, books, and influential reports have been prioritized for inclusion.

A systematic approach is used to extract relevant facts from the selected sources. Key facts, along with authors, publication year, studies method, prime findings, and theoretical backgrounds, turned into recorded for every Article. This concerned labeling the extracted facts to derive significant insights and facilitate comprehensive information on the study’s subject matter.

To ensure the reliability and credibility of the covered literature, a capable review of the literature is made. The Articles have been critically evaluated primarily based on the strength of their research design, the validity of the findings, and the recognition of the authorship assessment helped to decide the general primary findings and reliability of the literature covered in the evaluation.

The findings from the literature were interpreted and analyzed to identify theoretical frameworks. Connections among extraordinary research were collected, and variations within the findings were thoughtfully examined. Through this system, a complete understanding of monitoring structures in schooling emerged, supplying a foundation for the following sections of the thesis.

3.2. APPLICATIONS OF XR

The most difficult task for the professional working in the educational sector is to be students entertained and engaged during the lectures. Peggy Lisenbee illustrates teachers' perceptions of students' independent use of technology in a classroom. 42% percent of the teachers found that using technological methods helps students to learn more and 89% of the teachers stated that scientific tools were more operative for students to validate their understanding of concepts taught in a classroom [82]. As students are not able to follow the traditional whiteboards due to the generation gap, these technologies can help to connect the bridge between generations. VR can offer a completely fully immersed environment for the student in a form of a student classroom where education becomes fun, relevant, and also engaging at the same time. The students can be provided with multiple tasks, and simulations and teachers can implement creative Learning procedures. The most important point here to make is that theoretical knowledge is not always understandable. All students have a different way of Learning but the most common is the physical experience which helps to understand the real concept of some equipment or machine. With VR it is possible to simulate things along with the theory taught to make a better understanding.

The University of Michigan, for example, has redesigned the withdrawn Ford Nuclear Reactor in Virtual Reality. Nuclear engineering students can now enter the reactor and operate under the supervision of Professor [83]. This study helped students to obtain data from nuclear reactors and how it works. Nuclear engineering, radiation health and safety, nuclear physics, and reactor operations.

Figure 3.1: Virtual Elo [84].

Virtual Elo, shown in Figure 3.1. is short for Virtual Elocution. In this project, they investigate the use of Virtual Reality in therapy against public speaking anxiety. Traditional exercises for training coping mechanisms include exposure to triggering situations, e.g., talking to people in public, giving a speech, making phone calls, etc. [85]. The project is a prototype that uses VR in therapy for individuals having public speaking anxiety as there is a virtual audience created. Students can gain knowledge of therapeutic applications and techniques. They also get the opportunity to work with the audience perception model, involving human behavior. Overall, the project helps students to apply technical skills to equip them to work on real-world applications in healthcare.

Teachers' guidance on classrooms for the use of augmented technology in 2016 was offered by The International Society for Technology in Education (**ISTE**) [86]. The scope of AR in classrooms is growing with time as it allows teachers and students to learn together. It allows both to create their experiences by making animated images.

xR is not a new technology to education, it has quickly evolved from research and development into the mainstream. As distance Learning has become very common after the COVID 19 but xR is a step further. Students are being taught at schools and colleges about xR as it has the potential to understand and learn complex subjects better. Virtual reality for social and emotional learning, the virtual reality experience uses ordinary situations and spaces to effectively teach social skills and also supports teachers' professional learning. Advanced virtual reality features allow students to immerse themselves in a virtual environment to develop skills that meet classroom requirements [87]. This helps users to complete the scenes in a virtual environment where they learn social competency skills and also allows students to understand the potential impact of VR-based intervention on social interaction and communication.

Buń, Paweł, Górski, et al. presented a paper regarding the accuracy measurements of systems that are used in simulations. The simulations are made more realistic using tracking but it is also important to consider that where a small mistake from the operator can risk lives. The author's research over the years concluded that these tracking systems are not easy to be directly employed in the educational area like schools and universities due to budget reasons. Specialized systems are more precise and have more effectiveness but at the same time, it is not cheap for just Learning purposes. The research presented in the paper is an attempt of answering the question that how the immersion level related to the accuracy of position and orientation representation of a real object in a virtual environment impacts the assessment of expediency of a given tracking system in training applications. The paper presents the results of a study of the precision of one of the tracking systems available on the market – the PST 55 system. This device uses retro-reflective markers, which are widely used also in measurement systems not devoted to use in the VR system. Utilizing the TRITOP industrial photogrammetric measurement system from GOM, the actual position of monitored objects was confirmed. The comparison study's findings made it possible to determine how accurate the tracking system under review is and how immersion levels may be affected by accuracy. [88]

"Multi-Sensor Eye-Tracking Systems and Tools for Capturing Student Attention and Understanding Engagement in Learning: A Review." Wang, Yuehua, et al. It offers a comprehensive examination of the high-tech system and research for boosting education and capturing the interest of learners. It emphasizes sensors and devices, ranging from multi-sensor professional gaze tracking equipment to inexpensive systems. They also investigated system infrastructure, data formats, data processing methodologies, and essential technologies that can assist students in learning settings. [89].

However, nowadays as technology has advanced and new tracking opportunities are available in the market, one such tracking system in biomedical applications is reviewed by Sorriento, Angela, et al. Their article provides a comprehensive review of optical and electromagnetic tracking systems, specifically in the context of computer-assisted image-guided surgery. The review highlights the main features, advantages, and limitations of both technologies. The optical spectrum is used in Optical Tracking Systems (OTSs) to track the location and positioning of surgical instruments. They do, however, necessitate an even line of vision between the optic indicators and the camera sensor, limiting their application in contexts with rigidly defined operating theaters. Electromagnetic Tracking Systems (EMTSs) identify the location of electromagnetic sensors using electromagnetic field generators. Because EMTSs do not require a direct line of sight, they are more adaptable. Because traditional mechanical trackers are not practicable for the user (described in section 2.3. Multi-Sensor Technology), the study seeks to provide a critical comprehension of both optical and electromagnetic tracking systems. It typically provides a grasp of the functioning principles, mistakes, and protocol validation for both technologies involved. Furthermore, the paper discusses biomedical solutions that are appropriate. Finally, the paper provides a comprehensive comparative examination of the state-of-the-art in optical and electromagnetic tracking systems, as well as their potentials and limitations for medical applications. This evaluation offers a well-intentioned understanding for researchers and practitioners in assessing and choosing the most suitable tracking system for precise biomedical applications [90].

Duan and X evaluated magnetic tracking and positioning in endoscopy in another study. Both the flexible endoscope and the robotic capsule demonstrate advances in magnetic tracking methods [91]. Reduced radiation, precise therapy, minimally invasive surgery, and more comfortable examination are all benefits of proper placement for patients and professionals [92]. This evaluation is based on a complete review of previous years' literature, fundamental theories, and appraising commercial goods as an advantage for exploring magnetic technology.

In another tracking system literature, Wang, Cheng, et al. used an endoscopy-based navigation system to deal with the navigation problems between complex bronchus networks in the lungs. A primitive technique using bronchoscope tracking generates numerous errors, the author identified the errors that were countered during the study and used a new approach to make the navigation easier [93]. The plan specifically comprises three segments: a structure-aware bronchoscope tracking module, an anatomical structure classification module, and an RGB-D picture domain conversion module [94]. The image-based technique was used to improve the quality and performance of tracking. The technique involved is to convert the RGB-based module into the virtual module and depth image of bronchoscope images. Then, the two forms of structure are made, namely structureless and a rich structure of the bronchus network. An improved video-CT bronchoscope tracking was the last approach to estimate the camera pose. This approach showed relatively better performance and higher accuracy than the current tracking system.

Bald, Christine et al. Their paper presents the first approach to automatically localize the ultrasound head during measurement [95]. The proposed technology is based on coils placed around the patient's bed and a magnetic 3D sensor connected to the ultrasound head. In addition to some processing steps, the predictive localization algorithm is based on trilateration followed by a least-average approach to refine the estimate. In initial proof-of-concept measurements, average accuracies of 2.85 cm and 8.94 were achieved using fixed positions and orientations of the ultrasound head. In addition, a measurement with a moving ultrasound head has been provided, which shows the capability of the system in real-time. [96].

At St. George’s Hospital in London, patients getting anaesthesia for their operations were offered the option to use a VR headset before and during the operation, which helped them to a soothing virtual space. The results were amazing: all participants reported that using the headset improved their overall hospital experience, as a result, 94% felt more at comfort, and 80% experienced less physical pain. The patients were so occupied in the virtual world that they were often unconscious of being in the operating room. [97].

There are a lot of studies and state of art systems available in the market. XR empowers and engages learners by giving them more control over their Learning process and turns the role of the educator from communicating knowledge to controlling students to explore and learn. XR lets students study experientially, and according to Dale’s Cone of Experience, people remember and retain 90% of what they learn from doing [98].

The improvement of fundamental motor abilities is one of the most popular uses of VR in sports. Virtual reality (VR) can be used to build virtual simulations that let athletes practice motions and techniques in a secure setting. A soccer player can practice passing and receiving in a virtual field, whereas a basketball player can practice dribbling, shooting, and passing in a virtual court. Through this sort of training, athletes can advance their abilities without running the danger of injury or exhaustion. [99].

3.3. XR IN EDUCATION INDUSTRY

As xR is an important technology for now and mainly for future, it is also essential to learn the key concept and understand the importance of it. For this purpose, an Augmented and Virtual Reality Lab is made at the location of Bernburg-Strenzfeld campus. Mobile equipment facilitates usage at different university locations. The 3D Workshop and Presentation rooms house 6 HP workstations each, enabling efficient 3D modelling. A Dual Extrusion FDM printer with a 215 x 215 x 200 mm print volume supports STL, OBJ, and 3MF files through CURA. Additionally, a 3D scanner is available. 6 Motion-Capture Systems, using Kinect for Xbox, capture gestures and speech on Windows PCs. A Multiuser VR-Illusions room in Bernburg lets 5 users engage in real-time VR sessions with VR HMDs and hand tracking. Mobile setups at all campuses include 3 VR-Ready workstations, diverse VR HMDs, mobile devices, and a Samsung Gear 360 camera. [100]. These labs offer research and development options for the students along with offering space for them to work in teams for complex tasks. Students get new ways of Learning xR technologies, gaining experiences, and making products or business processes. This xR is also used for interdisciplinary subjects which helps students to learn another subject through xR.

TUM Extended Reality Laboratory (TUM XR-Lab) is located at the new TUM Campus Heilbronn. This lab consists of xR technologies such as Virtual and Augmented Reality. The With Lab, TUM Campus Heilbronn gains a place where regional companies can inform themselves about new technologies and participate in events Around Learning technologies and in workshops for employee development. TUM XR-Lab was set up in collaboration with the Centre for Digital Leadership Development (CDLD). In close collaboration with various stakeholders, experts, and user groups TUM XR-Lab will develop prototypical new formats and services to improve teaching and Learning [101]. This lab actively helps students to engage, learn and experience VR and AR. Different prototypes will be developed, tested and integrated into other study programmes to improve Learning as well as teaching.

In many municipalities, participation in urban transformation has taken place through various multimedia engagement initiatives. However, despite the wide range of communication channels provided by multimedia approaches, a significant portion of individuals are still marginalized in terms of participation or incomplete participation. In the collaborative research project XR-Part, both technical and process innovations for social participation are being further developed, interlinked, tested and evaluated. XR-Part combines various virtual and augmented reality solutions within the Augmented Reality Platform (XR) into a system consisting of three components: (1) a three-dimensional virtual that can temporarily hold joint events at any time; Experience and meeting space, (2) an augmented reality application that allows you to view the contents of the project in various three-dimensional ways, and (3) an online participation tool that supports on-site participation. Since online participation is less attractive, these projects help bring people together and plan the creation of real urban spaces on a local level. With these solutions, you can address target groups that are poorly or unattained in your planning process, make your visions and ideas for spatial change concrete and clear, simplify experimentation in space, and ultimately achieve sustainability It should be possible to enable the Learning process. [102].

The University of Bremen hosts the Computer Graphics and Virtual Reality (CGVR) Research Laboratory, where they conduct fundamental and applied research in the field of visual computing. This includes various aspects of computer graphics including visualization, modelling, simulation, animation and also computer vision, which involves extracting information from other types of images. like RGB, depth, etc. Naturally, these fields involve a wide range of disciplines such as computer science and mathematics. MultiVR Labs specializes in pioneering research in the field of visual computing, with a particular focus on virtual reality and computer graphics [103]. Emphasis on discovering new techniques, algorithms, methods and technologies and their application in different fields. Learning visual computing, performing comprehensive analysis, and performing task-specific modelling emphasise students to lean xR in detail. Students are also able to learn management ability through individual or team-based tasks.

At University Weimar, Critical XR Lab is a beginner module that offers an Introduction to the Unity game engine. The course is taught in the Digital Bauhaus Lab, where participants can experiment with diverse Virtual Reality and Augmented Reality technologies and ideas [104]. The students will be able to learn, work and understand the Unity interface. It will enable them to make strategies to deal with the difficulties faced when working with VR/AR technologies.

The XR Lab of the Faculty of Mechanical Engineering and Mechatronics at Karlsruhe University of Applied Sciences focuses on the research and development of Virtual and Augmented Realities and Mixed Reality (MR) solution approaches in the industrial field [105]. There students and researchers have the chance to learn the xR and apply it in different industrial domains.

For a long time, Virtual, Mixed and Augmented Reality (VR/MR/AR) have been the main application areas, but ART tracking systems can do more than that: wherever precise and reliable tracking of objects or human beings is needed, ART optical tracking technology can be helpful in many parts of the product developing process, in analysis and research as well as in various other professional use scenarios. Every ART system is powered by DTRACK control software at its core. This is primarily intended as means of providing the camera with an easy and precise setup that guarantees exceptional operational stability. The Controller is used to transmit both 3DOF and 6DOPF data from the cameras to the DTRACK system. Real-time orientations and positions are computed, resolved, and seamlessly transferred to your 3D software through this method [106]. ART system is a marker-based tracking, it utilizes markers namely active and passive to track the position or orientation of the tracked object. Passive markers are mainly retroreflective markers which use reflected light as an input when it's back to the source while active markers emit their infrared light. It also offers high accuracy and precision for precise monitoring.

Lastly, the main concept behind this technology is the tracking. Without tracking a human immersed environment cannot be established. Tracking allows the system to replicate human movements, therefore xR can be implemented to learn new aspects of tracking and how it can be innovative for future xR applications. For this purpose, a teaching tracking system was designed to help the student understand the basics of tracking and utilize the generated data for near future xR application which in a way helps the educational institutions to transform the entire education sector into a lively virtual environment.

A low-cost tracking system for Learning was previously designed at Hochschule Anhalt with the initiative to help students learn about this concept. The system design was magnificent and very simple even can be used by students on its own with a simple protocol measure. The system used multi-sensor technology in combination making the complex task simplified. The data generated through sensors was professionally handled and used for acquisition. The system was designed to give students different tracking methods strengths and weaknesses. This illustrates the possible field of application. The hardware included a magnetic sensor, and the acoustic sensor and the optical sensor were connected but the code script for acoustic is available and the other was not completely implemented. The data was processed using Arduino UNO and Raspberry Pi. The system structure was made of aluminum forming a boundary for working. Within the frame, the target can move to be moved Around. The acoustic or ultrasonic sensor and the magnetic sensor were attached to the movable target. The sender and receiver HC-SR04 ultrasonic sensor was controlled by Arduino UNO. This sensor measures the time it takes to receive the signal and send it to Arduino UNO for further processing, ultimately Raspberry Pi calculates the distance between the target and sensor using a circle intersection formula. The calibration of this sensor was also implemented which can be accessed through the menu. The graphical representation provides the data generated and also access to software log entries.

3.4. XR TECHNOLOGIES

As discussed earlier about xR details and how it is learned and taught, there is another domain where xR meets the industry. xR is a universal term for immersive technologies such as AR, VR, and MR, these technologies make an extension of reality by simulating digital materials. One such VR system is Oculus.

A powerful PC is required to connect the Oculus Rift Virtual Reality headset to perform complex calculations and produce visuals. To determine the user's location while they are interacting with the virtual environment, the system's Constellation positioning technology uses gaze tracking and IR LED sensors. The headset has a 2060 x 1200 OLED display with a 90Hz refresh rate. The tracking area is 5' by 11' and the field of vision is 110 degrees. Magnetometers, gyroscopes, and accelerometers are examples of sensors. The device allows for complete 360-degree location tracking. The controllers are wireless, but a lengthy cord connects the headset to the computer. Another requirement is a computer with enough processing power to support virtual reality, which can be quite expensive [107]. With Oculus, mainly the applications involve gaming but there is also something with video streaming applications. It is also used for working or studying.

A completely immersive Virtual Reality headset is the HTC Vive. The HTC Vive tracks and maps your movement around the room using two sensors in each corner. The controllers are wireless, but a lengthy cord connects the headset to the computer. Another requirement is a computer with enough processing power to support virtual reality, which can be quite expensive. SteamVR, a virtual reality-based gaming platform, powers the Vive. The HTC Vive has a lot of potential applications in business, education, and your personal life. To produce experiences that would otherwise be impossible, virtual reality can be specifically tailored for human interaction [108].

Microsoft HoloLens is a fully self-contained wireless holographic computer. It gives users a glimpse into so-called “Mixed Reality”. This means that people with Microsoft HoloLens can see digital content called holograms in the real world and interact with it using voice and gestures. holograms are graphics created with light that appear to float three-dimensionally in space. The potential of the HoloLens is almost unlimited and possible applications across all industries are conceivable. This diversity makes the HoloLens particularly interesting for companies and convinces developers and corporate customers to work with innovative technology [109]. This is used for digital businesses, education product sales, and visual support for doctors. Knowledge transfer, remote maintenance, and interior design applications are also aided by HoloLens.

For advanced VR users, VARjo produces the most immersive virtual and mixed-reality products and services. Their retinal resolution equipment is used to design cars, conduct ground-breaking research, and teach astronauts, pilots, and nuclear power plant operators. Virtual, augmented, and Mixed Reality is brought to life in lifelike visual quality by VARjo XR-3, VR-3, and Aero. With the highest resolution (60 pp) and the largest field of view in the market, the VARjo VR-3 establishes a new benchmark for VR headsets. Professionals and cutting-edge VR users can both benefit from VARjo XR-3's most immersive Mixed Reality experience and VARjo Aero's generational leap in visual fidelity [110].

3.5. SUMMARY

Extended reality which is an umbrella term for immersive technologies like Virtual Reality, Augmented Reality, and Mixed Reality offers a wide range of applications across multiple domains and industries such as health, biomedical engineering, teaching, Learning, gaming, or sports. Virtual Reality environments are mainly used in simulations, gaming, and training sessions whereas Augmented Reality composes digital items in the real world, like navigation, education, and industrial tasks. Mixed Reality merges both the virtual and real world, where users can interact with physical surroundings. These technologies play an important role in multiple healthcare research systems and achieve to solve complex medical tasks. This also gives the gaming industry and Learning sector a new way of thinking and doing things. Systems like Oculus, HTC Vive, and others offer high-quality immersive environments as well as VR experiences. Together these technologies have given humans a different way of thinking, learning, and understanding this world.

4. METHODOLOGY

4.1. PROBLEM ANALYSIS

To improvise and adapt this concept for better Learning something or a system should be developed. The system would be able to get the idea of how tracking is working, how to track an object, and how to use it for the benefit of human beings. Tracking needs to be fully understood for multiple types for the betterment of training. Therefore, it is important to get to know how tracking works.

To develop such a system, an existing system is taken into consideration which is low-cost and multi-sensor tracking. The problem that occurred after carefully examining was that the device was not able to provide proper tracking of the optical sensor nor there is an option to calibrate. To determine the pros and cons, multi-sensor tracking technologies are better. More missing functionalities can help in understanding the tracking in detail.

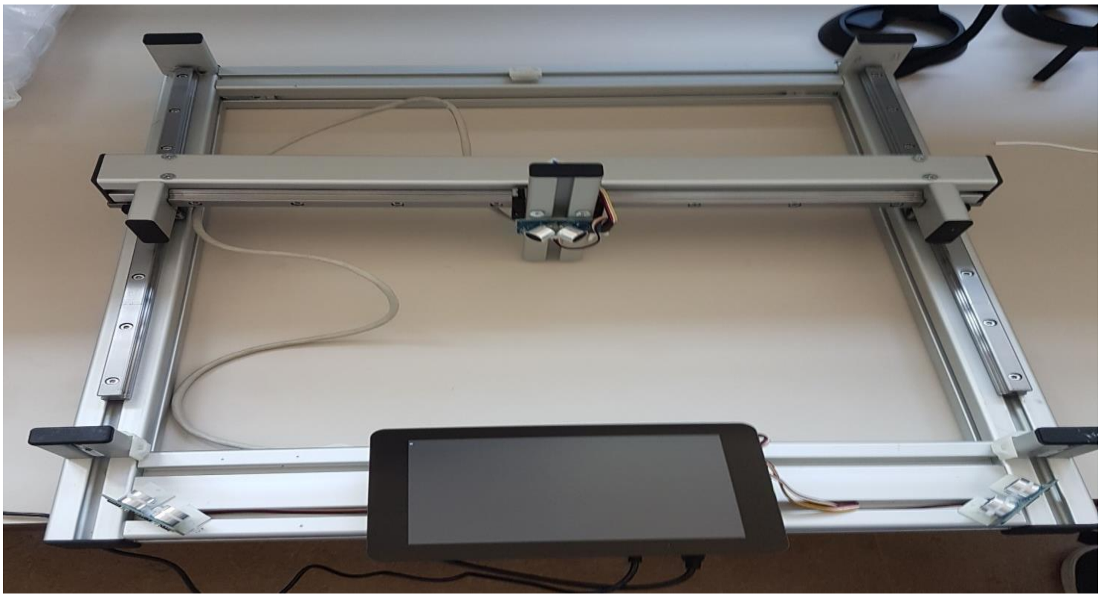


Figure 4.1: Learning tracking System (existing).

Figure 4.2: Optical Sensor in learning Tracking System.

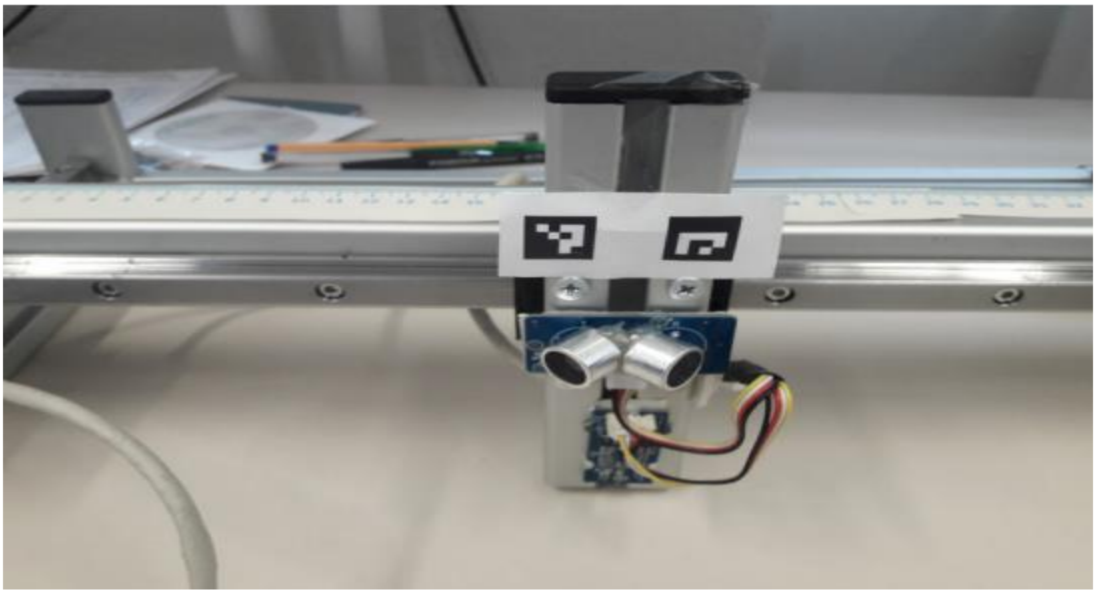


Figure 4.3: Connections of Acoustic and Magnetic sensors.

