



UNIVERSITÀ
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
DEPARTMENT OF INDUSTRIAL ENGINEERING
FONDAZIONE BRUNO KESSLER
Doctorate Program in Industrial Innovation

HOW THE BLEND OF PHYSICAL AND DIGITAL WORLDS IS RESHAPING THE INDUSTRIAL FIELD

Lorenzo Orlandi

Academic Advisor

Prof. Nicola Conci

Università degli Studi di Trento

Industrial Tutor

Dr. Daniele Sevegnani

Arcoda S.r.l.

March 2025

Abstract

These days, the evolution of technology is transforming the way we perceive the world. More and more companies are having the opportunity to improve efficiency and safety by adopting new technologies, particularly in the realm of extended reality and the digitalization of 3D content.

Arcoda is a company specialized in cartography applications as its core business. These applications typically consist of a map with digital content overlaid, such as markers representing the location of points of interest, such as cars or electric cabins. Arcoda sees extended reality as the natural evolution of its product. Transitioning from map-based visualization to overlaying digital content directly onto the real world offers multiple benefits. The most exciting advantage is the ability to reconstruct 3D contents and interact with them directly in real-world spaces.

The focus of this thesis is to address the current challenges in industrializing these technologies. After identifying the specific requirements of the company's use cases, the next step is to develop solutions that meet these needs. Specifically, we will explore the following key areas: experimenting with smart glasses and mixed reality for industrial use cases; identifying algorithms for spatial calibration and temporal synchronization between two devices for 3D georeferenced data acquisition and augmented reality purposes; and conducting a qualitative analysis of 3D reconstruction algorithms.

Keywords

Mixed Reality, Location-based Augmented Reality, Smart glasses, Real-Time Kinematic Global Navigation Satellite System (RTK-GNSS), 3D Reconstruction, Neural Radiance Field(NeRF), 3D Gaussian Splatting(3D-GS)

Acknowledgments

First and foremost, I would like to thank my family and my girlfriend, Francesca, for their continuous support throughout this journey.

I am also grateful to the company Arcoda s.r.l. and the Multimedia Signal Processing and Understanding Lab (MMLab) research group for giving me this incredible opportunity, especially to my company advisor Daniele Sevegnani and my academic advisor Nicola Conci, without whom this experience would not have been possible.

Further thanks go to the people I have met during my university path, who have collaborated with me, particularly the students that have collaborated in the projects and thesis and the other members of my research group.

Finally, I would like to thank my work colleagues for their support and help, with special mention to my brother Francesco, with whom I have always been able to discuss and collaborate on the most complex decisions regarding the final development of the product derived by this work.

Contents

1	Introduction	1
1.1	The Context	1
1.1.1	The PhD in Industrial Innovation	1
1.1.2	The Research Topic	2
1.2	The Problem	3
1.3	Innovative aspects	9
1.4	Structure of the Thesis	10
2	State of the Art	11
2.1	Hardware analysis	11
2.1.1	Hardware for 3D Reconstruction	11
2.1.2	Hardware for Mixed Reality purposes	15
2.2	Capturing the 3D scene	18
2.2.1	Spatial Calibration analysis	20
2.2.2	Major Works in the Field	21
2.3	3D Reconstruction algorithms	28
2.3.1	Background	28
2.3.2	Photogrammetry	29
2.3.3	Neural Radiance Fields	29
2.3.4	3D Gaussian Splatting for Real-Time Radiance Field Rendering	31

3	The Problem	33
3.1	The Solution at a Glance	33
3.2	Hardware Identifications	36
3.3	All Components as a Rigid-body	36
3.4	Spatial Calibration	38
3.5	Temporal Synchronization	39
3.6	3D Reconstruction of a Scene	40
4	Capturing 3D scenes	43
4.1	GNSS System	45
4.1.1	Principles of Localization	46
4.1.2	The GNSS standard	47
4.1.3	Real Time Kinematic GNSS	49
4.2	Used Materials	50
4.3	Proposed Solution	53
4.3.1	Spatial Calibration between the two devices	53
4.3.2	Temporal Alignment between the two devices	58
4.3.3	Obtaining the Transformation Matrix between the camera of the Smartphone and the RTK-GNSS device	63
4.4	Results	65
4.5	Discussion	72
4.6	Extended Reality	73
5	3D Reconstruction	79
5.1	Methodology	81
5.1.1	Dataset acquisition	82
5.1.2	Methodologies employed	83
5.1.3	Comparative analysis framework	88
5.2	Discussion	89

6	Meta-Trainer	91
6.1	Meta-Trainer	92
6.2	Method	95
6.2.1	The Meta-Trainer avatar	95
6.2.2	The Application Workflow	96
6.2.3	Workout Session	99
6.3	Scores and Feedback	100
6.4	Results	101
6.4.1	Hardware Setup	101
6.4.2	Application	101
7	Conclusion	105
7.1	Capturing 3D scene	105
7.2	3D Reconstruction	106
7.3	Meta-Trainer	107
7.4	Future research and consideration	107
	Bibliography	109

List of Tables

4.1	Detailed explanation of the relevant terms in Eq. (4.3) . . .	66
4.2	Comparison of the proposed solution with other state-of-the-art approaches for location-based AR. Our translation accuracy is estimated at 2m distance. [KEYS: Trans. acc is the translation accuracy, Rot. acc. is the rotation accuracy, Coverage refers to the scalability of the solution, and Dynamic reports the robustness to environmental visual variations]	71
5.1	From this table, we describe the number of images and their resolution for all acquisitions, then we also include, for each scene, the accuracy of translation with respect to the GCS of the images.	83

List of Figures

1.1	Reality–virtuality continuum, image taken from [1]	2
1.2	This is the final goal of our system, the acquisition of the real world, the reconstruction, and finally the visualization in AR	3
2.1	Scanner are used in the field, Image taken by [2]	12
2.2	An iPhone with LiDAR, with the representation of the depth map, image taken by [3]	13
2.3	An example of images drone acquisition, for 3D reconstruction, image taken by [4]	14
2.4	Smart glasses VS Smartphone for Location-based Augmented reality [5]	16
2.5	Xreal smart glasses setup	17
2.6	Test on the Hololens gesture recognition capability	17
3.1	Custom bracket for the camera and RTK-GNSS device (left), with an indication of the reference coordinate systems (right), for the RTK-GNSS and the smartphone. The X-axes are in red, the Y-axes in green, and the Z-axes in blue.	34
3.2	The type of MoCap that we have used in our work	35
3.3	A screenshot acquired from the final application that shows the over-imposition of the holograms in the real world . .	37

3.4	An example of visual pattern used for the extrinsic calibration of cameras	39
3.5	A screen acquired from a 3D-GS viewers about an excavation	40
4.1	Architecture of the proposed solution: we present all the components of our system and we highlight the role of the spatial and temporal calibration in our flow. $T(MCST \rightarrow MCSP)$ represent the spatial transformation between the RTK GNSS and the smartphone; $AR, GR, POSE$ represent, respectively, the ARcore packets from the smartphone, the RTK packets, and the final pose calculated, which will be assigned to the image at each time instant.	46
4.2	Scheme about the RTK system, Image taken by [6]	51
4.3	Custom 3D printed structure for the creation of the rigid body of the RTK-GNSS device within the MoCap reference system.	55
4.4	Custom bracket for the camera and RTK-GNSS device (left), with an indication of the reference coordinate systems (right), for the RTK-GNSS and the smartphone. The X-axes are in red, the Y-axes in green, and the Z-axes in blue.	59
4.5	Calibration using the MoCap, mapping the coordinate systems of the two devices: Red = latitude, green = altitude, blue = longitude.	60
4.6	Yaw curves sampled at 5Hz, for the smartphone and the Stonex RTK-GNSS receiver, before and after adjustment. .	61
4.7	Distribution histogram of the packets jumped to achieve an optimal alignment on the curves at 5Hz shown in Fig.4.6 .	62

4.8	The RTK-GNSS device is positioned in the environment to acquire the point location (left); the red hologram is virtually positioned, without spatial calibration (center) and with spatial calibration (right).	65
4.9	Error measurement process: the QR code top-left finder pattern is placed on the acquired spot, and the red holograms are placed using the acquired GCS pose.	67
4.10	Graphical representation showing how the error along the azimuth direction is computed, in particular in this case we present two possible cases of the <i>Asset2D</i> position. In the first case the $pde_1(1)$ (number of pixel) will have a negative value, for the second case indeed, the value of $pde_1(2)$ will be positive.	68
4.11	Acquisition process: images are collected through the smartphone coupled with the RTK-GNSS receiver, and positioned in the environment, according to their camera pose.	69
4.12	3D digitalization and overlay of the virtual georeferenced information (left) with real-world content (right).	70
4.13	(left) A scenario is shown, where the real scene is over-imposed with the virtual (pre-acquired) counterpart. Here, the specific case study is the excavation. The zoom of the red square area is also shown (right). The spatial accuracy of the over-imposition is very good.	72
4.14	In this example, the hologram of the excavation is overlaid on the scene (left) after the excavation has been covered (right).	73
4.15	An example of interaction using gesture with Hololens . . .	76
4.16	The helmet as a support for the combination of the Stonex and the Hololens	77

4.17	The visualization of the holograms of the underground net from the Hololens	78
5.1	Overview of the proposed methodology	81
5.2	System acquisition composed by Stonex and Smartphone for obtaining high accuracy georeferenced images	82
5.3	Comparison of the mesh obtained with the proposed methodologies on the playgrounds dataset. Other scenes can be found at: https://zenos4mbu.github.io/photogrammetry_nerf.github.io/	84
5.4	Comparison of the cloud-to-cloud distance of the proposed methodologies on the playground dataset.	85
5.5	Comparison of the cloud-to-cloud distance of the proposed methodologies on the excavation sites dataset. Other scenes can be found at: https://zenos4mbu.github.io/photogrammetry_nerf.github.io/	86
6.1	The preview of the application Trainreal on the Xreal store	92
6.2	Workflow of the application.	97
6.3	The hardware utilized consists of Nreal Light smart glasses, which are paired with a smartphone. With this setup, the user can visually observe the trainer during the exercise (b) and receive real-time feedback (c). This feedback enables the user to make immediate adjustments to their movement, ensuring proper form. The analysis of movement includes tracking the user's head movement (d) as a key factor. . .	98
6.4	Starting interface when wearing the smart glasses.	98
6.5	Interface feedback of the back squat	103

Chapter 1

Introduction

1.1 The Context

1.1.1 The PhD in Industrial Innovation

This thesis represents the final achievement of my career in the industrial doctoral course 'Doctorate Program in Industrial Innovation', cofounded by the University of Trento and the Fondazione Bruno Kessler(FBK). As the program's official headline states, 'Companies are the key players' in this interdisciplinary course that maintains a close connection with the corporate world. The author holds an "executive PhD" position, meaning that their activities are primarily conducted within their employing company. Unlike traditional doctoral studies, this program focuses on achieving high-impact applications in the industrial sector. Consequently, the research work has been paired with intensive development efforts, leading to the effective implementation of a solution for the industry sector. As a result of this research, the company is now selling an innovative product on the market, and some clients have already adopted it in their business processes.

1.1.2 The Research Topic

Arcoda [7], the company I work for, operates in the field of cartography. It is doing a study on mixed reality (MR) and 3D digitalization as an evolution of its current product offerings. Specifically, the focus is on developing a system for 3D georeferenced reconstructions and location-based augmented reality applications, which fall under the field of MR.

MR represents the seamless integration of real and virtual environments, blurring the lines between the physical and digital worlds. It encompasses a broad spectrum of experiences, from augmented reality (AR), where digital elements enhance the real world, to fully immersive virtual reality (VR). Within this spectrum, marker-based and location-based AR stand out as pivotal technologies. Marker-based AR relies on specific visual markers to trigger the display of digital content, providing a precise and interactive experience. In contrast, location-based AR uses GPS and other spatial data to overlay information on the real world, creating context-aware experiences tied to specific locations. For better understanding, the space between the real and virtual world is called Reality–virtuality continuum postulated by Milgram [8], you can see the Fig. 1.1



Figure 1.1: Reality–virtuality continuum, image taken from [1]

As the demand for more immersive and accessible MR experiences grows, smart glasses have emerged as a key hardware solution. These devices allow users to interact with digital content in a hands-free manner, making AR

experiences more intuitive and seamless. The integration of smart glasses into MR ecosystems opens new possibilities for interaction and information visualization.

The representation of 3D scenes is another critical area of development, with various algorithms pushing the boundaries of realism and accuracy. Traditional methods, like photogrammetry, reconstruct 3D models from photographs, creating detailed and accurate representations of the physical world. More recently, neural radiance field (NeRF) [9] have introduced a new way of representing scenes using neural networks, capturing complex light interactions, and providing highly realistic images. The evolution of these techniques has led to advancements such as 3D Gaussian Splatting(3D-GS) [10], which further refines scene representation by improving the efficiency and quality of 3D representation by exploiting the adoptions of Gaussians as the primitive element.

1.2 The Problem

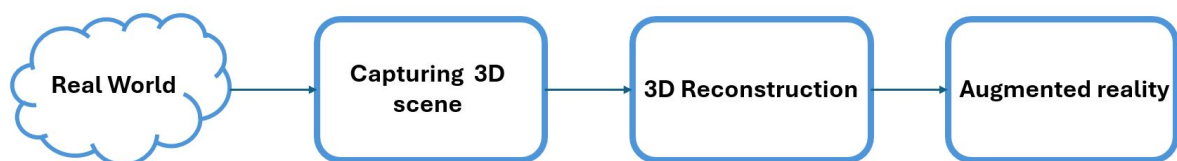


Figure 1.2: This is the final goal of our system, the acquisition of the real world, the reconstruction, and finally the visualization in AR

The goal of Arcoda in the domain of 3D digitalization is to look for a device solution that is able to serve both AR and data acquisition for generating the 3D reconstruction of the scene. In the first scenario, the

clients have selected the excavation.

