BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

VISUALIZATION OF DETECTED OBJECTS FROM DRONE IN MICROSOFT HOLOLENS 2

ZOBRAZENÍ OBJEKTŮ DETEKOVANÝCH DRONEM V BRÝLÍCH MICROSOFT HOLOLENS 2

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTOR PRÁCE
AUTHOR **FILIP OSVALD**

SUPERVISOR **Ing. DANIEL BAMBUŠEK**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Bachelor's Thesis Assignment

153848

Institut: Department of Computer Graphics and Multimedia (UPGM)
Student: **Osvald Filip**
Programme: Information Technology
Specialization: Information Technology
Title: **Visualization of Detected Objects from Drone in Microsoft HoloLens 2**
Category: User Interfaces
Academic year: 2022/23

Assignment:

1. Study the concept of augmented reality and its use in the visualization of georeferenced object data. Get familiar with Microsoft HoloLens 2 headset and explore its use for visualizing the locations of drone-detected objects.
2. Select appropriate methods and tools, and design a user interface that uses HoloLens to display the detected objects of interest.
3. Implement the proposed application.
4. Conduct experiments and evaluate the capabilities of the final solution.
5. Create a video presenting the key features of the final solution.

Literature:

- SCHMALSTIEG Dieter, HÖLLERER Tobias. *Augmented Reality: Principles and Practice.* Addison-Wesley, 2016. ISBN 978-0321883575.
- SEDLMAJER Kamil, BAMBUŠEK Daniel a BERAN Vítězslav. *Effective Remote Drone Control Using Augmented Virtuality.* In: Proceedings of the 3rd International Conference on Computer-Human Interaction Research and Applications 2019. Vienna: SciTePress - Science and Technology Publications, 2019, s. 177-182. ISBN 978-989-758-376-6.
- VÁCLAVÍK, Marek. *Vizualizační nástroj pro pilota dronu v Microsoft HoloLens 2*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. 2021-06-16. Vedoucí práce Bambušek Daniel. Dostupné z: https://www.fit.vut.cz/study/thesis/23524/
- HARTSON Rex. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. 2012. ISBN 9780123852427.
- Dále dle pokynů vedoucího.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Bambušek Daniel, Ing.**
Head of Department: Černocký Jan, prof. Dr. Ing.
Beginning of work: 1.11.2022
Submission deadline: 31.7.2023
Approval date: 19.6.2023

Abstract

Drones have become increasingly valuable in various fields, such as surveillance, disaster management and search and rescue operations. However, transmitting the data and geographic information captured by drones to ground personnel presents a significant challenge. The task becomes even more complex when multiple drones are used, making the processing of information by a single drone operator highly inefficient.

The objective of this thesis is to develop a system capable of displaying objects detected by a drone in Augmented Reality. The thesis assesses the usability of such a system for information conveying and identifies suitable devices and technologies that can be incorporated into the system.

Abstrakt

Drony se stávaji stále cennějšími v různých oblastech, jako je bezpečnost, řízení krizových situací a záchranné operace. Přenos dat a geografických informací zachycených drony k pozemnímu personálu však představuje významné výzvy. Úkol se stává ještě složitějším při použití více dronů, což způsobuje, že zpracování informací jedním operátorem dronu je velmi neefektivní.

Cílem této práce je vyvinout systém schopný zobrazovat objekty detekované dronem v Rozšířené Realitě. Práce posuzuje použitelnost takového systému pro přenos informací a identifikuje vhodná zařízení a technologie, které lze do systému začlenit.

Keywords

drones, object detection, HoloLens 2, Unity, augmented reality

Klíčová slova

drony, detekce objektů, HoloLens 2, Unity, rozšířená realita

Reference

OSVALD, Filip. *Visualization of Detected Objects from Drone in Microsoft HoloLens 2.* Brno, 2023. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Daniel Bambušek

Rozšířený abstrakt

Díky rychlému pokroku v technologii dronů a rozšířené reality se otevírají nové možnosti pro zlepšení interakce člověka s prostředím. Bezpilotní letadla (UAV) či drony nacházejí uplatnění v různých oborech, včetně záchranářství, zemědělství a fotografie. Rozšířená realita propojuje fyzický a digitální svět, čímž vytváří poutavé a interaktivní zážitky. Microsoft HoloLens 2, jedny z předních brýlí pro rozšířšnou realitu, nabízí skvělou platformu pro zkoumání nových aplikací, které využívají schopností dronů i rozšířené reality.

Tato práce má za cíl navrhnout a vyvinout systém, který integruje drony s headsetem Microsoft HoloLens 2 pro rozšířenou realitu a umožňit uživatelům vizualizovat detekované objekty z perspektivy dronu v reálném prostředí. Kombinací leteckého pohledu dronu a prostorového pochopení poskytnutého headsetem pro rozšířenou realitu mohou uživatelé získat lepší situační povědomí, zlepšit rozhodování a zvýšit svou schopnost reagovat na různé situace. Tento systém má potenciál ovlivnit různé oblasti, včetně záchranářství, inspekce infrastruktury a monitorování životního prostředí.

Hlavními cíli této práce jsou:

- Analyzovat potřebné součásti, zařízení a technologie potřebné pro integraci dronu s brýlemi Microsoft HoloLens 2 pro rozšířšnou realitu do systému.
- Identifikovat výzvy a problémy spojené s vývojem navrhovaného systému. Prozkoumat existující řešení a přístupy, které lze přizpůsobit tomuto systému.
- Vyvinout prototypový systém, který demonstruje proveditelnost vizualizace detekovaných objektů z dronu v headsetu Microsoft HoloLens 2 pro rozšířenou realitu.
- Vyhodnotit výkonnost a použitelnost vyvinutého systému a navrhnout oblasti pro budoucí zlepšení.

Práce je strukturována následovně:

Kapitola 2 poskytuje základní informace o rozšířené realitě. Kapitola 3 poskytuje přehled o brýlích Microsoft HoloLens 2 a obdobných řešeních. Kapitola 4 pojednává o použití dronů s moderními technologiemi. Kapitola 5 diskutuje o potřebných součástech, zařízeních a technologiích pro navrhovaný systém, řeší výzvy a problémy a zkoumá kompatibilní existující řešení. Kapitola 6 detailně popisuje implementaci prototypového systému, včetně volby návrhu, metodologií a nástrojů použitých při jeho vývoji. Kapitola 7 hodnotí vyvinutý systém, jeho výkonnost, použitelnost a omezení, jak by omezení šli spravit a diskutuje rozšíření do budoucna. Závěrečná kapitola 8 shrnuje práci a zmiňuje klíčové poznatky.

Tento projekt demonstruje koncept pro systém detekce objektů dronem a jejich vizualizace v rozšířené realitě pomocí brýlí Microsoft HoloLens 2. Systém umožňuje uživatelům lépe identifikovat a lokalizovat detekované objekty v reálném čase. Ačkoliv současná implementace má svá omezení, jako jsou přesnost, viditelnost a synchronizační problémy, poskytuje základ pro budoucí zlepšení a rozšíření.

Potenciální aplikace takového systému jsou rozsáhlé, od záchraných operací přes inspekci infrastruktury až po monitorování životního prostředí. Zlepšením přesnosti, spolehlivosti a použitelnosti systému by se mohl stát cenným nástrojem pro různé odvětví a použití, automatizací některých úkolů a uvolňit ruce uživatelů pro provádění jiných činností.

Visualization of Detected Objects from Drone in Microsoft HoloLens 2

Declaration

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Daniela Bambuška. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

. . . .
Filip Osvald
July 31, 2023

Acknowledgements

Acknowledgment goes to my supervisor Daniel Bambušek for guidance and for making this thesis possible.

Contents

List of Figures

Chapter 1

Introduction

Rapid drone technology advancements and augmented reality have opened up new possibilities for enhancing human interaction with the environment. Unmanned aerial vehicles (UAVs) or drones, have paved their way in various fields, including surveillance, agriculture, and photography. Meanwhile, augmented reality technology has matured, blending the physical and digital worlds to create immersive and interactive experiences. The Microsoft HoloLens 2, one of the leading augmented-reality headsets, provides an excellent platform for exploring novel applications that leverage the capabilities of both drones and augmented reality.

This thesis aims to design and develop a system that integrates drones with the Microsoft HoloLens 2 augmented reality headset, enabling users to visualize detected objects from a drone's perspective in real-time. By combining the drone's aerial view with the spatial understanding provided by the augmented reality headset, users can gain better situational awareness, enhance decision-making, and improve their ability to interact with the environment. This system has the potential to impact various fields, including search and rescue operations, infrastructure inspection, and environmental monitoring.

The main objectives of this thesis are to:

- Analyze the necessary components, devices, and technologies required to integrate a drone with the Microsoft HoloLens 2 augmented reality headset.
- Identify the challenges and problems associated with the development of the proposed system.
- Investigate existing solutions and approaches that can be adapted for this system.
- Develop a prototype system that demonstrates the feasibility of visualizing detected objects from a drone in the Microsoft HoloLens 2 augmented reality headset.
- Evaluate the performance and usability of the developed system and suggest areas for future improvements.

This thesis is structured as follows:

- **Chapter** 2 provides information about augmented reality.
- **Chapter** 3 provides an overview of the Microsoft HoloLens 2 headset and discusses similar solutions.

- **Chapter** 4 discusses the use of drones with modern technologies.
- **Chapter** 5 discusses the necessary components, devices, and technologies for the proposed system, addressing the challenges and problems faced and exploring compatible existing solutions.
- **Chapter** 6 details the implementation of the prototype system, describing the design choices, methodologies, and tools used in its development.
- **Chapter** 7 discusses the performance, usability, and limitations of the system, how these limitations could be resolved and future improvements
- **Chapter** 8 concludes the thesis, summarizing the key findings.

The work presented in this thesis creates a foundation for future research and development in the area of using drones and augmented reality for autonomous object detection and visualization.

Chapter 2

Augmented reality

Augmented Reality (AR) superimposes virtual objects or information onto the user's view of the physical world. Unlike virtual reality, augmented reality does not immerse the user in a completely virtual environment but instead augments the real world with digital content. Augmented Reality applications can be experienced through common consumer devices such as smartphones and tablets, through specialized devices including AR glasses, or integrated into common objects, for instance, motorcycle helmets and car windshields. AR, has been applied in many areas, including navigation, education, and retail [2].