In this system shown in Figures 4.1 and 4.2, the working sensor was only connected but was not completely implemented. The magnetic sensor and optical sensors were left out for implementation along with the system calibration. As we have discussed earlier in Literature review that xR is taught in multiple ways using multi-sensor technology concerning those systems this system lacks clarity for the integration of various sensor types and their data synchronization. This system is not completely capable of teaching xR tracking concepts in detail including sensor fusion/multi-sensor technology, pose estimation, latency, and synchronization (details in section 3.3 XR in education Industry). The evaluation of the system during a lecture or concerning other systems was not perform therefore it lacks qualitative analysis.

The implementations should be done:

- Optical camera complete calibration
- Magnetic sensor full implementation and calibration
- Multi-sensor calibration and tracking to be shown graphically

4.2. APPROACH

To understand this concept and enhance tracking in contrast to immersion and various factors in future systems, a low-cost system for understanding tracking was previously designed using a single ultrasonic sensor technology. The approach used in this study moves Around the modification of the existing tracking system in terms of precision and accuracy. Instead of following a traditional method like surveys or interviews, this study involves a practical implementation and enhancement of the system to meet the highest degree of efficiency. Therefore, the main goal is to develop and update the existing system to a more better tracking system.

The approach sticks to the research objectives and scope of this study which is to enhance and modify the existing system by adding multiple sensors and refining the data acquisition and analysis methods. Following the calibration and calculation update, it gives more strength to the results generated to be more accurate. The system is designed for the educational purpose so the aim is to provide complete accuracy, precision, and overall effectiveness for students to learn.

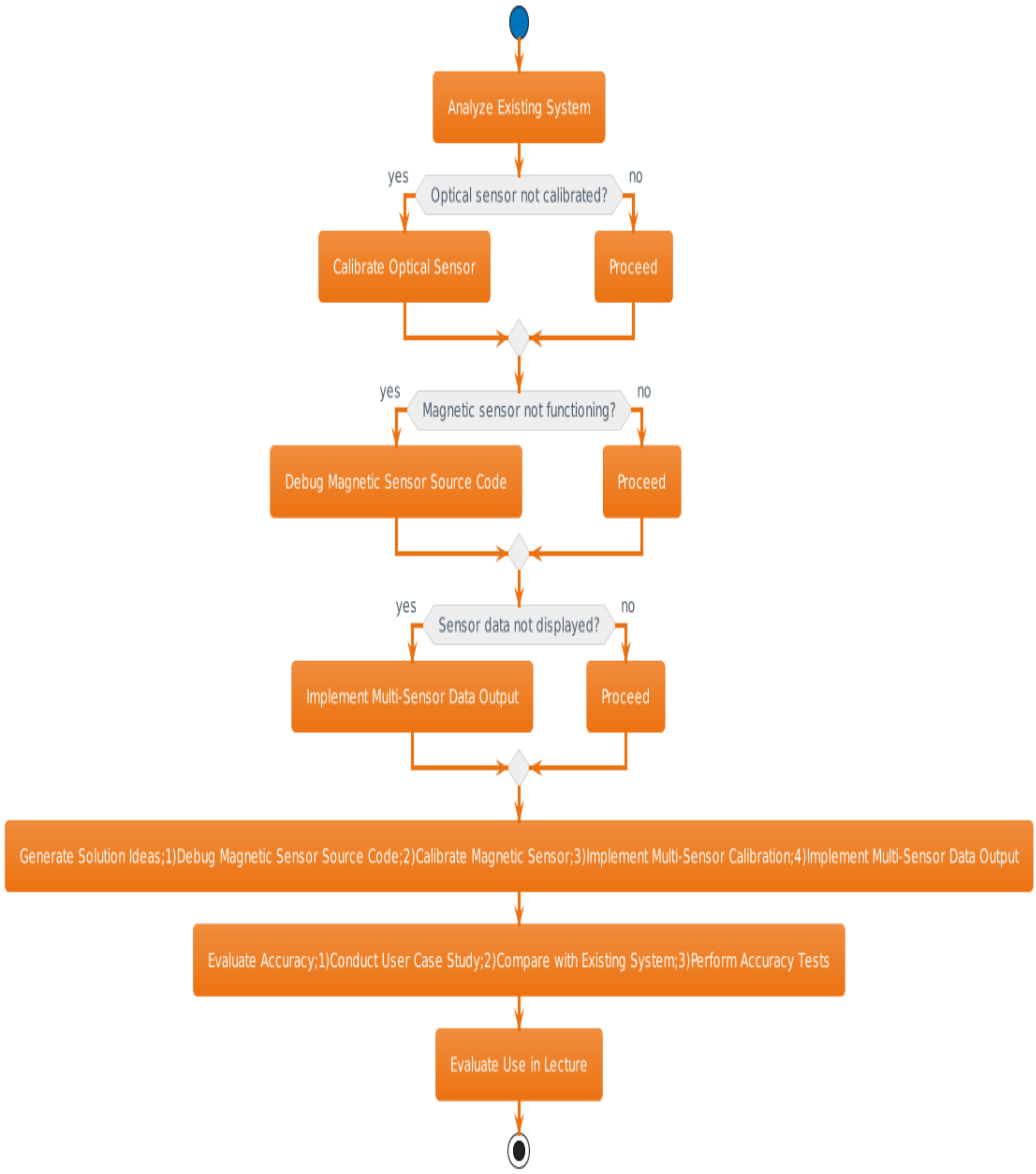


Figure 4.4: Activity chart for approach used to complete this existing system.

In Figure 4.4, the initial phase of the approach is used to check and identify the flaws in the existing tracking system and to look around the area for improvement. After analyzing and testing modifications were made to include a multi-sensor technique with upgraded data acquisition techniques. The second phase is to calibrate and check of all the sensors are implemented properly because calibration makes the sensor configured to provide proper output. After implementing the calibration, the next step is to implement all sensor calibration at once which is important to save time if single sensor calibration is not needed. The next phase is to generate solutions by first debugging the source code provided by the previous system and checking if it needs to be free from bugs or errors. The implementation of multi-sensor output shown on the graph is also an essential need so to compare all sensors together. For the accuracy of the completed system, it is important to conduct a user case study because it is mainly for Learning purposes and to check the accuracy of each sensor in comparison with an existing system. In the end, the evaluation will be done by using this system in lecture as it is the teaching tracking system so it has to be compatible and easy to use for the teacher.

4.3. IMPLEMENTATION

4.3.1. OVERVIEW OF TEACHING TRACKING SYSTEM

A detailed implementation of the system is shown in an activity diagram below:

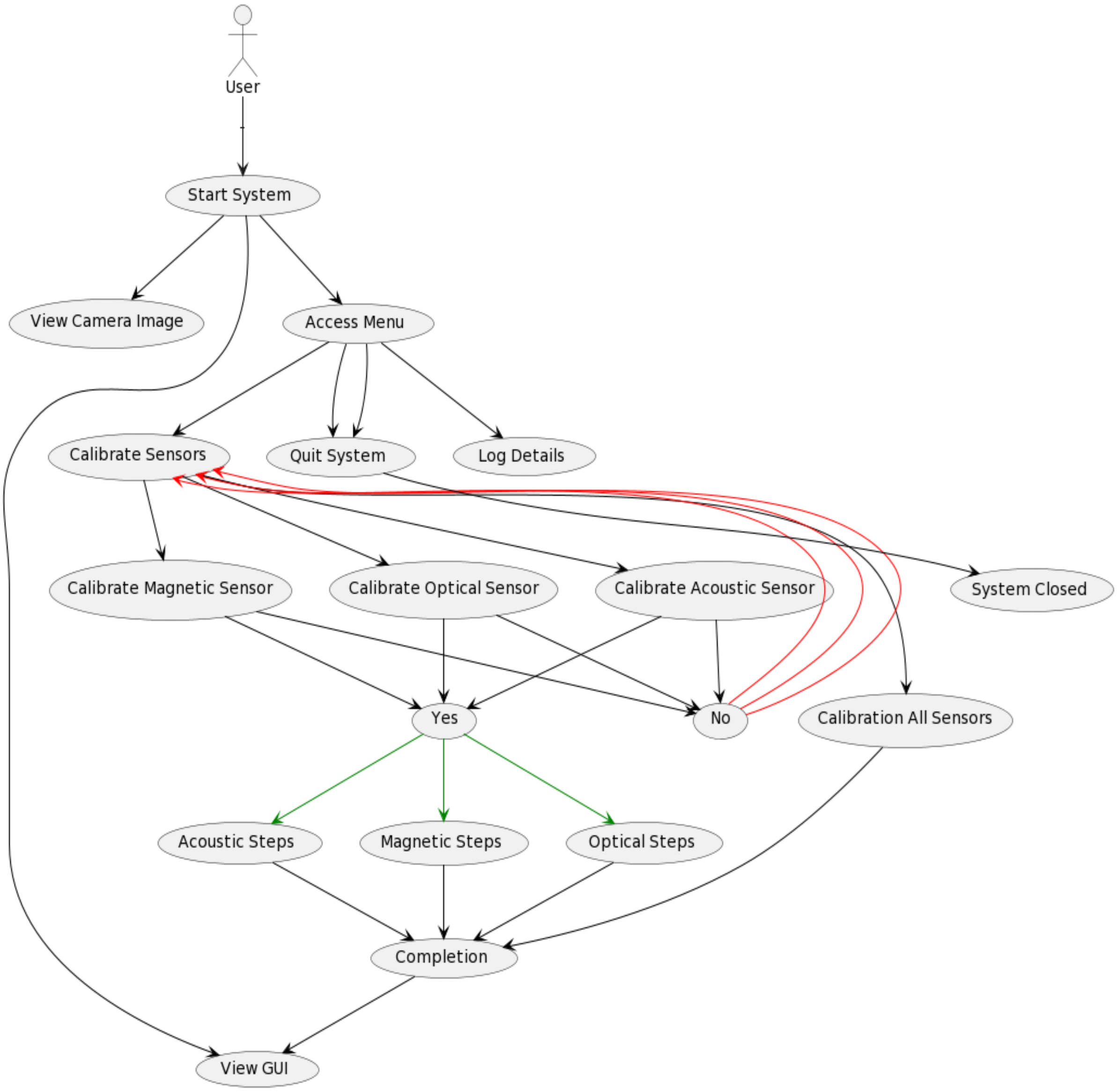


Figure 4.5: Teaching Tracking System Activity Diagram.

In Figure 4.5 The implementation of this system involved multiple steps are shown. The system starts after the user turns on the system. It takes a bit of time to start the software based on the Rasbian platform. After clicking on the start function the tracking system starts to function. Two screens can be simultaneously seen on the LCD which are the camera image and GUI (Graphical User Interface). On the GUI the left part shows the graph with tracking data and on the right top, numerical values of data can be seen. On the right bottom is the menu including multiple options: Log option, calibration option, and quit option. If the user presses the log button, a log window with details appears. If the user presses the quit button, the system is closed. If the user presses the calibration button, it gives more options to calibrate all sensors, calibrate the acoustic sensor, calibrate the magnetic sensor, and calibrate the optical sensor. If the user presses the calibration all sensors button, the calibration popup appears and one by one calibration of all sensors starts, after completing the instructions the calibration will be completed. If the user presses the calibration acoustic sensor button, the popup shows to move the gondola in a different direction in order to record the reading, the data is then sent to raspberry pi where the necessary calculations are performed and finally shown on the graph. The magnetic sensor also follows the same procedure if the user opts to press the magnetic sensor calibration button. Fort the option of optical sensor calibration is a bit different, after the button is pressed, a certain popup appears which needs to be followed. The optical sensor preloads the default images and do the necessary calculation. In the end, the system can be closed by the quit button.

Components	**Functionality**
MPU-9250	Magnetic Sensor
HC-SR04	Acoustic/Ultrasonic Sensor
Picamera V2	Raspberry Pi Camera Module
Raspberry Pi	For data handling and calculations
Arduino UNO	For sensors connections and integrations
Python 3.10.4	Programming language for scripting
Raspbian	Raspberry Pi OS
Arduino IDE	Platform for Arduino Programming

Table 4.1: Components and IDE used in developing Teaching Tracking System.

The above components mentioned in table 4.1 are essential components of a teaching tracking system designed for educational purposes. The system tracks motion and orientation using the

MPU-9250 magnetic sensor, allowing for more engaging learning experiences. The HC-SR04 acoustic/ultrasonic sensor aids in obstacle detection, improving movement safety. The Picamera V2 records visuals, allowing for real-time demonstrations and visual aids in the classroom. The system, which is powered by a Raspberry Pi, analyses data, makes calculations, and maintains interactions. Sensor integration is facilitated by the Arduino UNO, while data analysis and control are handled with Python 3.10.4 scripts. The Raspberry Pi OS, Raspbian, provides the required environment, while the Arduino IDE is used to construct control logic. This full setup enables instructors to construct interactive teaching sessions that are engaging and dynamic.

4.3.2. SOFTWARE PROGRAMMING AND ALTERNATIVE SOLUTIONS

In this chapter, the programming solution used in this low-cost tracking system using multiple sensor techniques will be discussed. In the previous section, the approach used and why Python is chosen as to be the main programming language was deliberated.

ASPECT	PYTHON	C#	C++
EASE OF LEARNING	Easiest to learn	Learning curve is moderate	Learning curve is the highest
SYNTAX	Easy to read	Moderate	More complex
DEVELOPMENT SPEED	Requires less coding	Moderate	Slower due to complexity
COMPLEXITY	Best for simple 2D games	Moderate complexity	Suitable for complex games
PERFORMANCE	Slower and less Efficient	Efficient	Efficient
MEMORY MANAGEMENT	Handled by interpreter	Managed by language/runtime	Direct control
FRAMEWORKS	PyGame, Pyglet	Unity (popular), Godot	Unreal Engine
COMMUNITY SUPPORT	Moderate	Large and well-developed (Unity)	Moderate
INDUSTRY USAGE	Limited usage in industry	Widely used in professional studios	Used in AAA games
LEARNING RESOURCES	Limited resources	Abundant resources	Available resources
DIRECT HARDWARE ACCESS	Limited	Limited	Full Control

Table 4.2: Comparison between Python, C# and C++ [111].

Python is an easier computer programming that is easier to read and understand as compared to C# and C++ as shown in table 4.1. It does not require a longer time to write and complete the product code. It is also considered slow when comes to complex tasks. As an interpreted language, therefore it does not require high specifications for variables, classes, or memory. Python is considered good for 2D games; hence it proves to be a good choice for game development. On the other hand, C# has a higher curve for Learning compared to Python but is easier than C++. C# is faster to learn and is also not much complex compared to C++. It requires high specifications for classes, variables, and memory management. This language is the main language used by Unity Framework which is a popular game engine [112]. C++ is a lower-level language compared to C# and Python. It requires a longer time for Learning and use. C++ gives the user more control which makes the user a technical programmer.

For this teaching tracking system Python is the first end choice to program with its ease of Learning and rapid prototyping. Python has a variety of libraries and frameworks that supports development such as Pygame or Pyglet. Python's rapid development capabilities and ease of use make it attractive for indie game developers who often work with smaller teams and tighter budgets. Python's simplicity makes it a great choice for educational purposes, such as teaching programming and game development to students.

4.3.3. MAGNETIC SENSOR TRACKING AND CALIBRATION

The magnetic sensor used for this system is MPU9250 (details in 2.3.2 Magnetic Sensor). The magnetic sensor is attached to a moveable gondola which can be moved freely in a given space. This sensor senses the position which is then integrated to Arduino UNO and then to Raspberry Pi for further classification and calculations using Python as a programming language. The final tracking output is shown on a GUI window. The first phase of any sensor is to be calibrated; therefore, the following snippet shows a general version of how this sensor is being calibrated in a teaching tracking system:

```
class MagneticSensor:
    def __init__(self, conf):
        # Initialization of attributes and configuration parameters
    def start_calibration(self):
        # Reset calibration state and initialize data structures
    def calibrate (self, value):
        if self.calibration_state.get_state () == self. calibration_state.
ACCUMULATING_1:
            # Accumulate readings for "front" position
            # Calculate mean values after enough readings
            # Transition to the next calibration state
        elif self.calibration_state.get_state () == self.
calibration_state. ACCUMULATING_2:
            # Accumulate readings for "back" position
            # Calculate mean values after enough readings
            # Calculate offset, scale, and max values
            return self. offset, self. offset, self. scolex, self. scale,
self.max_x, self.max_y
```

Listing 1: Magnetic Sensor Calibration.

The initialization and configuration of the magnetic sensor as shown in Listing 1. form the core of the calibration procedure. The **MagneticSensor** class methodically specifies configuration parameters including offsets, scales, and maximum values, frequently taking them from a configuration file. These variables are crucial because they make it easier to compensate for sensor biases and set the stage for precise location calculations.

There are two separate steps to the calibration process: **ACCUMULATING_1** and **ACCUMULATING_2**.

The accumulation and computation of calibration values based on sensor readings are handled by the calibrating method. It begins by examining how the calibration process is currently progressing. It accumulates raw sensor values (x, y, and z) for the "front" position if it is in the **ACCUMULATING_1** state. The mean values for each axis are calculated once 100 readings have been accumulated. It then moves on to the following calibration state.

The approach keeps accumulating readings for the "back" position in the **ACCUMULATING_2** stage and computes the mean values similarly. Based on the calibration values received from the "front" and "back" positions, it then determines offset, scale, and maximum values. These computations are used to account for aberrations in sensor measurements.

Calculating the offset involves calculating the average difference between the respective mean values for the "front" and "back" positions. The calibration process engineers offset values for the x and y axes. This tactical move is effective in reducing typical hard iron distortions in sensor measurements.

Scale Calculation: A complimentary procedure is carried out to calculate the scale factor. The normalizing of the average differences between the mean values of "front" and "back" positions the x and y-axis-governing factors. The scale factors that result is essential for reducing soft iron distortions.

Maximum Value Calculation: In parallel, the calibrated values are used to derive the maximum values for the placeholder algorithm. The scaffolding for the placeholder orientation calculation is provided by these maximal points.

The calibration procedure moves on to the next stage after determining the offset, scale, and maximum values. The calculation and tracking of magnetic sensor data will only be accurate after performing the calibration. The following snippet from the code gives an overview that how magnetic sensor data is organized and calculated:

```
# Constructor for MagneticSensor class, initializes attributes and calibration parameters
class MagneticSensor:
    def __init__ (self, conf):
        pass # Placeholder, no specific initialization code provided
    def _readCb (self, raw):
        # Reading callback method triggered when new sensor data is available
        value = conn. getMagneticField () # Get magnetic field data
        self. calibrate(value) # Initiate calibration process
        position = self. calculate_position(value) # Calculate corrected position
        if position is not None:
            self. pass_to_gui (position + value) # Pass combined position and sensor value to GUI
    def calculate_position (self, values):
        # Calculate Position: Adjust raw sensor values and estimate orientation
        return (x + 130, y + 300) # Return shifted position values
    def pass_to_gui (self, data):
        # Pass Data to GUI: Store data in the queue for GUI presentation
        pass # Placeholder, no specific action defined in this snippet
```

Listing 2: Magnetic Sensor attributes and calibration parameters.

The magnetic sensor class plays a key role in managing and interpreting data from the magnetic sensor within the system. This class is designed to process incoming magnetic field data using a variety of methods to ensure accuracy and meaningful interpretation. During initialization, the class defines important parameters and properties, including configuration options, data ordering, and calibration-related properties. The calibration process is a key aspect and the placeholder calibration values are stored in a dictionary. A callback function is triggered when raw sensor data arrives. This function acquires the magnetic field data and continues to calibrate and calculate the sensor position. The calculated position is then integrated with the sensor data and stored in a queue, allowing easy access to a graphical representation of the user interface. This integration ensures accurate readings by eliminating offsets and scaling factors. Finally, the MagneticSensor class ensures that the sensor data is accurately processed, calibrated, and effectively presented in the user interface to provide meaningful communication. Its features contribute to a complete workflow that transforms raw sensor output into valuable information for users.

4.3.4. ACOUSTIC SENSOR TRACKING AND CALIBRATION

The sensor used for acoustic sensor tracking is HC-SR04 (details in 2.3.1 Acoustic Sensor). The acoustic sensor is attached to the moveable gondola and two more acoustic sensor are connected to each corner forming a triangle for calculating distances. The acoustic sensor tracking involves the calibration and calculations of important parameters used for the tracking of an object. It involves the configuring of acoustic sensor system to accurately measure distance and derive essential parameters. The following code shows a general overview of how calibration of acoustic sensor is done in this system.

Calibration procedures play a central role in improving the accuracy and reliability of acoustic sensor. The process begins with the initialization of the essential components, including setting up a log manager responsible for managing system logs. Temporary variables are introduced to facilitate the calibration process. The sensor system is then started with the appropriate configurations and connections established to communicate with the acoustic sensor. To ensure transparent operation, a dedicated thread generates dummy sensor values, mimicking real-world data, during the sensor initialization phase. The calibration process itself is initiated by resetting the calibration state and preparing time values for data accumulation. Through simulation of dummy sensor values, different scenarios are explored, including different sensor locations in the field. Prediction functions are used to calculate the parameters associated with these fictitious values, thus facilitating the position estimation and calibration simulations.

```
def start(self):
    # ...
    self. dummyActive = True
    dummyThread = threading. Thread (target=self. ReadCb_dummy)
    dummyThread. Start ()
```

Listing 3: Dummy values for Acoustic Sensor.

In listing 3. It is shown that dummy values are used as the basis for the calibration. These simulated values are used to accumulate calibration data, with particular emphasis on data collection from the front and rear sensors. After a considerable volume of data is obtained, the calibration mean values are meticulously calculated and stored.

```
def calibrate (self, value):
    if self.calibration_state.get_state () == self. calibration_state.
ACCUMULATING_1:
    elif self.calibration_state.get_state () == self. calibration_state.
ACCUMULATING_2:
               distance_1 = math. sqrt ((self. calibration_x_offset + self.
    left_sensor_x_offset) **2 + (self. sensor_y_offset + self.
    calibration_y_offset_1) **2
            self. calibration_y_offset_2) **2)
            # same for distance_3 and distance _4/ distanceif = distance_4
- distance_3
            sonicspeed_1 = distancedif / timedif
            sonicspeed_2 = distancedif / timedif
            overhead_1 = statistics. mean(...)
            overhead_2 = statistics. mean(...)
            self. sonic. speed = statistics. mean ((sonicspeed_1,
sonicspeed_2))
```

Listing 4: Acoustic sensor calculations.

When switching to actual sensor values, the system moves into the data collection phase. Actual sensor readings are collected and extensive testing is performed to identify missing or incomplete data points as shown in listing 4. The collected sensor values are then processed, where calibration techniques are applied. The calibrated values are analysed to determine the exact location of the sound source, contributing to accurate spatial inference. Completing the calibration process includes extensive calculations to fine-tune the sensor system's performance. The resulting correction values are used to calculate key parameters including speed of sound and processing cost. Distances between different sensor positions are meticulously calculated from accumulated calibration data. This information is used to determine the speed of sound, which greatly improves the accuracy of real-time measurements. In addition, processing costs are calculated to refine the data interpretation. The resulting calibration results are carefully recorded, encapsulating important details for further analysis and validation. Key parameters, such as calculated sound velocity and overload condition, are presented to facilitate a comprehensive evaluation of the calibration process.

The calibration process culminates in successfully improving the accuracy and reliability of the acoustic sensing system, positioning it for accurate and consistent interpretation of real-world data across applications. different.

```
def calculate_position (self, values):
    # Adjust time-of-flight values by subtracting overhead times
    val1 -= self. overhead_left
    val2 -= self. overhead_right
    # Calculate distances from time-of-flight using sonic speed
    distance_left = val1 * self. sonic_speed
    distance_right = val2 * self. sonic_speed

    # Compute potential x-coordinate of the source using triangulation
    x = (self. sensor_distance**2 - distance_right**2 + distance_left**2) /
(2 * self. sensor_distance) + self. left_sensor_x_offset
        # Calculate potential y-coordinate using geometry
        y = math. sqrt (distance_left**2 - x**2) - self. sensor_y_offset
```

Listing 5: Position calculation for acoustic tracking.