An important requirement for this solution is achieving an accuracy of 10 centimeters in georeferencing the model.

There are hardware options like 3D laser scanners and drones that can perform 3D reconstruction very precisely, but they are not suitable for AR purposes. Moreover, they are quite costly, and the use of drones is often prohibited in urban areas.

One possible solution is to combine two devices: a smartphone or smart glasses and an Real-Time Kinematic Global Navigation Satellite System (RTK-GNSS) device. The RTK-GNSS can achieve centimeter-level accuracy, while the smartphone or smart glasses can collect images for 3D reconstruction and display holograms in the real world.

The problems that I will go through during this thesis are:

- **Comparison between smart glasses and smartphones:** When considering the comparison between smart glasses and smartphones, it may initially seem that smart glasses offer a more ideal solution, especially when analyzing their potential to enhance productivity in various industries. However, a deeper, step-by-step evaluation is essential to determine whether the hardware is truly ready for industrialization, particularly in the context of fieldwork by utility operators. Smart glasses, such as Microsoft's HoloLens, represent a significant advancement over traditional mobile devices by providing a hands-free, immersive experience that integrates AR into real-world tasks. The key advantage of smart glasses lies in their ability to offer critical information directly in the user's field of view. This means utility operators, for example, can receive real-time data, instructions, and visual overlays, all while keeping their hands free to perform the necessary manual tasks. This is a stark contrast to smartphones, where users must divert attention away from the task at hand to interact

with a screen, potentially leading to safety risks or inefficiencies in high-stakes environments.

One of the most promising aspects of smart glasses is their ability to leverage gesture controls, voice commands, and eye-tracking to operate, reducing the need for physical touch. In industries like utilities, where operators may be working in challenging environments such as construction sites, underground facilities, or electrical substations, this hands-free functionality is critical. Gesture controls allow users to interact with the device in a natural, intuitive way, minimizing the need for additional tools or interfaces. For example, simple hand gestures can be used to navigate menus, zoom into detailed schematics, or control devices remotely, all without needing to take the gloves off or stop what they are doing.

Moreover, with information directly overlaid in the field of view, operators can access a variety of real-time data such as schematics, maintenance records, diagnostics, or troubleshooting information. This immediate access to information can significantly reduce downtime, prevent errors, and increase efficiency by ensuring that workers have the information they need right when they need it. For instance, while performing a complex repair, a utility worker can see step-by-step instructions, schematics of the system they are working on, or even a remote expert's guidance without ever having to look away from their task.

However, it is crucial to assess whether the hardware powering these devices is ready for the demands of industrial applications. The current generation of smart glasses, while promising, still faces challenges in areas such as battery life, durability, processing power, and field visibility and weight. For utility operators, who may be working in extreme conditions (rain, snow, or in low-light environments), these

devices need to be rugged enough to handle the elements while remaining lightweight and comfortable for prolonged wear.

The durability of smart glasses, as well as the quality and brightness of the display, are key factors to consider. Operators will need a device that offers clear, readable information in diverse lighting conditions and environments. Additionally, battery life remains a concern; the hardware must be able to support a full workday of use without requiring frequent recharging, especially in scenarios where downtime is costly or dangerous.

- **Analysis of the algorithms to combine the outputs of the two devices:** The integration of RTK-GNSS and smartphones for precise localization and image capture presents an exciting opportunity for creating highly accurate 3D representations. By combining the precise positioning data from the RTK-GNSS with the smartphone's camera for capturing images, users can generate high-resolution, georeferenced 3D models of objects or environments. However, as promising as this integration is, it introduces significant challenges in terms of spatial and temporal calibration that must be addressed to successfully combine the outputs of the two devices.

One of the primary challenges in integrating the RTK-GNSS and smartphone is the temporal synchronization of the data streams. RTK-GNSS systems provide highly accurate positional data that is typically output at a specific frequency, often in the range of 1 Hz to 10 Hz, depending on the system and settings. Meanwhile, the smartphone's camera operates at a much higher frame rate, usually ranging from 30 Hz to 60 Hz or even higher in some cases. This disparity in data output rates creates a problem for aligning the two streams in time. To combine the RTK-GNSS data with the images captured by the smartphone's camera, it is crucial to synchronize the timestamps of

both data streams. Any mismatch in timing between the GNSS position data and the camera images can lead to inaccuracies in the resulting 3D models. If, for example, the positional data does not correspond to the exact moment an image is taken, the final model may contain errors in spatial alignment, leading to misrepresentations of the real-world environment.

In addition to temporal synchronization, spatial calibration is another significant challenge when combining the outputs from the two devices. RTK-GNSS systems rely on satellite-based positioning, providing a precise 3D geospatial location in a world coordinate system, usually referenced to a global standard such as World Geodetic System 1984 (WGS84) or a local coordinate frame. In contrast, the smartphone's camera captures 2D images from its own local coordinate system, with pixel coordinates that must be transformed into a 3D spatial framework.

Since the RTK-GNSS and the smartphone operate using different coordinate systems, it is essential to align their spatial references before their data can be effectively combined. This requires a calibration process to establish the relationship between the RTK-GNSS system's coordinates and the smartphone's camera frame. The primary issue here is that the RTK-GNSS system gives a precise 3D position in world coordinates, while the smartphone's camera gives a 2D image, often accompanied by metadata such as its intrinsic parameters (focal length, principal point, etc.) and extrinsic parameters (position and orientation). To combine these two data sources, the camera's position and orientation must be estimated relative to the RTK-GNSS system's global reference.

- **Analysis of the algorithms for 3D reconstruction:** Photogrammetry is a crucial technique for 3D reconstruction of real-world en-

vironments using 2D images captured by sensors such as cameras or drones. When these images are georeferenced with RTK-GNSS data, it becomes possible to create highly accurate 3D models, both in terms of geometry and spatial positioning. RTK-GNSS data provides extremely precise geographical information, allowing each point in the 3D model to be linked to an exact location within a global or local coordinate system. This ensures a high level of accuracy for applications in mapping, design, and environmental monitoring.

However, to improve the visual quality and efficiency of 3D reconstruction from georeferenced images, advanced algorithms based on NeRF and 3D-GS paradigm are emerging.

NeRF is a deep learning-based algorithm that represents a 3D scene volumetrically rather than using traditional polygonal meshes. Using a set of georeferenced 2D images, NeRF reconstructs the 3D scene by modeling how light interacts with objects, allowing for photorealistic images to be generated from any viewpoint. The strength of NeRF lies in its ability to capture fine details, including complex lighting effects, textures, and subtle gradients, making it especially useful for applications like virtual reality, urban modeling, and realistic scene simulation.

On the other hand, 3D-GS is a technique that uses point clouds represented by "Gaussian splats," which describe the spatial distribution of each point. Unlike traditional meshes, which try to build a solid surface, 3D-GS creates a smoother and more continuous representation of the scene by using a series of points with Gaussian distributions to represent the geometry and visual features of objects. This method is advantageous in contexts where fast and precise reconstruction is needed, such as 3D scanning projects or terrain analysis.

The integration of RTK-GNSS data with these advanced techniques

ensures that the resulting 3D models are not only visually detailed and realistic but also spatially accurate, with precise positioning in the real world.

1.3 Innovative aspects

The research project has the final goal of generating a prototype that can support the field inspection activities of technicians, enabling them to visualize in AR the underground services previously photographed and cataloged during construction or maintenance (with open excavation). The processing of a three-dimensional model of the work area from 2D photographic records allows for:

Visualization of the three-dimensional data in the back office, to perform subsequent measurements or conduct virtual site inspections remotely. The visualization on-site of the underground service using holograms superimposed on the real scene, thanks to the potential offered by AR technologies. The use of the module derived from this research project will allow utility companies to conduct as-built surveys entirely digitally, abandoning the traditional approach based on manual data collection and paper surveys, a process generally costly in terms of resource use. The digitization of the process will also help minimize, and potentially eliminate, the errors that occur with manual surveys, which can be significant when the graphic complexity of the survey to be reproduced increases. A manual error results in the generation of incorrect reports, consequently compromising the work carried out by technical offices.

1.4 Structure of the Thesis

In Chapter 2, I describe the state of the art (SOTA) regarding hardware for MR purposes. I then analyze the SOTA for performing spatial calibration, which is essential for creating a system that captures 3D scenes and enables AR with multiple devices. Finally, I review the current algorithms used for 3D representation of scenes.

In Chapter 3, I delve deeper into the problems to be solved explaining in more detail what are the challenges. Chapter 4 presents a proposed solution for capturing 3D scenes and location-based AR applications using a smartphone and an RTK device. I analyze the spatial calibration and temporal synchronization between the devices to achieve centimeter-level accuracy, finally I introduce the realm of extended reality(XR), focusing specifically on MR and its associated hardware. In Chapter 5, I discuss the final reconstruction of the scene based on the data acquired with the system presented in Chapter 4.

In Chapter 6, I introduce the project meta-trainer where I present the advantages of MR devices and showcase a real solution developed for the Xreal smart glasses [11], namely, the application called Trainreal, which is now available on the Xreal store lab [12].

Finally, in Chapter 7, I summarize the conclusions of my work during the Ph.D., analyzing future perspectives.

Chapter 2

State of the Art

In this chapter, I analyze the state of the art both hardware and software

- **Hardware analysis:** Analysis of the hardware used for 3D reconstruction and for MR purposes
- **Capturing 3D scene:** Analysis of the algorithms adopted for combining more devices for AR purposes and for capturing 3D scenes with high accuracy.
- **3D reconstruction algorithms:** In this part, I analyze the algorithms that are at the state of the art in 3D representation

2.1 Hardware analysis

2.1.1 Hardware for 3D Reconstruction

Creating 3D reconstructions requires a range of hardware components that capture spatial data with high precision and details. For selecting the right hardware the considerations of our use cases are crucial. A first analysis of the possible hardware for 3D reconstruction:

- 3D scanners like the BLK360 [13] and RTC360 [14] from Leica offer excellent accuracy, with precision in the range of millimeters. However,

they present significant challenges in terms of cost and practicality for field operations. These devices are expensive, making them a substantial investment. Moreover, their use in the field is not very practical. Operators need to move the scanners to various points to capture the entire volume, which can be cumbersome and inefficient. This limitation becomes particularly pronounced in excavation scenarios, where frequent repositioning is required, reducing overall practicality and efficiency.



Figure 2.1: Scanner are used in the field, Image taken by [2]

- Commercial hardware like for example, the Apple device iPhone equipped with LiDAR (Light Detection and Ranging) offers an accessible option for 3D reconstruction as explained in this article [3], combining affordability with acceptable precision for many tasks, achieving accuracy within centimeters. This makes it a cost-effective alternative for those needing reliable 3D scanning without the high investment of

specialized devices. The output provided by the iPhone includes both a depth image and an RGB image, facilitating detailed and textured 3D models. However, it remains a device that not everyone possesses, limiting its widespread application. Despite its potential, the reliance on owning an iPhone means it cannot be universally adopted across all operators and organizations, especially in the Utility sectors which are the most common clients of Arcoda.

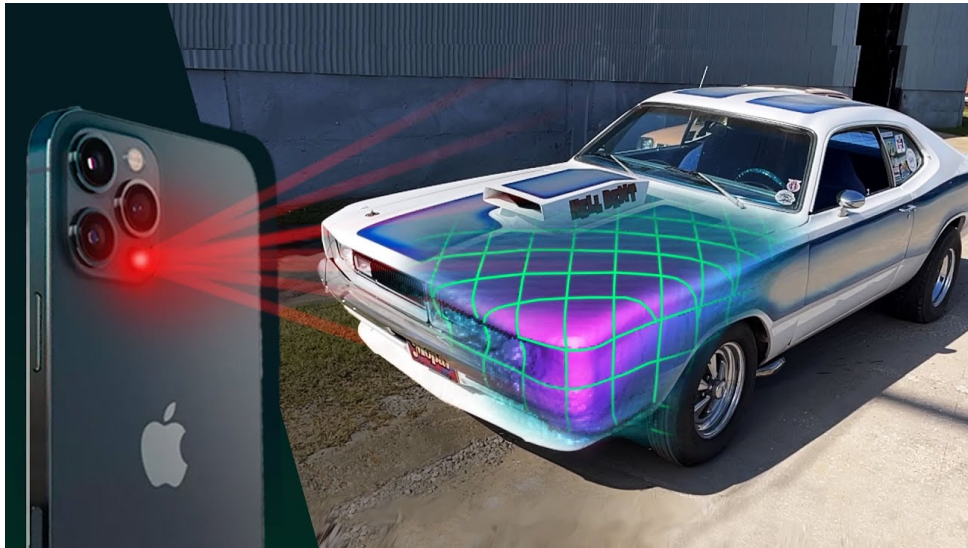


Figure 2.2: An iPhone with LiDAR, with the representation of the depth map, image taken by [3]

- A simple smartphone represents the best-case scenario in terms of cost, as most companies already provide one to their operators. It captures a stream of RGB images with all the sensor's information, but it does not have a depth camera in most cases. However, the challenge lies in the need for robust software to ensure reliable 3D reconstruction. This software must effectively utilize the smartphone's internal sensors to extract scale information and other necessary data.

While this approach leverages existing hardware, the reliance on software presents a hurdle that must be addressed to achieve accurate and practical 3D reconstructions.