2.1 Augmented reality in smartphones

Augmented reality devices require a display, sensors such as camera, accelerometer and GPS, a processor and input devices. As smartphones contain such hardware, they are the most common platform for augmented reality applications. Their widespread adoption has been significantly driven by the rise of AR-infused mobile games. These mobile games demonstrated the potential of AR to engage users in novel and interactive experiences.

Beyond entertainment, AR applications on smartphones have found a solid foothold in the commercial sector. Retailers and service providers are increasingly utilizing AR technology to enhance customer interactions by presenting augmented representations of products or services (see Figure 2.1). This allows customers to visualize potential purchases in a more realistic and immersive way, often resulting in more informed decision-making.

In addition to its utility in the gaming and commercial industries, AR technology on smartphones has also been employed for more specialized tasks such as astronomic observations. Applications are available that leverage the power of AR to facilitate the location of celestial bodies, including satellites and planets. By simply pointing their smartphones towards the sky, users can obtain detailed and accurate information about various celestial objects, effectively transforming these devices into pocket-sized planetariums.

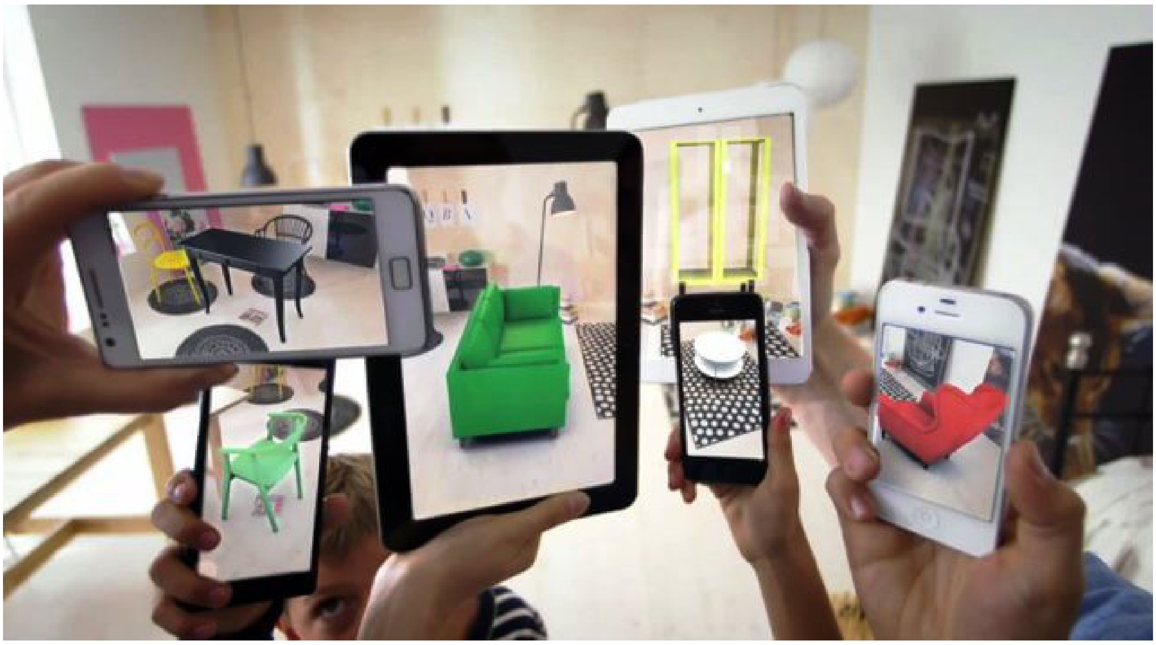

Figure 2.1: Showcasing products in AR.[1]

2.2 Wearable augmented reality devices

Augmented reality's expanding influence extends to wearable devices, notably in the form of glasses and helmets. These innovations have provided an entirely new dimension to the user experience, allowing for hands-free, immersive engagement with the digital world.

AR glasses, a growing sector within wearable technology, have opened up a plethora of possibilities across multiple industries. With AR glasses, users can overlay digital information onto their natural field of view. In a day-to-day context, this could mean seeing navigation instructions directly in your field of vision while walking, or in a professional setting, obtaining real-time data feeds overlaid on physical equipment in a manufacturing plant. The potential applications are broad, from aiding in complex surgical procedures (as shown in Figure 2.2) to facilitating immersive language learning experiences.

Similarly, AR-integrated helmets are transforming tasks that require protective headwear. Fields such as construction, firefighting, and aviation are benefiting greatly from these advancements [5]. For instance, AR-integrated helmets can provide construction workers with real-time structural information or alert firefighters to potentially hazardous conditions within smoke-filled environments. In the aviation industry, pilots can use these helmets to access critical flight data without averting their gaze from the flight path or see through the fuselage of their plane.

The integration of AR technology into wearable devices like glasses and helmets enhances safety measures by keeping users' hands-free while they interact with their environment. This feature is particularly beneficial in sectors where manual engagement is necessary or where distraction could result in significant risks.

---

[1] Image source: `http://bitly.ws/PbfL`

Figure 2.2: Medical simulation using Hololens 2.[1]

---

[1] Image source: https://www.caehealthcare.com/hololens/

Chapter 3

Microsoft HoloLens 2

Microsoft HoloLens 2 headset is an advanced augmented reality head-mounted display designed to blend digital information with the user's physical environment. This chapter briefly describes this device, its outdoor usage, and its usage with drones similar to this thesis.

3.1 Interaction possibilities

One of the key improvements of the HoloLens 2 over its predecessor is the enhanced interaction capabilities. The device supports instinctual interactions, which allow users to engage with holograms using natural hand gestures and movements. The system recognizes gestures such as air tap, pinch, and push, and is capable of tracking individual finger movements, allowing for more precise manipulation of virtual objects. The HoloLens 2 also incorporates eye-tracking technology, enabling the device to determine where the user is looking and adjust holograms accordingly, providing a more intuitive experience. This feature also allows for gaze-based interactions, such as scrolling through menus or selecting objects by looking at them. The device also supports voice commands, allowing users to control holograms and navigate menus without the need for physical inputs. This hands-free interaction method is particularly useful for physically challenged users.

3.2 Applications

The HoloLens 2 has a wide range of potential applications across various sectors, including healthcare, education, and construction. In healthcare, the device can be used for surgical planning, medical training, and remote consultations. For example, physicians can overlay 3D models of patients' anatomy onto their bodies during surgery, providing a more accurate and immersive visualization of the surgical field. In education, the HoloLens 2 can be employed to create immersive learning environments, facilitating a better understanding of complex concepts. Additionally, the device can be utilized for remote collaboration, allowing students and educators to interact in real-time from across the world. In the industrial sector, the HoloLens 2 has applications in fields such as maintenance and inspection. Workers can access schematics overlaid onto physical equipment, reducing errors, and increasing efficiency.

3.3 Outdoor use of HoloLens 2

HoloLens 2 features 6 degrees of freedom using inside-out tracking. This allows the usage of the device without external tracking devices. However, the device is meant to be used indoors and does not always function properly outdoors. The tracking cameras work best in consistent lighting which isn't guaranteed outside with changing weather conditions.[1] Furthermore the holograms aren't bright enough for outdoor use in bright conditions. The outdoor use of HoloLens is experimental.

3.4 HoloLens 2 and drones

Work by Martin Kyjac

This work focuses on allowing easier control of the drone with the help of the Microsoft HoloLens 2 headset. The project explores the idea of displaying relevant information, such as the drone's flight data and an indicator of the drone's position to the user in the Microsoft HoloLens 2 headset [8].

Drone-Augmented Human Vision

This article explores the use of drone-augmented human vision to enhance spatial understanding and improve interaction with drones in indoor environments. By providing an exocentric[2] view through a see-through display, the system simulates X-ray vision, allowing users to explore occluded environments more intuitively. The user interface reduces cognitive load by delegating flight control to the drone's autopilot system, allowing users to focus on exploration [6]. This aims to allow users to focus on other tasks while object detection and localization happen autonomously.

Mixed Reality Drone Path Planning

This study demonstrates the innovative use of HoloLens 2 in creating drone pathways through hand gestures. Users are able to efficiently generate and eliminate waypoints for a drone using simple gestures. The spatial map feature of HoloLens 2 is utilized to avoid any obstacles. While currently confined to indoor utilization due to dependence on SteamVR tracking[3], it provides a glimpse into the potential interaction between HoloLens and drones [1].

---

[1] https://learn.microsoft.com/en-us/hololens/hololens-environment-considerations

[2] https://www.usabilityfirst.com/glossary/exocentric-viewpoint/index.html

[3] https://partner.steamgames.com/vrlicensing

Chapter 4

Advanced use of drones

This chapter discusses the various applications of drones with some modern technologies and discusses the advantages that drones bring into these areas and what limitations are still present.

4.1 Georeferencing with drones

Georeferencing is a technique in the field of geospatial science that involves assigning real-world coordinates to spatial data, enabling the integration of various datasets into a common coordinate system. This section discusses the benefits, limitations, and different strategies for collecting and processing geospatial data with drones.

Benefits of using drones for Georeferencing

Advancements in drone technology and their decreasing price has made them the best tool for georeferencing. Some of their advantages are:

- High-resolution imagery: Even newer consumer-level drones are capable of capturing high-resolution images, which allows for the identification of smaller features and finer details. This improves the usability of georeferencing and enables more precise analyses.
- Rapid data acquisition: Drones can cover large areas in a relatively short amount of time and can be automated, significantly increasing data acquisition speed compared to aerial photography and ground-based methods.
- Cost-effectiveness: Georeferencing with drones is more cost-effective [4] than other methods, such as aerial photography and satellite imagery.
- Flexibility and frequency: Drones can be deployed in various environments, under different conditions, and at various times, providing greater flexibility for geospatial data collection. Furthermore, they can be deployed more frequently. A few clouds can completely block the view of satellites or aircraft depending on altitude.

Georeferencing methods

There are several methods for georeferencing with drones. The two most common methods are:

1. Direct georeferencing: This method uses data from a GPS receiver from the drone or a separate high-precision receiver attached to the drone, which records the drone's position and orientation during image acquisition. The GPS coordinates are then embedded into the captured images' metadata and used for direct georeferencing.