After moving the gondola to any position, the real-time data is transferred to Arduino UNO which acts as a slave to Raspberry Pi for the calculation of parameters. This snippet in Listing 5 encapsulates important functionality for handling acoustic sensor data and communicating with the graphical user interface. The reading method derives time-of-flight values from an acoustic sensor. The **calculate_position** method uses time-of-flight, sound velocity, and sensor geometry data to estimate the location of the sound source. The **predict values** method generates predicted time-of-flight values based on a given position and other parameters. Finally, the **pass_to_gui** method uses a queuing mechanism to transfer the processed data to the graphical user interface. These methods collectively contribute to the accurate collection, analysis, and presentation of acoustic sensor data. This code snippet introduces the **calculate_position** method, which is important for determining the position of a sound source based on time-of-flight data. The method involves adjusting the time-of-flight values, calculating the distance, and using geometry to estimate the position of the source (x, y). The calculated position is returned or None if the geometry constraints are not respected.

4.3.5. OPTICAL SENSOR TRACKING AND CALIBRATION

In the optical sensor tracking and calibration, the integration of advanced techniques and algorithms plays an important role in ensuring accurate and reliable measurements. This section delves into the complex process of optimizing optical sensor systems for accurate data acquisition and tracking. Using the most advanced methods, the sensor is fine-tuned to interpret real-world situations with greater fidelity. The following discussion sheds light on important aspects of optical sensor monitoring and calibration, shedding light on the fundamentals, methodology, and technical intricacies that improve accuracy. sensor accuracy and delivers consistent results. Before starting the calibration, the optical is provided with a pre-calibration code which turns on as soon as the system is turned on as shown in listing 6. A window pops up for the optical senser tracking the ArUco marker attached to the moveable gondola. This code works independently of the camera calibration process and serves a different purpose. While the camera calibration focuses on the internal and external parameters of the camera to correct distortion and obtain accurate measurements, the code provided helps to measure the distance between ArUco markers in an image. These measurements are important for spatial inference and are used to confirm the accuracy of camera calibration. This listing 6. below shows the code which is therefore placed before the calibration code to ensure that the measured distances are obtained before the calibration is performed, allowing the calibrated camera to be evaluated using the distance measurements.

```
def saveMarkers ():
# Generates and saves ArUco marker images

def find_marker (image, debug=True):
# Detects ArUco markers in an image and returns positions

def measureDistances (image, debug=True):
# Calculates distances and center positions between detected markers
# Other code and explanations...
```

Listing 6: Calibration of optical sensor.

Table 4.2 elaborates on how the system starts by importing the required libraries and configuring the necessary settings. It establishes a connection to a predefined marker dictionary, determines marker lengths and distances, and generates ArUco maps for calibration. ArUco detection-related settings are configured, such as mapping and detection settings. There are some default parameters set already from the preloaded images of ArUco markers to make the calibration quicker.

camera_matrix	813.2316651804839	0.0	18.8091945202511
	0.0	812.4656520524218	239.0215181258818
	0.0	0.0	1.0
dist_coeff	-0.10152261503170229	0.3596072448749952	-0.0002385073606136108
	0.0021837521467388604	0.0632101247576826	

Table 4.3: Pre-load camera matrix and distortion coefficient values.

The core of the code revolves around calibrating the camera using ArUco markers. It loads the calibration images, detects angles and marker IDs, and accumulates them. The number of detected markers for each image is recorded in a counter array. The calibration process calculates the camera matrix and distortion coefficients. These results are then stored in a YAML file for future use. The final calibrated camera matrix is displayed to the user. The following table shows all the parameters which were used to configure the calibration set.

Parameter	**Value**
root	Path object pointing to the root directory of the script.
calibrate_camera	True
calib_imgs_path	Path to the directory "aruco_data" within the script's root directory.
markerLength	3.75
markerSeparation	0.5
aruco_dict	Predefined dictionary DICT_6X6_1000
board	Grid board of size 4x5 with marker length and separation.
arucoParams	Detector parameters for marker detection.

Table 4.4: Parameters setting for Optical tracking.

In the parameters, the root is used to get the absolute path of the directory containing the script file. **Calibrate_camera** indicates to control camera calibration or verify results. **Calib_imgs_path** builds the folder path containing the calibration images. **Marker length** and **markerSeparation** sets the length and distance of the markers of ArUco markers for detection, **aruco_dict** creates an ArUco array with the specified dimensions and marker information. **aucoParams** create a setting for ArUco marker detection.

```
def calibrate_camera_and_display_output ():
    if calibrate_camera:
        img_list = [cv2.imread(str(fn)) for fn in calib_imgs_path.
glob('*.jpg')]
        corners_list, id_list, first = [], [], True
        # Detect markers in calibration images
        for im in tqdm(img_list):
        # ... (detect markers and accumulate corners_list, id_list,
counter)
        # Calibrate camera using detected markers
        ret, mtx, dist, _, _ = aruco. calibrateCameraAruco (corners_list,
        id_list, counter, board, img_gray.shape, None, None)
        # Display calibration output
        output = f"Camera matrix:\n {np. round (mtx, 2)} \nStored in
        calibration. yaml along with distortion coefficients:\n{dist}"
calibrate_camera_and_display_output ().
```

Listing 7: Calibration method and output display for optical tracking data

In listing 7, the basic calibration process includes loading the calibration image, detecting the marker, and calculating the distortion factor and camera matrix. Corner points and IDs of detected markers are accumulated. The number of detected markers for each image is recorded in a counter array. Using this data, the code proceeds to calculate the camera matrix (**mtx**) and the distortion factor (**dist**) using the function aruco. calibCameraAruco.

In the provided listing 7, the internal and external calibration is done mainly in the function **Caliber_camera_and_display_output ().** Below is a breakdown of the key code steps involved in internal and external calibration:

Intrinsic calibration involves estimating the camera's intrinsic parameters, including the camera matrix (**mtx**) and the coefficient of distortion (**dist**). These settings correct lens distortion and project the 3D scene onto the 2D image plane. Here, the function **aruco. CalibCameraAruco ()** estimates the camera matrix (**mtx**) and distortion factor (**dist**) using the angles of the detected ArUco marker (**corners_list**) and their respective identifiers (**id_list**). Calculated **mtx** and **dist** parameters help correct lens distortion and are important for accurate image conversion.

External correction includes an estimate of the real-world camera position and orientation for each image taken. This step allows for precise mapping of 3D points on the 2D image plane. Although the code provided does not directly calculate the external parameters, it sets the stage for future external calibration. The **rvecs** (rotation vector) and **tvecs** (translation vector) returned by the **aruco. calibreCameraAruco ()** function can be used to calculate the external parameters.

The **rvecs** and **tvecs** contain information about camera rotation and translation, which is necessary for accurate real-world camera positioning. In summary, the function **Calib_camera_and_display_output ()** in the provided code mainly performs the intrinsic correction by estimating the camera matrix (**mtx**) and the distortion factor (**dist**). In addition, it provides the necessary information (**rvecs and tvecs**) for potential external calibration, even though the code does not do this step explicitly.

```
def measureDistances (image, debug=True):
    # Detect ArUco markers in the image
    if markers [0] and markers [1]:  # If two markers are detected
        # Calculate the Euclidean distance between the marker centers
        distance = np. sqrt ((markers [0][0] - markers [1][0]) **2 +
(markers [0][1] - markers [1][1]) **2)
        distance mm = distance * 0.1 # Convert distance to millimeters
        # Calculate the X-coordinate of the center between the markers
        centerX = (markers [0][0] + markers [1][0]) / 2
        centerX_mm = centerX * 0.1 # Convert X-coordinate to millimeters
        centerY = (markers [0][1] + markers [1][1]) / 2 # Calculate the Y-
coordinate of the center
```

Listing 8: Distance measurement in optical tracking.

Listing 8. shows the Distance measuring function that takes an image as input and detects ArUco markers using the **find_marker** function. If two markers are detected, it calculates the distance between their centers and their respective horizontal axis positions. These measurements are then converted to millimeters and scaled. Debug images are also added to the image for illustrative purposes. Finally, the function returns the normalized X coordinates and center distance as a fraction of the image width, or None if no marker is detected. The procedure shows the complete calculation steps needed to calculate the position of the tracking object. The calculated results are then forwarded to the GUI for the representation of data.

4.3.6. DATA COLLECTION AND INTEGRATION

To collect the data, teaching tracking system utilizes multiple sensors such as acoustic, magnetic and optical to capture real time information about the object's orientation, position, movement and location. This also uses real time continue monitoring of the object to get the precise information. For the data collection, Python script is used in this system to integrate the sensors and get specified information with reference to multiple parameters. The data about the moving objects distance to origin, optical tracking and positioning is made possible using the particular sensor. These sensor sense and transmit the data to the Arduino UNO or directly to Raspberry Pi in case of optical sensing.

The Arduino UNO starts the serial communication, sets the LED pin as an output, and initializes the sensors. Specifically, it sets up the MPU9250 sensor which is a magnetic sensor, with a specific address, and calibrates the accelerometer and gyroscope. It first checks if a specified time interval has passed. If it has, the Arduino UNO updates the sensor values, reads the ultrasonic sensor data, and formats the collected data into a string, which is a readable text form.

The provided listing 9 showcases a streamlined process of establishing a serial connection with Arduino-based sensor systems, receiving sensor data, and efficiently extracting specific measurements for analysis and application. The SerialConnection class provides methods to connect to available ports. The open () method is used to establish a connection by specifying a port or by searching for available ports. After a successful connection, a serial communication link is initialized with the selected or default baud rate.

```
class SerialConnection:
    def open (self, port=None, baudrate=1000000):
        # ... connection code ...
```

Listing 9: Connection establishment

```
class ArduinoSlave (SerialConnection):
    def _readSensors(self):
                # ... data extraction and processing ...
    def addRecvCallback (self, cb):
        self. RecvCbs.append(cb)
```

Listing 10: Sensor data reception and processing.

The ArduinoSlave class provides methods for extracting and processing measurements from specific sensors. For example, getAcousticRTTs () retrieves audio time measurements, getMagneticField () retrieves magnetic field measurements, getAccelValues () accesses accelerometer values, and getTemperature () is designed for tempreture readings degree.

```
class ArduinoSlave (SerialConnection):
    def getAcousticRTTs(self):
        # ... extraction code ...

    def getMagneticField(self):
        # ... extraction code ...

    def getAccelValues(self):
        # ... extraction code ...

    def getTemperature(self):
        # ... extraction code ...
```

Listing 11: Specific sensor measurements.

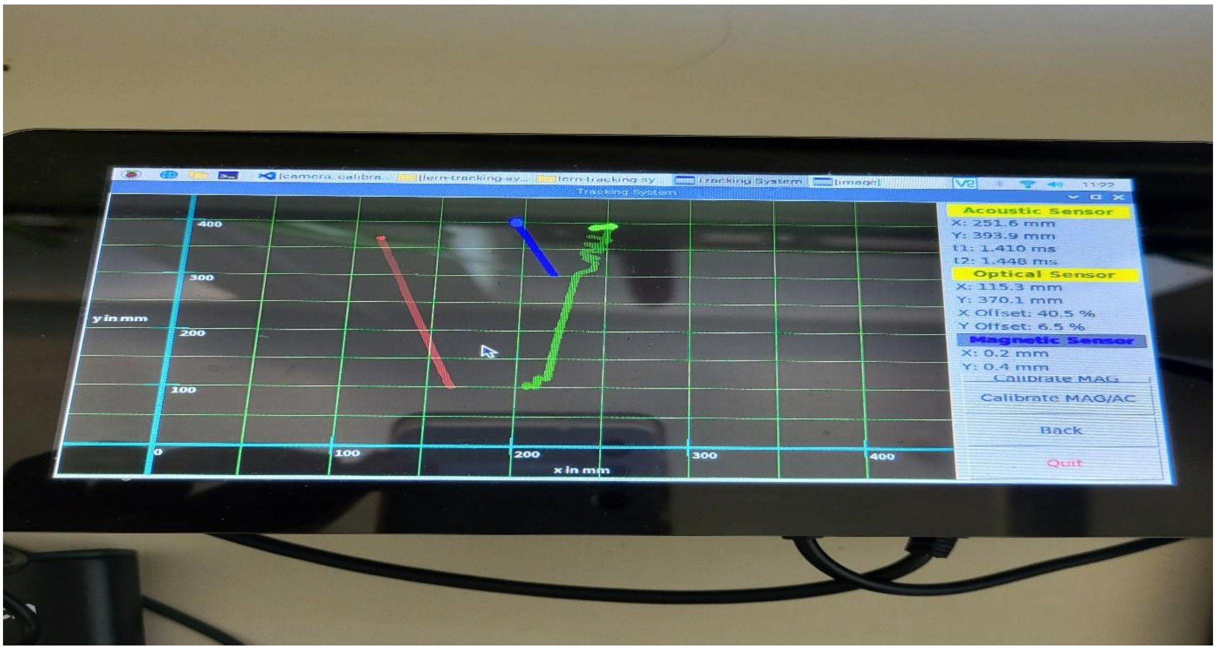
In the main execution block, an ArduinoSlave instance is created. The callback function is defined to handle specific sensor data using the methods provided by the ArduinoSlave class. This callback is registered using the addRecvCallback () method. The serial connection is established using the open () method and the permanent loop ensures continuous data processing.

The ‘serial communication’ is used to establish communication between raspberry pi and Arduino Uno as shown in listing 11. Opening and closing of connections along with transmittance of data and checking the connection status is done using this class. The Arduino Uno sends the data of acoustic and magnetic sensors for further calculation, acquisition, and calibration. The raspberry pi enters in a loop with Arduino UNO where it continuously sleeps for 1 second. The wakeup is followed whenever new data is received. It allows real-time interaction with Arduino UNO and enables further data process.

In summary, the code exemplifies the efficient establishment of serial connections with Arduino-based sensors, continuous reception of sensor data, and specialized methods for extracting and utilizing specific measurements. This integration framework provides a robust foundation for further analysis and diverse applications.

4.3.7. GRAPHICAL REPRESENTATION OF DATA

One more important aspect of implementation is the representation of data gathered, analyzed, and interpreted. For presentation, the graphical presentation is most commonly used among systems. Other presentations can also be possible but a systematic and easy-to-understand presentation of complex data is to make it available in the form of a graph.



Acoustic Sensor
X: 251.6 mm
Y: 393.9 mm
t1: 1.410 ms
t2: 1.448 ms
Optical Sensor
X: 115.3 mm
Y: 370.1 mm
X Offset: 40.5 %
Y Offset: 6.5 %
Magnetic Sensor
X: 0.2 mm
Y: 0.4 mm

Figure 4.6: Graphical representation of data gathered from sensors on an LCD in Low-cost tracking system.

The above figure shows the interpreted results in a graphical form. The coordinates determine the specified area of the movement of object and the multi colors shows the tracking of sensors in that area such as green for acoustic sensor, purple for magnetic sensor and red for optical sensor. It determines the position, movement and orientation of the sensors in the system. The system proves to be user friendly and easy to understand by determining specific color to the specific sensor. The movement and tracking of sensor studies is possible by moving the sensors in specified area and to understand the differences between the tracking of sensors. This GUI window also shows the values in numerical form on the right side of the screen where the sensors along with their offsets are continuously changed as the sensor position is manipulated. The colored line shows that the green color specifies for acoustic sensor, dark blue for magnetic sensor and red for the optical sensor. These colors are adjustable and can be amended easily from users' choice.

The scale is set to mm scale due to multiple factors. Precision is possible using millimeter scale compared to other units, this system deal with precise measurements so the smaller units provide more fine details. It simplifies the data complexity, calculations and analysis across different sensor outputs. It also provides good balance between granularity and scalability, allowing precise tracking of small.

4.4. EVALUATION OF THE IMPLEMENTATION: A USER CASE STUDY

Has already been discussed in the previous chapters the complete data acquisition process and the description of the sensor used to build a user-friendly, easy-to-handle, affordable, and more accuracy in terms of tracking, low-cost tracking system. This system is aimed to provide the learner, educator, and students with a brief description of the concept of tracking. The tracking, as discussed earlier is a very important concept behind xR systems. To evaluate this tracking system a study for evaluation was made consisting of a group of students.

The case study incorporates n=5 students that participated in the following questionnaire survey about the evaluation and usage of the Learning tracking system. The goal of the study was to investigate the understanding of the tracking concept through this system and several characteristics such as ease of use, attitude towards the system, subjective norm, and system accessibility, Attitude has been identified as a cause of intention. Behavioral intention is a measure of how likely they are to utilize the system. Subjective norms and system accessibility can only be measured by the difficulty of accessing this Learning tracking system.

Quantitative research involves collecting numerical data and analysing it using statistical methods. The goal is to produce objective, empirical data that can be measured and expressed numerically [113]. Qualitative research focuses on a multi-method approach, accompanied by an interpretive, naturalistic approach to its subject. That is, qualitative researchers study things in their natural environment, trying to understand or interpret phenomena based on the meanings people give them. [114]. This study was made using quantitative and qualitative questions. Quantitative questions elicit responses that can be easily quantified, such as yes/no answers or rating scale choices, while qualitative questions invite open-ended responses that may require thematic analysis or interpretation. The original questionnaire (attached 8.3 QUESTIONNAIRE) is here dived into quantitative and qualitative questions with the respective question numbers.

Quantitative Questions:

1. Have you used the tracking system developed for this project? (Yes / No)

This first question serves as a filter, it was asked to that whether the participants have interacted with the system which help to understand their experience.

3. How would you describe your experience using the tracking system? (Choose one option)
   a) Very positive b) Positive c) Neutral d) Negative e) Very negative

The third question provides an assessment of the participant’s overall experience and offers a general understanding of user sentiments.

4. Did the tracking system enhance your understanding of the concept of tracking in xR environments? (Yes/No)

The fourth question assess the educational value of the system for understanding the tracking concept. It helps to find out whether the system has fulfilled its intended purpose of education or not.

6. Were you able to easily interpret and analyse the tracked data presented in graphical or numerical form? (Yes/No)

The sixth allows to indicate if participants found the data visually and analytically accessible or not. It also provides insights into the system's usability.

7. Did the combination of three sensors improve the tracking system's performance? (Yes/No)

The seventh question is the importance to evaluate the impact of sensor combination measuring the effectiveness of the multi-sensor approach and inform about design modifications.

8. Would you recommend the tracking system to other students for Learning about tracking in xR environments? (Yes/No)

This question shows the satisfaction of the participant and their willingness to endorse this system to the other students.

Qualitative Questions:

5. Which sensor(s) did you find most effective in tracking the object's movement?

This qualitative question gets insights into users' views about the sensors that stood out in terms of performance and accuracy.

9. What suggestions do you have for further improving the tracking system? (Open-ended response)

The ninth question is open-ended which invites participants to give their views or detailed feedback about the improvement and suggestion for this system.

2. Which aspects of the tracking system did you find most interesting or useful?

The second question uncovers the features that participants found appealing. It helps to identify the strength and potential understanding of the system

Before handing over the questionnaire, the students were briefly explained about the usability of the system by the subject supervisor. The students were also explained about the usage protocols to use the system. The students were given a chance to operate the system on their own, identifying multiple aspects and capabilities of the Learning tracking system. After careful consideration and using the system, the participants were asked to fill out the above questionnaire without any compensation.

4.5. COMPARISON OF ART AND LOW-COST TEACHING TRACKING SYSTEM

In this study, an efficient and accurate tracking system developed by ART (Advanced Realtime Tracking) named SMARTTRACK3/M (Details in 3.3 XR In Education Industry) was used for evaluating the precision and tracking of our low-cost Learning tracking system. The DTRACK is the software solution widely used in multiple industries such as VR, AR, Robotics, etc for motion capture, tracking, or position tracking. In contrast to this system, low-cost Learning tracking systems also offers good accuracy and offers multiple sensors tracking technology although it is mainly usable for Learning perspective. One major difference is the multi-camera setup, DTRACK supports multi-camera setups, which enables the system to get a wide field of view and track multiple objects, whereas low-cost Learning tracking systems have only one camera sensing and can only track a single object at one time. The following table shows a systematic comparison of both the system used in this study of evaluation.

Feature	**SMARTTRACK3/M3/M**	**Low-Cost Teaching Tracking System**
Tracking Technology	Marker-based	Ultrasonic, Magnetic, and Marker-based
Tracking Accuracy	High precision and sub-millimetre up to 2x2 meters	?
Multi-Camera Setup	Yes	No
Real-Time Data Streaming	Yes	Yes
Integration with Software	Yes, supports Industry Standards	Python script with Visual Studio
Cost	Relatively expensive	cost-effective
Tracking Applications	VR, AR, robotics, animation, etc.	Educational and basic tracking tasks
Implementation Complexity	Advanced, requires Expertise	Moderate, suitable for Beginners

Table 4.5: Comparison between ART system and Low-cost tracking system.

Table 4.4 is a conclusion table for features of ART SMARTTRACK3/M3/M and low-cost Learning tracking system. shows a comparative comparison of the ART system with a teaching tracking system. As discussed in the Literature review Chapter, the ART system is a marker-based system, which means it uses markers for tracking the object whereas the teaching tracking system uses sensor technology for tracking an object. Tracking accuracy indicates the difference between the actual 3D position of an object and the position reported by tracking measurements [115].

In this case ART shares a high accuracy whereas tracking system accuracy is not yet known. This study also helps to find out the accuracy of each sensor concerning the ART system. Both systems have real-time data streaming and have integration with software.

To perform the evaluation, the setup was this system can be easily shown in the Figure 4.7. On the left side is the ART system and on the right is the Low-cost tracking system. The ART system is calibrated and fixed to a place to not get distorted data. The procedure follows using a needle-shaped structure having spheres on it. This needle is placed on the tracking object, in our case Gondola. The gondola is moved in different positions to get positional data. The reading is generated by low-cost tracking systems as well as DTRACK respectively. The result generated is compared from both systems to evaluate the accuracy and efficiency of our system.

Figure 4.7: Evaluation setup from left to right: ART SMARTTRACK, Performer and Teaching Tracking System.

5. RESULTS AND DISCUSSION

5.1. USER CASE STUDY

After completing the necessary protocols and getting details about the teaching tracking system from the Professor of Virtual reality, the participants were allowed to use the system with ease. The students operated the system from calibration to data acquisition with every sensor. Each student was requested to fill out the questionnaire to evaluate the usage ability, concept building, and understating of the concept of tracking for students. The following results discuss the details of the evaluation of the system concerning student learning.