- Drones as proposed in [15] used for 3D reconstructions typically capture data from altitudes ranging between 50 and 120 meters an example 2.3, acquiring high-resolution RGB imagery. They are designed for covering extensive areas, making them ideal for large-scale surveys rather than for achieving high-ground resolution. Consequently, while drones can efficiently collect data over broad regions, the level of detail might not be sufficient for applications requiring fine ground resolution. Additionally, drones cannot be used in densely populated urban areas due to regulatory restrictions, limiting their application to more open and less regulated environments.

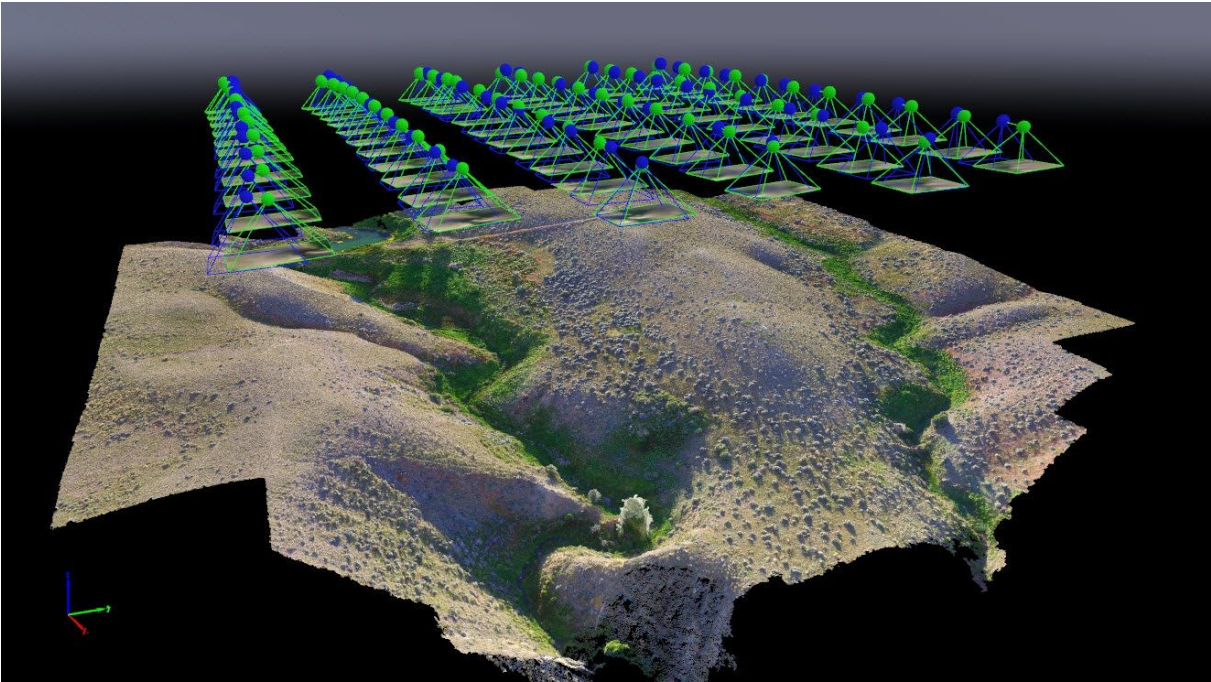


Figure 2.3: An example of images drone acquisition, for 3D reconstruction, image taken by [4]

2.1.2 Hardware for Mixed Reality purposes

Lastly, I analyze the hardware currently available for rendering the reconstructed model in the real world. The use of smart glasses and smartphones for outdoor AR applications offers distinct advantages and challenges. Smart glasses, particularly those with optical see-through technology, provide a more immersive and hands-free AR experience. These glasses overlay digital information directly onto the user's field of view, allowing for seamless integration of virtual elements with the real world. This can be particularly beneficial for navigation and situational awareness, as users can receive real-time directions or information without needing to be distracted by external stimuli. However, smart glasses often come with limitations such as short battery life, higher cost, and potential issues with the field of view compared to smartphones. On the other hand, smartphones offer a more versatile and cost-effective solution. They have larger screens, more powerful processing capabilities, and longer battery life, but require users to hold or interact with the device, which can be cumbersome and less immersive.

Now, I provide an in-depth analysis of the most valuable smart glasses for the use cases that I have analyzed. I am focusing on the analysis of optical see-through devices because they are safer, and they guarantee that users do not lose touch with reality through their semi-transparent screen where holograms are projected. Starting from Xreal [11] represents a new generation of smart glasses that emphasize lightweight design and external computational power through integration with compatible smartphones. This approach is the same for the Original Design Manufacturer (ODM) of Snapdragon [16], which similarly uses lightweight smart glasses connected to mobile devices to leverage advanced computing power without burdening



Figure 2.4: Smart glasses VS Smartphone for Location-based Augmented reality [5]

the user. A distinctive feature of Xreal is its enhanced robustness for outdoor scenarios, thanks to darker lenses that improve visibility by reducing glare in bright environments. Another big player in the smart glasses world is Microsoft's HoloLens 2 [17], which takes a completely different approach compared to the previous smart glasses. It is a standalone device with all the necessary hardware integrated, providing a complete AR experience but with the trade-off of being bulkier and heavier. While HoloLens 2 is suitable for industrial indoor applications, Xreal and Snapdragon Space are ideal for users who prefer more portable and versatile solutions, allowing them to utilize their smartphone's power for AR applications.

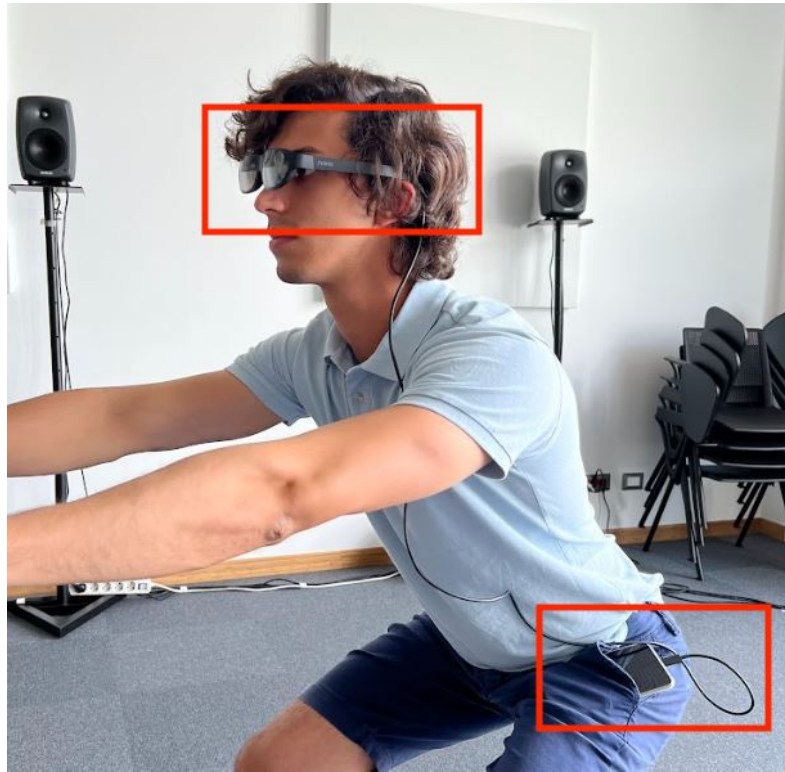


Figure 2.5: Xreal smart glasses setup



Figure 2.6: Test on the HoloLens gesture recognition capability

The smart glasses offer a new way to interact with the digital information, particularly through gestures. This method of interaction is more natural than the usual way of interacting with smartphones. The device needs to understand the precise pose of the hands, in the 3D world. The algorithm of the Hololens 2 has shown more precision compared to the Xreal (which is the new name the old one is Nreal) and Qualcomm's Snapdragon Spaces on this task because it has more cameras including one depth camera which represents a big advantage in the pose estimation process. Despite that, the other two devices are closing the gaps by improving their algorithms, exploiting the depth extracted from the stereo vision, or using AI.

2.2 Capturing the 3D scene

Capturing the 3D scene involves collecting information about the environment from different perspectives usually appearance and spatial data. This can be achieved by combining different hardware spatially and temporally, for example, an RTK-GNSS device and a smartphone, the first device offers an accurate positioning around sub-centimeters, by utilizing satellite signals corrected through reference stations, this ensures accurate positioning with respect to the geographic coordinates system (GCS). The smartphone instead is the device responsible for the acquisition of the images, but it can also catch spatial data. Usually, the spatial data are collected by exploiting algorithms like Vi-SLAM, which combines Inertial Measurement Unit (IMU) sensors and images to track the smartphone movement in 6-DoF (position and rotation). Combining these two devices in a rigid body the final goal is to estimate the pose or the position and the rotation of the smartphone relative to the RTK-GNSS system. This information allows us to estimate the pose of the images collected by the smartphone

with respect to the RTK-GNSS system reference, this data helps the final reconstruction algorithm to achieve better accuracy in the final 3D representation and also locates the representation relative to the GCS. For this purpose, I analyze the SOTA regarding this list of topics:

- Spatial calibration procedures between smartphone or smart glasses and the RTK-GNSS receiver;
- Improvement of relocalization for location-based AR for centimeter accuracy;
- Calibration exploiting markers and motion capture systems between the camera and GNSS receiver;
- Temporal synchronization between smartphone/smart glasses and RTK-GNSS receiver.

The combination of RTK-GNSS and Vi-SLAM [18] is primarily driven by the demand for precise integration of vision-based solutions with positioning systems, as required in fields like robotics and autonomous driving.

Vi-SLAM, which enables simultaneous localization and mapping, uses the input from IMU and from the camera to capture visual information and then processes this data to build a 3D map of the environment. The system tracks key features within the images as the camera moves, identifying changes in perspective to estimate the device's movement (localization) and reconstructing a spatial model of the environment (mapping). When the algorithm starts working on the device, it can track its movement concerning the device's initial position in 6-DoF. This algorithm can exploit different sensor data to achieve its goal like RGB camera, depth camera, and IMU.

When the Vi-SLAM is integrated with RTK-GNSS, it can directly interface with the real-world GCS. It is important to recognize that the com-

combination of these two distinct technologies can introduce errors. Therefore, it is crucial to rigorously assess the accuracy of this combined system, especially concerning the orientation relative to the gravitational axis. This accuracy is crucial, particularly in the context of location-based AR applications, because it can introduce a misalignment of holographic data in the real world.

2.2.1 Spatial Calibration analysis

To address the spatial calibration procedure, various solutions have been proposed in the literature; the authors in [19] and [20] introduce an AR application which, coupled together with a tablet enclosure withstanding all the necessary sensors, allows to visualize the utility lines underground with a very high degree of accuracy. Their goal is to couple the GNSS receiver and a smart device.

Another solution is proposed in [21], where Patias et al. state that, to achieve a seamless AR experience, they require a precise pose estimation. Here, all the parameters of the optical system are needed, including the GCS position and the values for the IMU (pitch, yaw, roll). Putting together the position from the GNSS device and the orientation from the IMU allows to represent a 6-DoF pose with respect to the sensors. The authors still rely on hand-eye calibration to obtain the pose of the camera system, leading to high precision in rotation (around 1 degree), but quite poor in translation (below 20cm).

In [22], Mercier et al. discuss the problem of location accuracy in the fruition of AR and compare experiences based on the smartphone GCS data with the ones based on RTK-GNSS receiver data. For a precise orientation, they record seven positions from the GNSS system, and their displacement is compared with the local AR displacement estimated by the SLAM algorithm. Then, they use the Helmert transformation to de-

rive a rotational matrix and a 3D translation vector. In this way, they do not require any magnetometer or IMU, and they can correlate the AR positions with the GCS. Unfortunately, this is a very recent work and the results are only preliminary. For example, no detail is provided in terms of the calibration between the RTK-GNSS receiver and smartphone. The most relevant works dealing with calibration are the ones proposed in [23]. Here, the author uses a MoCap with retro-reflective markers to obtain a reliable transformation matrix between a robot and a camera, reaching centimeter-level accuracy. In these two works they exploit visual markers like ChArUco [24] targets, on top of which the retro-reflective markers are attached. Below I provide a more in-depth explanation of each paper cited previously.

2.2.2 Major Works in the Field

”LARA: a location-based and augmented reality assistive system for underground utilities’ networks through GNSS”

In the paper by E. Stylianidis et al. in [19], the authors present the LARA project, co-funded by the European Commission, which addresses the growing need for accurate positioning systems in the utilities and construction sectors. These industries require precise tools to facilitate work and minimize damage to existing infrastructure during maintenance or expansion projects. The LARA project aims to develop both hardware and software solutions to meet these needs, using a tablet paired with an external RTK-GNSS receiver and an IMU [19].

The core of the LARA system is its ability to handle geospatial data for underground utilities, such as water, gas, electricity, and sewerage, stored in a GIS geo-database. The GNSS and IMU components are used to precisely determine the user’s position and orientation. The AR component renders 3D models of the underground utilities, allowing users to visualize

the infrastructure beneath the ground.

The project emphasizes the importance of integrating GIS capabilities, which are essential for managing and visualizing geospatial data. These tools are also cost-effective, helping companies save 10 to 30 percent on operational expenses. The focus is on developing a mobile GIS solution that enables on-site data capture, storage, manipulation, and analysis.

For pose estimation, the LARA system utilizes a high-end RTK-GNSS receiver and a high-precision IMU, which together provide accurate positioning in 6-DoF: latitude, longitude, altitude, pitch, yaw, and roll. This setup can achieve up to 20 cm accuracy in translation and 1 degree in rotation. If the pose accuracy is insufficient, a photogrammetry module allows users to manually define control points to refine the positioning.

Once the user's pose is accurately determined, the AR engine uses this information to render holograms of the underground utilities in the correct position and orientation within the scene. This step also involves the use of a calibration matrix to align the camera with the GNSS receiver.