2. Ground control points (GCPs) [7]. In this method, a set of known geographic coordinates are marked on the ground using visible landmarks. These landmarks (GCPs) are then identified in the drone-captured images and used to georeference the rest of the dataset.

3. Simultaneous Localization and Mapping (SLAM): SLAM is a computational method that involves constructing or updating a map of an unknown environment while simultaneously tracking the drone's location within it. This technique can be used in situations where GPS data might be unreliable or unavailable, such as in indoor environments or areas with significant signal interference.

   SLAM relies on a variety of sensor data, such as those from inertial measurement units (IMUs), cameras, and LiDAR, to estimate the drone's trajectory and create a map of its environment. It identifies features within the environment and tracks these features across a sequence of captured images, creating a 3D map. This map, combined with the tracked movements of the drone, provides means to georeference the captured data without relying on external positioning sources like GPS or GCPs [3].

Usages of obtained data

Once the captured images have been georeferenced, they are processed and analyzed for various uses. They can be stitched into one continuous orthomosaic (see Figure 4.1). Orthomosaic is a seamless and highly detailed map-like image. They are usually created using an orthographic (top-down) view which enables precise measurements. Another possible use of georeferenced images is obtaining elevation and topography data. Georeference images can also be used for feature classification, identifying and classifying specific features and objects in the orthomosaic, such as roads, buildings, and vegetation.

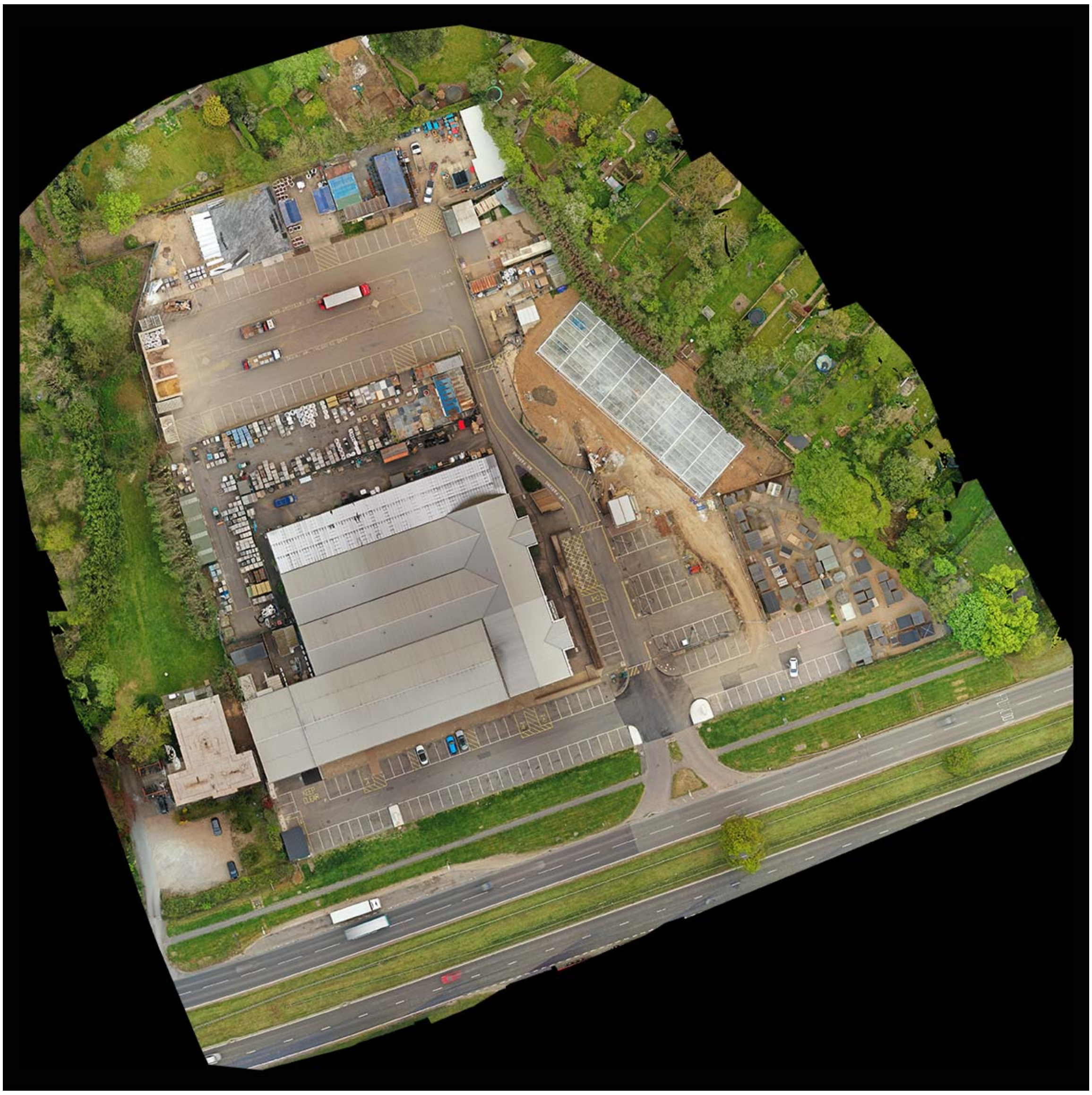

Figure 4.1: Drone captured orthomosaic.[1]

Limitations

Despite the numerous advantages, using drones for georeferencing is not without limitations.

- Laws and restrictions: Drones are subject to various regulations, including airspace restrictions and privacy concerns, which may limit their use in certain areas or circumstances. For example, restrictions that apply while flying close to people.[1]
- Weather conditions: Bad weather conditions, such as strong winds, heavy rain, or fog, negatively affects the performance of drones, resulting in lower-quality imagery and

---

[1] Image source: https://bit.ly/3nE5kOo

[1] https://www.easa.europa.eu/en/light/topics/flying-drones-close-people

potential errors in georeferencing. However, weather affects other methods as well, in some circumstances to a higher degree (for example satellites and clouds).

- GPS accuracy: The accuracy of the GPS used for georeferencing is crucial. Low-quality GPS data will lead to errors in the final geospatial dataset. These solutions such as using RTK[2] (Real-time kinematic) positioning modules to improve the accuracy.

4.2 Computer vision and drones

This section discusses the usage of computer vision and drones in conjunction and looks at multiple examples where this proved useful.

Obstacle avoidance

Obstacle avoidance is a critical aspect of drone navigation that ensures safe and efficient operation, especially in complex challenging environments. Computer vision plays an important role in helping drones to detect and avoid obstacles in real-time, reducing the risk of collisions and improving overall flight safety.

Search and rescue

In search and rescue missions, drones are already significantly useful. With computer vision, their capabilities to quickly scan large areas, locate missing individuals or objects, and provide essential information to rescue teams are enhanced. Thermal imaging sensors can also be used to detect heat signatures of humans or animals, significantly improving the chances of a successful rescue. Larger drones can be modified to deliver and drop medical supplies or rescue equipment.

Surveillance

Similar to search and rescue, drones with computer vision can be employed to detect threats or anomalies in large areas. This can be useful, for example, in a military setting, border patrol, or in large facilities such as warehouses and power plants to prevent accidents.

Agriculture

Drones combined with computer vision technology can be employed for precision agriculture, which includes tasks such as crop health monitoring, pest detection, and yield estimation [11]. Multispectral imaging and machine learning algorithms allow the identification of stressed, damaged, or diseased plants, allowing faster intervention and maximizing crop yield [9].

Inspection

The high demand for inspecting buildings, bridges, and large-scale infrastructure such as power lines can be a time-consuming process when conducted manually. However, the implementation of computer vision technology and drone automation can significantly improve

---

[2] `https://en.wikipedia.org/wiki/Real-time_kinematic_positioning`

the speed and efficiency of these inspections. By training computer vision algorithms to search for cracks and structural damage, drones equipped with this technology can identify and report any issues, simplifying the inspection process and saving time and resources [10].

Chapter 5

Analysis and design

This chapter discusses the necessary components, devices, and technologies required to visualize detected objects from a drone in the Microsoft HoloLens 2 augmented reality headset, addressing the challenges and problems faced by this system while considering compatible existing solutions.

5.1 Components requirements

The primary components of this system include a drone, augmented reality glasses, and a server to facilitate communication between these devices. This section outlines the general requirements for each component.

Drone requirements

The drone must be capable of the following:

- Self-positioning
- Video streaming
- Transmitting flight data to the server
- Having a known field of view (FOV)
- Downward facing camera or gimbal

The drone needs a positioning system such as GPS to know its position and needs to send this information and its altitude and heading to a server. It needs to send its video to the server for object detection to run. Lastly, its field of view has to be known to precisely calculate an object's position in the drone's video feed. The camera has to face directly to the ground so that the center of the image will be equal to the ground position of the drone.

Augmented Reality Headset requirements

The augmented-reality headset needs the following capabilities:

- Six degrees of freedom (6DOF)

- Inside-out tracking
- Connection to the server

Six degrees of freedom are required to keep track of the position of the headset. Inside-out tracking is necessary while targeting outdoor use. Connection to the server allows the device to receive data from the drone.

Server requirements

The server is required to perform the following tasks:

- Hosting a server for video streaming.
- Hosting a server for drone's flight data transmission.
- Performing real-time object detection in the video stream.

5.2 Drone analysis

DJI drones, used for testing, support RTMP video streaming and have GPS capability. While flight data cannot be sent through consumer applications, it can be accessed via the software development kit (SDK). The DJI streamer application[1] has the capability of sending flight data to a WebSocket server. The app sends the following data required for the system in JSON format:

- Drone Id
- Altitude
- Latitude
- Longitude
- Compass

The app was implemented using DJI's MSDK and UXSDK. It is compatible with DJI's drones Mavic PRO, Mavic Mini, and others. However, the DJI Mini 2 drone used for most of the testing does not support UXSDK and therefore requires a different solution. DJI provides a sample app for the MSDK, which can be adjusted to send flight data to a WebSocket server in the same format as the DJI streamer. This solution is compatible with all of the mentioned drones. DJI also lists the field of view value on their website. However, during testing, it was found that these values represent the maximum possible values and are not valid in all situations.