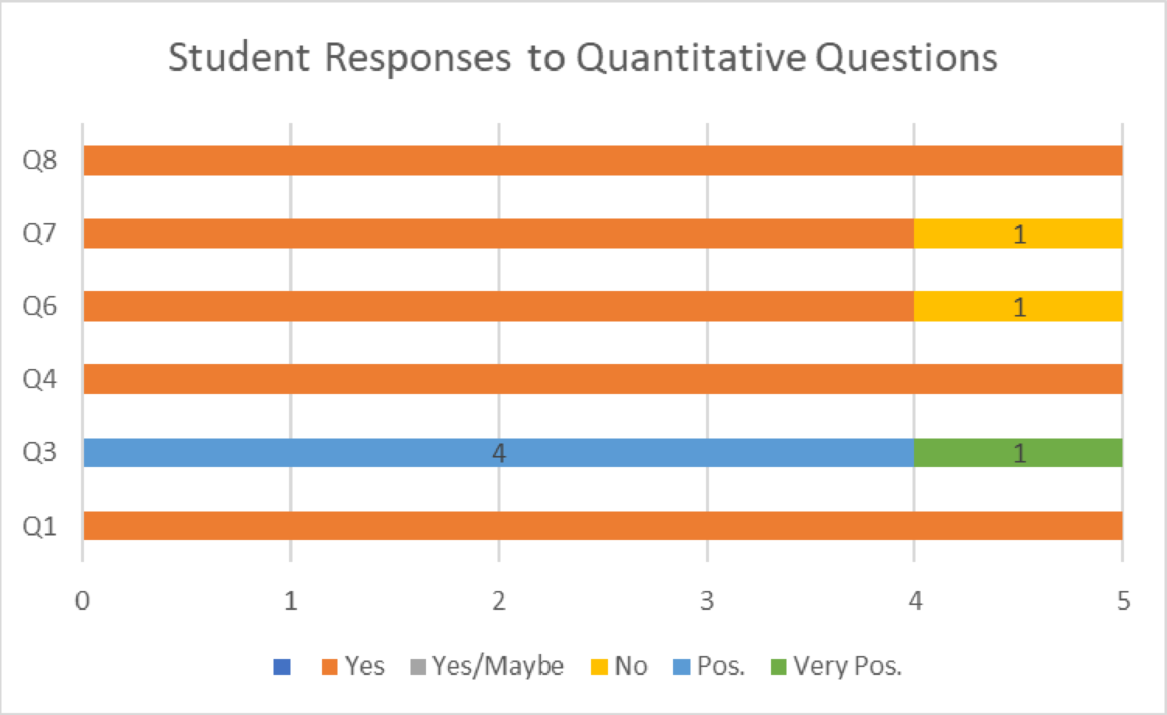


Figure 5.1: Stacked bar graph for quantitative questions responses.

A stacked bar chart briefly illustrates participant response to the tracking system in xR environments. In particular, each participant participated in the system, representing full adoption (100% "yes" responses). Most students expressed their experience positively. Four participants reported a positive experience, representing an 80% positive response, while a very positive response from one participant indicated exceptional satisfaction, representing a 20% positive response. In addition, all respondents confirmed the educational value of the monitoring system as they unanimously agreed that it increased their understanding of monitoring in xR settings (100% “Yes” responses). When it comes to interpreting the data, the graph highlights that most people found it accessible. Four students reported that the monitored information was easy to understand, with 80% responding positively, emphasizing the user-friendly design of the system. However, an individual participant's response to difficulties provides valuable information about potential usability improvements (20% negative response rate). The multisensory approach produced interesting results. Four participants (80%) believed that the combination of three sensors improved the performance of the system, indicating its effectiveness. However, the disagreement of one student (20%) provides a valuable perspective for further research and possible improvement of the sensor approach. Finally, the unanimous acceptance of the educational value of the monitoring system is significant. All participants enthusiastically recommended the system to fellow students who want to learn about monitoring in xR environments (100% "Yes"). This consensus underlines its potential as an effective tool in educational settings.

For the answers to question 2 and question 9 the responses from students are gathered here as a word cloud: "to learn about the weak points of the tracking system and how to counter them “, „most interesting part was the behavior of magnetic sensor", "the way someone can distort the values of sensors using noise, etc.", "not sure", "using in games" "adjustments and rotation of sensors", "add more sensors from different directions", "to add a rotation of sensors", "more sensitivity to sensors, more accuracy", "optimized algorithm calculates accurate orientation"

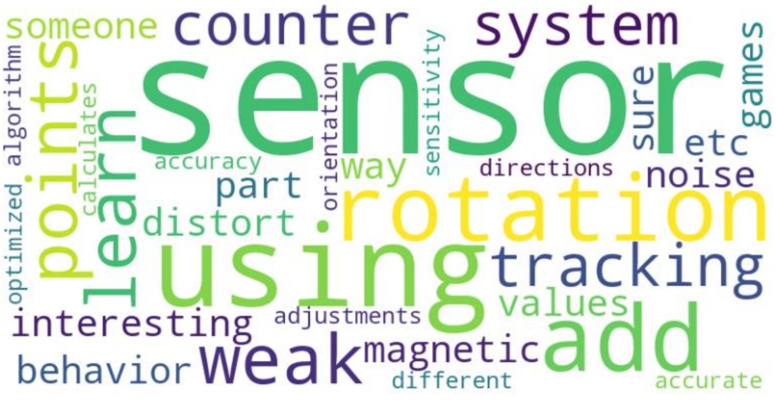


Figure 5.2: Word cloud for the responses of students.

In Figure 5.2 participants' responses to question 2 revealed a diverse perspective on their experiences with the monitoring system. Some were interested in the behavior of the magnetic sensor, while others sought to exploit system weaknesses and countermeasures. Gaming applications were considered as well as recommendations for adjusting and rotating the sensors to improve performance. The answers to question 4 mainly focused on recommendations to improve accuracy through sensor sensitivity and optimized algorithms. These diverse reviews highlight proactive research, technical understanding, and interest of participants in improving tracking system capabilities.

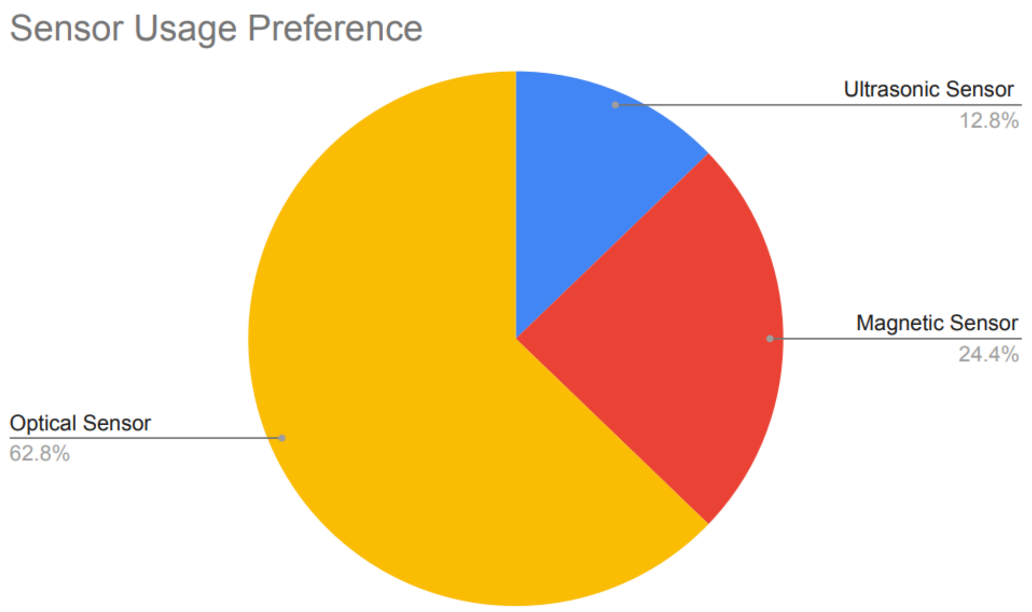


Figure 5.3: A pie chart shows the sensor using qualitative performance based on tracking data.

In Figure 5.3. it is shown that this system comprises 3 different sensors namely Ultrasonic or acoustic sensor, magnetic sensor, and optical sensor. The sensor collaboration accuracy was highly achieved in this system which helped the students to learn tracking from different aspects. On the behalf of student's response, a pie chart was generated to understand which sensor was easier to use and understandable for the students. The optical sensor proved to be easier and more interpretable by students in contrast with magnetic and ultrasonic sensors. The visibility of tracked objects was more profound according to students. On the other hand, magnetic sensors and ultrasonic sensors also contributed to the understanding of the tracking concept. Overall study shows that this system was able to provide a proper understanding of tracking concepts, calibration procedures, data acquisitions technique, and interpretation of results

5.2. DISCUSSION OVER COMPARISON OF ART AND TEACHING TRACKING SYSTEM

To assess the effectiveness and performance of the teaching tracking system, a practical evaluation is performed. The following test and exercise are done about ART SMATTRACK3.

Test for Accuracy for Acoustic, magnetic, and optical sensors: Evaluate the tracking system's accuracy by contrasting the positions it returns with the locations that are known. To determine the tracking errors, utilize a calibration target with known coordinates. The created system's repeatability has been assessed to make an accurate claim. For the quality of the localized data, the tracking performance's accuracy and precision are crucial. As a result, a validation procedure must be carried out [116]. The test is conducted in 4 quadrants as shown:

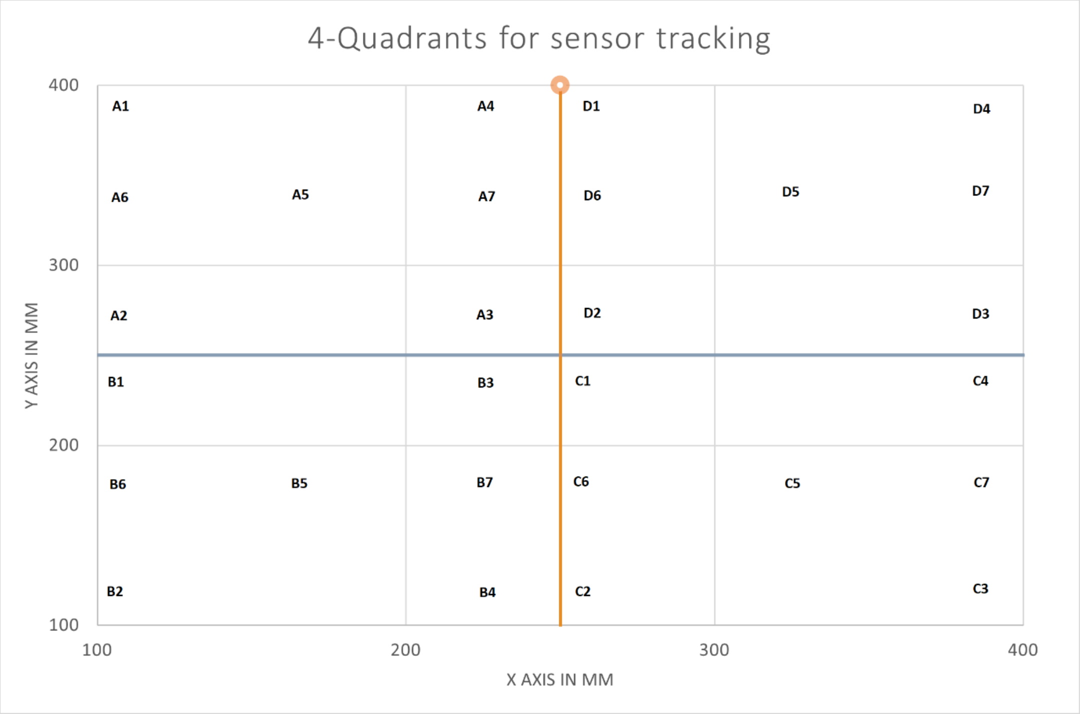


Figure 5.4: Quadrant for data measurement.

In Figure 5.4 it is shown that the original grid for tracking is always from 0 in mm but here the grid of 100x400 is chosen because of the tracking area. The sensor moving capability is restricted under 100mm from origin which is why a minimum distance of 100mm is chosen. The quadrants are divided into 4 equal quarters and in every quadrant 7 readings are noted to get minimum distance from one another.

Readings were measured at every point for all sensors and ART SMARTTRACK3/M3. This table shows every single reading noted concerning the ART SMARTTRACK3/M3 system. These readings were used for the accuracy test. All the values of sensors and ART are referenced to every single point in the above-mentioned quadrant. The gondola was moved in different directions for the measurement of data. There were a few processes and calculations that must be made to determine the accuracy of each sensor in the various quadrants. Since the system uses both the X and Y axes, accuracy and precision will be determined. Here is a detailed explanation of how to accomplish it:

1. Accuracy:

Accuracy measures how close the sensor readings are to the true values or the reference system (in this case, the readings from the ART SMARTTRACK3/M system).
To calculate accuracy, the Mean Absolute Error (MAE) formula is used:
MAE=n1∑i=1n∣Reference$i$−Sensor$i$∣
Where:

- n is the number of data points.
- Reference$i$ is the reading from the reference system for the ith data point.
- SSensor$i$ is the reading from your sensor for the ith data point.

The lower the MAE value, the closer the sensor readings are to the reference system, indicating higher accuracy.

The results shown in the scattered graphs are distributed by the quadrants using different graph grids. The area of the graph would be considerable only between 100 mm and 400 mm, therefore any value apart would not be considered as an actual or accurate value. The graph shows the results of sensors and reference values from ART system for calculating the accuracies of tracking capabilities of sensors at multiple positions.

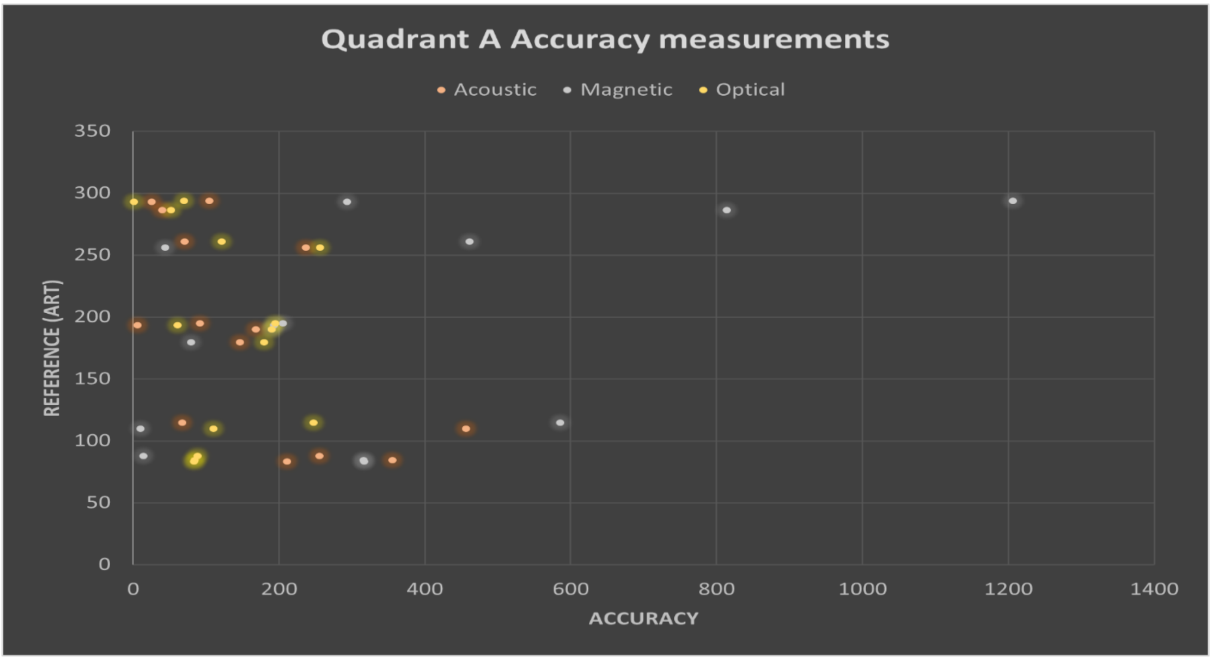


Figure 5.5: Quadrant A accuracy measurements.

The scatter chart shown in Figure 5.5 gives a thorough overview of sensor performance along many axes. Accuracy variations can be seen when looking at the data from the Acoustic sensor. Accuracy results for the Y-axis in the area of 100 mm to 400mm in particular show fluctuation, ranging from significantly lower to very close to reference values. Similar results may be seen from the Y-axis Acoustic sensor, which exhibits differing degrees of accuracy about various reference points. The graph covers an area of 400mm because the original axis of the system is a 400mm grid, therefore, values apart that are considered zero or not trackable.

The magnetic sensors, on the other hand, constantly display high levels of precision along both axes. The X and Y-axis Magnetic sensors' data points are noticeably not concentrated around the reference values rather it is spread all over the graph, demonstrating their accuracy and high fluctuations in values with a small movement due to magnetic field-related information.

The results from the optical sensor show a range of accuracy levels. The accuracy of the X-axis optical sensor fluctuates, with some data points well aligned with reference values and others falling far short. The precision of the Y-axis optical sensor varies, with some spots being precisely aligned and others straying from the reference.

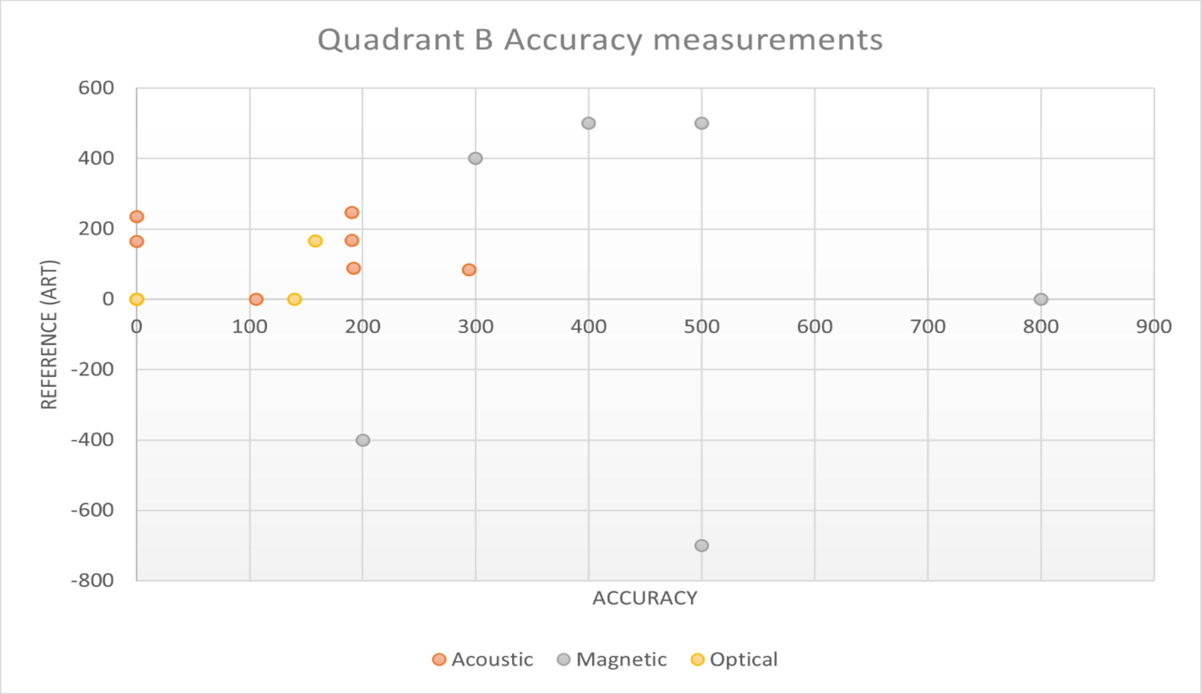


Figure 5.6: Quadrant B accuracy measurements.

The scatter graph in Figure 5.6 offers a perceptive representation of the connection between accuracy and reference values for various sensors. The chart's data points provide useful information about how well each sort of sensor works in terms of measuring data reliably.
The graph above clearly shows the bottom left corner of the original graph. Most of the values of sensors lie before 200 mm which secures quadrant C axis values forming the points B1 to B7 as shown above.

When examining the data from the Acoustic sensor, a dispersed pattern of data points can be seen between 0 mm to 300 mm. This suggests that the accuracy of this sensor varies. While some spots are more evenly distributed and indicate deviations from the reference, others are closer to the reference values as between 0mm to 200 mm at the Y axis and indicate correct measurements. This variation implies that the accuracy of the Acoustic sensors varies depending on the type of measurement.

The Magnetic sensors, on the other hand, display a distinct spread of data points that are far from their respective reference values even in the negatives which shows that at these positions the magnetic field is not in the range or either disturbed by external factors. This spread implies that these sensors don't always deliver precise measurements. While no places match the reference, all are dispersed, which suggests difficulties in gathering accurate data. This emphasizes how difficult it is for magnetic sensors to measure data precisely in this situation.

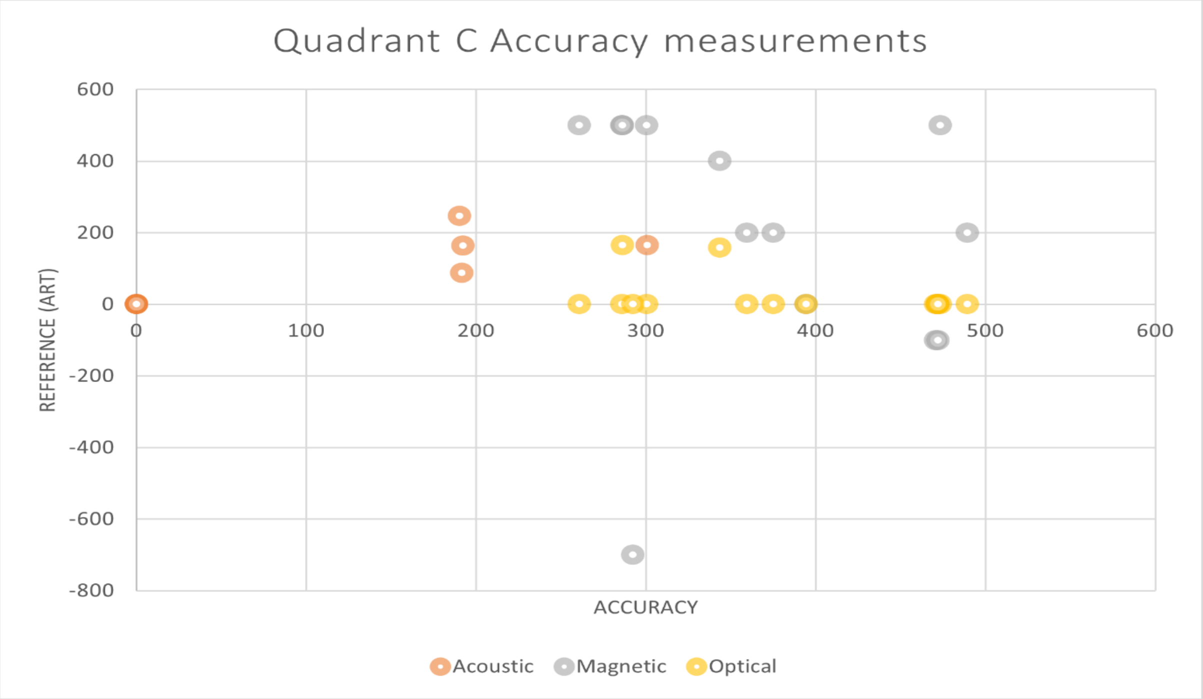


Figure 5.7 : Quadrant C accuracy measurements.

As in quadrant C, it is clearly shown in the graph in Figure 5.7 that the most values lie between the grid of 200 mm and 400mm. It is because that quadrant C is the bottom-right part of the original graph axis forming an axis more than the value 200mm. This quadrant includes the measured points from C1 to C7 as shown above. Let's talk about the outcomes based on the scatter graph, where the accuracy is displayed on the x-axis and the reference is plotted on the y-axis for the provided data.

The scatter graph effectively conveys how accuracy and reference values interact for various sensor kinds and axes. The graph reveals clear patterns and trends as we look further into it. There is a noticeable tendency among the Acoustic sensors (both X and Y axes) that higher reference values are associated with improved accuracy. This indicates that the precision of measurements increases as the reference value rises, according to the positive correlation. The magnetic sensors (X and Y axes), in contrast, display a more dispersed distribution, with accuracy values that range across a variety of reference values. Unexpectedly, situations occur when higher accuracy and lower reference values are connected, emphasizing instances of accurate measurements even for relatively low references.

The optical sensors, in contrast, stand out since their accuracy levels are typically very low. The majority of the points are grouped together at the origin, showing measurement inaccuracy when compared to other reference values. The constant lack of precision over a range of reference values raises the possibility that these sensors are underperforming or that they need to be recalibrated in order to improve accuracy.