Overall, the LARA system significantly enhances on-site work by providing a clear, intuitive visualization of underground utilities through AR, improving both safety and efficiency. However, the paper lacks detailed technical information, particularly regarding the calibration process between the camera and the GNSS receiver used in the AR engine.

"Augmented Reality Geo-visualization for Underground Utilities"

In their updated paper, E. Stylianidis et al. build on their previous work by focusing more on the AR aspect of the LARA project [20]. They start by comparing popular AR Software Development Kits (SDKs) such as Wikitude, Vuforia, Layar, ARKit, and ARCore. The paper details how camera pose estimation is improved using hand-eye calibration, which accurately transfers pose data from the RTK-GNSS receiver and IMU to the camera's

coordinate system, achieving up to 1 cm accuracy in translation and 1 degree in rotation. This refined pose information is then utilized in their AR engine, which is based on Unity. The system integrates the camera pose with mobile GIS data, rendering 3D models and overlaying them onto the camera feed to provide an augmented view.

The paper also addresses some limitations of the current LARA system. Despite improvements, the translation error remains under 10 cm, which, while better than smartphone GNSS devices, is still higher than the potential accuracy of the RTK-GNSS receiver. Additionally, the system currently operates at a frame rate of just 1 frame per second, which limits the AR experience. The authors aim to enhance this to 25 Hz for a smoother AR experience but note that achieving this depends on the effective fusion of IMU and GNSS data—a challenge they acknowledge but do not fully elaborate on.

”Robust pose estimation through Visual-GNSS mixing”

In this paper P. Patias et al., dive deeper into the pose estimation and the calibration procedure adopted for Project LARA [21].

They state in more detail that, given a GNSS and an IMU sensor, it is possible to compute the camera pose through a calibration procedure. In this specific case, they define the ”lever arm offset” as the spatial vector difference between the center of the RTK-GNSS receiver’s phase and the camera system’s origin. In the same way, they define the ”bore-sight calibration” as the rotation misalignment between the IMU and the camera frame. According to the authors, the former calibration component can be computed using land surveying techniques or photogrammetry approaches, while the latter can be calculated by aerial triangulation or photogrammetry approaches.

Later the authors discuss how the calibration procedure for the project

LARA was performed using a photogrammetry approach under laboratory conditions for the lever arm offset. Unfortunately, they do not give any information on how the calibration component for the bore-sight misalignment was computed.

”BiodivAR: A Cartographic Authoring Tool for the Visualization of Geolocated Media in Augmented Reality”

In their study, J. Mercier et al. explore how using only GNSS affects the user experience in location-based AR applications compared to RTK-GNSS [22]. They confirm a key point of this thesis: the accuracy of smartphone internal GNSS receivers is often inadequate for effective location-based AR. While internal GNSS can reach about 1 meter accuracy in ideal conditions, it can easily degrade to over 10 meters, making AR applications impractical. This inaccuracy is a significant factor in the usability issues commonly associated with AR technology.

To address this, the authors suggest using a low-cost external RTK-GNSS receiver for improved accuracy. They avoid using an external IMU due to the potential inaccuracies in yaw angle from small-scale sensors. Instead, they continue using the internal IMU for pitch and roll, which are reliable due to their dependence on gravity. For yaw, they use a novel method combining SLAM data with RTK-GNSS geolocation data. By logging displacement and absolute geolocation data, and using geometric similarity transformations on seven data points, they can accurately determine orientation and position without needing magnetometers. However, the paper does not address calibration issues, either spatial or temporal, and relies on the average accuracy of their uncalibrated solution.

”A High-Precision Localization Device for Outdoor Augmented Reality”

In this paper, M. Stranner et al., introduce the sensor-cube, a system developed appositely to empower location-based AR applications with a high-end sensor unit built for an affordable price [25]. This unit poses itself as a solution to the inaccurate consumer-grade sensors present in smartphones, which, according to the authors are insufficient to handle most of the location-based AR experiences. Indeed, this sensor allows it to reach centimeter-level accuracy thanks to the combination of sensors such as a differential GNSS receiver, an altimeter, an inertial measurements unit, and a WiFi radio module for communication. This device is small, it weighs only 40 grams and it can be easily attached to the device used to run the AR experience.

The tests carried out by the authors on the field show very good results for their proposed solution, with an accuracy that is only 2 cm distant from the one of a differential GNSS receiver. Unfortunately, this accuracy refers to the quality of the position measured by their sensor cube, but no calibration was performed between the sensor cube and the smartphone. Indeed, the authors state that performing a calibration between the two devices would be beneficial and that this will be one of their future works.

”Extrinsic Calibration of Camera and Motion Capture Systems”

In their paper, P. Ganesh et al. introduce a new calibration method that uses MoCap systems, MoCap markers, and a ChArUco board to estimate the positional and rotational offsets between a camera’s optical frame and a nearby reference frame defined by a MoCap marker [23]. ChArUco refers to a checkerboard pattern that combines both ArUco markers and a traditional checkerboard, used for extrinsic camera calibration and pose estimation. Their method involves calculating four key poses to determine the

transformation matrix between the camera's optical frame and the camera marker frame. The four poses are the pose of the camera optical frame w.r.t. a fiducial marker ChArUco board placed in front of it, the pose of the fiducial marker ChArUco board w.r.t. some MoCap markers installed over it, the pose of the MoCap markers on the fiducial marker ChArUco board w.r.t. the MoCap global reference system and the pose of the camera marker frame w.r.t. the MoCap global reference system.

- Camera to fiducial marker(ChArUco): First, they determine the camera's pose relative to the ChArUco board placed in front of it using OpenCV to process the camera's video stream. This gives a transformation matrix, representing the camera's pose in the ChArUco board's reference frame.
- ChArUco to MoCap markers: Next, they manually measure the transformation matrix between the ChArUco board and the MoCap markers attached to it. This matrix updates the camera's pose from the fiducial marker's reference frame to the MoCap markers' frame.
- MoCap markers to MoCap system: The transformation matrix between the MoCap markers and the global MoCap system is computed directly using the MoCap system, which then updates the camera's pose to the global MoCap reference frame.
- Camera marker frame to MoCap system: Finally, they compute the transformation matrix for the camera marker frame relative to the MoCap system.

The final calibration transforms are calculated by combining the transformation matrix obtained in each step, which allows them to convert the pose from the camera marker frame to the camera's optical frame. To ensure accuracy, the final calibration is averaged over more than 1000 mea-

surements, removing outliers to reduce noise. This averaged calibration matrix is then used to refine the pose estimations and calculate the reprojection error, evaluating the calibration’s accuracy. The reprojection error is the Euclidean distance between the projected 3D point using the camera estimated parameters and the actual 2D point in the image captured by the camera.

Despite its effectiveness, the authors note challenges including time synchronization between MoCap and the data acquired by the OpenCV, matrix inversion approximations, noise amplification, and manual estimation in the second step. Their work informs this thesis, which aims to simplify the calibration procedure.

ChArUco board-based omnidirectional camera calibration method

Gwon et al. in [24] propose a novel omnidirectional camera calibration method utilizing a ChArUco board-based cube structure to simplify the traditionally complex calibration procedures. Their approach employs ChArUco markers, which combine ArUco markers with a chessboard pattern, to accurately determine 2D and 3D coordinates of feature points. This method eliminates the need for extensive adjustments to mirror angles or multiple calibration boards, allowing for both intrinsic and extrinsic calibrations. Intrinsic calibration is achieved by estimating the perspective projection matrix for each camera, while extrinsic calibration can be performed even if only part of the calibration structure is visible.

Experimental results demonstrate the effectiveness of their approach, yielding an average re-projection error of 0.37 pixels for intrinsic parameters. For extrinsic parameters, the method achieved a mean absolute error of 0.90° in rotation and 1.32 mm in translation, outperforming conventional methods in terms of accuracy and reliability.

2.3 3D Reconstruction algorithms

3D reconstruction is crucial in fields like construction, excavation, and worksite management. In our scenario we focus on the Utilities use cases, so the excavation one. Employing multi-view reconstruction techniques, scenes are captured from various angles using 2D images. This enables detailed monitoring of the project progress and provides the ability to virtually navigate sites, both during and after completion, utilizing georeferencing and virtual reality. I focus on a comparison of three approaches for 3D scene representation: the photogrammetry method, which offers an explicit representation of the 3D structure using point cloud and mesh; NeRF, which relies on a neural network for an implicit representation aimed at view synthesis; and the latest evolution in view synthesis, 3D-GS, which creates an explicit 3D representation using Gaussians.

2.3.1 Background

Photogrammetry enables the conversion of 2D images into comprehensive 3D models, including point clouds and textured meshes, enabling advanced spatial analyses.

However, the application of photogrammetry reconstruction encounters several challenges, particularly when dealing with objects characterized by complex optical properties such as high absorbency, reflectivity, or scattering. These methods can also suffer from variance in lighting conditions, including shadows, glare, or inconsistent illumination, as well as from surfaces with uniform or repetitive textures and complex shapes or geometries.

In this context, NeRF-based technologies that rely on neural networks emerge as cutting-edge solutions for solving some of these inherent limitations. They rely on an innovative scene representation that contains particles, each of them characterized by density and color. One of the

most promising approaches is NeRFacto, a variation of InstantNGP available in NeRFstudio.

Another alternative to the NeRF-based approach is 3D-GS which solves the same task but uses an explicit representation made by Gaussian without using a neural network.

2.3.2 Photogrammetry

Among the most common photogrammetry solutions for 3D view reconstruction, I focus on COLMAP [26] because of its open-access policy, its widespread adoption, and an active community working on it. COLMAP is an open source state-of-the-art photogrammetry software that has become a critical tool for 3D reconstruction from 2D imagery. Developed by Johannes L. Schönberger, it has gained recognition for its robust and automated structure-from-motion (SfM) and multi-view stereo (MVS) pipeline. COLMAP excels through its integration of advanced computer vision algorithms, such as the Scale-Invariant Feature Transform (SIFT) for feature detection and matching, and bundle adjustment for optimizing camera parameters and 3D structures. Its user-friendly design supports a wide range of applications, including cultural heritage documentation, robotics, and other computer vision research. Notably, COLMAP’s flexibility allows it to scale from small datasets to large, complex scenes involving thousands of images, providing high accuracy and adaptability. This makes it a benchmark tool in photogrammetry, often serving as the foundation for comparative studies and further algorithmic advancements in 3D reconstruction.

2.3.3 Neural Radiance Fields

NeRF have emerged as a significant advancement in the field of 3D scene reconstruction. The scene is represented with a novel 5D function. This

function correlates each spatial point (x, y, z) with the radiance emitted in any direction, defined by azimuthal and polar angles (θ, ϕ) . The outcome, characterized by volume density σ and RGB color values c , varies with the viewing direction. This relation is formulated through the Multi-Layer Perceptron (MLP) F_θ , expressed as:

$$F_\theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma) \quad (2.1)$$

where $\mathbf{x} = (x, y, z)$ denotes the coordinates within the scene, and $\mathbf{d}(\theta, \phi)$ represents the 3D Cartesian unit vector indicating the direction. The color $\mathbf{c} = (r, g, b)$ shifts with the viewing angle, while σ , denoting volume density, remains invariant. The usage of neural volume rendering pipelines, over traditional point clouds or meshes, enables the modeling of variations in color and illumination. Since its debut in 2020, NeRF has found application in diverse research areas, from urban mapping and robotics to human pose estimation. Its adaptability includes evaluating 3D reconstructions through metrics such as noise level, geometric accuracy, and the requisite number of images for effective reconstruction. InstantNGP [27], short for Instant Neural Graphics Primitives, is a variant that enhances NeRF’s framework to expedite scene reconstruction significantly. By refining the neural network’s architecture and computations, InstantNGP facilitates quicker achievement of high-quality results, positioning it as a viable option for real-time applications. This advancement expands NeRF’s capability in 3D reconstruction, offering increased applicability and efficiency in capturing intricate scenes.

Building upon these foundations, NeRFStudio introduces an innovative platform designed to streamline and enhance the creation and manipulation of NeRF-based models. At the heart of NeRFStudio’s methods is the NeRFacto model, which incorporates insights from state-of-the-art research, including pivotal works such as MipNeRF-360, Instant-NGP, NeRF-W, and

Ref-NeRF. NeRFacto’s focus is on optimizing camera views and sampling processes, utilizing a piece-wise sampler and a proposal network sampler for efficient and effective scene reconstruction. Additionally, it employs a novel scene contraction technique and a refined NeRF field that incorporates appearance embeddings and normal predictions, all implemented in PyTorch for ease of use and customization. This method not only accelerates the reconstruction process but also improves the fidelity and versatility of the generated scenes, marking a significant leap forward in the practical application of NeRF technologies.

NeRFStudio introduces an innovative platform, leveraging the NeRFacto model, to streamline NeRF-based model creation and manipulation. NeRFacto integrates insights from very recent research, including MipNeRF-360 [28], Instant-NGP [27], and Ref-NeRF [29], focusing on optimizing camera views and sampling processes. It employs a piece-wise sampler and proposal network sampler for efficient scene reconstruction, alongside a scene contraction technique and refined NeRF field with appearance embeddings and normal predictions. Implemented in PyTorch, this approach accelerates reconstruction, enhancing scene fidelity and versatility, representing a significant advancement in NeRF technology’s practical application.