5.3 Determining drone's field of view

Drone manufacturers, such as DJI, usually provide field of view (FOV) values on their websites (e.g., DJI Mini 2). However, these values are often the maximum possible values only valid in specific modes and may not apply in other situations. It is based on photo

---

[1] `https://github.com/robofit/drone_dji_streamer`

mode, where the whole sensor is used. In video mode, part of the sensor may be used for image stabilization, resulting in lower FOV values. Thus, it becomes necessary to measure the FOV manually. Two methods were used to measure the FOV of the DJI Mini 2:

Wall Measurement Method

1. Place the drone on an elevated surface facing a flat wall.
2. Measure the distance between the drone and the wall.
3. Mark the vertical and horizontal edges of the image on the wall.
4. Measure the distance between the marked points on the wall.
5. Use trigonometry to calculate horizontal and vertical FOV.

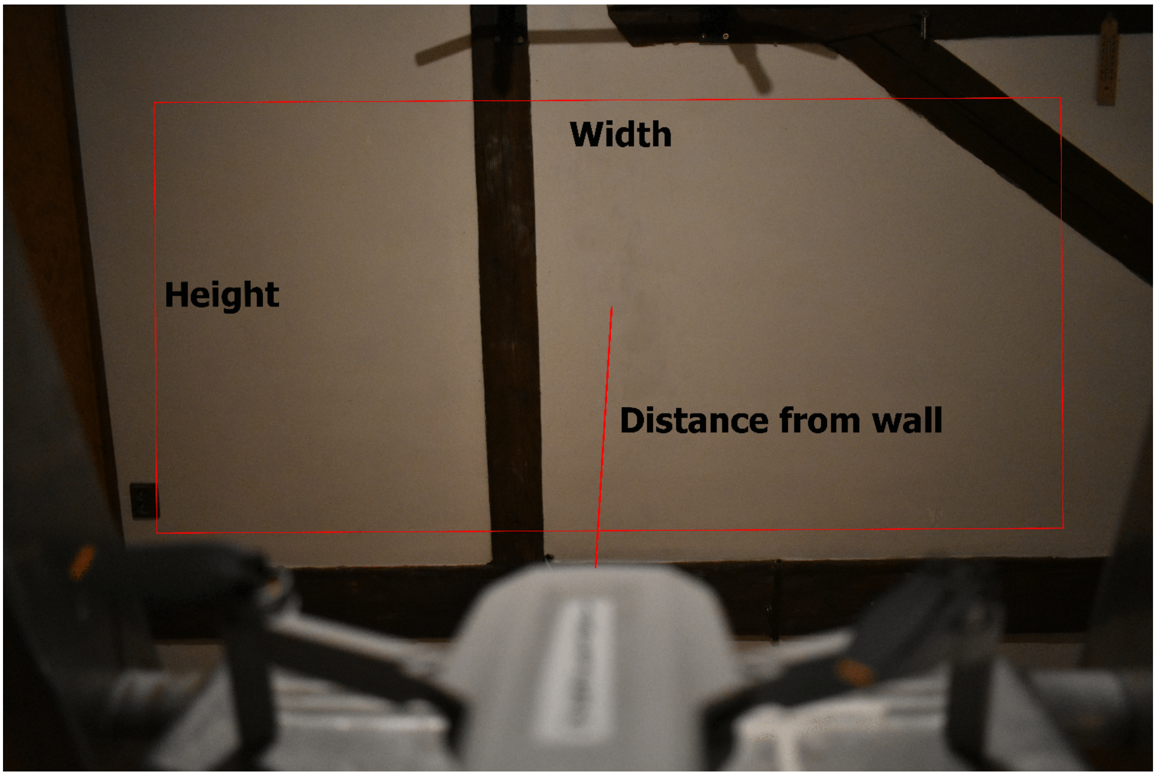

Figure 5.1: Wall FOV measurement

In this test, the following values were measured:

- Distance from wall: 2.381 meters
- Height: 1.597
- Width: 2.900

Using right-angle trigonometry, we can calculate half of the vertical or horizontal field of view:

$$\tan\theta = \frac{\text{opposite}}{\text{adjacent}} = \text{half horizontal FOV} = \frac{\text{half width}}{\text{distance from wall}} \tag{5.1}$$

This gives a value of **31.341** degrees as half of horizontal FOV, which is **62.682** degrees for the whole horizontal FOV. The vertical FOV is calculated in the same manner.

$$\text{half vertical FOV} = \frac{\text{half height}}{\text{distance from wall}} \tag{5.2}$$

And gives a value **18.54** degrees for half the vertical FOV or **37.08** degrees for the whole vertical FOV. This method has limitations, such as the size of the wall and the accuracy of the measurement device.

Landmark Measurement Method

1. Fly the drone outdoors and identify two spaced landmarks.
2. Position the drone so that each landmark is at one edge of the video feed.
3. Note the drone's altitude.
4. Use a tool to measure the distance between the landmarks (for example Google Earth's[2] measure function).
5. Apply trigonometry to calculate horizontal and vertical FOV, as in the previous method.


F/2.8, SS 346.69, ISO 100, EV 0, DZOOM 1.000, GPS (16.6100, 49.2196, 27), D 55.71m, H 74.60m, H.S 0.00m/s, V.S -0.00m/s

Figure 5.2: Drone photo for measurement

---

[2] `https://earth.google.com/`

Figure 5.3: Google Earth screenshot for measurement.

Figure 5.2 shows the actual photo from a drone with flight data in subtitles. Here only the height (H) or altitude value is important. Figure 5.3 shows the approximate distance along the top edge of the drone's image. The calculation is done in the same manner as in the previous method:

$$\text{half vertical FOV} = \frac{\text{half distance}}{\text{altitude}} \tag{5.3}$$

Which gives a value of **33.4** degrees for half of horizontal FOV or **66.8** degrees for the whole horizontal FOV and **21.1** degrees for half of vertical FOV or **42.2** degrees for the whole vertical FOV. This method has limitations, such as the availability of suitable landmarks and differences in terrain elevation. A large flat space such as a field would be ideal, however, such locations conversely tend to have fewer landmarks.

Conclusion

Using these two methods, we obtained two estimates, **62.682** and **66.8** degrees for the horizontal field of view and **37.08** and **42.2** degrees for the vertical field of view. These values are very different from DJI's advertised **83** degrees.[3] However, even from these two methods, we get results that are 4 to 5 degrees apart and will result in different accuracy. Further testing or information provided by the manufacturer would be needed to enable better accuracy. We can expect FOV to be similar in drones that share the same sensor size, such as DJI Mavic Mini and Mini 2 (see Figure 5.4).

---

[3] https://www.dji.com/mini-2/specs

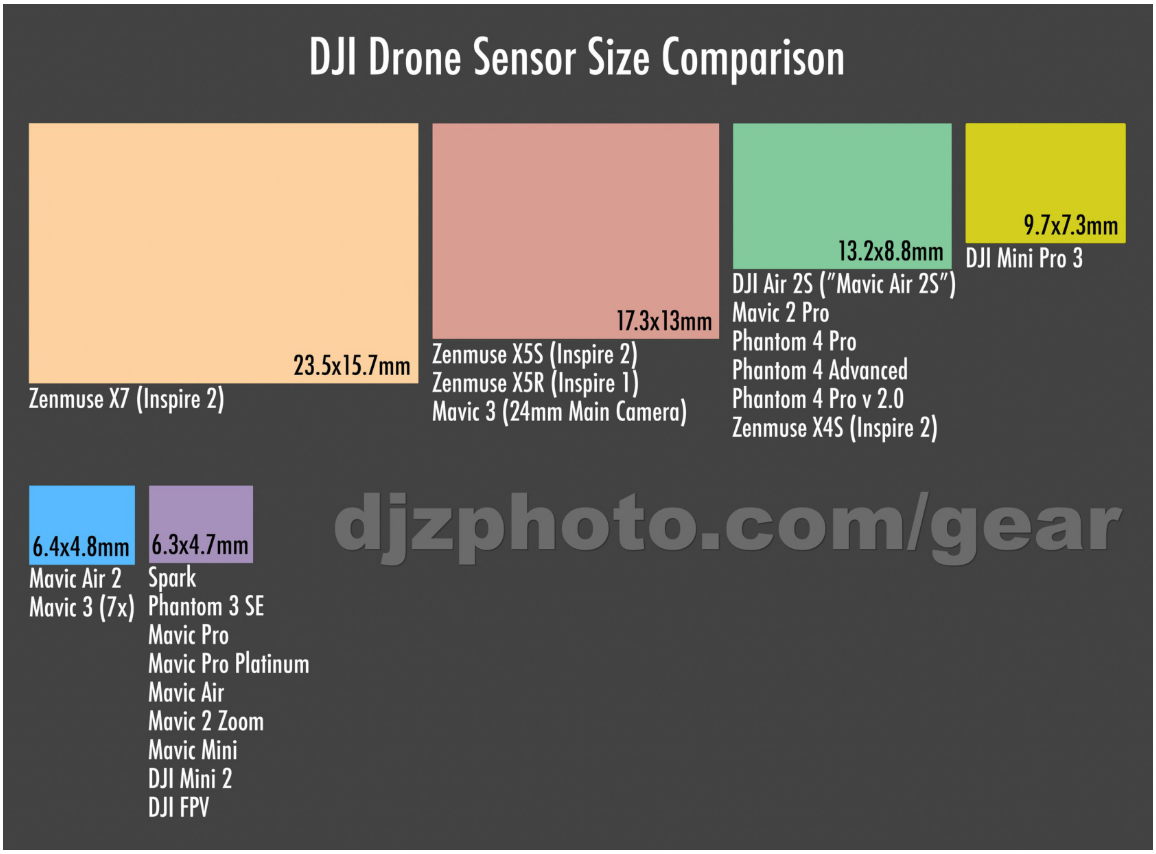

Figure 5.4: Sensor sizes of DJI drones.[1]

5.4 Headset analysis

The Microsoft HoloLens 2 headset allows for six degrees of freedom, is inside-out tracked, and can connect to an external server. It does not have GPS capability and therefore, an initial calibration with the drone's position data and orientation is required.

Mapbox

The Mapbox[4] software development kit for Unity is a powerful tool that provides an interface to Mapbox's data and services, allowing developers to create interactive and customizable maps for Unity applications. This software development kit (SDK) has proven to be of immense utility in creating location-based games, virtual and augmented reality experiences, and architectural visualizations among other applications.