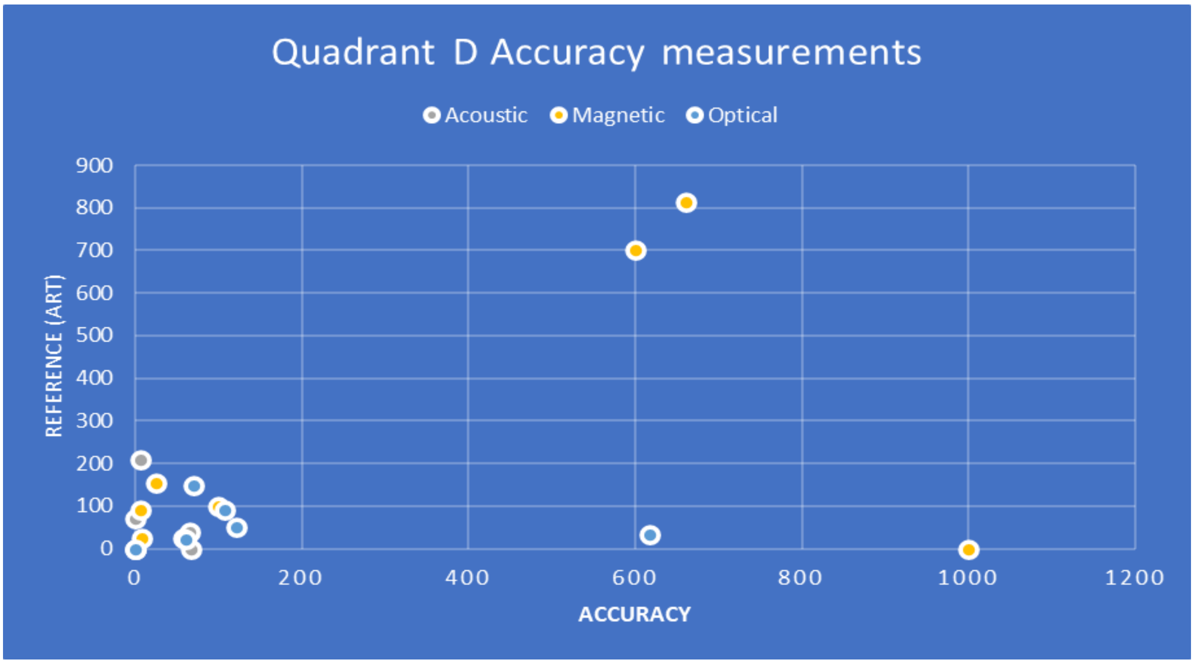


Figure 5.8: Quadrant D accuracy measurements.

The given data as shown in the graph in Figure 5.8, consists of measurements made by several sensors along various axes, together with accompanying accuracy measurements and reference values. We made sporadic graphs showing the correlation between accuracy and reference values to assess the results. The sensor types (Acoustic, Magnetic, and Optical) and axis (X, Y) are used to categorize the data points. For the Acoustic sensor type, there appears to be an overall moderately favorable association between accuracy for the X and Y axes and reference values. Although there is significant data fluctuation, accuracy tends to grow as reference values do. There is less of a consistent relationship between accuracy and reference values for magnetic sensor types. There appears to be a positive link in some instances while a weak or even negative correlation is present in others. This shows that factors other than just the reference values have an impact on the accuracy of the magnetic sensor data.

A mixed relationship exists between accuracy and reference values for optical sensor types. Some data points show a positive link, while others don't clearly reveal a trend. This suggests that reference values may not have a significant impact on accuracy measures for optical sensors.

In summary, the analysis of the scatter chart provides important insights into how various sensors function along various axes. Particularly in the data from the Acoustic sensor, a thorough overview of accuracy fluctuations is obvious. Notably, the Acoustic sensor's Y-axis accuracy fluctuates between 100 and 400 mm, showing variable degrees of precision in comparison to reference values Due to the system's initial axis, the chart's range is 400 mm; values outside of that are regarded as zero or untraceable.

In contrast to the findings, magnetic sensors consistently do not exhibit high levels of precision along both the X and Y axes. In contrast to the other sensors, the X and Y-axis Magnetic sensors' data points are widely scattered on the graph, demonstrating their accuracy and susceptibility to value variations even with little motions caused by magnetic field influences. This implies that although the magnetic sensors are capable of making exact measurements, their readings can be affected by minute variations in the magnetic field around them.

The optical sensors, however, show a range of accuracy levels. The accuracy of the X-axis optical sensor fluctuates, with some data points well aligned to reference values and others drastically out of alignment. Similar variations in precision may be seen in the Y-axis optical sensor, with some locations aligning well with references and others deviating. This suggests that optical sensors may have trouble providing reliable and precise results since their accuracy fluctuates across different measurement settings.

The scatter chart also highlights key geographic areas of importance. For instance, Quadrant B in Figure displays the various accuracy ranges of several sensor kinds. Moderate precision is demonstrated by optical sensors, large variations are seen by magnetic sensors and mixed precision is seen by acoustic sensors. Quadrant C in Figure, which corresponds to the bottom-right corner of the original graph axis, emphasizes the predominance of values between 200 mm and 400 mm. The scatter chart provides a clear picture of the sensor performance and successfully captures these subtleties.

The dispersed pattern of the Acoustic sensor data displays varied accuracy levels upon closer inspection. A few places, especially those between 0 mm and 200 mm on the Y-axis, closely coincide with reference values in all quadrants, indicating precise readings. Other places, on the other hand, show deviations, indicating changes in measurement precision. The magnetic sensor data points, on the other hand, are dispersed widely from their reference values and even include negative values. This dispersion highlights the difficulties in taking accurate measurements. This shows that magnetic sensors have trouble consistently providing accurate readings at different sites, potentially because of outside magnetic forces. The data from optical sensors also shows various degrees of accuracy. While some points hardly deviate from the reference values, others are in close agreement. This shows that optical sensors provide a moderate level of precision but may have trouble providing precise readings in various environments.

The scatter chart also reveals remarkable patterns when sensor accuracy is contrasted with reference values. Higher reference values and increased accuracy are positively correlated for Acoustic sensors, especially for the Y-axis. This suggests that greater measurement precision and higher reference values are related. Magnetic sensors, in contrast, display a more dispersed distribution with varied accuracy levels over various reference values. Unexpectedly, situations occur where greater accuracy is attained while having lower reference values, indicating the capability of providing accurate measurements even with relatively low references. However, optical sensors constantly display lower levels of precision, which are mainly centered around the origin. This suggests that optical sensors have difficulty providing accurate measurements across different reference values. In general, the scatter chart successfully illustrates the intricate connections between accuracy, reference values, and sensor behavior.

In conclusion, the scatter chart sheds important light on how accuracy, reference values, and sensor types interact. The patterns seen emphasize how crucial it is to take particular sensor behaviors and features into account when interpreting the data. Understanding the complex relationships is aided by this depiction, which also assists in decision-making for sensor applications.

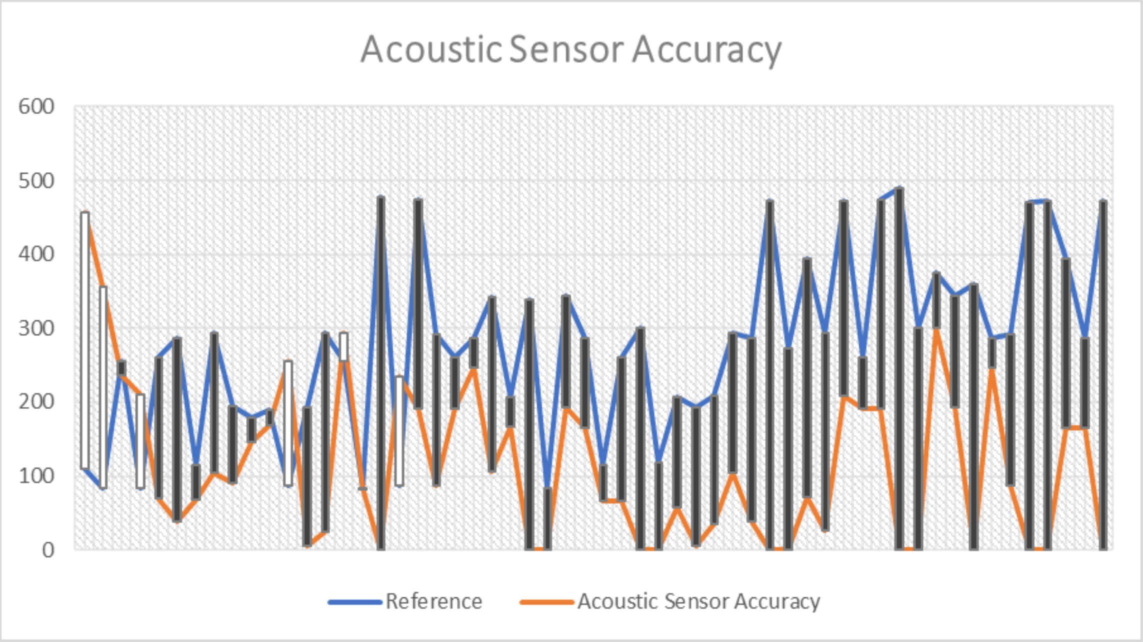


Figure 5.9: Line Graph showing Acoustic Sensor accuracy with Reference values.

The acoustic sensor's readings in the graph in Figure 5.9, which are based on sound waves and are affected by physical factors, are closely tied to the values of the ART system. The line graph shown previously demonstrates how the acoustic sensor closely matches the ART values between 100mm and 300mm. However, as the ART values rise above this range, a glaring disparity between the two systems' measures starts to appear. This discrepancy may be ascribed to potential noise or other influences that affect how accurate the readings from the acoustic sensor are.

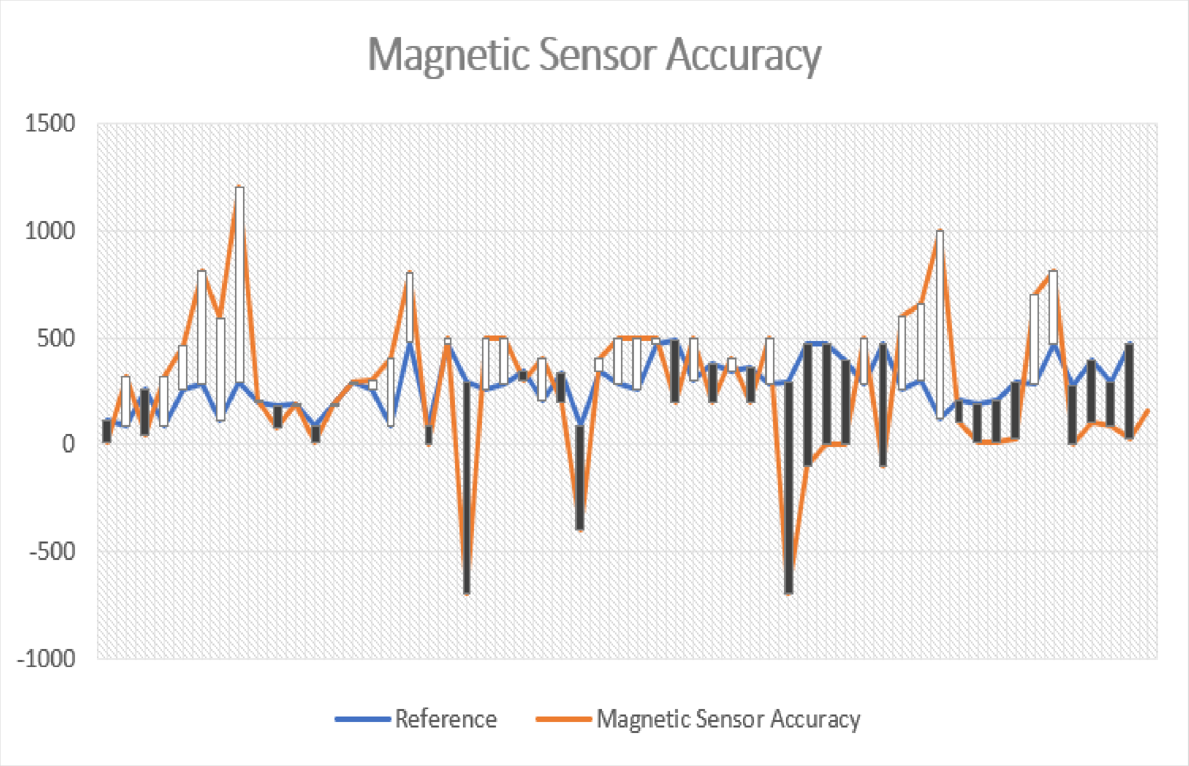


Figure 5.10: Line Graph showing Magnetic Sensor accuracy with Reference values.

In contrast to the other sensors, the magnetic sensor's behavior shown in the graph in Figure 5.10 exhibits a distinctive pattern. Although its data seems to be near the reference values of the ART system, a substantial dispersion becomes apparent as the reference values rise. The data from the magnetic sensor shows a linear trend, the blacking bars showing the negtive turns and white bars showing the positive turns, suggesting some degree of agreement between the sensor's observations and the ART reference values. However, once the reference values rise above a particular threshold, this regularity is broken, leading to a wider scatter of data points. This occurrence indicates that the magnetic sensor's precision may be at its best inside a particular range of reference values, but that its measurements become less accurate outside of that range.

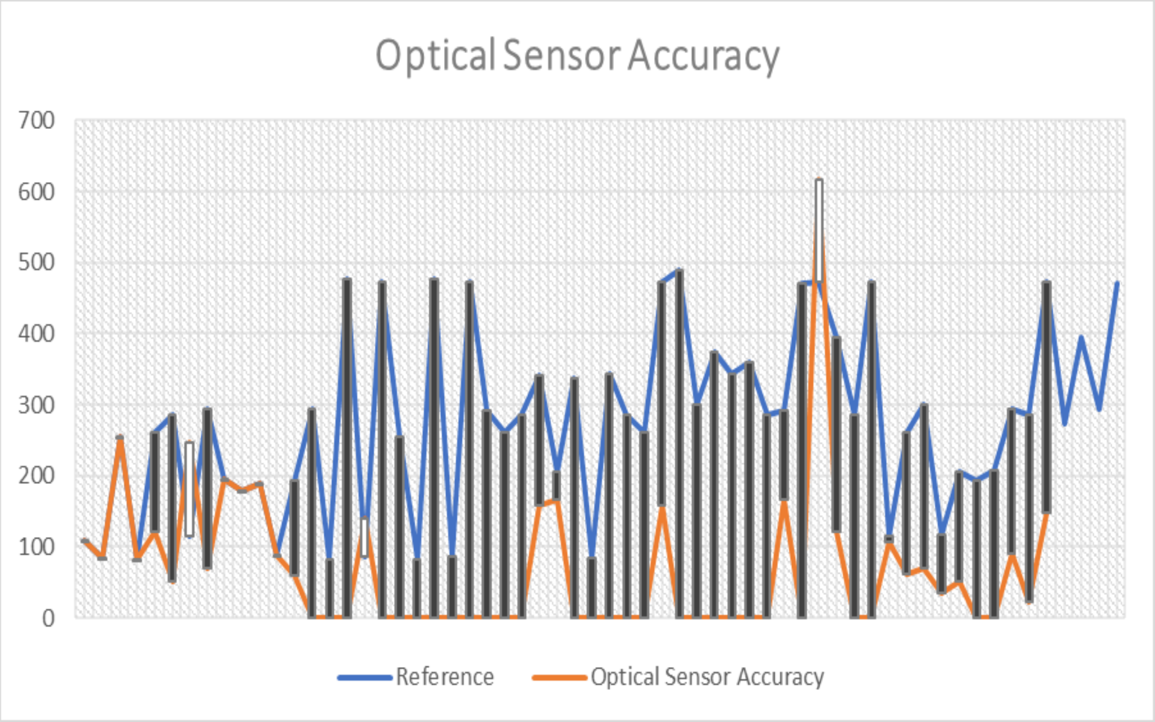


Figure 5.11: Line Graph showing Optical Sensor accuracy with Reference values.

A weak association between the optical sensor's data and the reference values may be seen in the graph in Figure 5.11. This means that the optical sensor can only track objects within a certain spatial range. The data points don't reveal any substantial variances, but they do indicate a clear contrast when compared to the reference values of the ART system. It becomes clear that the optical sensor's tracking performance is best in a small space for example between 100 to 300 mm, maybe indicating that it is suited for particular localized applications. The data also hints at some degree of opposition or divergence in respect to the ART values, pointing to the possibility that the optical sensor's behavior may display oscillations that differ from the reference pattern.

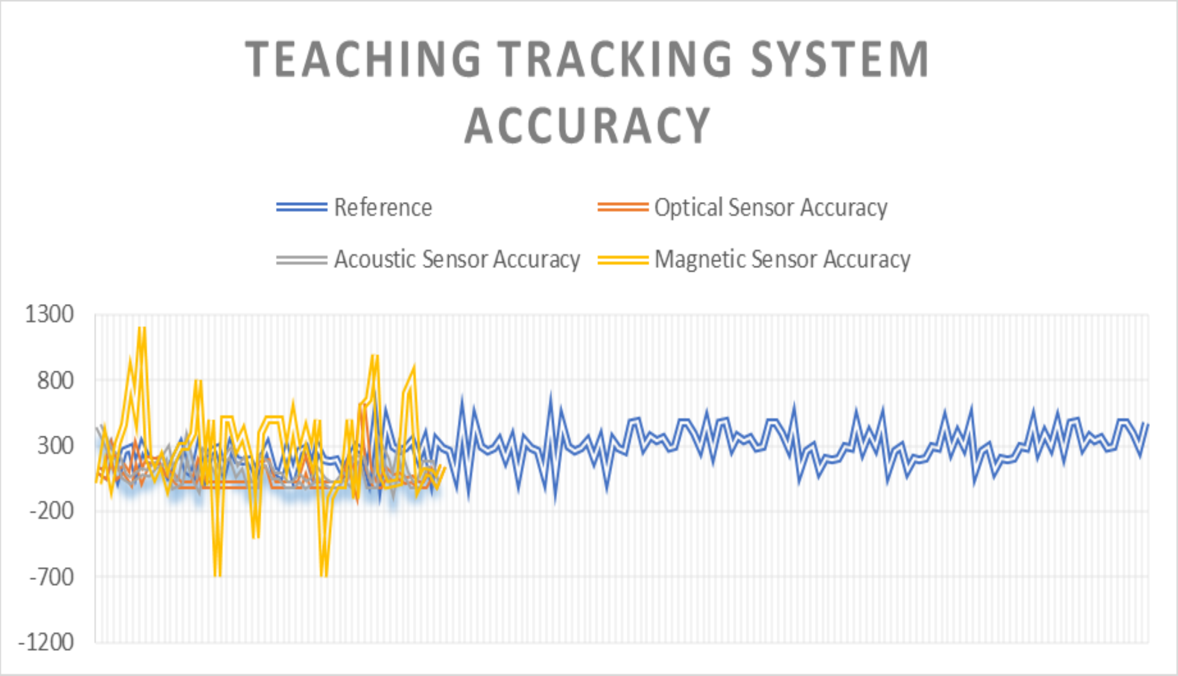


Figure 5.12: Teaching tracking system accuracy line graph consisting of Acoustic sensor, Magnetic sensor, Optical sensor and Reference values (ART SMARTTRACK3/M).

By looking at the graph in Figure 5.12 an insightful grasp of the teaching tracking system's effectiveness can be gained by evaluating its accuracy using a variety of sensors, specifically acoustic, magnetic, and optical ones. The results from the acoustic sensor, which relies on sound waves and is affected by physical conditions, show a strong association with the values of the ART system. The acoustic sensor closely matches ART values in the ranges of 100mm and 300mm, as shown in the line graph. Deviations outside of this range, however, point to possible external factors that could affect how accurate acoustic sensor results are. This shows that even while the sensor efficiently collects data when it is in its ideal range, outside influences may cause accuracy to vary.

The magnetic sensor, in comparison, has a distinctive behavior. Although its data initially looks to be close to the reference values of the ART system, a significant dispersion becomes apparent as the reference values rise. The linear trend of the data points to a relationship between sensor measurements and ART reference values, but this coherence decreases beyond a particular threshold. As a result, accuracy appears to decline outside of a certain reference value range while precision appears to be at its best within that range. This suggests that the performance of the magnetic sensor is susceptible to changes in the reference values, which is crucial to take into account for applications requiring consistent measurements.

5.3. TEACHING TRACKING SYSTEM LIMITATIONS

The use of multiple sensors, calibration processes, and data acquisition techniques can be challenging for students therefore these must be taught briefly to understand the system. As this is a cost-effective system which is not accurate as the high-cost systems are but it generates an idea and learning of the tracking system, therefore the system needs to be calibrated every time which can be time-consuming. While working with multiple sensors there might be challenges in achieving precise alignment between sensors. The system may require regular maintenance and updates to keep up with evolving technologies and potential changes in teaching methodologies.

6. CONCLUSION

6.1. SUMMARY OF KEY FINDINGS

In conclusion, a thorough evaluation of the characteristics of a tracking system revealed a nuanced understanding of its effectiveness and potential applications. Scatterplot analysis explored sensors behaviour and highlights strengths and challenges on different axes. The accuracy of the acoustic sensor varied and its Y-axis shows variations between 100- and 400-mm. Magnetic sensors can measure accurately, but were sensitive to magnetic field effects. Optical sensors showed a variable accuracy, indicating difficulty in obtaining consistent results. These findings highlighted the need for tailored sensor selection based on application requirements.

Additionally, evaluating participant responses with stacked bar graphs provided valuable information about the user experience and educational value of the system. The positive experiences and unanimous support of participants demonstrated the success of the system in increasing understanding in xR environments. Although challenges in data interpretation and the multi-sensor approach were identified, these findings provide opportunities for system improvement and optimization.

A tracking system incorporating student feedback and sensor analysis demonstrates its educational potential and practical applicability, making it a valuable tool for tracking in xR environments. The combination of technical views and user perspectives provided a comprehensive overview of the functions of the system and contributes to its continued development and use both in educational environments and elsewhere.

6.2. IMPLICATION OF SYSTEM FOR LEARNING PERSPECTIVE

The evaluation of the learning effect of the monitoring system is investigated. The analysis includes both the technical efficiency of the system and its pedagogical value. The overview of system suitability for various applications and help educators and learners make informed decisions about sensor selection based on accuracy requirements. Participant responses, such as the stacked bar chart, highlight the system's role as effective learning.

The distribution of sensor accuracy along an axis not only talks about the technical implementation, but also gives teachers a nuanced picture of the strengths and limitations of the sensors. This enables the design of a customized curriculum that meets the learning objectives.

Similarly, feedback from participants shows that the system fulfils its educational goal of increasing understanding of monitoring in xR environments. The unanimous recommendation emphasizes its potential contribution to the educational experience.

This evaluation covers both technical and pedagogical perspectives and emphasizes the importance of the system as a learning system that benefits both students and teachers. By integrating this knowledge, teachers can effectively use the capabilities of the tracking system to improve student understanding and engagement in xR environments. This section provides a comprehensive overview of how the system translates into meaningful learning outcomes.

7. BIBLIOGRAPHY

[1] A. Burova et al., “Utilizing VR and gaze tracking to develop AR solutions for Industrial Maintenance,” Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020. doi:10.1145/3313831.3376405

[2] What is Extended Reality (XR)? | IxDF https://www.interaction-design.org/literature/topics/extended-reality-xr Accessed: 2023-02-11

[3] J. Bloomberg, “Digitization, Digitalization, And Digital Transformation: Confuse Them at Your Peril,” Forbes, May 17, 2018. Available: https://www.forbes.com/sites/jasonbloomberg/2018/04/29/digitization-digitalization-and-digital-transformation-confuse-them-at-your-peril/ Accessed: 2023-02-11

[4] “Tracking definition and meaning | Collins English Dictionary,” www.collinsdictionary.com. https://www.collinsdictionary.com/dictionary/english/tracking Accessed: 2023-08-14

[5] B. Marr, “The Fascinating History and Evolution of Extended Reality (XR) – Covering AR, VR And MR,” Bernard Marr, Jul. 02, 2021. https://bernardmarr.com/the-fascinating-history-and-evolution-of-extended-reality-xr-covering-ar-vr-and-mr/ (accessed May 12, 2023).