2.3.4 3D Gaussian Splatting for Real-Time Radiance Field Rendering

3D-GS [10], a novel approach to scene representation, contrasts with neural fields by optimizing an explicit point-based scene model. Each point in this representation is associated with various attributes: a position $p \in \mathbb{R}^3$, opacity $o \in [0, 1]$, third-degree spherical harmonics (SH) coefficients

$k \in \mathbb{R}^{16}$, 3D scale $s \in \mathbb{R}^3$, and 3D rotation $R \in SO(3)$ represented by 4D quaternions $q \in \mathbb{R}^4$. Rendering to the image plane involves accumulating the color c_{GS} from correctly-sorted points using the equation:

$$c_{GS} = \sum_{j=1}^{N_p} c_j \alpha_j \tau_i \quad \text{where} \quad \tau_i = \prod_{i=1}^{j-1} (1 - \alpha_i) \quad (2.2)$$

with c_j determined by SH coefficients k and α_j calculated from the projected 2D gaussian with covariance $\Sigma' = JM\Sigma M^T J^T$, incorporating per-point opacity o , viewing transformation M , and Jacobian J of the affine approximation of the projective transformation. The 3D covariance matrix Σ ensures positive semi-definiteness through the scale matrix $S = \text{diag}(s_1, s_2, s_3)$ and rotation R , following $\Sigma = RSS^T R^T$.

Building upon the principles of 3D-GS, SuGaR [30] leverages gaussian functions to model object surfaces within a scene, achieving precision in handling occlusions and detailed surface texturing through gaussian "splats" projected onto a volume grid. Each splat influences the volume's density and color, based on its spatial location and gaussian distribution, described mathematically as:

$$G(\mathbf{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right) \quad (2.3)$$

In Eq. (2.3) $\mathbf{x}$ denotes a point in space, μ the mean location (center of the splat), and Σ the covariance matrix shaping the Gaussian distribution. SuGaR's method for accumulating multiple splats across a scene constructs a volumetric representation capturing density and color information, enabling precise shading and depth rendering.

Chapter 3

The Problem

3.1 The Solution at a Glance

To provide clarity, the first part presents briefly the final solution to better understand the problems I have encountered. The final solution that I have built consists of a smartphone combined with an RTK-GNSS device. As an RTK-GNSS device I started to build the prototype for the Stonex S850A [31], I bound the two devices creating a first structure with a 3D printer and the result is in Fig.5.2

Starting from this prototype where the devices are locked in a unique rigid body, the next step is to perform spatial and temporal calibration or synchronization. Our solution can be summarized as follows:

- Spatial calibration of the RTK-GNSS and the camera of the smartphone, conducted offline using a MoCap in Fig. 3.2, which is a device composed by a series of synchronized cameras, which are able to track some retroreflective markers with millimeter accuracy.
- Analysis and characterization of the temporal delay between the packets coming from the RTK-GNSS device and the packets generated by the smartphone;
- Computation of the geometric transformation between the camera

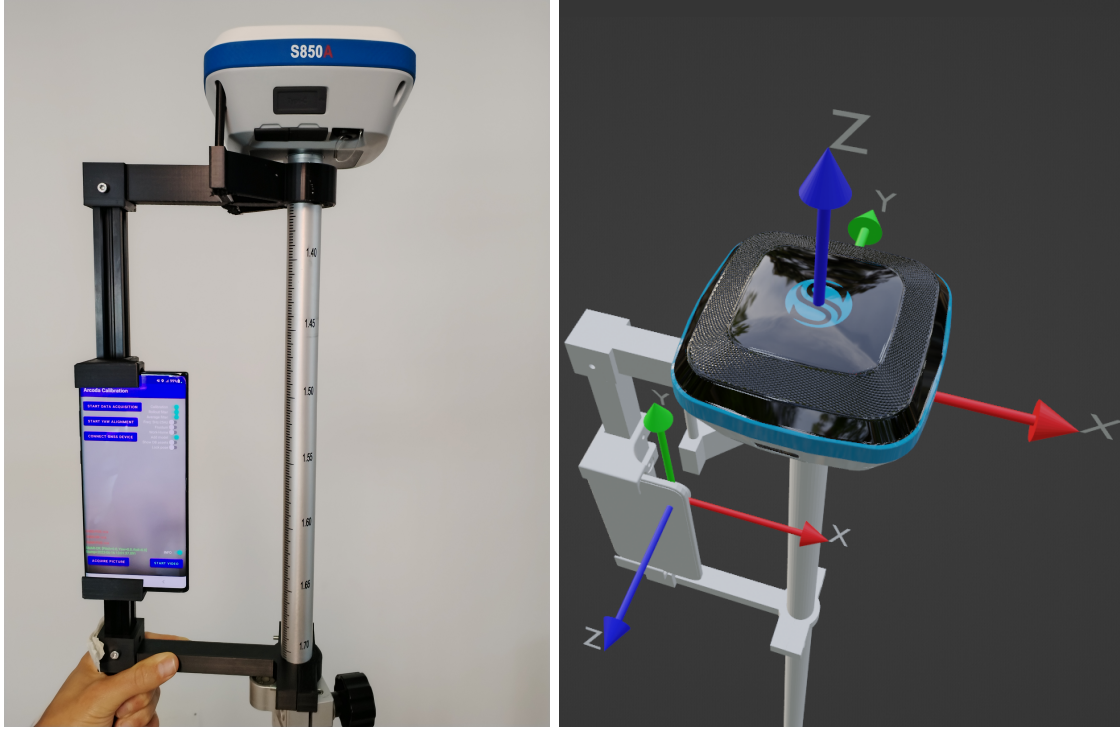


Figure 3.1: Custom bracket for the camera and RTK-GNSS device (left), with an indication of the reference coordinate systems (right), for the RTK-GNSS and the smartphone. The X-axes are in red, the Y-axes in green, and the Z-axes in blue.

of the smartphone and the GNSS(Global Navigation Satellite System) device, exploiting Visual Simultaneous Localization And Mapping (Vi-SLAM) [32] to reconstruct the relative translation when the RTK-GNSS device is covered and it is not able to provide a reliable GNSS position.

Finally, once the prototype is ready I can use the data acquired by the device (images with geographic coordinates system information) to create a 3D reconstruction.

I evaluate algorithms for 3D reconstruction, the goal is to understand which one works better for our use case and evaluate the possibility of adopting it for industrialization purposes. For this reason, I have chosen

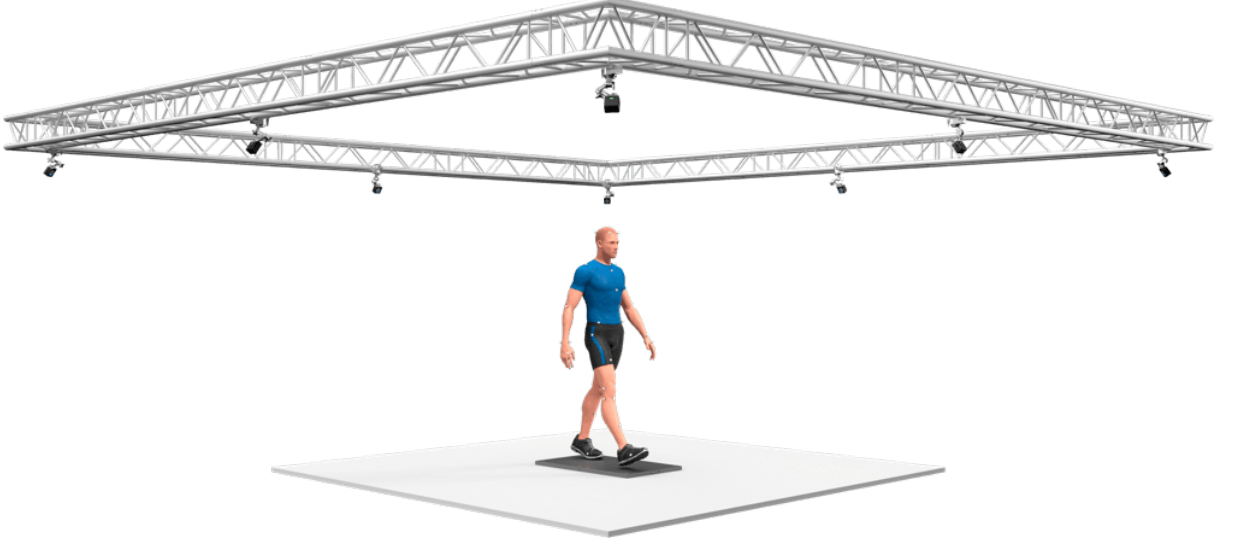


Figure 3.2: The type of MoCap that we have used in our work

3 algorithms: COLMAP [33] which is based on photogrammetry, Instant-NGP [27] which is an evolution of NeRF and finally a method-based on 3D-GS called Surface-Aligned Gaussian Splatting (SuGar) for Efficient 3D Mesh reconstruction and High-Quality Mesh Rendering) [30]. I evaluate these three methods in terms of qualitative reconstruction, hardware needs, and reconstruction time.

After a brief introduction to the solution, the remainder of this chapter presents in-depth analysis of the problem encountered in this thesis, offering a detailed examination of the issue. Below is a list of the main challenges that I have addressed:

- Identification of the hardware; given the constraints the final prototype has to be usable by an operator in the field, can get a 3D reconstruction with centimeter-level accuracy; and finally be able to display AR content over imposed on the real world.
- Combination of hardware for collecting image streams, and at the same time getting the relative pose with respect to the GCS requires constructing a structure that holds the two devices in a unique rigid

body.

- Identification of the type of spatial calibration, starting from the hypothesis that this procedure has to be performed only once.
- Estimation of the temporal delay between the two sources of information: the smartphone and the RTK-GNSS device
- Selection of the best algorithm for 3D reconstruction to adopt considering our scenarios.

3.2 Hardware Identifications

The analysis of the possible hardware for acquisition involves the verification of the usability of this hardware on the field. It has to comply with several requisites, including:

- The possibility of acquiring a streaming of images
- The capability to acquire high-accuracy GNSS data
- The capability to display digital information over-imposed on the field
- Usability everywhere especially in the outdoor scenario without security problems

3.3 All Components as a Rigid-body

The devices to be combined are the RTK-GNSS receiver, which is responsible for acquiring positional data relative to the GNSS reference system, and a smartphone, which gathers GNSS information from the RTK device while simultaneously capturing image streams.



Figure 3.3: A screenshot acquired from the final application that shows the over-imposition of the holograms in the real world

To merge these devices I need to design a structure that securely holds both devices in a fixed position. This structure has to ensure that if the devices are separated and later reassembled, they will return to their exact original positions. This consistency allows for a stable transformation matrix that relates the pose of the RTK-GNSS receiver's center of phase to the pose of the smartphone's camera. By applying the translation vector and rotation matrix defined by this transformation matrix computed during the spatial calibration procedure we can accurately transition between the poses of the two devices.

This ability to switch between device poses will be crucial in the subsequent process, enabling the placement of AR assets in the space based on RTK-GNSS receiver measurements, and their visualization from the smartphone's current camera pose.

3.4 Spatial Calibration

The spatial calibration problem involves calibrating the two devices, and aligning their respective coordinate systems so they can operate in the same physical space. In this case, the two coordinate systems are the camera, which is used for frame acquisition, and the RTK-GNSS device, which acquires the GNSS position.

The final goal is to estimate the camera's pose relative to the RTK-GNSS system reference, and ultimately relative to the GCS.

This problem is quite common in fields such as AR, robotics, or multi-camera systems. The general approach for calibrating two devices involves the following steps:

- **Define a Common Coordinate System** Select one device to be the reference and align the other device's coordinate system to this one. Another possibility could be to use an external reference system where both the devices have ground points, that could be exploited for estimating their pose inside this external coordinates system reference.
- **Collect Correspondence Points:** Identify or create known physical points in the environment that are visible or measurable by both devices. These points should have fixed and known locations in the world coordinate system. Once the points are fixed, the final step is to collect those points and record the position from both devices.
- **Compute the Transformation:** At this point, the goal is to retrieve the transformation matrix, which consists of three components: translation, rotation, and scale factor.

The transformation can be summarized with the following formula:

$$\mathbf{P}_{\text{device2}} = \mathbf{R} \cdot \mathbf{P}_{\text{device1}} + \mathbf{T}$$

where:

- $\mathbf{P}_{\text{device1}}$ is a point in the first device's coordinate system,
 - $\mathbf{P}_{\text{device2}}$ is the corresponding point in the second device's coordinate system,
 - $\mathbf{R}$ is the rotation matrix,
 - $\mathbf{T}$ is the translation vector.
- **Apply and validate the transformation:** Convert the points (or the poses in the case of 3D models) from one system to the other and validate the accuracy of the transformation

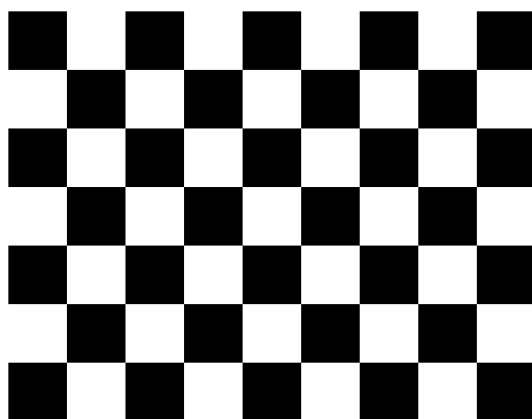


Figure 3.4: An example of visual pattern used for the extrinsic calibration of cameras

3.5 Temporal Synchronization

Since the goal is to combine the data provided by two different devices, it is very important to sync the two streams of data at the temporal level. The first challenge I need to address is estimating the delay between the two data sources. This involves characterizing the delay from a statistical viewpoint. As the next step, I must develop an algorithm that allows us to

fuse these two types of data, taking into account the statistical distribution of the delay.