The SDK provides access to Mapbox's rich datasets, including high-quality satellite imagery, terrain data, and street-level data. Developers can access this data via Mapbox's APIs and integrate it into their Unity applications.

One of the primary features of the Mapbox SDK for Unity is its ability to customize map styles. Developers can customize colors, icons, and labels, as well as the visibility of

---

[1] Image source: https://bit.ly/41jLqWC

[4] https://www.unity.com/

features on the map. This adaptability allows for the creation of maps that align with a project's specific aesthetic and functional requirements.

Another significant feature is the ability to convert real-world geographical coordinates to Unity's Cartesian coordinate system. This georeferencing ability enables developers to create accurate representations of the world or parts of it within the Unity engine, as well as position Unity GameObjects in real-world locations.

This feature allows this system to convert real-world geographical coordinates, obtained from the drone, to Unity coordinates to allow calculations between the headset's and drone's position.

5.5 Server analysis

The server can use a solution such as the Node-Media-Server[5] for hosting the RTMP video server. The data transmission of the server only needs to resend messages to all its other clients. Older version of the RoboFIT's drone_server [6] can be used. Lastly, the YOLOv5[7] library is the best option for real-time object detection as it provides a pre-trained model capable of detecting common objects. This model is not trained on data taken from drones therefore its precision in correctly identifying objects is significantly lowered, however, this does not affect the object locating aspect required for testing.

---

[5] https://www.npmjs.com/package/node-media-server?activeTab=readme

[6] https://github.com/robofit/drone_server

[7] https://github.com/ultralytics/yolov5

Chapter 6

Implementation

6.1 Overview

The drone transmits video feed and flight data to the server via the user's phone. Additionally, the user's location data can be sent to the server if enabled. The server hosts a RTMP server instance, an object detection process, and a WebSocket server instance. The drone's flight data and detection results are sent from the server to the headset where they are visualized. An overview of the entire system is illustrated in Figure 6.1.

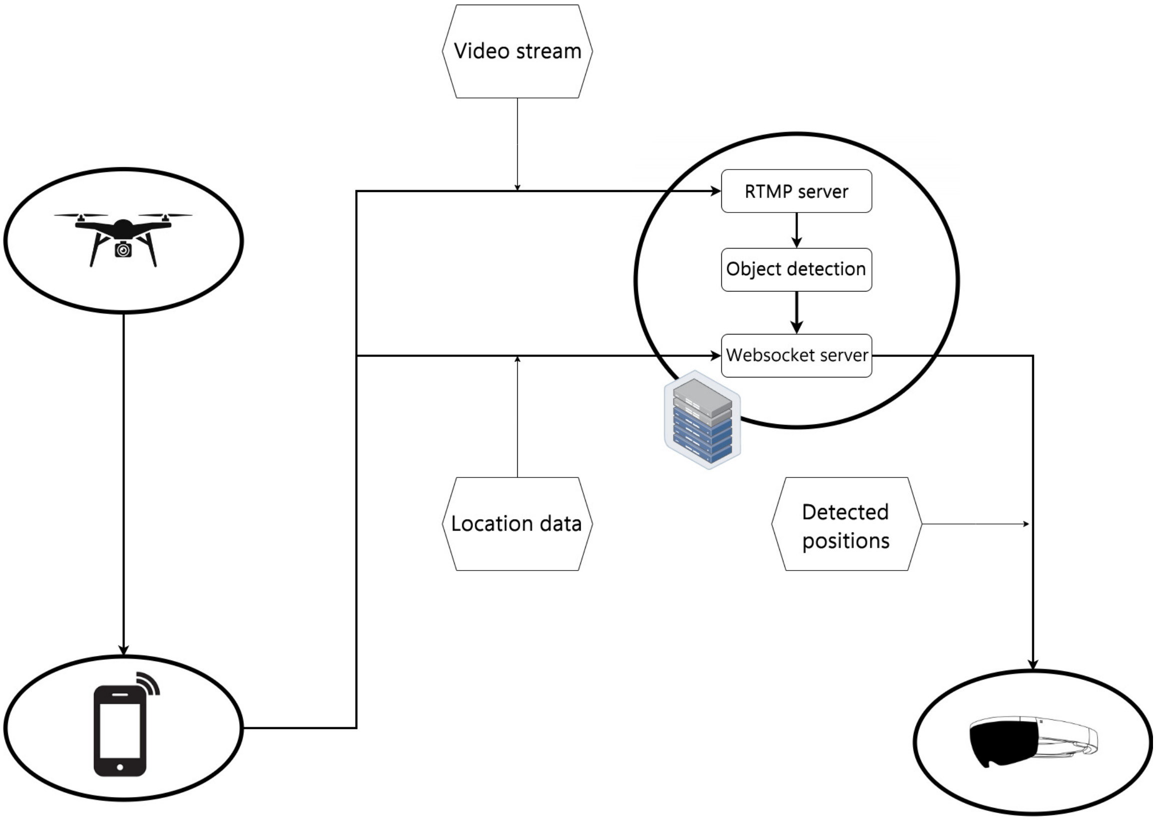

Figure 6.1: System overview.

6.2 Drone

DJI Streamer

The DJI Streamer application (`https://github.com/robofit/drone_dji_streamer`) implements the capability to send flight data to a WebSocket server. When using this application with this system it is advised to change the interval at which flight data is sent from 100 milliseconds to 1000 milliseconds to prevent buffering in the headset when computing object positions due to it's limited computational power. This can cause higher latency in the system. The delay is set in the **MainActivity.java** file via a variable named **delay**. The flight data is sent in JSON format with a specific structure. The JSON object contains the following keys:

- DroneId: A string representing the unique identifier of the drone.
- Altitude: A floating-point number, the altitude of the drone in meters.
- Latitude: A floating-point number, the latitude of the drone's position.
- Longitude: A floating-point number, the longitude of the drone's position.
- Pitch: A floating-point number, drone's pitch angle in degrees.
- Roll: A floating-point number, drone's roll angle in degrees.
- Yaw: A floating-point number, the drone's yaw angle in degrees.
- Compass: A floating-point number, drone's compass heading in degrees.

An example of the JSON format adhering to the specification is shown below:

```
{
   "DroneId": "DJI-MAVIC_PRO",
   "Altitude": 15.6,
   "Latitude": 49.22720288202451,
   "Longitude": 16.597363361433125,
   "Pitch": 4.9,
   "Roll": -0.1,
   "Yaw": -33.9,
   "Compass": -33.9
}
```

The app is built using DJI's MSDK (Mobile SDK) and UXSDK (User Experience SDK). The MSDK is a platform that allows developers to build custom applications for DJI products. The UXSDK is a suite of user interface components and tools that simplifies the development of DJI app interfaces. Some devices support only the MSDK and not the UXSDK. The app is therefore limited to devices that support UXSDK. On the Android operating system, the DJI streamer app works with most DJI devices but excludes DJI Mini 2, Mini SE and Air 2S. Because most of the testing was done with DJI Mini 2 a different solution was introduced.

Modifications to DJI Sample App

The DJI Sample App is designed to showcase the capabilities of the MSDK, aimed at developers who wish to better understand MSDK functionalities, and create custom applications for DJI drones. The app offers options for drone control and sending video to an RTMP server. To use the SDK, developers are required to register on DJI's website[1] to obtain an SDK key. The app requires a registering action done by clicking the Register button on the main screen at every start. Several adjustments have been made to the DJI Sample App to fulfill the system requirements:

- **Added WebSocket Capability**: WebSocket functionality has been integrated to enable communication between the app and a WebSocket server. The code for WebSocket functionality is taken from the DJI Streamer App. The data sent follow the same JSON format as the DJI Streamer app with changes limiting the interval at which data is sent to the server from 100 milliseconds to 1000 milliseconds.

- **GUI changes**: The Live stream page was changed in files **LiveStreamView.java** and **view_live_stream.xml**. A text input field has been added for entering the WebSocket server IP address. Additionally, buttons have been included to connect to the WebSocket server and to send a reset message.

- **Sending Location Provider Data**: The app adds the capability to send location data from the user's phone to the WebSocket server. Google's FusedLocationProvider[2] is utilized to achieve this functionality. The FusedLocationProvider is a location API provided by Google that intelligently combines GPS, Wi-Fi, and cellular network data to provide accurate location information while optimizing battery usage. Additionally, a compass heading of the mobile device is sent along with location data. This data is sent only when all the sensors are enabled. This data is logged on the server and is more accurate than the drone's GPS. It can be used to see the GPS deviation or the georeference recorded images more accurately.