[6] R. Skarbez, M. Smith, and M. C. Whitton, “Revisiting Milgram and Kishino’s Reality-Virtuality Continuum,” Frontiers in Virtual Reality, vol. 2, p. Figure 1, Mar. 2021, doi: https://doi.org/10.3389/frvir.2021.647997.

[7] MILGRAM, P., AND KISHINO, F. "A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS." IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS, E77-D (12), 1994, PP. 1321–1329.

[8],[10],[41],[50],[57],[115] G. Burdea and P. Coiffet, Virtual reality technology. [Indianapolis, Indiana]: [Wiley-IEEE Press], 2017.

[9] R. T. Azuma, “A Survey of Augmented Reality,” Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355–385, 1997, Doi: https://doi.org/10.1162/pres.1997.6.4.355

[11] G. Nageswara Rao, K. Aruna Kumari, D. Ravi Shankar, and K. G. Kharade, “A comparative study of augmented reality-based head-worn display devices,” Materials Today: Proceedings, pp. 40–43, Feb. 2021, Doi: https://doi.org/10.1016/j.matpr.2020.12.400.

[12] S. Lemmer, “What is the difference between XR, AR, VR and MR | HYVE Innovation,” HYVE, Aug. 06, 2020. https://www.hyve.net/en/blog/all-about-virtual-reality/ (accessed May 10, 2023).

[13],[14],[15] Creating Full Sensory Experiences: The Future of AR/VR/MR/XR | Radiant Vision Systems https://www.radiantvisionsystems.com/blog/creating-full-sensory-experiences-future-ar/vr/mr/xr Accessed: 2023-08-14

[16],[17] R. Klamma, “Perceptual Foundations of Augmented Reality | The Open Augmented Reality Teaching Book - Create and Code Augmented Reality!” The Open Augmented Reality Teaching Book. https://codereality.net/ar-for-eu-book/chapter/foundations/design/perception/ Accessed: 2023-08-14

[18] The role of perceptual manipulations of XR in neurological rehabilitation | Frontiers Research Topic
https://www.frontiersin.org/reseARch-topics/48098/the-role-of-perceptual-manipulations-of-xr-in-neurological-rehabilitation
Accessed: 2023-08-14

[19] Seth, A., Vance, J.M. & Oliver, J.H. Virtual Reality for assembly methods prototyping: a review. Virtual Reality 15, 5–20 (2011). https://doi.org/10.1007/s10055-009-0153-y

[20] S. M. LaValle, Virtual Reality. Cambridge: Cambridge University Press, 2023, pp. 38–40. Accessed: May 06, 2023. [Online]. Available: https://www.cambridge.org/core/books/virtual-reality/0EC0542C3688B5CA3E9733011DC8DEC2

[21] W. R. Sherman and A. B. Craig, Understanding virtual reality: interface, application, and design. San Francisco, Ca: Morgan Kaufmann, 2003, pp. 70–80. Accessed: Aug. 1, 2023. [Online]. Available: http://www.arise.mae.usp.br/wp-content/uploads/2018/03/Understanding-Virtual-Reality-Interface-Application-and-Design-The-Morgan-Kaufmann-Series-in-Computer-Graphics-.pdf

[22] “3DoF vs 6DoF in VR: What is the difference?” Smart VR Lab, Mar. 04, 2021. https://www.smartvrlab.nl/3dof-vs-6dof-in-vr/ (accessed May 06, 2023).

[23] “VIVETM | Discover Virtual Reality Beyond Imagination,” Vive.com, 2011. https://www.vive.com (accessed Mar. 12, 2023).

[24] “Pimax | VR Headsets, HMDs & Equipment,” pimax.com. https://pimax.com/ (accessed Jun. 10, 2023).

[25] “VIVE Wireless Adapter | VIVE European Union,” www.vive.com. https://www.vive.com/eu/accessory/wireless-adapter-full-pack/ (accessed Jun. 12, 2023).

[26] “Overview of Google Cardboard - Cardboard Manufacturer Help,” support.google.com. https://support.google.com/cardboard/manufacturers/answer/6321454?hl=en (accessed Jul. 09, 2023).

[27] S. Greenwold, In Form. Massachusetts Institute of Technology: Simon Greenwold, 2003. Accessed: Apr. 12, 2023. [Online]. Available: https://acg.media.mit.edu/people/simong/thesis/SpatialComputing.pdf

[28] “System - Definition and Examples - Biology Online Dictionary,” Biology Articles, Tutorials & Dictionary Online, Feb. 14, 2021. https://www.biologyonline.com/dictionary/system (accessed May 12, 2023).

[29] “Tracking,” Wikipedia, Apr. 24, 2023. https://en.wikipedia.org/wiki/Tracking (accessed Jul. 12, 2023).

[30] “How do common virtual reality tracking systems work?” www.mechatech.co.uk. https://www.mechatech.co.uk/journal/how-do-common-virtual-reality-tracking-systems-work (accessed Apr. 02, 2023).

[31] J. Steinbring, C. Mandery, F. Pfaff, Florian Faion, T. Asfour, and U. D. Hanebeck, “Real-time whole-body human motion tracking based on unlabelled markers,” pp. 1–2, Sep. 2016, Doi: https://doi.org/10.1109/mfi.2016.7849550

[32] “A Comprehensive Guide to Asset Tracking Technologies (2021),” WISER Systems. https://www.wisersystems.com/blog/asset-tracking-technologies-comprehensive-guide (accessed Jun. 10, 20203).

[33] Anonymous, “GS1 Barcodes - Standards | GS1,” www.gs1.org, Jan. 14, 2015. https://www.gs1.org/standards/barcodes (accessed Jun. 19, 2023).

[34] Centre for Devices and Radiological Health, “Radio Frequency Identification RFID,” U.S. Food and Drug Administration, Sep. 17, 2018. https://www.fda.gov/radiation-emitting-products/electromagnetic-compatibility-emc/radio-frequency-identification-rfid (accessed Jul. 09, 2023).

[35],[36] S. Manifold, “A Comprehensive Guide to Asset Tracking Technologies,” Ubisense, Oct. 27, 2022. https://ubisense.com/guide-asset-tracking-technologies/ (accessed Aug. 01, 2023).

[37] “Bluetooth Low Energy,” Android Developers. https://developer.android.com/guide/topics/connectivity/bluetooth/ble-overview (accessed Jul. 02, 2023).

[38] Cisco, “What Is Wi-Fi?” Cisco. https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html (accessed Jul. 16, 2023).

[39] “What is LPWAN (low-power wide area network)? - Definition from WhatIs.com,” IoT Agenda. https://www.techtarget.com/iotagenda/definition/LPWAN-low-power-wide-area-network (accessed Aug. 02, 2023).

[40] R. Mobus and U. Kolbe, "Multi-target multi-object tracking, sensor fusion of radar and infrared," IEEE Intelligent Vehicles Symposium, 2004, Parma, Italy, 2004, pp. 732-737, Doi: 10.1109/IVS.2004.1336475

[42] “MPU-9250 Product Specification Revision 1.1 MPU-9250 Product Specification.” Accessed: Jul. 20, 2023. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf

[43] “picamera — Picamera 1.13 Documentation,” Readthedocs.io, 2013. https://picamera.readthedocs.io/en/release-1.13/ (accessed Jun. 08, 2023).

[44] ElecFreaks, “Ultrasonic Ranging Module HC -SR04,” 2011. Accessed: May 22, 2023. [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf

[45] “I2C Communication Protocol,” GeeksforGeeks, Feb. 01, 2021. https://www.geeksforgeeks.org/i2c-communication-protocol/ (accessed Feb. 07, 2023).

[46] M. Grusin, “Serial Peripheral Interface (SPI) - learn.sparkfun.com,” Sparkfun.com, 2019. https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all (accessed May 27, 2023).

[47] “Difference between Accuracy and Resolution: Understand these Metrology Terminologies,” www.rvmagnetics.com. https://www.rvmagnetics.com/accuracy-vs-resolution-understand-these-metrology-terminologies-110 (accessed Jul. 20, 2023).

[48] “What is field of view (FOV)?” WhatIs.com. https://www.techtarget.com/whatis/definition/field-of-view-FOV (accessed Jun. 20, 2023).

[49] “A Beginner’s guide to frame rates in movies | Adobe,” www.adobe.com. https://www.adobe.com/creativecloud/video/discover/frame-rate.html (accessed Jul. 19, 2023).

[51] “Measure Distance with a Sonar Sensor on an Arduino,” Allaboutcircuits.com, Jan. 16, 2016. https://www.allaboutcircuits.com/projects/measure-distance-with-a-sonar-sensor-on-an-arduino/ (accessed Jun. 01, 20203).

[52] “HC-SR04-P Ultraschall-Modul Entfernungsmesser,” Roboter-Bausatz.de. https://www.roboter-bausatz.de/p/hc-sr04-p-ultraschall-modul-entfernungsmesser (accessed Jul. 14, 2023).

[53] “How Does an Ultrasonic Sensor Work? - Lesson,” TeachEngineering.org. https://www.teachengineering.org/lessons/view/umo_sensorswork_lesson06 (accessed Jul. 10, 2023).

[54] “MIYBot,” hackaday.io. https://hackaday.io/project/25984-miybot (accessed Jul. 12, 2023).

[55] HowToMechatronics, “Arduino Radar Project - HowToMechatronics,” HowToMechatronics, Jul. 28, 2015. https://howtomechatronics.com/projects/arduino-radar-project/ (accessed Jun. 09, 2023).

[56] Admin, “Arduino Ultrasonic Range Finder with HC-SR04 Sensor on OLED,” How to Electronics, Apr. 03, 2019. https://how2electronics.com/arduino-ultrasonic-range-finder-hc-sr04-oled-display/ (accessed Jun. 18, 2023).

[58],[59],[60] WatElectronics, “Magnetic Sensor: Working, Types, Interfacing & Its Applications,” WatElectronics.com, Aug. 13, 2022. https://www.watelectronics.com/magnetic-sensor (accessed Jun. 08, 2023).

[61] P. F. Schewe, “Elucidating the Hall Effect,” www.aps.org, Dec. 1998. https://www.aps.org/publications/apsnews/199812/hall.cfm (accessed Jul. 03, 2023).

[62] “4.3. Optical tracking — MPHY0026 documentation,” mphy0026.readthedocs.io. https://mphy0026.readthedocs.io/en/latest/tracking/optical.html (accessed Jun. 18, 2023).

[63] S. Meeks, “Patient Positioning Using Optical and Ultrasound Techniques.” Accessed: Jul. 19, 2023. [Online]. Available: https://www.aapm.org/meetings/03SS/Presentations/Meeks.pdf

[64] “Optical Tracking in Medical Physics Advanced Lab Course Experiment at the Department of Medical Physics,” 2021. Accessed: Jun. 14, 2023. [Online]. Available: https://www.med.physik.uni-muenchen.de/studium_lehre/advanced_laboratory_courses/p2/P2_manual.pdf

[65] “Image plane,” Wikipedia, Oct. 22, 2020. https://en.wikipedia.org/wiki/Image_plane (accessed May 10, 2023).

[66] S. Skwiot, “What Is Focal Length and How to Use It - 42West,” 42 West, the Adorama Learning Center, Mar. 01, 2021. https://www.adorama.com/alc/understanding-focal-length-and-how-to-use-it/ (accessed Jul. 10, 2023).

[67] onexute, “Raspberry Pi NoIR Camera Module V2.1,” onexute. https://onexute.com/en-de/products/raspberry-pi-noir-camera-module-v2-1-8mp-1080p-sensitive-webcam-rpi (accessed Jun. 14, 2023).

[68] “Raspberry Pi V2.1, 8MP, 1080p, Camera Module: Amazon.de: Computer & Accessories,” www.amazon.de. https://www.amazon.de/Raspberry-Pi-V2-1-1080P-Kamera-Modul/dp/B01ER2SKFS (accessed Jun. 17, 2023).

[69] E. Odemakinde, “Human Pose Estimation with Deep Learning - Ultimate Overview in 2021,” viso.ai, Jan. 08, 2021. https://viso.ai/deep-learning/pose-estimation-ultimate-overview/ (accessed Jul. 02, 2023).

[70] “OpenCV: Detection of ArUco Markers,” docs.opencv.org. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html (accessed Jul. 29, 2023).

[71],[72] “OpenCV: Detection of ArUco Markers,” docs.opencv.org. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html (accessed Jun. 04, 20203).

[73] S. D. Team, 22.3 Statistical vs. Computational Efficiency | Stan User’s Guide. Accessed: Jul. 19, 2023. [Online]. Available: https://mc-stan.org/docs/2_22/stan-users-guide/statistical-vs-computational-efficiency.html

[74] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, “Fiducial Markers for Pose Estimation,” Journal of Intelligent & Robotic Systems, vol. 101, no. 4, Mar. 2021, Doi: https://doi.org/10.1007/s10846-020-01307-9.

[75] “Arduino Uno 3D-Modell - TurboSquid 1829290,” www.turbosquid.com, Dec. 17, 2021. https://www.turbosquid.com/de/3d-models/arduino-uno-3d-model-1829290 (accessed Jul. 19, 2023).

[76] “What is UART (Universal Asynchronous Receiver/Transmitter)? | Definition from TechTarget,” WhatIs.com. https://www.techtarget.com/whatis/definition/UART-Universal-Asynchronous-Receiver-Transmitter (accessed Jun. 10, 2023).

[77] P. Gupta, “What is Arduino UNO? | Amazing 11 Features of Arduino UNO,” EDUCBA, Aug. 13, 2020. https://www.educba.com/what-is-arduino-uno/ (accessed Aug. 01, 2023).

[78] H. Tao, “Building Custom Real-Time Sensors for Virtual Reality Applications,” pp. 1–5, Jan. 2017, Accessed: Jul. 07, 2023. [Online]. Available: https://escholarship.org/content/qt83z2g6vq/qt83z2g6vq_noSplash_06bb4e3531130602a2839b635e745d2f.pdf

[79] C. BasuMallick, “What Is Raspberry Pi? Models, Features, and Uses,” Spiceworks, Aug. 26, 2022. https://www.spiceworks.com/tech/networking/articles/what-is-raspberry-pi/ (accessed Jul. 17, 2023).

[80] “Raspberry Pi 4 Model B (Raspberry-PI-4-4GB) od 1 629 Kč,” Zboží.cz. https://www.zbozi.cz/vyrobek/raspberry-pi-4-model-b-raspberry-pi-4-4gb/ (accessed Jun. 07, 2023).

[81] “LPDDR,” Wikipedia, Mar. 29, 2022. https://en.wikipedia.org/wiki/LPDDR (accessed Jul. 01, 2023).

[82] P. Lisenbee, “Generation Gap Between Students’ Needs and Teachers’ Use of Technology in Classrooms,” Journal of Literacy and Technology, vol. 17, no. 3, pp. 101–119, 2016, Accessed: 2023. [Online]. Available: http://www.literacyandtechnology.org/uploads/1/3/6/8/136889/jlt_v16_3_lisenbee.pdf

[83] A. S. is a freelance journalist with an M. A. in journalism from the U. of Georgia, “VR, AR and XR: What’s the Difference, and How Can They Transform Online Learning?” Technology Solutions That Drive Education, Dec. 20, 2021. https://edtechmagazine.com/higher/article/2021/12/vr-ar-and-xr-whats-difference-and-how-can-they-transform-online-learning-perfcon (accessed Jun. 02, 2023).

[84],[85] “VirtualElo!” hci.uni-wuerzburg.de. https://hci.uni-wuerzburg.de/projects/virtualelo/ (accessed Jul. 14, 2023).

[86] M. Yoder, “ISTE,” www.iste.org, May 19, 2016. https://www.iste.org/explore/toolbox/use-these-resources-to-bring-augmented-reality-to-your-classroom?articleid=739 (accessed Jul. 14, 2023).

[87] “Project VOISS – Giving Students a Voice,” Virtual Reality for Social and Emotional Learning. https://www.projectvoiss.org/#:~:text=Virtual%20Reality%20for%20Social%20and (accessed Jul. 05, 2023).

[88] P. Buń, F. Górski, R. Wichniarek, W. Kuczko, A. Hamrol, and P. Zawadzki, “Application of Low-cost Tracking Systems in Educational Training Applications,” Procedia Computer Science, vol. 75, no. 1877-0509, pp. 398–407, 2015, Doi: https://doi.org/10.1016/j.procs.2015.12.263.

[89] Y. Wang, S. Lu, and D. Harter, “Multi-Sensor Eye-Tracking Systems and Tools for Capturing Student Attention and Understanding Engagement in Learning: A Review,” IEEE Sensors Journal, vol. Vol 2 (3), pp. 1–1, 2021, Doi: https://doi.org/10.1109/jsen.2021.3105706.

[90] A. Sorriento et al., “Optical and Electromagnetic Tracking Systems for Biomedical Applications: A Critical Review on Potentialities and Limitations,” IEEE Reviews in Biomedical Engineering, vol. 13, pp. 212–232, 2020, Doi: https://doi.org/10.1109/rbme.2019.2939091.

[91],[92] X. Duan, “Magnetic tracking and positioning in endoscopy,” Elsevier eBooks, no. 9780128225325, pp. 287–338, Jan. 2022, Doi: https://doi.org/10.1016/b978-0-12-822532-5.00013-3.

[93],[94] W. Cheng et al., “Anatomy aware-based 2.5D bronchoscope tracking for image-guided bronchoscopic navigation,” Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, vol. 11, pp. 1–8, Dec. 2022, Doi: https://doi.org/10.1080/21681163.2022.2152728.

[95],[96] C. Bald, R. Bergholz, and G. Schmidt, “Automatic Localization of an Ultrasound Probe with the Help of Magnetic Sensors,” Current Directions in Biomedical Engineering, vol. 8, no. 2, pp. 317–320, Aug. 2022, Doi: https://doi.org/10.1515/cdbme-2022-1081.

[97] “VR headsets relaxing patients during surgery at St George’s,” St George’s University Hospitals NHS Foundation Trust, Dec. 26, 2019. https://www.stgeorges.nhs.uk/newsitem/vr-headsets-relaxing-patients-during-surgery-at-st-georges/ (accessed Jul. 14, 2023).

[98] Strath.ac.uk, 2023.https://www.strath.ac.uk/humanities/schoolofeducation/blog/remembering90ofwhatyoudo/ (accessed Jul. 10, 2023).

[99] M. Frąckiewicz, “The Role of Virtual Reality in Sports Training and Analysis – TS2 SPACE,” ts2.space, Mar. 24, 2023. https://ts2.space/en/the-role-of-virtual-reality-in-sports-training-and-analysis/ (accessed Jul. 19, 2023).

[100] “Augmented- and Virtual Reality Laboratory,” www.hs-anhalt.de. https://www.hs-anhalt.de/en/university/institutions/forschungs-transfer-und-gruenderzentrum/gz-entwurf/inkubatoren-coworking-spaces/inkubatoren/augmented-and-virtual-reality-laboratory.html#panel2-Augm (accessed Jul. 29, 2023).

[101] “TUM XR-LAB,” TUM Executive & Professional Education. https://www.lll.tum.de/about/digital-tools/xr-lab/ (accessed Jul. 29, 2023).

[102] “Seite nicht gefunden,” isp.fh-erfurt.de. https://isp.fh-erfurt.de/en/xr-pARt (accessed Jul. 28, 2023).

[103] “Home - U Bremen,” cgvr.cs.uni-bremen.de. https://cgvr.cs.uni-bremen.de/ (accessed Jul. 30, 2023).

[104] “GMU: Critical XR Lab - Medien Wiki,” www.uni-weimar.de. https://www.uni-weimar.de/kunst-und-gestaltung/wiki/GMU:Critical_XR_Lab (accessed Jul. 28, 2023).

[105] “Die HKA - Die Hochschule Karlsruhe: XR Lab Lab for Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) Karlsruhe University of Applied Sciences,” www.h-ka.de. https://www.h-ka.de/en/iaf/facilities/xr-lab (accessed Jul. 30, 2023).

[106] “DTRACK | Advanced Realtime Tracking GmbH & Co. KG,” ar-tracking.com. https://ar-tracking.com/en/product-program/dtrack (accessed Jul. 16, 2023).

[107] “What is Oculus Rift? - Definition from WhatIs.com,” WhatIs.com. https://www.techtarget.com/whatis/definition/Oculus-Rift (accessed Jul. 18, 2023).

[108] “7 Things about HTC Vive 1 -What is it?” Accessed: Jul. 26, 2023. [Online]. Available: https://www.gvsu.edu/cms4/asset/7E70FBB5-0BBC-EF4C-A56CBB9121AECA7F/7_things_about_htc_vive.pdf

[109] J. Ronsdorf, “Microsoft erklärt: Was ist Microsoft HoloLens? Definition & Funktionen | News Center Microsoft,” News Center Microsoft Deutschland, Dec. 04, 2020. https://news-microsoft-com.translate.goog/de-de/microsoft-erklaert-was-ist-microsoft-hololens-definition-funktionen/?_x_tr_sl=auto&_x_tr_tl=nl&_x_tr_hl=de. (Accessed Jul. 11, 2023).

[110] “The world’s most advanced virtual and mixed reality devices.,” Varjo.com. https://varjo.com/ (accessed Jul. 18, 20203).

[111] Vinay R, “C# vs. Python vs. C++ for Game Development - DZone Open Source,” dzone.com, Jun. 13, 2021. https://dzone.com/articles/c-vs-python-vs-c-for-game-development (accessed Jul. 15, 2023).

[112] N. Torska, “C++ Vs C#: Comparing Two Popular Programming Languages,” MarketSplash, Apr. 19, 2023. https://marketsplash.com/tutorials/cpp/cplusplus-vs-csharp/ (accessed Jul. 19, 2023).

[113] S. McLeod, “What’s the Difference between Qualitative and Quantitative research?” Simply Psychology, Jul. 31, 2023. https://www.simplypsychology.org/qualitative-quantitative.html (accessed Jul. 20, 2023).

[114] N. Denzin and Y. Lincoln, “Handbook of Qualitative Research,” British Journal of Educational Studies, vol. 42, no. 4, p. 2, Dec. 1994, doi: https://doi.org/10.2307/3121684.

[116] A. Wilken, F. Heilemann, Lezgin Turgut, and G. Helfrich, “CONCEPT FOR DIGITALISATION OF AN INSPECTION PROCESS USING HYBRID TRACKING OF PART AND PROBE FOR FUTURE MAINTENANCE AND DIGITAL TWINS,” 2021, pp. 5–7. Accessed: Jul. 27, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:244683493

[117] “Visual Studio vs Visual Studio Code - What’s Best In 2023?” www.turing.com. https://www.turing.com/kb/ultimate-guide-visual-studio-vs-visual-studio-code#why-should-vs-code-be-your-choice? (Accessed Aug. 03, 2023).

8. APPENDICES

8.1. SOFTWARES IDEs

In this subsection, the software integrated development environments (IDEs) used for this project will be discussed: **Visual Studio Code (VS Code) for Raspberry Pi and Arduino**. These IDEs have played an important role in supporting the development and programming of projects. Below, are the features and configurations of each IDE, highlighting their contributions to this work.

VISUAL STUDIO CODE (VS CODE) FOR RASPBERRY PI:

Visual Studio Code is a versatile and widely used code editor that offers a wide range of features, extensions, and integrations. When used for Raspberry Pi development, it provides an efficient environment for coding and project management. Remarkably quicker and quicker to start, VS Code offers IDE-like flexibility while remaining streamlined and user-friendly, making it the ultimate choice for web development, supported by extension tools and software support [117].

Python development:

Visual Studio Code (VS Code) greatly simplifies Python development on Raspberry Pi. Its code editor with syntax highlighting, autocomplete, and snippets simplified the coding process. The built-in terminal comes in handy for running scripts and managing packages. Overall, VS Code has greatly improved Python development on the Raspberry Pi platform. There were some package versions requirements used in programming and development as dependencies for this project, namely:

- numpy: 1.17.4
- pyglet: 1.5.15
- opencv_python: 4.4.0.44
- Pillow: 8.3.1
- pyserial: 3.5

- NumPy: A library for numerical computations in Python, providing support for arrays, matrices, and mathematical functions.
- pyglet: A multimedia library for Python that aids in creating games, multimedia applications, and graphical user interfaces.