3.6 3D Reconstruction of a Scene

The ability to digitize a scene in 3D has unlocked the potential not only in terms of visualization but also for interaction, taking measurements, create projects directly in the 3D digital world. As more companies delve into this world, they increasingly seek solutions that leverage these types of content. The research community is dedicating significant resources to developing new algorithms to meet this demand. Today, frameworks such as photogrammetry, which has long been known for reconstructing 3D scenes, are widely used. In recent years, new algorithms like NeRF, created for view synthesis, have emerged, followed by 3D-GS, which builds on its foundation.



Figure 3.5: A screen acquired from a 3D-GS viewers about an excavation

In our context, we have to analyze the state of the art and identify the

algorithm for 3D representation, that best suits our specific requirements and use cases. The algorithm must be applicable in contexts involving external acquisitions, such as those found in excavation sites. It must be robust, capable of generating a measurable model, and maintain an acceptable level of visual quality. Additionally, the output format should be compatible and supported in real-time on the device used for AR content management.

Chapter 4

Capturing 3D scenes

The 3D digitalization of contents and their visualization using AR has gained significant interest within the scientific community. Researchers from various fields have acknowledged the potential of these technologies, actively exploring the ability to provide users with easy access to digitized information, by seamlessly integrating content directly into their field of view. One of the most promising ways to approach the problem in outdoor scenarios consists of the so-called location-based AR, where contents are displayed to the user by integrating satellite positioning (GNSS) and inertial (IMUs) sensors. Although the number of application fields is numerous, the accuracy of the over-imposition of contents in the virtual view still hinders the widespread adoption of such technologies. In this chapter, I propose the combination of a GNSS device equipped with RTK, and a regular smartphone. To this aim, we implement a novel offline calibration process that leverages the potential of a MoCap. The proposed solution is capable of ensuring temporal consistency and allows for real-time acquisition at centimeter-level accuracy. AR enhances the perception of the world through the integration of virtual objects inside the user's field of view. It can be exploited in many diverse fields, including gaming analysis [34], construction [35], education [36], cultural heritage [37]. When dealing with

AR, two main approaches can be adopted: location and marker-based solutions. Marker-based AR [38] requires the user to point at a predetermined physical object, such as a code or an image, so that the device recognizes it and overlays the desired content. Location-based AR [39], instead, exploits a combination of the device's geographic location and IMU to locate the device (i.e., the smartphone) in the environment, and consequently display and position the virtual content correctly. Nowadays, location-based solutions suffer from several limitations; in particular, the GPS is affected by multipath errors and atmospheric conditions including ionospheric and tropospheric delays [40]. Moreover, the accuracy of the available IMUs is often sub-optimal for orientation calculations.

The primary objective of this chapter is to develop a system that can accurately estimate the orientation and position of the camera with respect to the GCS. The goal is to achieve high positioning accuracy and superimpose holograms with centimeter-level precision in outdoor scenarios. By attaining this level of accuracy, we enable a wide range of applications, including the collection of datasets consisting of images and their respective poses, accurately positioned in the GCS. There are many solutions based on Visual Position Systems (VPS) [41] to obtain global localization (as the works in [42] [43]). These solutions, however, require prior knowledge of the environment because they rely on the visual characteristics of the scene, and it is not always possible to gather this piece of information.

Therefore, while it would be convenient to use a single device, such as a smartphone, it is worth noting that the enclosed internal GPS is unfortunately not accurate enough. Smartphones use a single satellite network to provide location data and can reach an accuracy in the range of a few meters. On the other hand, RTK-GNSS positioning, uses a combination of GPS and Global Navigation Satellite systems (GNSS) such as GLONASS,

Beidou, and Galileo, together with a base station, which sends the correction of the measurement to the RTK device. This correction mitigates the imprecision of a standard GNSS device, allowing it to reach accurate positioning of the objects. An RTK system needs specialized hardware and software to work, and it is not currently available in consumer-grade smartphones. However, integrating two different hardware components, in our case a smartphone and an RTK-GNSS device, poses several challenges, in particular, related to spatial calibration and temporal consistency.

Our contributions can be summarized as follows:

- Spatial calibration of the RTK-GNSS and the camera of the smartphone, conducted offline using a MoCap system [44];
- Analysis and characterization of the temporal delay between the packets coming from the RTK-GNSS device and the packets generated by the smartphone;
- Computation of the geometric transformation between the camera of the smartphone and the GNSS device;
- Evaluation of the performances of our system in terms of orientation accuracy around the gravity axis.

The source code of the project is made available in our GitHub repository¹.

4.1 GNSS System

This section delves into the key concepts of localization using GNSS technologies, providing a foundation for understanding how the integration of

¹<https://github.com/Lorenzoarc/Spatial-Temporal-Calibration-for-Outdoor-Location-Based-Augmented-Reality>

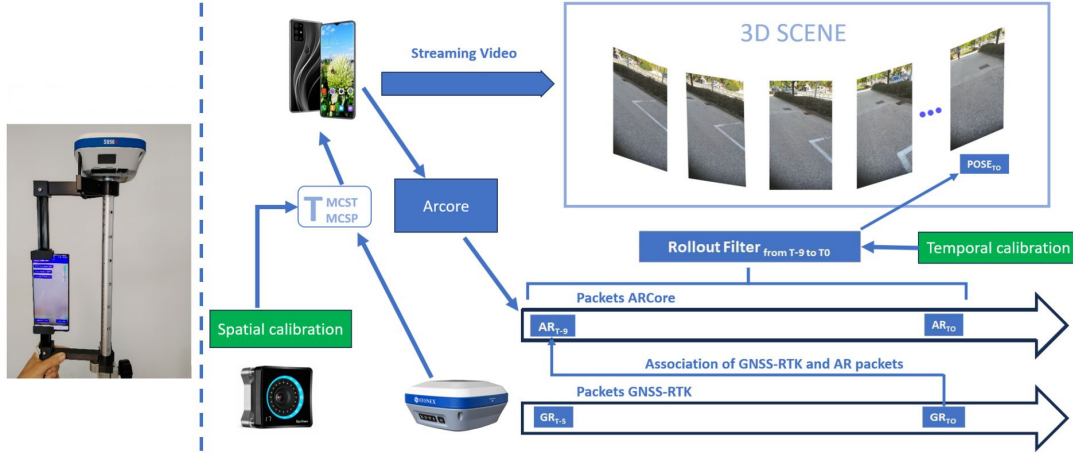


Figure 4.1: Architecture of the proposed solution: we present all the components of our system and we highlight the role of the spatial and temporal calibration in our flow. $T(MCST \rightarrow MCSP)$ represent the spatial transformation between the RTK GNSS and the smartphone; $AR, GR, POSE$ represent, respectively, the ARcore packets from the smartphone, the RTK packets, and the final pose calculated, which will be assigned to the image at each time instant.

Vi-SLAM with satellite positioning can enhance the accuracy of location-based AR technologies. By exploring these concepts, the reader will gain insight into why fusing these technologies is advantageous for improving precision in AR applications. Additionally, the same system can be utilized for the acquisition of a 3D dataset composed of images and relative poses.

4.1.1 Principles of Localization

Localization is essential across a wide range of applications in various fields. In the context of this thesis, accurate localization is a critical component for achieving stable and reliable results with the employed technology.

For the acquisition of the 3D dataset composed of images and poses having the current location of the camera is a key point. Also, the location-based AR depends on comparing the user's current location with the pre-defined locations where AR content is intended to appear. After this com-

parison, the distance between these positions can be broken down along different axes, with the resulting values used as coordinates to appropriately augment the AR content.

To determine the user's current position in the world, Global Navigation Satellite Systems (GNSS) can be utilized. These satellite-based systems offer positioning, navigation, and timing services globally. GNSS works by using a network of satellites that transmit signals to receivers on the ground, allowing them to calculate their precise location. The most well-known and widely used GNSS system is the Global Positioning System (GPS), developed and operated by the United States. [45].

The position obtained from a GNSS system is typically expressed in terms of latitude, longitude, and altitude, based on a specific GCS known as the World Geodetic System 1984 (WGS84). WGS84 is the most widely used GCS for GNSS positioning, providing a global reference framework that defines the Earth's shape and its relation to the coordinate grid. It employs a three-dimensional coordinate system, with latitude, longitude, and ellipsoidal height as its primary coordinates. While there are other GCS instances, such as the North American Datum 1983 (NAD83) and the European Terrestrial Reference System 1989 (ETRS89), they are less commonly used compared to WGS84. [46].

4.1.2 The GNSS standard

GNSS technology determines a receiver's position on Earth using a method called trilateration. This involves measuring the distances between the receiver and several satellites and then calculating the location by finding the point where these distance spheres intersect. Each sphere is centered on a satellite, with a radius equal to the distance between the satellite and the receiver.

To perform this calculation, the receiver connects with GNSS satellites,

which continuously transmit signals containing precise timing information and their orbital parameters. The receiver on the ground captures these signals and compares them with a locally generated replica, based on a unique pseudo-random noise (PRN) code assigned to each satellite. The difference between the received signal and the expected local signal reveals the time it took for the signal to travel from the satellite to the receiver.

Knowing the time it took for the signal to travel allows the receiver to calculate the distance to each satellite by multiplying this time by the speed of light—a process known as pseudo-range measurement. By measuring the distance to multiple satellites, the receiver can use trilateration to determine its exact coordinates, including latitude, longitude, and altitude.

The accuracy of the calculated position depends on the geometry of the satellites in view. Ideally, satellites should be well-distributed and spread apart to maximize accuracy. If the satellites are too closely grouped or aligned in one direction, the accuracy of the position estimate can decrease, leading to less precise results. [45].

GNSS positioning is influenced by various sources of error that can impact the accuracy and reliability of location calculations. These errors include atmospheric effects, clock discrepancies, multipath interference, and signal blockage. Each of these factors can introduce inaccuracies that affect the precision of the positioning data.

Atmospheric effects occur as the GNSS signals travel through the Earth's atmosphere. The ionospheric delay happens when ionized particles in the ionosphere slow down the signals, altering the measured distance between the satellite and the receiver. Tropospheric delay, on the other hand, is caused by water vapor and other atmospheric components in the troposphere, which can also distort the position calculations.

Clock errors arise from the imperfect synchronization of clocks in both the satellites and receivers. Satellite clock errors can result from imperfec-

tions in the satellite's atomic clock or relativistic effects, while receiver clock errors are due to the less accurate clocks used in the receivers. These errors can lead to inaccuracies in timing measurements, which in turn affect the positioning calculations. Additionally, multipath interference occurs when GNSS signals bounce off surfaces like buildings or trees before reaching the receiver, causing incorrect measurements. Signal blockage, caused by obstacles such as buildings, trees, or mountains, can weaken or block signals, resulting in incomplete or inaccurate positioning data. [47].

To improve GNSS positioning and reduce errors, two key methods are often used: differential correction and augmentation systems.

- Differential correction uses a network of fixed reference stations with known locations to monitor GNSS signals. These stations calculate the difference between their known position and the GNSS-derived position, and then send this correction to nearby receivers. This helps adjust for errors caused by factors like atmospheric conditions and clock inaccuracies.
- Augmentation systems add extra data to GNSS signals to enhance accuracy. For example, systems like WAAS in the United States and EGNOS in Europe use ground stations to gather GNSS data and send corrections via satellites back to receivers. This process helps correct errors in the signals, improving positioning accuracy.

4.1.3 Real Time Kinematic GNSS

GNSS positioning can be affected by various errors, impacting its accuracy. To improve precision, technologies like Real-Time kinematics (RTK) are used. RTK can achieve centimeter-level accuracy, far surpassing traditional GNSS methods that usually offer meter-level precision.

RTK works by setting up a real-time link between a base station and a rover receiver. The base station, located at a known position, measures GNSS signals to determine its own position and calculate the difference (differential correction) between this known position and the measured one. This correction accounts for errors such as clock discrepancies and atmospheric effects.

The base station sends these corrections to nearby rover receivers via radio or cellular networks. The rover applies these corrections to its GNSS data, significantly improving its positioning accuracy. RTK is valuable for tasks needing high precision, like surveying and autonomous navigation.

However, RTK's accuracy can be affected by factors like satellite visibility, environmental conditions, and distance from the base station. It's best to keep the base station within 25 km of the rover to minimize interference. In areas without nearby base stations, corrections can be estimated from other stations. Despite its high accuracy, RTK performance may decrease in obstructed or signal-interference environments [48] the Fig. 4.2 it is possible to see a scheme of the overall system.