---

[1] https://developer.dji.com/mobile-sdk/

[2] https://developers.google.com/location-context/fused-location-provider

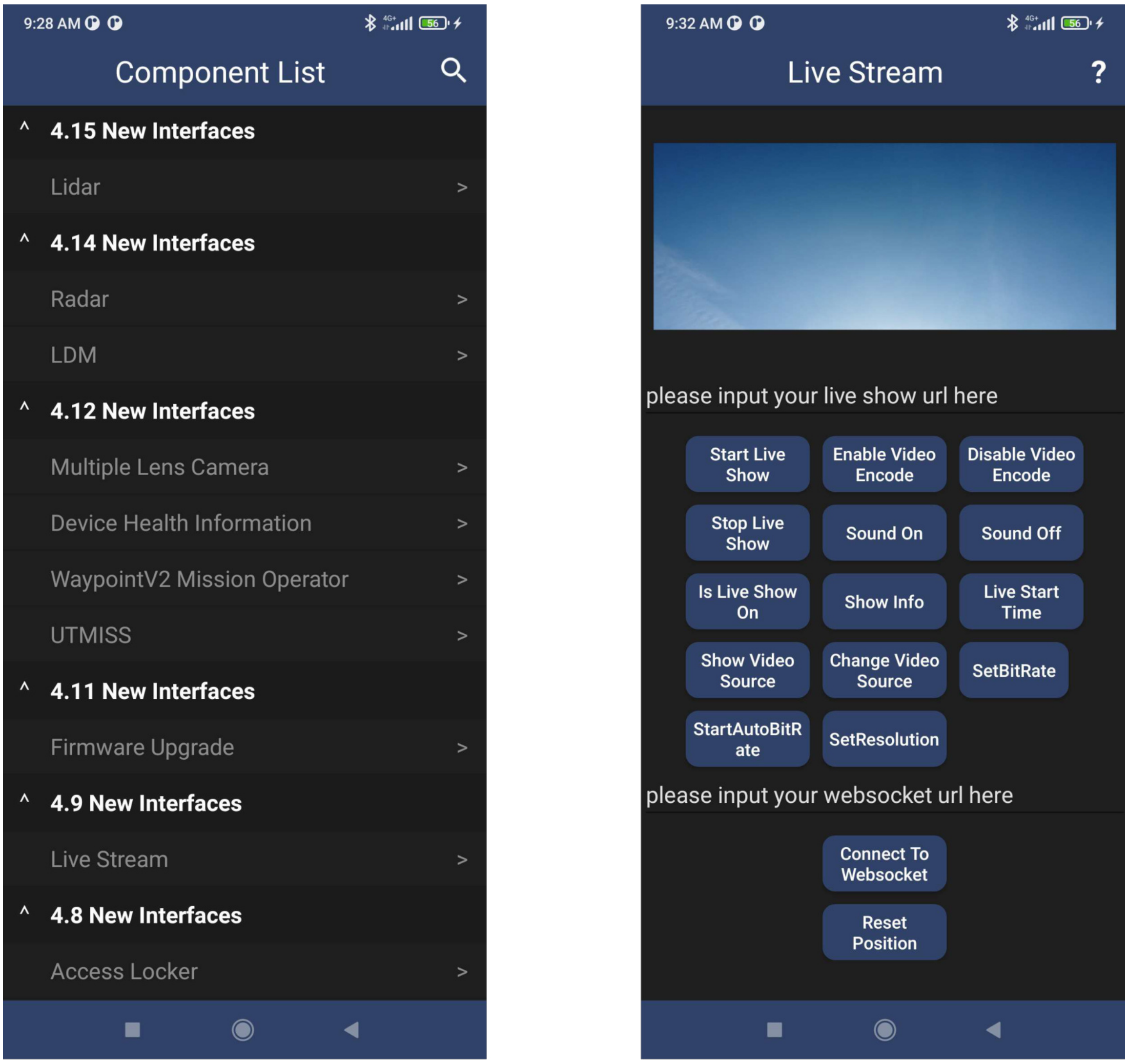

Figure 6.2: Adjusted DJI Sample App

6.3 Server

Websocket Server

The websocket server's requirements are straightforward: it must resend any received message to all other connected clients. This functionality is already implemented in the RoboFIT's websocket server, per its requirements, and therefore does not necessitate any modifications. The drone's flight data are transmitted to the server's IP address and a predefined port.

RTMP Server

For the RTMP server, the node-media-server implementation also remains unaltered. Similar to the WebSocket server, the drone publishes its video stream to the server's IP address and the specified port.

Object Detection with YOLOv5

Object detection is performed using the YOLOv5 library, a real-time object detection system developed by Ultralytics. The acronym „YOLO“ stands for „You Only Look Once“, which signifies the system's capability to detect objects in an image using a single forward pass of a neural network. This property enables real-time performance.

The library includes a pre-trained model for detecting common objects such as people, vehicles, clothing, or animals (see Figure 6.3). A limitation of this model is that it is trained on images of common objects captured from typical angles. To simplify object position calculations, the drone is required to transmit its video from a top-down (90 degrees down) view, keeping the center of the image as the reference ground position of the drone. This perspective can confuse the model, as most drone-captured images are not directly taken from a top-down view. For instance, the model may misidentify cars as cell phones (see Figure 6.4) or generate multiple detections for a single object (see Figure 6.5).

This issue is not significant for testing purposes, as in real-world applications, the model would be trained for a specific task, such as detecting a drowning person in water or heat signatures in an avalanche or similar use cases described in section 4.2.

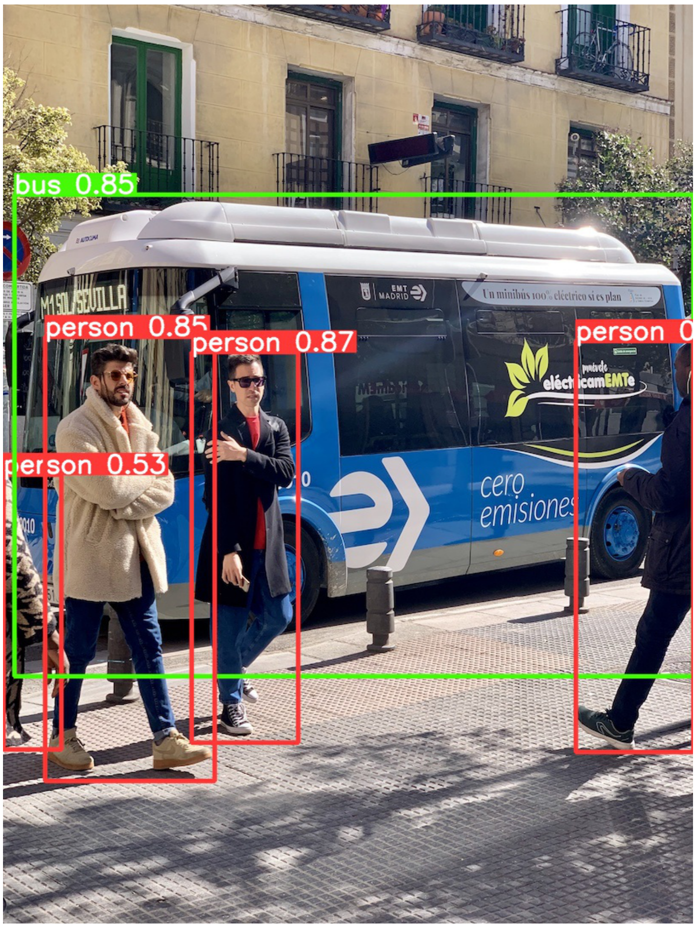

Figure 6.3: Common objects detected with the YOLOv5 library.[1]

---

[1] Image provided for testing with the YOLOv5 library: https://github.com/ultralytics/yolov5

Figure 6.4: Cars are detected incorrectly as cell phones when the source image is taken from a top-down view.

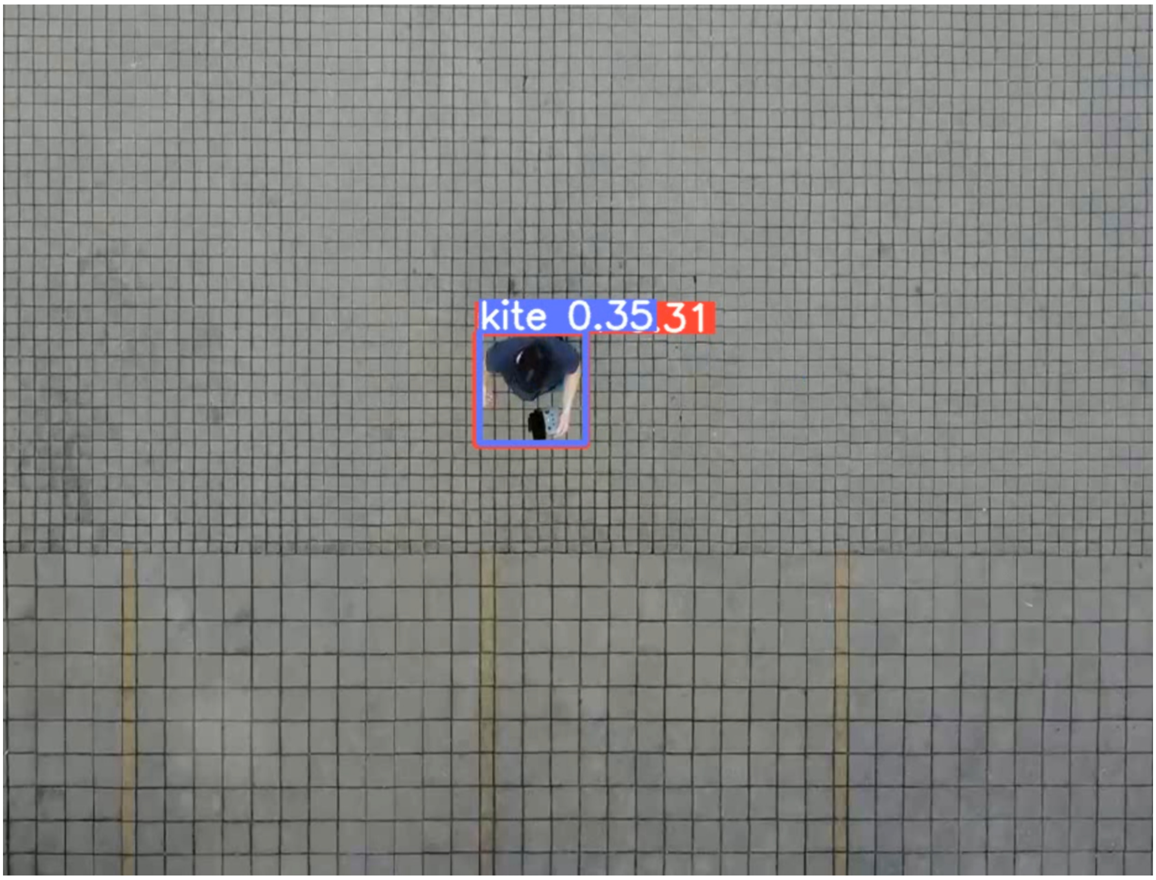

Figure 6.5: Multiple detection results on a single object on source image taken from a top-down view.

YOLOv5 source, result and adjustment

YOLOv5 supports various input sources, such as webcams, images, videos, YouTube links, and RTSP, HTTP, and RTMP stream links. The algorithm processes the source images and returns an identification (name) of the object while displaying a bounding box around it. For this system, two adjustments to the file **detect.py** are required:

1. The first adjustment involves obtaining the center coordinates of detected objects. This is calculated from the bounding boxes.

2. The second adjustment incorporates WebSocket connectivity into the system. The detection results are transmitted to the WebSocket server using a JSON format with a specific structure. The JSON object contains a key named `DetectedObjects`, which holds an array of detected object instances. Each instance in the array is an object containing the following keys:

    - `DetectedObjectName`: A string representing the name or class of the detected object (e.g., „person“, „fire hydrant“).
    - `x`: An integer representing the x-coordinate of the center of the detected object’s bounding box.

- `y`: An integer representing the y-coordinate of the center of the detected object's bounding box.