- OpenCV Python: OpenCV (Open Source Computer Vision Library) is used for computer vision tasks like image processing, object detection, and machine learning.
- Pillow: A Python Imaging Library (PIL) fork that adds enhancements for image processing tasks, such as image manipulation and editing.

- pyserial: A library that facilitates serial communication (communication between devices through a serial port) in Python

ARDUINO IDE:

The Arduino IDE served as a dedicated environment suitable for programming Arduino microcontrollers in this project. It has greatly simplified the coding process by providing a user-friendly interface, writing code efficiently, and uploading code to the board conveniently. The IDE also made it easy to monitor the behaviour of sensors, allowing them to debug and tweak the code as needed. This dedicated platform ensures a seamless experience for developing and deploying code to Arduino boards, keeping the project running smoothly.

8.2. ADDITIONAL FIGURES

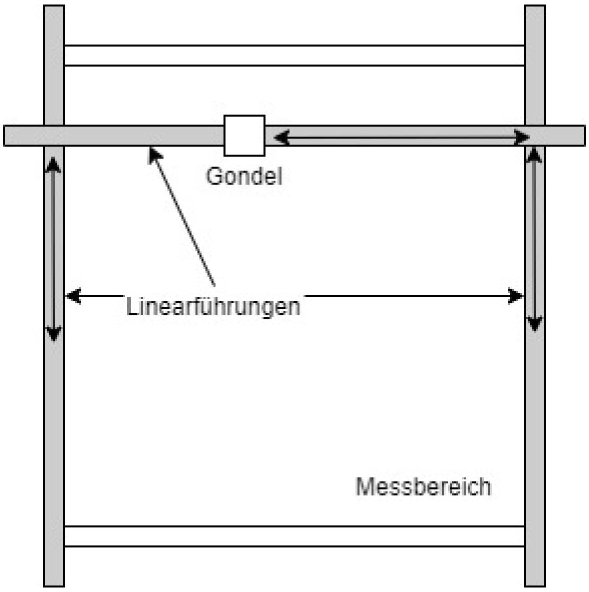


Figure 8.1 Aufbauen Stellung

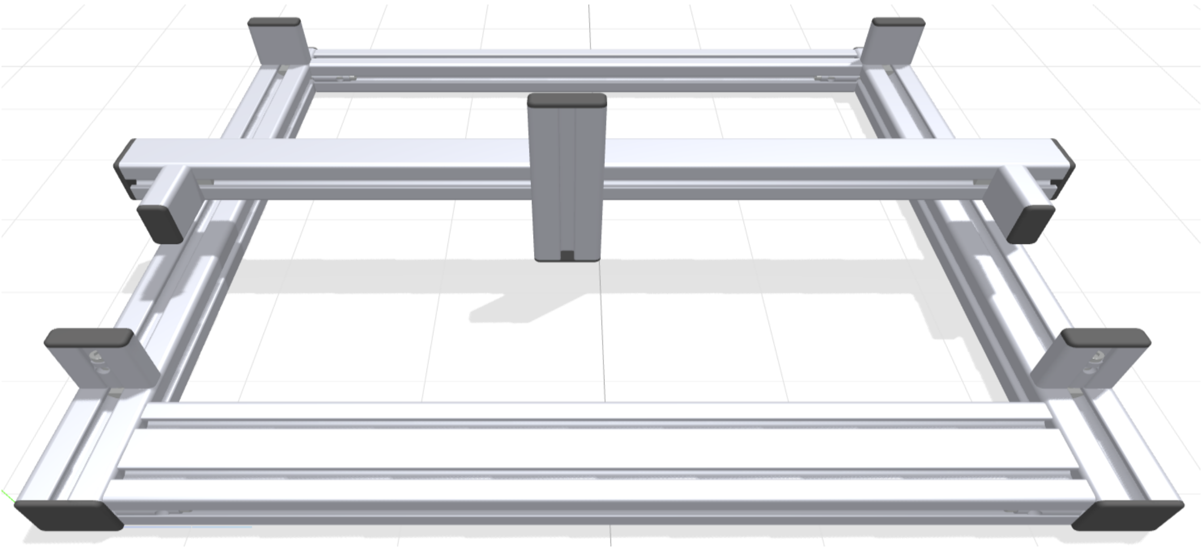


Figure 8.2 3D model of project

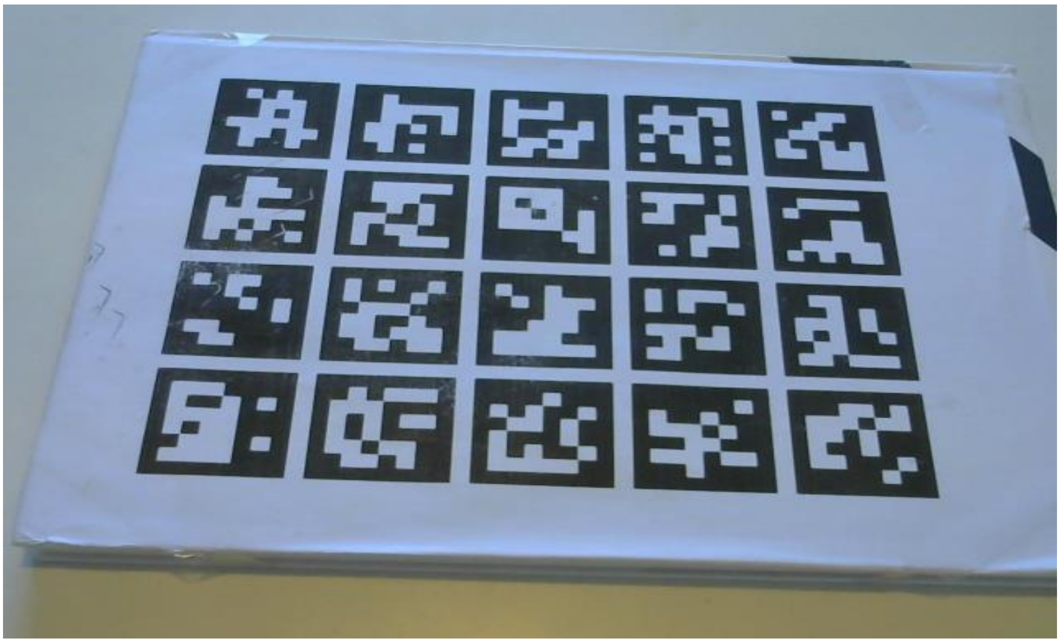
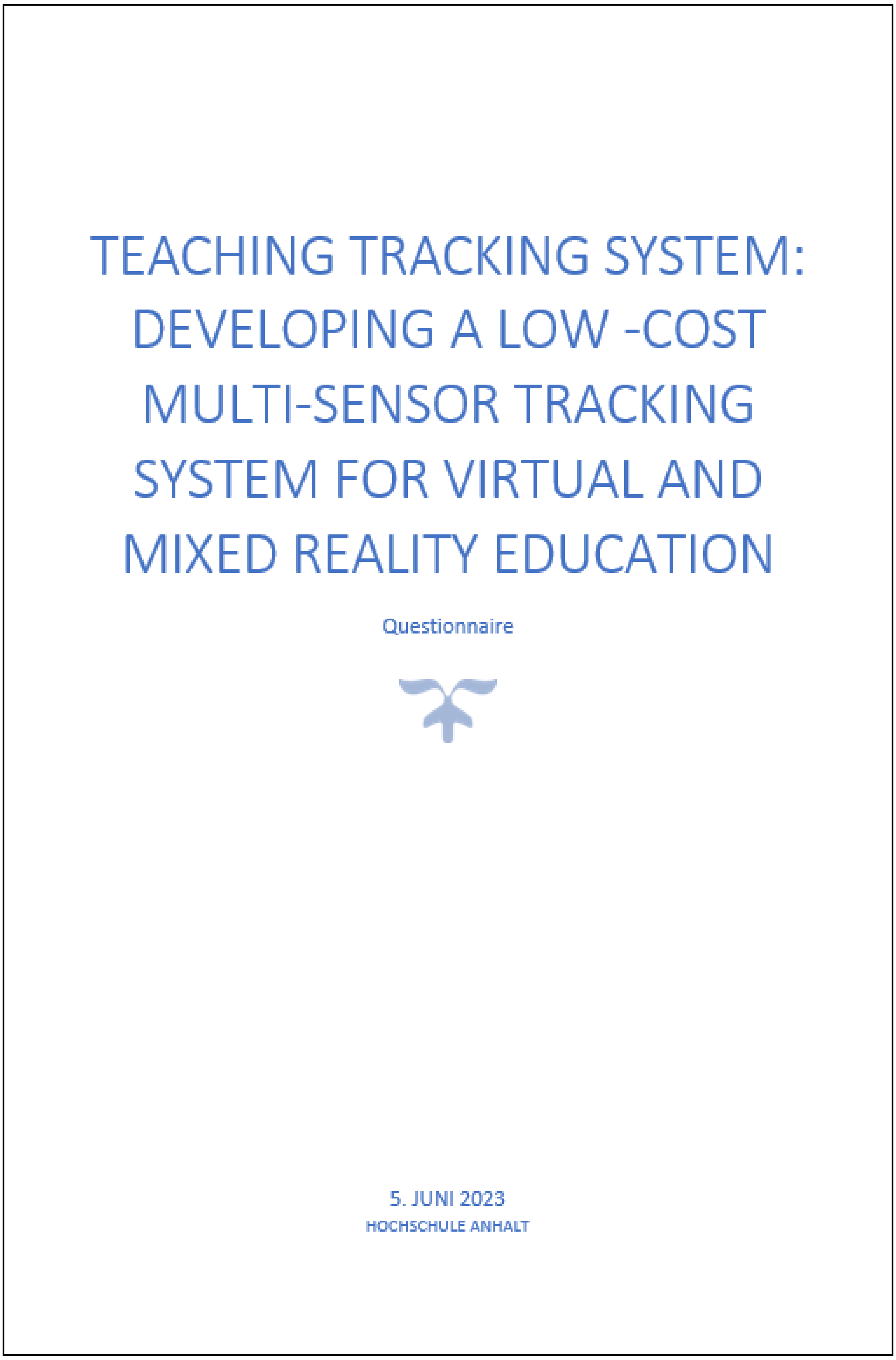


Figure 8.3 ArUco marker board used for optical sensor calibration

Figure 8.4 ArUco marker attached to gondola in system

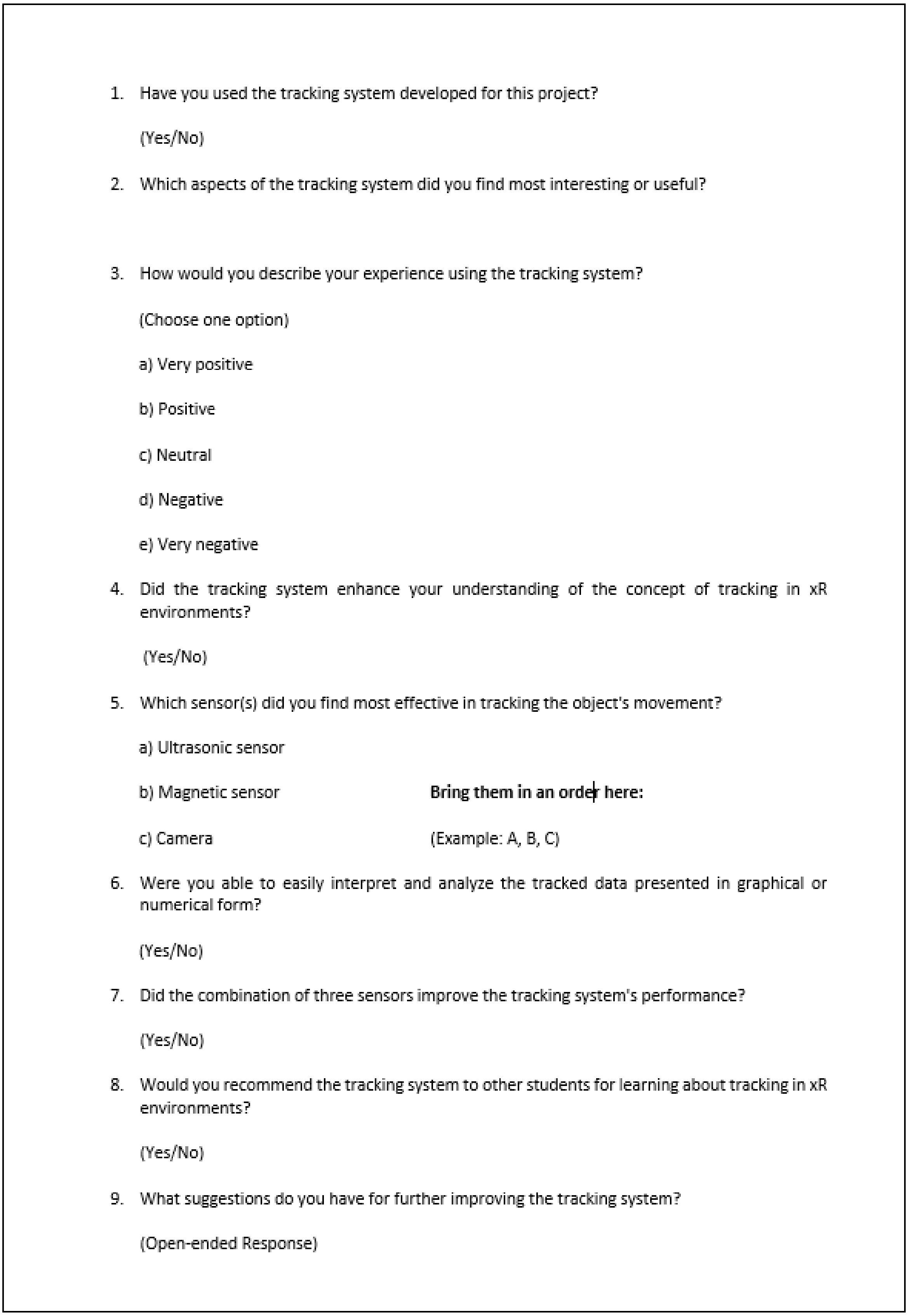
8.3. QUESTIONNAIRE



TEACHING TRACKING SYSTEM:
DEVELOPING A LOW -COST
MULTI-SENSOR TRACKING
SYSTEM FOR VIRTUAL AND
MIXED REALITY EDUCATION

Questionnaire

5. JUNI 2023
HOCHSCHULE ANHALT



1. Have you used the tracking system developed for this project?

   (Yes/No)

2. Which aspects of the tracking system did you find most interesting or useful?

3. How would you describe your experience using the tracking system?

   (Choose one option)

   a) Very positive

   b) Positive

   c) Neutral

   d) Negative

   e) Very negative

4. Did the tracking system enhance your understanding of the concept of tracking in xR environments?

   (Yes/No)

5. Which sensor(s) did you find most effective in tracking the object's movement?

   a) Ultrasonic sensor

   b) Magnetic sensor **Bring them in an order here:**

   c) Camera (Example: A, B, C)

6. Were you able to easily interpret and analyze the tracked data presented in graphical or numerical form?

   (Yes/No)

7. Did the combination of three sensors improve the tracking system's performance?

   (Yes/No)

8. Would you recommend the tracking system to other students for learning about tracking in xR environments?

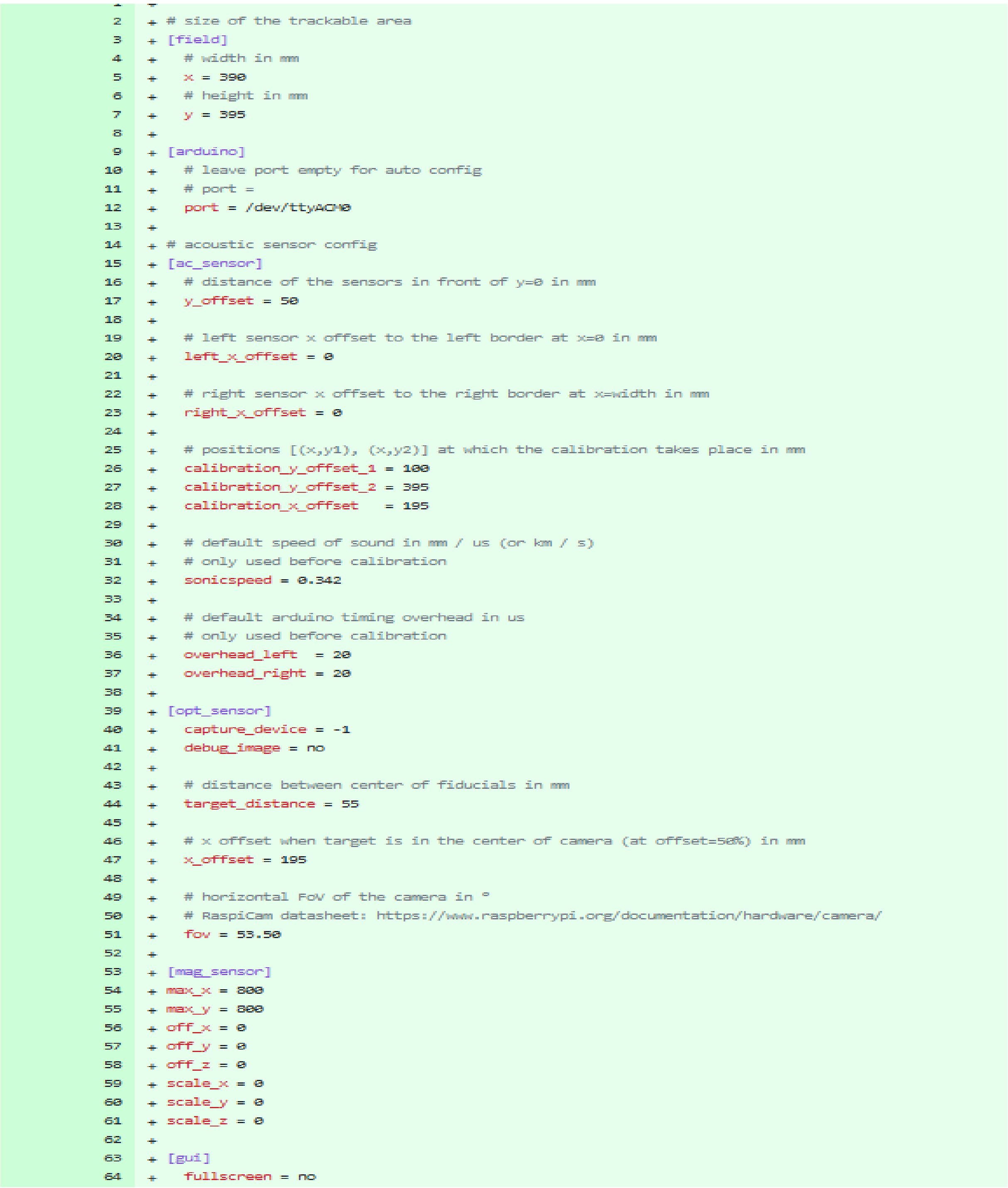
   (Yes/No)

9. What suggestions do you have for further improving the tracking system?

   (Open-ended Response)

8.4. CODING SCRIPTS

1. raspberry-pi/config.ini

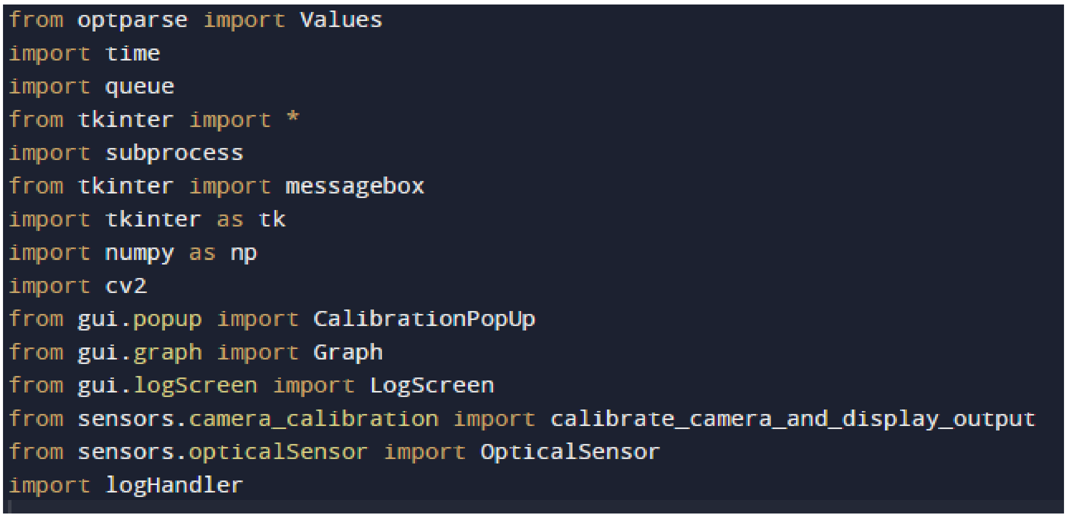


```
+
+ # size of the trackable area
+ [field]
+   # width in mm
+   x = 390
+   # height in mm
+   y = 395
+
+ [arduino]
+   # leave port empty for auto config
+   # port =
+   port = /dev/ttyACM0
+
+ # acoustic sensor config
+ [ac_sensor]
+   # distance of the sensors in front of y=0 in mm
+   y_offset = 50
+
+   # left sensor x offset to the left border at x=0 in mm
+   left_x_offset = 0
+
+   # right sensor x offset to the right border at x=width in mm
+   right_x_offset = 0
+
+   # positions [(x,y1), (x,y2)] at which the calibration takes place in mm
+   calibration_y_offset_1 = 100
+   calibration_y_offset_2 = 395
+   calibration_x_offset   = 195
+
+   # default speed of sound in mm / us (or km / s)
+   # only used before calibration
+   sonicspeed = 0.342
+
+   # default arduino timing overhead in us
+   # only used before calibration
+   overhead_left  = 20
+   overhead_right = 20
+
+ [opt_sensor]
+   capture_device = -1
+   debug_image = no
+
+   # distance between center of fiducials in mm
+   target_distance = 55
+
+   # x offset when target is in the center of camera (at offset=50%) in mm
+   x_offset = 195
+
+   # horizontal FoV of the camera in °
+   # RaspiCam datasheet: https://www.raspberrypi.org/documentation/hardware/camera/
+   fov = 53.50
+
+ [mag_sensor]
+ max_x = 800
+ max_y = 800
+ off_x = 0
+ off_y = 0
+ off_z = 0
+ scale_x = 0
+ scale_y = 0
+ scale_z = 0
+
+ [gui]
+   fullscreen = no
```