4.2 Used Materials

Before I go into the specific steps of the proposed approach, I describe the tools and devices that have been used. The devices used in this thesis are:

- Different Samsung devices like *Note 10*, *Note 10 plus*, *Samsung A33*, smartphones are based on the Android operative system and they don't have lidar cameras for depth info;
- *Stonex S850A* [31], external RTK-GNSS receiver used to get a precise pose in the GCS with accuracy up to 1cm in translation and 1 degree in rotation. It is installed on the top of a surveying pole with an adjustable height up to 2 meters;

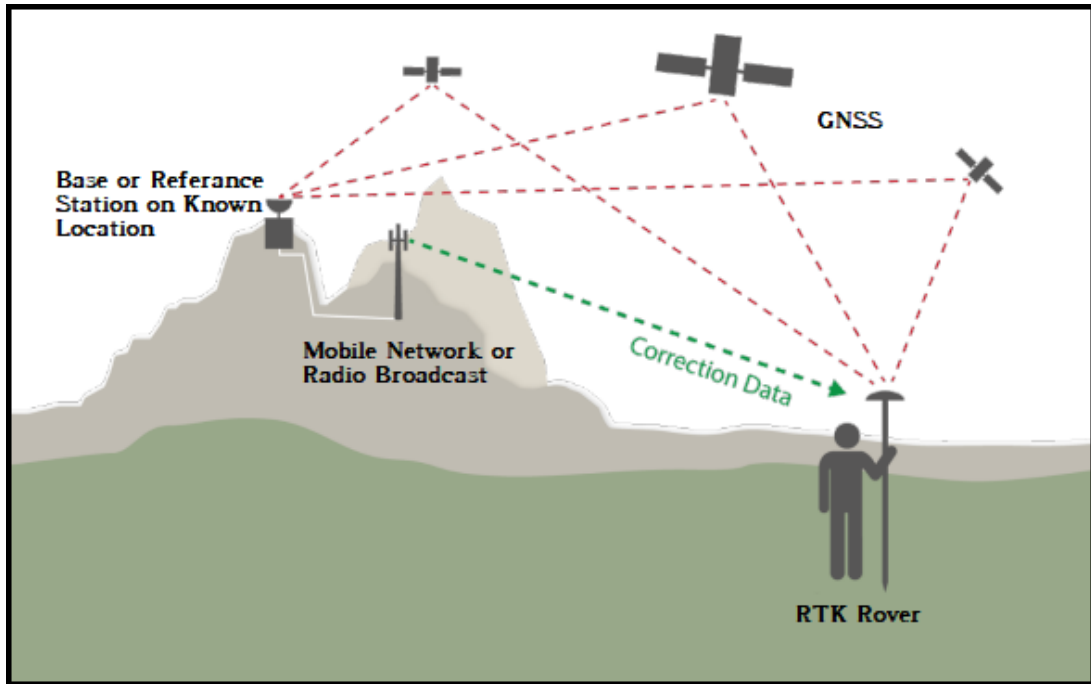


Figure 4.2: Scheme about the RTK system, Image taken by [6]

- *The Creality Ender 3* [49] is a highly regarded and cost-effective 3D printer, celebrated for its reliability and ease of use. It features a sturdy build with a heated print bed that enhances adhesion and reduces warping. The printer offers a generous build volume and employs a Cartesian coordinate system for precise movements. Compatible with various filament types such as *PLA*, *ABS*, and *TPU*, it comes with a user-friendly interface and works with open-source slicing software. Its straightforward assembly and balance of performance and affordability make it a popular choice for both beginners and experienced users;
- Optitrack rig (8 Prime22 cameras), motion capture system provided by the Multi-sensory Lab of the University of Trento to perform the spatial calibration between smartphone and RTK-GNSS.

The software used are:

- Blender [50], to develop the initial sketch of the prototypes and to examine and modify the first prototypes of our system;
- Solid-Edge [51], to refine the prototypes and take into account all the small details necessary before starting the 3D printing process;
- Cura [52], to convert STL models in gcode and to manage the 3D printer and the printing process;
- COLMAP was used to generate the 3D model and georeference it, leveraging its photogrammetry capabilities for accurate 3D reconstruction and spatial positioning;

On the programming side, the languages used are:

- Kotlin was used to develop the AR android app, handling device pairing and crafting the app interface, with all development work carried out in Android Studio;
- Python was employed to analyze data from the Motion Capture system, compute the calibration matrix, and conduct temporal calibration using two different approaches, all managed through *Jupyter Notebooks*

The frameworks and libraries used are:

- Google Android for managing the front end;
- Google ARCore [53] as Vi-SLAM for supporting the visualization contents;
- *Pandas, Numpy, Scipy and Matplotlib* to manage data and display the result.

4.3 Proposed Solution

Our solution consists of two modules, one for spatial calibration and one for temporal alignment. The two procedures are applied between the GNSS device and the camera of the smartphone.

For the spatial component, we propose to exploit a MoCap system, consisting of eight infra-red (IR) calibrated cameras, which can track moving objects with millimeter-level accuracy. To accomplish the task, we create two so-called *rigid bodies*, one for the smartphone and the other one for the GNSS device, using retro-reflective markers.

As for the temporal calibration, the problem is addressed through a synchronization technique, which measures the delay between the RTK-GNSS packets and the smartphone AR packets. A rollout filter is then implemented, allowing to alignment of the device's packets temporally. The alignment enables us to estimate the translation and rotation variations that occur between the last received RTK-GNSS packet and the current smartphone AR packet. This estimate is then used to adjust the hologram's position in the AR scene, leading to a smoother and more stable user experience.

In the specific implementation, and for reference purposes, we use an Android smartphone with ARCore [54], used for Vi-SLAM, and a Stonex GNSS receiver, the solution can be scaled to any other compatible device.

4.3.1 Spatial Calibration between the two devices

The main steps necessary to perform the spatial calibration, consist of:

- Generate, through the MoCap, a so-called rigid body for the RTK-GNSS using a pattern of markers. This rigid body must precisely align with the reference system of the RTK-GNSS, so we utilize the

3D model to create a 3D printed structure, depicted in Fig.4.3, where the markers are positioned accordingly.

- Generate a rigid body for the smartphone, which matches the camera smartphone reference system; in this case, the markers are placed on the surface of the smartphone, and in particular, one of them is placed on the smartphone camera, which represents the origin of the camera smartphone reference system. The others are spread on the smartphone surface.
- Create a 3D-printed structure that binds the smartphone and the RTK-GNSS devices.
- Place the smartphone, with its marker pattern, and the RTK-GNSS device, with the additional structure containing markers, in the MoCap reference system. Before calibration, both devices are mounted on the 3D bracket designed to interconnect them.
- The subsequent step involves acquiring the positions and orientations of the markers on both the RTK-GNSS and the smartphone using MoCap software and setting the two rigid bodies as seen in Fig.5.2, within the MoCap reference system.
- Finally, we extract the transformation matrix concerning the MoCap system reference between the two rigid bodies. This matrix represents the extrinsic calibration matrix between the RTK-GNSS and the smartphone camera.

Let us now discuss these steps one by one. To create a pattern of markers for the MoCap and define a reference system for both the RTK-GNSS device and the smartphone, it is crucial to position at least one retroreflective marker at the center of the phase of the device, with additional markers defining the 3D reference system and the object shape.

For standard-shaped smartphones, attaching markers to their surface is straightforward, especially around the accessible camera, which serves as the central point for the smartphone system reference. However, for RTK-GNSS devices with non-standard shapes, a different approach is necessary. Our solution involves using a 3D printed structure that can fit the RTK-GNSS, and which is provided with suitable mounts to affix the MoCap markers. This structure is used only in the calibration procedure and can be removed at runtime. In our implementation, the solution has been found through 3D printing, starting from the 3D CAD model of the RTK-GNSS, as shown in Fig.4.3.



Figure 4.3: Custom 3D printed structure for the creation of the rigid body of the RTK-GNSS device within the MoCap reference system.

Before proceeding with the explanation of the calibration process, let's discuss briefly the choices we made in designing the marker pattern. Five

markers were utilized for the RTK-GNSS receiver, one per side and one on the top, located in the center of the phase of the receiver (see Fig.4.3).

Similarly, also for the smartphone, four markers were affixed to the back surface. These markers were positioned on the main camera (aligned with the camera hole), the second camera, the third marker placed as low as possible under the camera, and the fourth marker on the opposite side of the camera. Notably, the markers were installed asymmetrically on both devices to prevent ambiguities in the MoCap’s rigid body recognition.

To ensure a stable translation vector and rotation matrix, an additional custom 3D-printed bracket was designed and deployed. This bracket serves the purpose of maintaining the same relative pose between the two devices post-calibration and throughout the entire acquisition process, as depicted in Fig. 5.2.

Now, with the smartphone and the RTK-GNSS bound together to a rigid structure, and all markers in place, we can proceed with the calibration by positioning the entire system within the MoCap reference system. In this phase, it is essential to ensure that all markers remain visible to the MoCap.

Next, we combine the two *rigid bodies*, according to [55]. One body group represents the markers of the RTK-GNSS device. By using these markers in conjunction with the 3D model, we calculated its center of mass, symbolizing the center phase of the RTK-GNSS receiver or the origin of the system reference.

The second body group consists of the markers on the smartphone, with its center of mass determined by the marker placed on the main camera. These centers of mass serve as the origins for the reference systems of their respective rigid bodies, facilitating translation and rotation calculations between different rigid bodies.

This procedure allows the position data (X, Y, Z), and the quaternions

(X, Y, Z, W) of each marker composing the two rigid bodies, to calculate the orientation and translation of the rigid bodies concerning the MoCap reference system.

The computed translation vector (connecting the origins of the two rigid bodies) and the rotation matrix (the product of the three rotation matrices around the X, Y, and Z axes) allow pairing the smartphone and the RTK-GNSS, as depicted in Fig. 4.5.

For each device, we extract the following information the SP_{Mocap} and $RTK-GNSS_{Mocap}$:

- SP_{Mocap} denotes the transformation matrix of the smartphone relative to the Mocap System reference, obtained from the markers comprising the smartphone's rigid body.
- $RTK-GNSS_{Mocap}$ represents the transformation matrix of the RTK-GNSS relative to the Mocap System reference, obtained from the markers comprising its rigid body.

$$RTK-GNSS_{Mocap} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$SP_{Mocap} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Given the two transformation matrices it is possible to calculate the final *Calibmatrix* as in Eq.(4.1), which allows for mapping coordinates among the two devices :

$$Calibmatrix = RTK-GNSS_{Mocap}^{-1} * SP_{Mocap} \quad (4.1)$$

To successfully calibrate the smartphone and the RTK-GNSS receiver, it is required that the two devices remain in the same relative pose after calibration, and throughout the entire acquisition session, thus obtaining a stable translation vector and rotation matrix. This required designing and deploying a custom 3D-printed bracket, as shown in Fig.5.2, to guarantee a rigid link between the devices. Through calibration, the objective consists of obtaining the transformation matrix between the center of the phase of the RTK-GNSS receiver, located in the lower center of the receiver, and the smartphone camera center. To this goal, we have employed the MoCap, so as to leverage its millimeter-accurate measurements.

4.3.2 Temporal Alignment between the two devices

The goal of the temporal calibration is to align in the best way possible the packets generated from the two devices. On the one hand, we have an Android smartphone running the app based on the ARCore framework; this generates packets with a frequency of 25 Hz. On the other hand, the external RTK-GNSS receiver outputs the position information with a frequency of 5 Hz. The fact that the two devices have a different output rate, would not cause major issues per se. Instead, the inconsistency is that, when an RTK-GNSS packet is received, it cannot be directly associated with a corresponding ARCore packet. This is due to the unavoidable latency introduced to receiving the signal from the RTK-GNSS system, processing it, and transferring it from the external receiver to the smartphone via

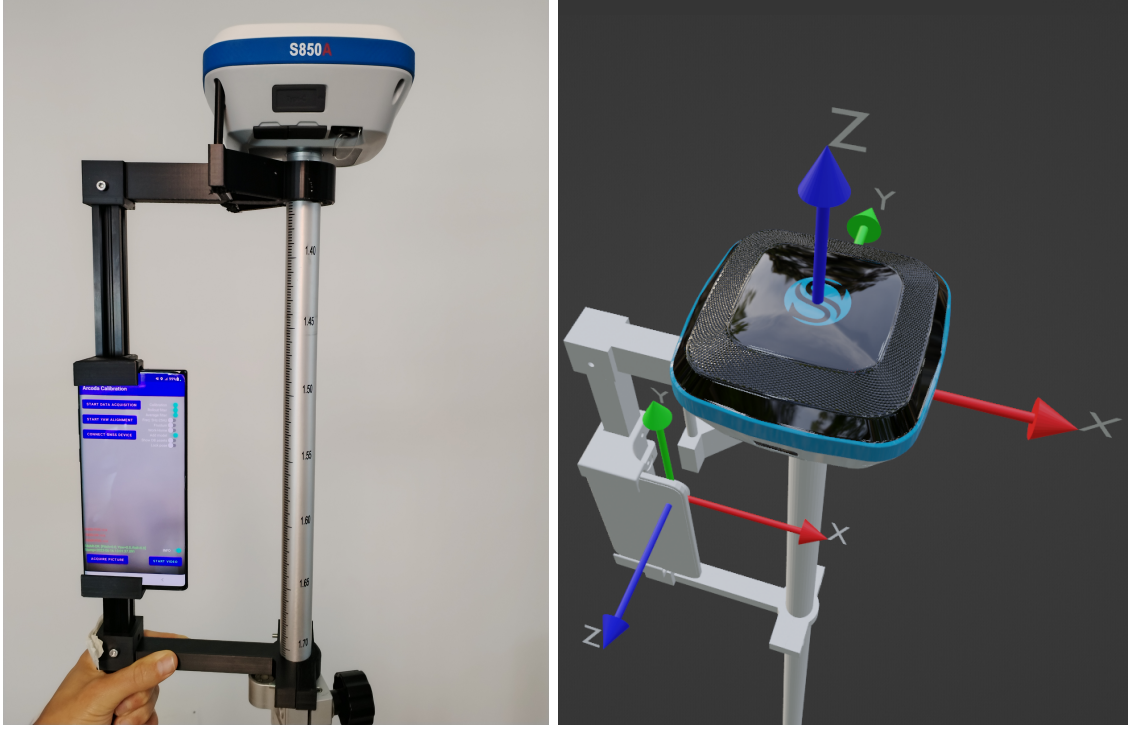


Figure 4.4: Custom bracket for the camera and RTK-GNSS device (left), with an indication of the reference coordinate systems (right), for the RTK-GNSS and the smartphone. The X-axes are in red, the Y-axes in green, and the Z-axes in blue.

Bluetooth.