Example of the detection result JSON message:

```
{"DetectedObjects": [
    {
        "DetectedObjectName": "person",
        "x": 970,
        "y": 416
    },
    {
        "DetectedObjectName": "fire hydrant",
        "x": 730,
        "y": 354
    }
]}
```

6.4 Headset

Server communication

The server communication script listens for incoming messages, which can be detected objects messages, drone flight JSON messages, or reset messages. Flight data JSON messages are deserialized into a drone class object, identified by its ID. Detected objects messages are deserialized into a list of objects detected by a specific drone. This design allows for potential future improvements, such as supporting object detection from multiple drones simultaneously.

During a session, drone objects are persistent, ensuring that their data remains available throughout the session. On the other hand, detected objects are not persistent. Due to the lack of continuity between previously detected objects and the current detection results, all detected objects must be deleted and recreated with each new detection update. This approach only displays the most recent object detection information.

Object position calculation

To accurately display the position of an object in the Unity project, it is essential to determine the user's location, the drone's location, and the object's position within the drone's image. The drone's flight data, which includes its GPS coordinates, serves as the real-world reference point. Mapbox is used to fuse the usage of real-world coordinates and unity coordinates.

Due to the absence of a compass and GPS capabilities in the Microsoft Hololens 2 headset, its position and orientation are synchronized with the drone's coordinates and heading when the user is standing directly beneath the drone. At this point, the drone and headset share the same coordinates and heading. This is done when the drone is first connected to the WebSocket server and again whenever a reset message is captured from the server (But due to an issue described later in 7.1 it is required to start the application while standing directly beneath the drone in flight). The drone's position is

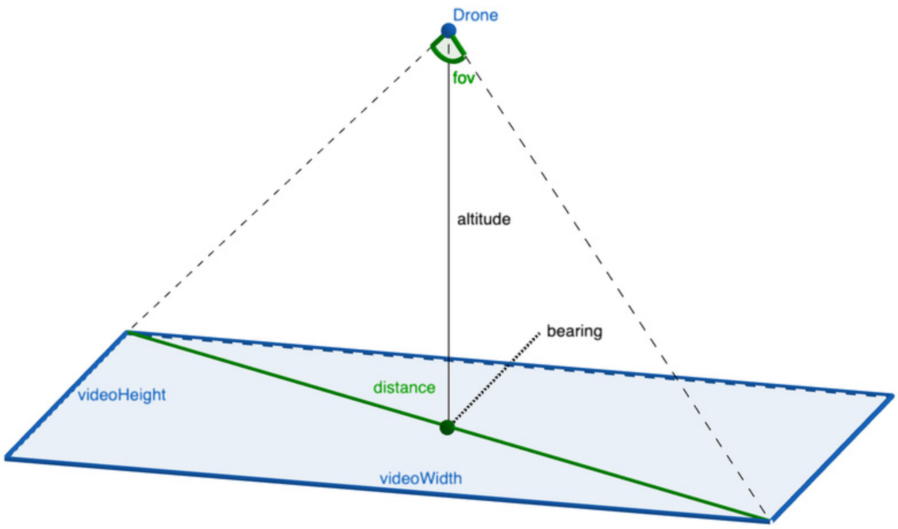

Figure 6.6: Diagram showcasing the ground sample distance calculation.[1]

then continuously updated using Mapbox, while the headset tracks its position utilizing its 6 degrees of freedom capability.

As the drone's video feed captures a flat area below it while facing directly downward (90 degrees), the center of the image corresponds to the drone's ground position (see Figure 6.6. The horizontal and vertical ground distance from edge to edge can be calculated using the drone's horizontal and vertical field of view (FOV) values and altitude, as described in section 5.3 but solving for the distance instead of the FOV.

$$\text{GroundDistance} = 2 * \text{Altitude} * \tan(\text{FOVinRadians}/2) \tag{6.1}$$

The DJI drone's RTMP stream resolution is 1920x1080, which allows for the calculation of the ground sample distance (GSD) in both vertical and horizontal domains. The GSD represents the real-world distance covered by one pixel.

Ground sample distance is calculated as follows:

$$\text{GSD} = \text{GroundDimension}/\text{ImageSizeInDimension} \tag{6.2}$$

With the knowledge of the detected object's x and y coordinates in the image, its position can be calculated by multiplying the distance from the center of the image in resolution by the GSD. This process is conducted separately for the vertical and horizontal domains to ensure higher accuracy. Now the coordinates represent the distance from the drone in meters which are equal to unity coordinates.

Since the drone's orientation may not always be the same, a rotation matrix is utilized to adjust the object's position accordingly. The rotation matrix pivots the x and y coordinates around the origin. Unity's support for local object coordinates, in this case, the drone, allows the calculation to be performed using zero as the origin. Finally, the coordinates are transformed into global Unity coordinates using Unity's built-in transform method.

---

[1] Image source: `https://blog.roboflow.com/georeferencing-drone-videos/`

Object Visualization

In the Hololens 2, the height is set to zero at the headset's position, necessitating an adjustment to remove this offset. However, the adjustment may not always be accurate. A 180-degree rotated cone-shaped object was chosen as an indicator for detected object locations, as it appears to point to the ground when above ground, and appears as a flat circle indicator when viewed from above. The object name, identified by the server's object detection, is displayed above the object, along with the distance to the object.

The text above the object is rendered using the TextMeshPro Unity package. As objects located farther away might be invisible or limited by the Hololens 2's resolution, the text is scaled to maintain a consistent apparent size at any distance (see Figure 6.7). When viewed in the headset, the text will appear farther but bigger, further allowing an intuitive distance judging. This effect is not possible to show in a flat image. Additionally, the text is rotated to always face the main camera. This ensures that the object's name and distance information remain visible and easily readable, regardless of the object's distance from the user or the user's viewing angle.

Figure 6.7: Scaled text shown on an object at distance

Figure 6.8: Scaled text shown close to user

Chapter 7

Testing

Testing was constrained by several factors, including limited access to the Microsoft Hololens 2 headset, weather conditions, and the drone's flight time. To mitigate some of these issues, an indoor testing method was employed. This method involved using OBS Studio[1] to create an RTMP stream representing the drone's video feed and a script to simulate the drone's flight data. While this approach facilitated progress, it had limitations in identifying specific issues, which were only discovered during proper outdoor testing. This chapter discusses these issues and their potential fixes and other improvements.

7.1 Issues identified during testing

Clutter

When a large number of objects were detected far from the user, the text indicators overlapped and became unreadable (see Figure 7.1). This could be resolved by imposing a limit of objects in a certain section of the field of view and showing a single object with information about the number of objects detected in this section. The limited computational power of HoloLens 2 will need to be taken into consideration.

Outside visibility

Microsoft HoloLens 2 is designed for indoor use. In good weather conditions suitable for drone flights, the sunlight sometimes made the headset and objects barely or not visible at all. Until a new headset is released made for outside use there is no direct solution for this issue. A lot of testing needed to be done recording in the headset and then reviewing the footage.

Drone synchronization

During indoor testing, the angle synchronization of the Unity scene with the real world could be tested and worked as expected. However, position synchronization could not be thoroughly tested due to the constraints of the physical room. While this error was not entirely resolved, it can be overcome by starting the application directly in the synchronization position below the drone, maintaining Unity's origin point in this spot. This is an

---

[1] https://obsproject.com/

error in the code related to the object position calculation relative to the user and could be fixed with future debugging.

FOV inaccuracy

The field of view (FOV) calculations differed by a few degrees, introducing more inaccuracy in object positioning. Using calibration methods for computer vision, further discussed in 7.2 for the cameras would be ideal to produce the most accurate results.

GPS inaccuracy

The GPS accuracy fluctuates during the drone's flight. Furthermore, the drone's GPS coordinates exhibited a deviation of about 15 meters when compared to the user's location data sent to the server from the phone using FusedLocationProvider. This deviation appears to be persistent and synchronization with the headset cancels it out. However, this issue would hinder the system's use for additional georeferencing or similar purposes.

Newer drones with more precise GPS modules or added tracking systems could resolve this issue. In addition, methods described in 4.1 could be used in conjunction to provide more accurate georeferencing capabilities for certain use cases.

Bandwidth limitations

To connect to the server when testing outdoors, cellular data were used for the connection. The upload speed limit of cellular data could not meet the required speed for handling the RTMP stream and caused severe lag, rendering the video feed unusable for object detection. This required the use of a laptop, connected to a local network created on the phone, to handle the RTMP server and object detection. An adjusted system overview diagram can be seen in Figure 7.2.

More expensive drones with computer vision capability would be able to do object detection onboard. As the RTMP stream is also the primary cause for latency, removing the requirement to stream video would reduce it significantly.

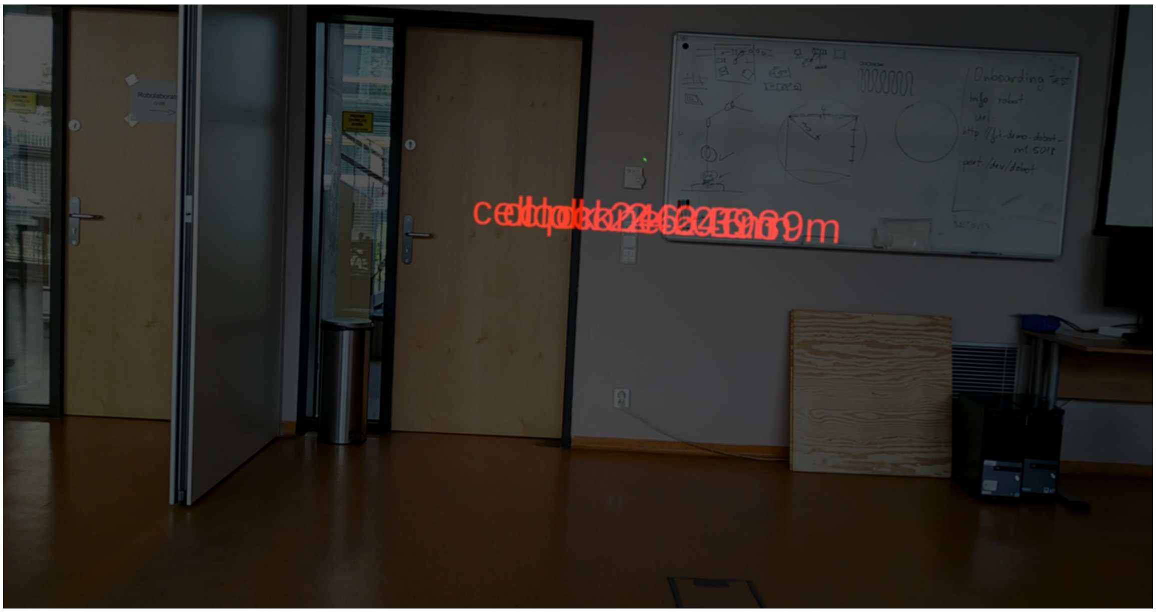

Figure 7.1: Multiple object detections cluttering the user's view.