2. Readme file for the project:

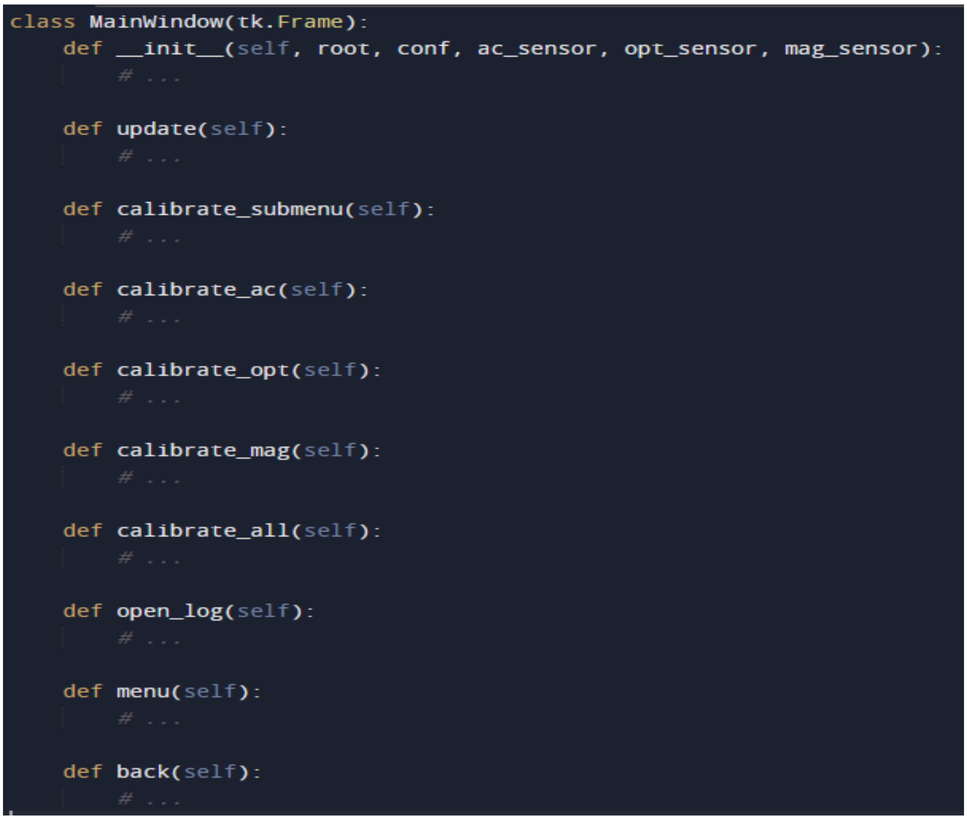
```
... ...  @@ -0,0 +1,18 @@
+ # Calibrate camera with arUco markers using opencv and python.
+ (Code developed and tested on opencv 3.3.1)
+
+
+ # camera_calibration
+ Code and resources for camera calibration using arUco markers and opencv
+
+ 1. Print the aruco marker board provided. (You can generate your own board, see the code "camera_calibration.py")
+ 2. Take around 50 images of the printed board pasted on a flat card-board, from different angles. Use Use data_generation node for this.
+ 3. Set path to store images first.
+ (See sample images of arUco board in aruco_data folder)
+
+ ## Calibrating camera
+ 1. Measure length of the side of individual marker and spacing between two marker.
+ 2. Input above data (length and spacing) in "camera_calibration.py".
+ 3. Set path to stored images of aruco marker.
+ 4. Set calibrate_camera flag to true.
```

3. Important statements for main window



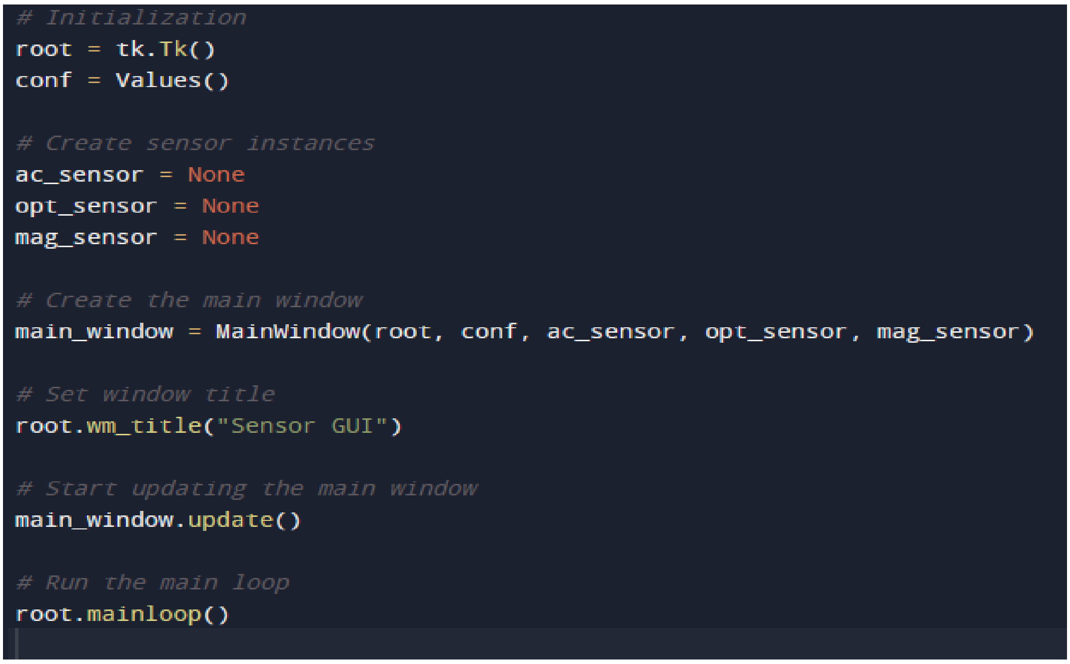
```python
from optparse import Values
import time
import queue
from tkinter import *
import subprocess
from tkinter import messagebox
import tkinter as tk
import numpy as np
import cv2
from gui.popup import CalibrationPopUp
from gui.graph import Graph
from gui.logScreen import LogScreen
from sensors.camera_calibration import calibrate_camera_and_display_output
from sensors.opticalSensor import OpticalSensor
import logHandler
```

3.1. Classes definitions



```
class MainWindow(tk.Frame):
    def __init__(self, root, conf, ac_sensor, opt_sensor, mag_sensor):
        # ...

    def update(self):
        # ...

    def calibrate_submenu(self):
        # ...

    def calibrate_ac(self):
        # ...

    def calibrate_opt(self):
        # ...

    def calibrate_mag(self):
        # ...

    def calibrate_all(self):
        # ...

    def open_log(self):
        # ...

    def menu(self):
        # ...

    def back(self):
        # ...
```

3.2. Initialization and layout

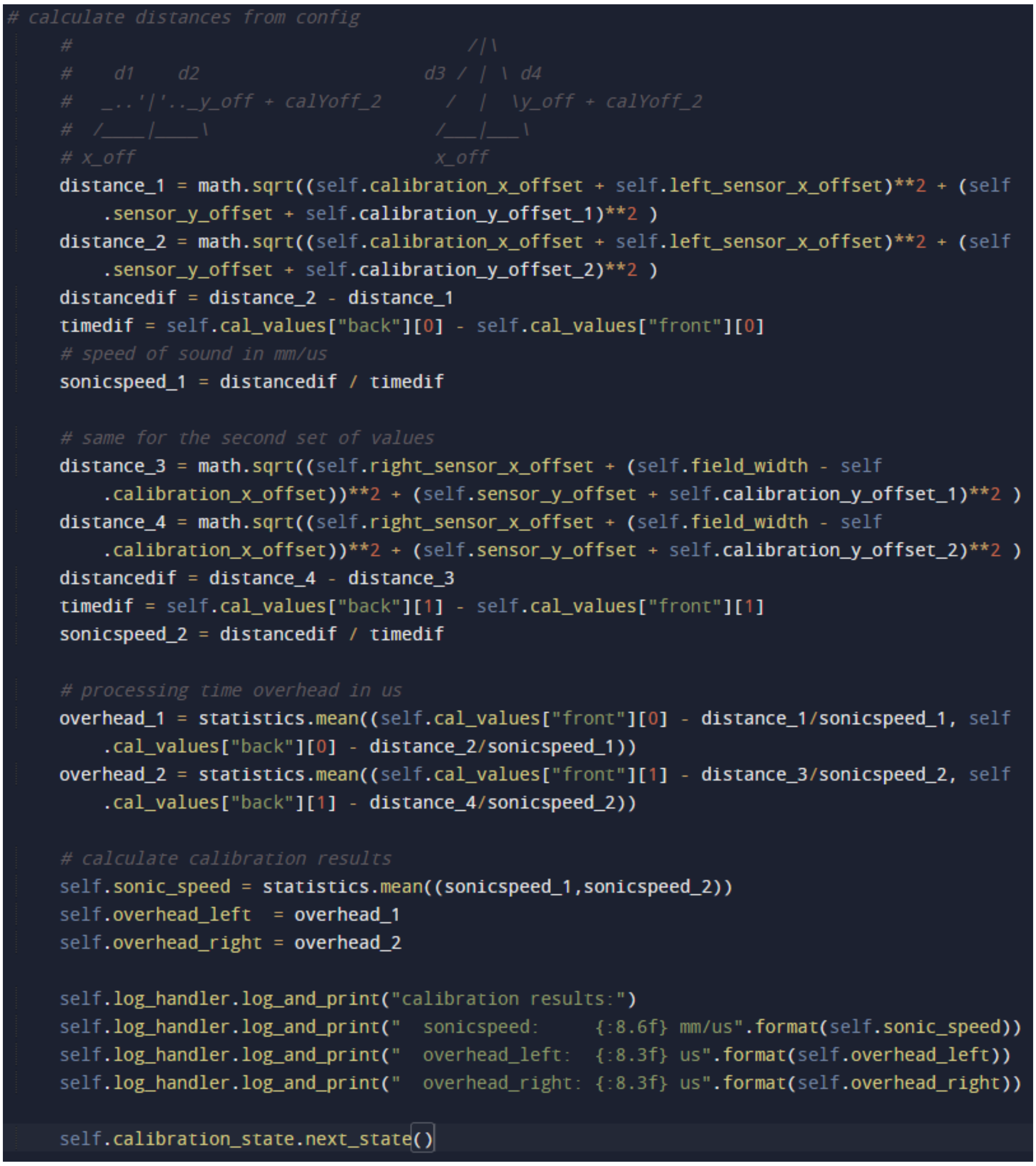


```
# Initialization
root = tk.Tk()
conf = Values()

# Create sensor instances
ac_sensor = None
opt_sensor = None
mag_sensor = None

# Create the main window
main_window = MainWindow(root, conf, ac_sensor, opt_sensor, mag_sensor)

# Set window title
root.wm_title("Sensor GUI")

# Start updating the main window
main_window.update()

# Run the main loop
root.mainloop()
```

4. Important code script for calibration popup for all sensors. It includes the calibration status and update for status after moving gondola to positions.

```
class CalibrationPopUp(tk.Frame):
    def __init__(self, root, calibration_state, conf):
        self.root = root
        self.font = pyglet.font.add_file("gui/SourceSansPro-Semibold.otf")
        self.calibration_state = calibration_state
        tk.Frame.__init__(self, root)
        self.pendingClose = False
        self.conf = conf

        self.instruction = tk.Label(self, text="Start Calibration", anchor="c", font=("SourceSansPro
            -Semibold", 18))
        self.instruction.pack(side="top", fill="both", expand=True)

        button = tk.Button(self, text="OK", command=self.calibration_state.next_state_gui, anchor="c"
            , height=1, width=5)
        button.pack(side="top", fill="both", expand=True)

        self.cs = Progressbar(self, orient='horizontal', mode='determinate')
        self.cs.pack(side="top", fill="both", expand=True)

        root.bind('<Escape>', self.close)

    def update(self):
        if not self.root.winfo_exists():
            return
        self.cs['value'] = self.calibration_state.progress
        if self.calibration_state.get_state() == self.calibration_state.WAITING_POS_1:
            text = "Move gondola to [" + self.conf["ac_sensor"]["calibration_x_offset"] + " , " +
                self.conf["ac_sensor"]["calibration_y_offset_1"] + "]!"
            self.instruction["text"] = text
        elif self.calibration_state.get_state() == self.calibration_state.WAITING_POS_2:
            text = "Move gondola to [" + self.conf["ac_sensor"]["calibration_x_offset"] + " , " +
                self.conf["ac_sensor"]["calibration_y_offset_2"] + "]!"
            self.instruction["text"] = text
        elif self.calibration_state.get_state() == self.calibration_state.CALIBRATION_DONE:
            self.instruction["text"] = "Calibration Done!"
            if not self.pendingClose:
                self.pendingClose = True
                self.root.after(1500, self.close)
        else:
            self.instruction["text"] = "Processing..."
```

5. Important classes used for acoustic sensor tracking calculations and calibrations from configuration values



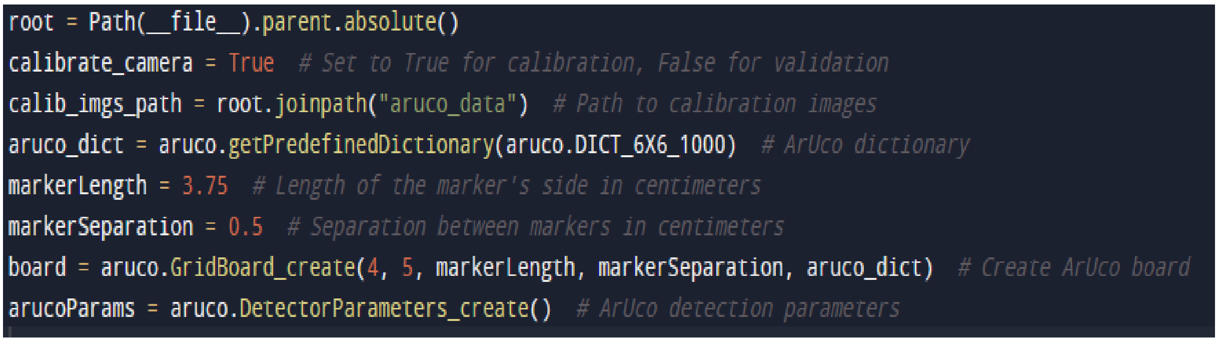
```
# calculate distances from config
    #                                     /|\
    #    d1    d2                     d3 / | \ d4
    #   _..'|'.._y_off + calYoff_2      /  |  \y_off + calYoff_2
    #  /____|____\                     /___|___\
    # x_off                            x_off
    distance_1 = math.sqrt((self.calibration_x_offset + self.left_sensor_x_offset)**2 + (self
        .sensor_y_offset + self.calibration_y_offset_1)**2 )
    distance_2 = math.sqrt((self.calibration_x_offset + self.left_sensor_x_offset)**2 + (self
        .sensor_y_offset + self.calibration_y_offset_2)**2 )
    distancedif = distance_2 - distance_1
    timedif = self.cal_values["back"][0] - self.cal_values["front"][0]
    # speed of sound in mm/us
    sonicspeed_1 = distancedif / timedif

    # same for the second set of values
    distance_3 = math.sqrt((self.right_sensor_x_offset + (self.field_width - self
        .calibration_x_offset))**2 + (self.sensor_y_offset + self.calibration_y_offset_1)**2 )
    distance_4 = math.sqrt((self.right_sensor_x_offset + (self.field_width - self
        .calibration_x_offset))**2 + (self.sensor_y_offset + self.calibration_y_offset_2)**2 )
    distancedif = distance_4 - distance_3
    timedif = self.cal_values["back"][1] - self.cal_values["front"][1]
    sonicspeed_2 = distancedif / timedif

    # processing time overhead in us
    overhead_1 = statistics.mean((self.cal_values["front"][0] - distance_1/sonicspeed_1, self
        .cal_values["back"][0] - distance_2/sonicspeed_1))
    overhead_2 = statistics.mean((self.cal_values["front"][1] - distance_3/sonicspeed_2, self
        .cal_values["back"][1] - distance_4/sonicspeed_2))

    # calculate calibration results
    self.sonic_speed = statistics.mean((sonicspeed_1,sonicspeed_2))
    self.overhead_left  = overhead_1
    self.overhead_right = overhead_2

    self.log_handler.log_and_print("calibration results:")
    self.log_handler.log_and_print("  sonicspeed:     {:8.6f} mm/us".format(self.sonic_speed))
    self.log_handler.log_and_print("  overhead_left:  {:8.3f} us".format(self.overhead_left))
    self.log_handler.log_and_print("  overhead_right: {:8.3f} us".format(self.overhead_right))

    self.calibration_state.next_state()
```

```
def calculate_position(self, values):
    val1, val2 = values
    val1 -= self.overhead_left
    val2 -= self.overhead_right
    distance_left = val1 * self.sonic_speed
    distance_right = val2 * self.sonic_speed
    # compute intersection of distance circles
    x = (self.sensor_distance**2 - distance_right**2 + distance_left**2) / (2*self.sensor_distance) +
        self.left_sensor_x_offset
    #print(x)
    if distance_left**2 - x**2 >= 0:
        y = math.sqrt(distance_left**2 - x**2) - self.sensor_y_offset
        #print(y)
        return (x, y)
    else:
        return None
```

And global connection setup

```
conn = globalArduinoSlave()
```

6. Camera tracking parameters and variables settings



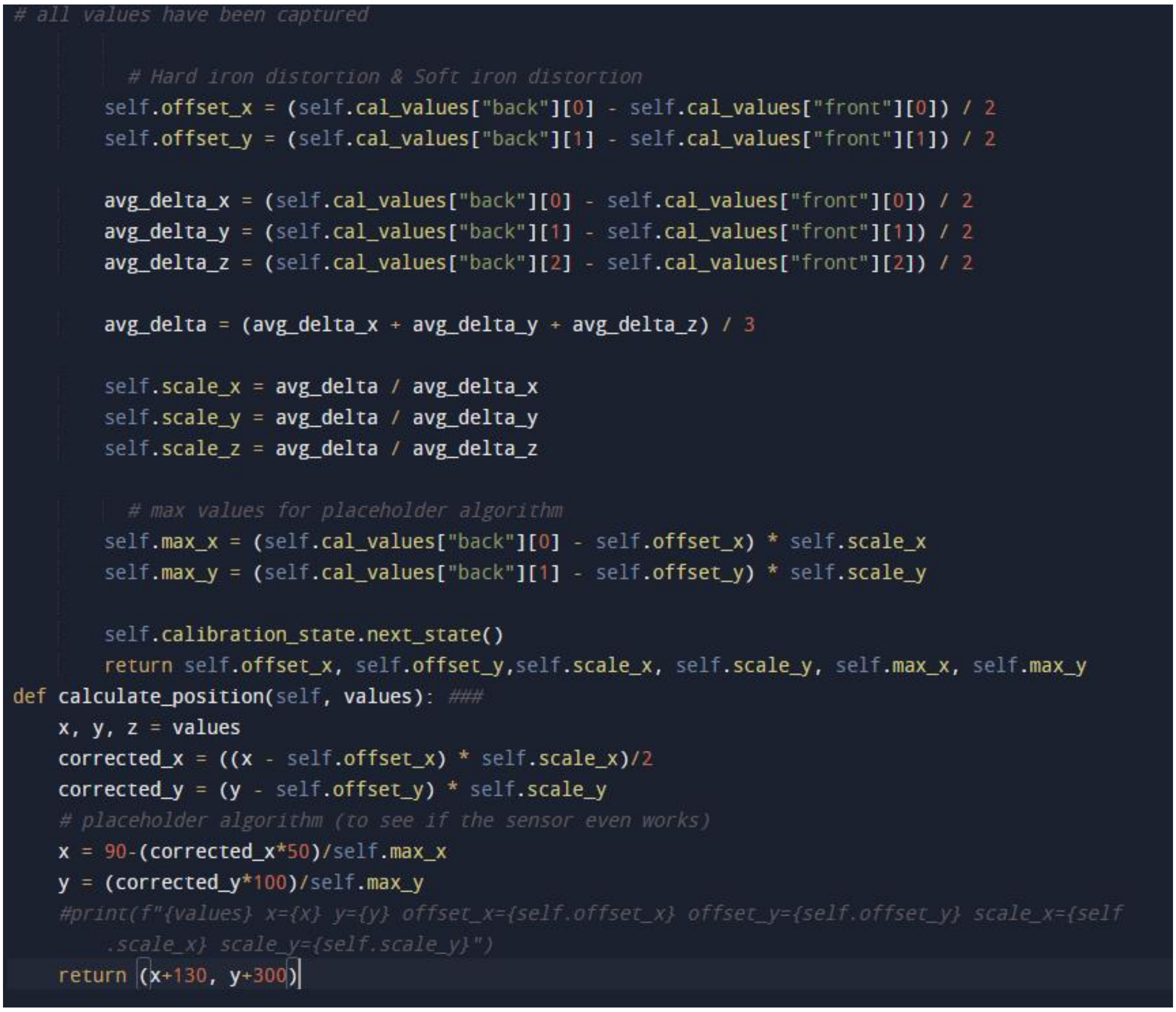
```
root = Path(__file__).parent.absolute()
calibrate_camera = True  # Set to True for calibration, False for validation
calib_imgs_path = root.joinpath("aruco_data")  # Path to calibration images
aruco_dict = aruco.getPredefinedDictionary(aruco.DICT_6X6_1000)  # ArUco dictionary
markerLength = 3.75  # Length of the marker's side in centimeters
markerSeparation = 0.5  # Separation between markers in centimeters
board = aruco.GridBoard_create(4, 5, markerLength, markerSeparation, aruco_dict)  # Create ArUco board
arucoParams = aruco.DetectorParameters_create()  # ArUco detection parameters
```

7. Camera calibration functions and main execution

```python
def calibrate_camera_and_display_output():
    # ...

    if calibrate_camera == True:
        # Load calibration images
        img_list = []

        # Detect markers in calibration images
        for idx, fn in enumerate(calib_fnms):
            # ...

        # Calibrate camera using detected markers
        counter, corners_list, id_list = [], [], []
        for im in tqdm(img_list):
            # ...

        ret, mtx, dist, rvecs, tvecs = aruco.calibrateCameraAruco(
            corners_list, id_list, counter, board, img_gray.shape, None, None
        )

        # Display and store calibration results
        data = {'camera_matrix': np.asarray(mtx).tolist(), 'dist_coeff': np.asarray(dist).tolist()}
        with open("calibration.yaml", "w") as f:
            yaml.dump(data, f)
        # ...

        output = "Camera matrix: \n" + "[" + str(np.round(mtx, 2)[0, 0]) + "      " + str(np.round(mtx, 2)[0
            , 1]) + "    " + str(np.round(mtx, 2)[0, 2]) + "\n" + str(np.round(mtx, 2)[1, 0]) + "      " +
            str(np.round(mtx, 2)[1, 1]) + "    " + str(np.round(mtx, 2)[1, 2]) + "\n" + str(np.round(mtx, 2
            )[2, 0]) + "         " + str(np.round(mtx, 2)[2, 1]) + "         " + str(mtx[2, 2]) + "]" +
            "\nAnd is stored in calibration.yaml file along with distortion coefficients : \n" + str(dist)
        messagebox.showinfo("Camera Calibration Output", output)

root = tk.Tk()
root.withdraw()
calibrate_camera_and_display_output()
cv2.destroyAllWindows()
```

```python
if __name__ == "__main__":
    # Saves ArUco markers to an image
    saveMarkers()

    # Captures frames from a camera and displays distance measurements
    cap = cv2.VideoCapture(1)
    while True:
        success, image = cap.read()
        print(measureDistances(image))
        cv2.imshow("image", image)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
```

8. Magnetic sensor important calibration and caluclation. The magnetci secor uses the same calibration popup and global connector



```
# all values have been captured

          # Hard iron distortion & Soft iron distortion
        self.offset_x = (self.cal_values["back"][0] - self.cal_values["front"][0]) / 2
        self.offset_y = (self.cal_values["back"][1] - self.cal_values["front"][1]) / 2

        avg_delta_x = (self.cal_values["back"][0] - self.cal_values["front"][0]) / 2
        avg_delta_y = (self.cal_values["back"][1] - self.cal_values["front"][1]) / 2
        avg_delta_z = (self.cal_values["back"][2] - self.cal_values["front"][2]) / 2

        avg_delta = (avg_delta_x + avg_delta_y + avg_delta_z) / 3

        self.scale_x = avg_delta / avg_delta_x
        self.scale_y = avg_delta / avg_delta_y
        self.scale_z = avg_delta / avg_delta_z

          # max values for placeholder algorithm
        self.max_x = (self.cal_values["back"][0] - self.offset_x) * self.scale_x
        self.max_y = (self.cal_values["back"][1] - self.offset_y) * self.scale_y

        self.calibration_state.next_state()
        return self.offset_x, self.offset_y,self.scale_x, self.scale_y, self.max_x, self.max_y
def calculate_position(self, values): ###
    x, y, z = values
    corrected_x = ((x - self.offset_x) * self.scale_x)/2
    corrected_y = (y - self.offset_y) * self.scale_y
    # placeholder algorithm (to see if the sensor even works)
    x = 90-(corrected_x*50)/self.max_x
    y = (corrected_y*100)/self.max_y
    #print(f"{values} x={x} y={y} offset_x={self.offset_x} offset_y={self.offset_y} scale_x={self
        .scale_x} scale_y={self.scale_y}")
    return (x+130, y+300)
```