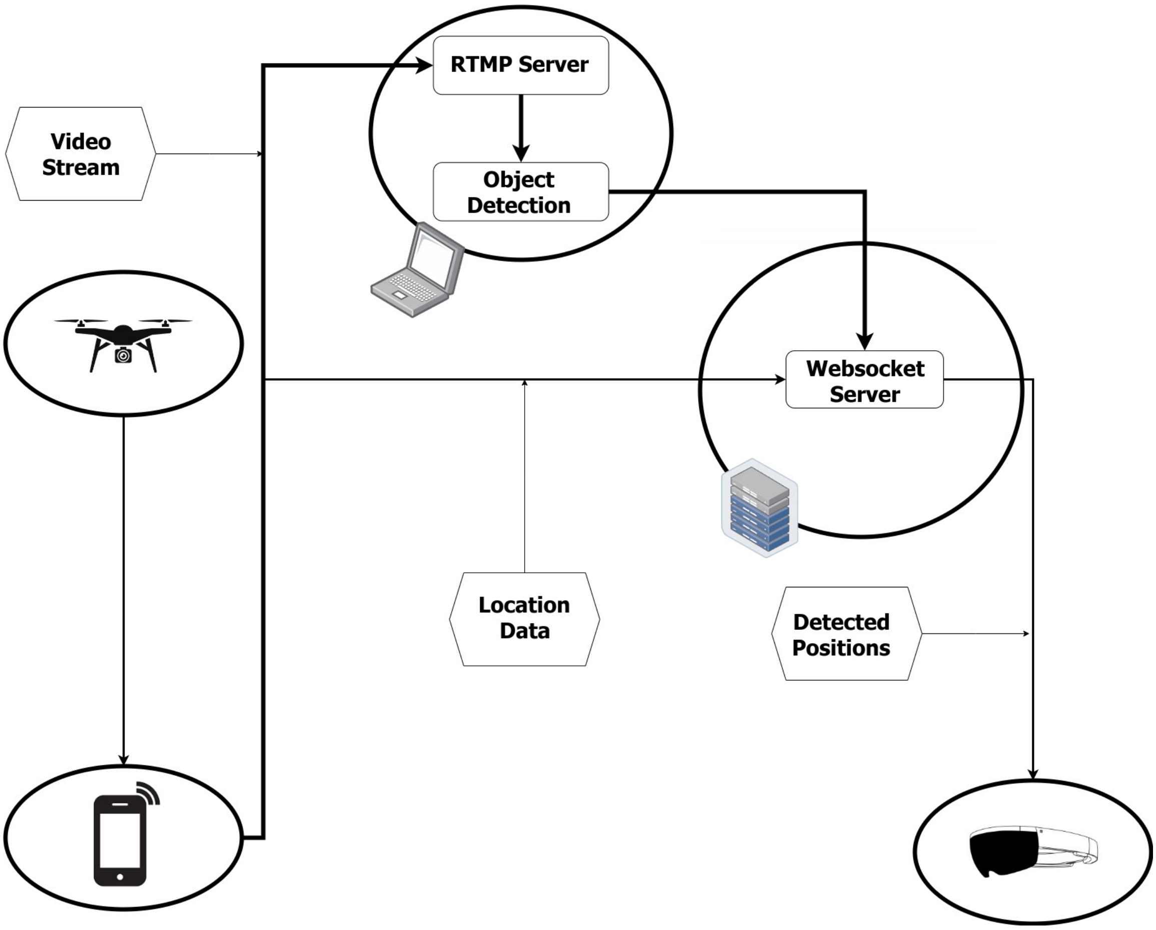

Figure 7.2: Adjusted system overview accounting for limited cellular data bandwidth.

With proper synchronization and altitude of above approximately 20 meters and depending on other circumstances such as GPS satellite count and accuracy. The accuracy was under 3 meters. This was tested by detecting the wearer of the headset and noting the distance shown on the detected object representing the wearer (see Figure 7.3).

Figure 7.3: Headset wearer indicator shown from the wearer's point of view.

7.2 Future Improvements

Camera calibration

Camera calibration in computer vision is a process that helps to estimate the intrinsic properties, extrinsic properties, and lens distortion of a camera.

Intrinsic parameters relate to the internal characteristics of the camera such as focal length, optical centers, and skew coefficient. These parameters are typically constant and unique to each camera.

Extrinsic parameters pertain to the position and orientation of the camera in the real world, defined by translation and rotation vectors.

The calibration process begins by collecting images of a known pattern, often a chessboard, from varying angles. From these images, feature points (like corners of the squares in a chessboard) are identified. Using the known real-world distances between these features, algorithms are utilized to estimate the camera parameters.

Once estimated, these parameters are refined through an error minimization process, ensuring the predicted points from the camera model align as closely as possible with the actual positions in the image.

The final product of this process is a camera matrix, which serves as a mathematical representation of the camera's characteristics. This matrix is used to correct distortion, determine real-world dimensions from pixel sizes, and perform other image-processing tasks [12].

Computer vision improvements

Training a tailored model for distinct use cases, like identifying individuals in search and rescue operations, or a generalized model designed specifically for drone-captured images, would significantly enhance usability.

Implementing a tracking solution that understands the continuous existence of objects in a video sequence could promote more consistent visualizations. This way, objects can possess a defined time of death, preventing them from disappearing simply because a detection is missed in single or several frames.

Other potential improvements

- High-precision localization: Exploring the use of higher-precision GPS modules or alternative localization methods, such as inertial navigation systems, could further increase the system's accuracy in tracking and positioning.
- Support for multiple drones: Adding support for multiple drones would enhance the system's capabilities, as automation is one of the key benefits of employing drones for various tasks.
- Improved headset technology: Future headsets designed for outdoor use could enable better visibility in bright conditions, expanding the range of applications, environments, and conditions in which the system could be effectively utilized.
- Improved user interface: Decluttering user's view when too many objects are detected. Showing indicators for objects outside the headset's field of view.

The system's overall performance and applicability could be significantly improved by addressing these areas, making it a more versatile and accurate solution for various scenarios.

Chapter 8

Conclusion

This project demonstrates a proof of concept for an AR-based object detection system using a DJI drone and Microsoft Hololens 2 headset. The system allows users to identify and locate detected objects in real-time, augmenting the object information onto their field of view. While the current implementation has limitations, including accuracy, visibility, and synchronization issues, it provides a foundation for future improvements and enhancements.

The potential applications of such a system are vast, ranging from search and rescue operations to infrastructure inspection and environmental monitoring. By improving the system's accuracy, reliability, and usability, it could become a valuable tool for various industries and use cases, automating certain tasks, while freeing the hands of users for performing other tasks.

In conclusion, this project highlights the potential of combining AR and drone technologies for object detection and visualization, providing a glimpse of what can be achieved in the future with this system with newer and better devices.

Bibliography

[1] Angelopoulos, A., Hale, A., Shaik, H., Paruchuri, A., Liu, K. et al. Drone brush: Mixed reality drone path planning. In: IEEE. *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2022, p. 678–682.

[2] Bellan, R. *Augmented reality finds a foothold in cars via safety features*. Feb 2022. Available at: `https://techcrunch.com/2022/02/10/augmented-reality-finds-a-foothold-in-cars-via-safety-features/`.

[3] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D. et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*. IEEE. 2016, vol. 32, no. 6, p. 1309–1332.

[4] Colomina, I. and Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS Journal of photogrammetry and remote sensing*. Elsevier. 2014, vol. 92, p. 79–97.

[5] Dianxi, Z., Danhongand, C. and Zhen, G. Design of Intelligent Firefighter Helmet Based on AR Technology. In: IOP Publishing. *IOP Conference Series: Earth and Environmental Science*. 2021, vol. 621, no. 1, p. 012176.

[6] Erat, O., Isop, W. A., Kalkofen, D. and Schmalstieg, D. Drone-augmented human vision: Exocentric control for drones exploring hidden areas. *IEEE transactions on visualization and computer graphics*. IEEE. 2018, vol. 24, no. 4, p. 1437–1446.

[7] Harwin, S. and Lucieer, A. Assessing the accuracy of georeferenced point clouds produced via multi-view stereopsis from unmanned aerial vehicle (UAV) imagery. *Remote Sensing*. Molecular Diversity Preservation International (MDPI). 2012, vol. 4, no. 6, p. 1573–1599.

[8] Kyjac, M. *Vizualizační nástroj pro pilota dronu v Microsoft HoloLens 2*. Brno, CZ, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Available at: `https://www.fit.vut.cz/study/thesis/24785/`.

[9] Lum, C., Mackenzie, M., Shaw Feather, C., Luker, E. and Dunbabin, M. Multispectral imaging and elevation mapping from an unmanned aerial system for precision agriculture applications. In: *Proceedings of the 13th International Conference on Precision Agriculture*. 2016, vol. 31, p. 4–10.

[10] Munawar, H. S., Ullah, F., Heravi, A., Thaheem, M. J. and Maqsoom, A. Inspecting buildings using drones and computer vision: A machine learning approach to detect cracks and damages. *Drones*. MDPI. 2021, vol. 6, no. 1, p. 5.

[11] Patrício, D. I. and Rieder, R. Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review. *Computers and electronics in agriculture*. Elsevier. 2018, vol. 153, p. 69–81.

[12] Qi, W., Li, F. and Zhenzhong, L. Review on camera calibration. In: IEEE. *2010 Chinese control and decision conference*. 2010, p. 3354–3358.

Appendix A

Medium contents

- xosval04.pdf – Thesis text.
- tex/ – LaTeX latex source files.
- src/ – source files of the system.
- video.mp4 – Video showcase of the system.