Therefore, to obtain a smooth AR experience it is important to quantify such temporal misalignment: considering the UNIX timestamp of the smartphone and the RTK-GNSS, respectively, namely $t_{android}$ and $t_{RTK-GNSS}$, we calculate the packet latency ΔT_{unix} as the difference between these two temporal markers as you can see from Eq.(4.2).

$$\Delta T_{unix} = t_{android} - t_{RTK-GNSS} \quad (4.2)$$

In principle, once a sufficient number of packets has been received, and all the packet latencies have been computed, it should be possible to use these values to build a distribution. Being, the UNIX timestamp across different devices is not reliable, which results in potential mismatches in

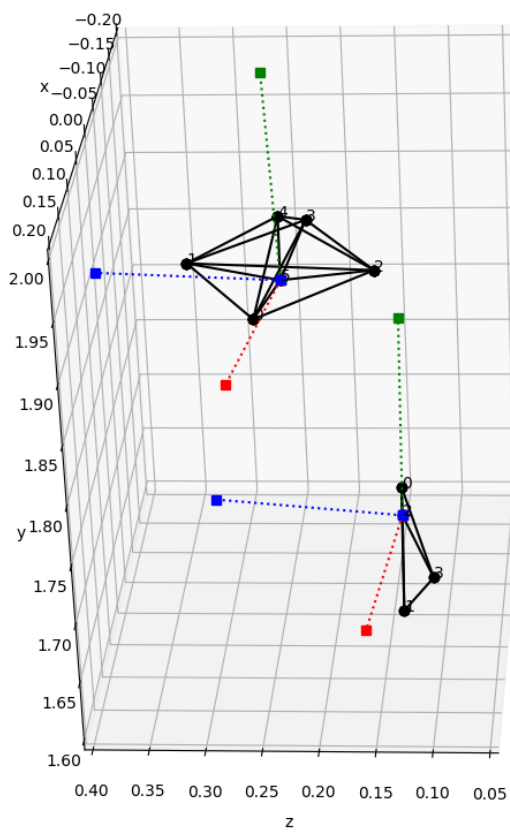


Figure 4.5: Calibration using the MoCap, mapping the coordinate systems of the two devices: Red = latitude, green = altitude, blue = longitude.

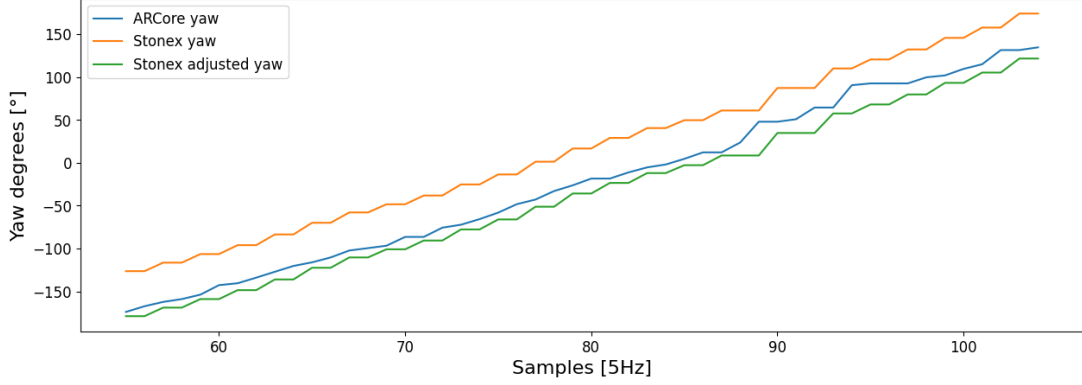


Figure 4.6: Yaw curves sampled at 5Hz, for the smartphone and the Stonex RTK-GNSS receiver, before and after adjustment.

the order of seconds, this is not a viable solution for real-time applications. Therefore, we use an experimental approach for measuring and modeling statistically the latency of the received GNSS packets. The data are relative to the ARCore *yaw* angle and to the RTK-GNSS receiver, *yaw* angle are logged while keeping the receiver in a fixed position for a few seconds. This allows obtaining a reliable measurement of the initial offset between the two angles. Then, we begin turning the two devices around the GNSS-receiver pole in a continuous fashion, similar to what would be performed in real scenarios. Once the data are gathered, they are used to plot the curves of the two *yaw* angles: the initial offset between the two measures is then applied to every point of the RTK-GNSS receiver *yaw* curve, to adjust it and align it to the ARCore *yaw* curve, as represented in Fig.4.6.

$$\begin{aligned}
 Asset_{AR} = & \left[R|T \right]_{AR}^{CAM_{t-9}} \cdot \left[R|T \right]_{CAM_{t-9}}^{CAM_{t0}} \cdot \left[R \right]_{CAM_{t0}}^{MCSP} \\
 & \cdot \left[R|T \right]_{MCSP}^{MCST} \cdot \left[R \right]_{MCST}^{ST} \cdot \left[R \right]_{ST}^{GCS} \cdot HAV(Asset_{GCS})
 \end{aligned} \tag{4.3}$$

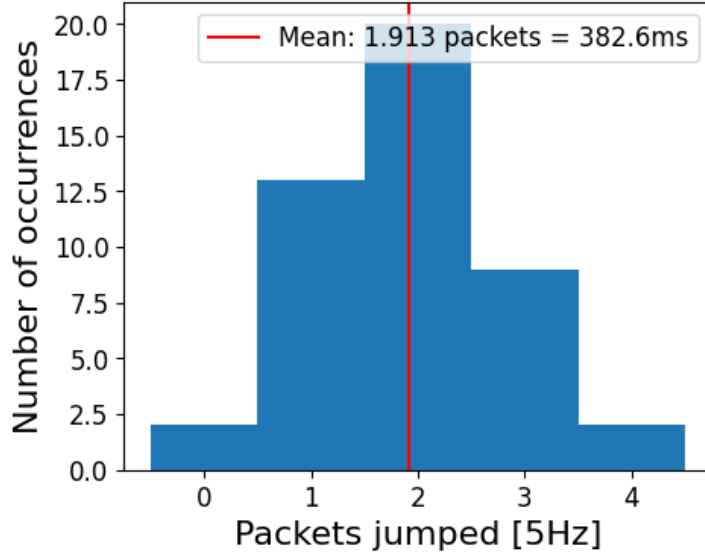


Figure 4.7: Distribution histogram of the packets jumped to achieve an optimal alignment on the curves at 5Hz shown in Fig.4.6

At this stage, without delay, the plots (the blue and the orange curves in Fig. 4.6) should overlap. However, in real cases, they are often translated by a certain offset due to the delay introduced by the GNSS device. Because of the presence of this GNSS delay, the two curves do not perfectly align when motion is present. This discrepancy allows us to estimate the temporal offset between them. To do so the curves used are the "ARCore *yaw* curve" and the "Stonex adjusted *yaw* curve" as shown in Fig.4.6. Once compared, a value of delay expressed in a number of packets is retrieved. This can later be converted into a proper delay, in seconds, by multiplying it by the inverse of the sampling frequency.

The delay estimation starts by comparing the value of a point on the Stonex curve with the value sampled at the same time location on the ARCore curve. This very same Stonex value is then compared also with the N previous points of the ARCore curve, where N is defined empirically and corresponds to the number of the past ARCore packets used for compar-

ison against the latest RTK-GNSS packet received. In short, if there is a delay in the yaw values from the RTK-GNSS receiver, then by examining the ARCore yaw values backward in time, it should be feasible to find an ARCore yaw value that aligns with the last RTK-GNSS yaw value. Once the corresponding pair is found it is possible to compute the delay. This computation is carried out for all the points in the Stonex curve. We can then define a histogram, which provides an effective representation of the delay between the devices. Using the data acquired on the field we built the histogram at 5Hz (see Fig.4.6). The results indicate that, on average, the latency between the devices is around 380ms, which equals around 9 ARCore packets (40ms for each packet). This value is later used to implement a roll-out filter, to align the current received RTK-GNSS position with the ARCore frame generated *9 frames earlier*. It is also possible to calculate the spatial error given the latency, as illustrated by Eq.(4.4). In this formula, the *SPPose* represents the pose of the smartphone at a specific time, while the *ErrorMotion* represents the error motion in terms of the pose.

$$ErrorMotion = SPPose_{t_0} \cdot (SPPose_{t-9})^{-1} \quad (4.4)$$

4.3.3 Obtaining the Transformation Matrix between the camera of the Smartphone and the RTK-GNSS device

After computing the spatial calibration matrix between the two devices and the delay for the roll-out filter, it is possible to build the final transformation matrix to represent the GCS assets in the AR space with centimeter accuracy.

The final formulation is reported in Eq. (4.3).

We describe hereafter all the steps involved in the final transformation, reading it from the input *Asset_{GCS}* toward the output *Asset_{AR}*:

- Haversine formula (HAV), to compute the linearized distance of the asset (in latitude, longitude, and altitude) in the GCS (assuming it to have origin in the current Stonex position).
- Stonex rotation matrix ($GCS \rightarrow ST$), to move the asset point from the linearized GCS (centered on the GNSS device) to the GNSS device (ST) reference system (pitch, yaw, roll w.r.t. GCS).
- MoCap rotation matrix ($ST \rightarrow MCST$), to rotate the Stonex reference system by -90° around the Z-axis, followed by another rotation of -90° around the X-axis. After that, the Stonex reference system (previously: Z=altitude, Y=latitude, X=longitude) is aligned with the reference system used during calibration with the Motion Capture system (Y=altitude, X=latitude, Z=longitude) and the asset point is represented w.r.t the MoCap Stonex (ST) reference system.
- Calibration Matrix ($MCST \rightarrow MCSP$), computed in section 4.3.1, to move the asset point from the MoCap Stonex (ST) reference system to the MoCap smartphone (SP) reference system.
- MoCap rotation matrix ($MCSP \rightarrow CAM_{t_0}$), to rotate the MoCap smartphone (SP) reference system by $+90^\circ$ around its Y-axis. The asset point is now represented w.r.t the current ARCore Camera reference system.
- RolloutFilter ($CAM_{t_0} \rightarrow CAM_{t-9}$), to move the asset point from the ARCore camera reference system of the current frame to the ARCore camera reference system, 9 frames behind. This accounts for the latency between the smartphone and the Stonex.
- CameraPose ($CAM_{t-9} \rightarrow AR$) referring to the 9-th packets in the past, to move the asset point from the ARCore camera reference system delayed by 9 frames, in the ARCore origin reference system.

4.4 Results

To validate our architecture, we have chosen to test the entire system in a real scenario. First, we have used the RTK-GNSS receiver to acquire the GCS pose of a specific spot outdoors, leveraging its centimeter-level accuracy. At this point, we can qualitatively and quantitatively evaluate the spatial and temporal calibration for the given RTK-GNSS/ smartphone ensemble.

For the qualitative evaluation of the spatial calibration, we use the previously computed transformation matrix, calculated by the MoCap to visualize a hologram over-imposed on the point acquired. As can be seen in Fig.4.8, the hologram (in red) is displayed accurately also at quite large distances, over 3 meters from the point of acquisition. This confirms the accuracy (and relevance) of the spatial calibration performed in the laboratory, and also of the entire transformation pipeline.

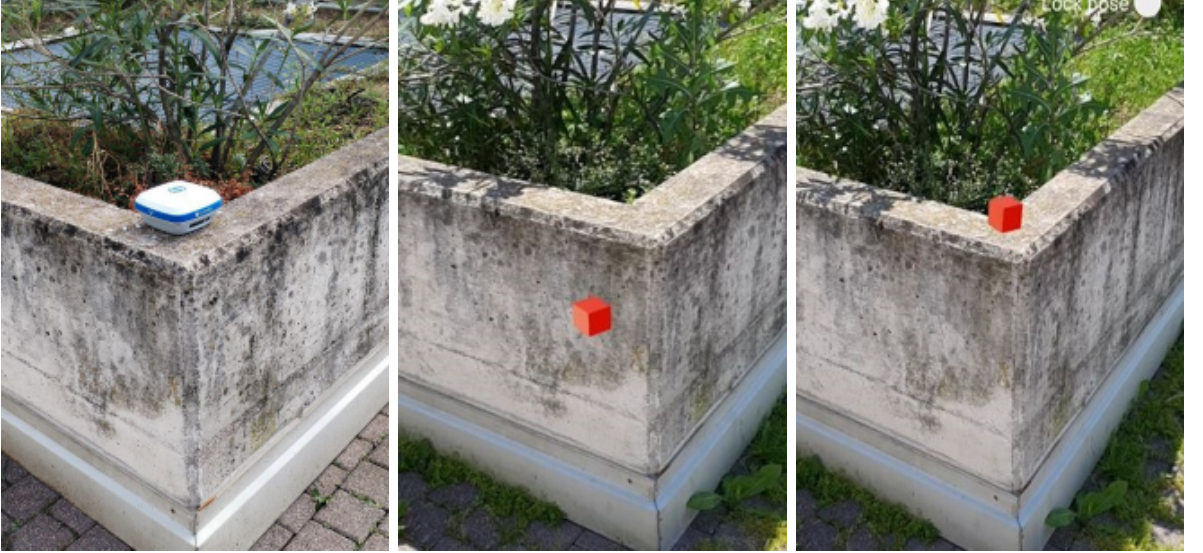


Figure 4.8: The RTK-GNSS device is positioned in the environment to acquire the point location (left); the red hologram is virtually positioned, without spatial calibration (center) and with spatial calibration (right).

As for the quantitative evaluation of the spatial calibration, we compute

Term	Description
GCS	Geographical coordinates reference system
$Asset_{GCS}$	Position of the asset with respect to GCS
HAV	Haversine formula used for calculating the meter distance between two GNSS locations for AR purposes
ST	Pose of the GNSS device with respect to the GCS
$MCST$	Pose of the GNSS device with respect to the MoCap reference system
$MCSP$	Pose of the smartphone with respect to the MoCap reference system
CAM	Pose of the smartphone camera with respect to the ARCore reference system
AR	ARCore reference system
$Asset_{AR}$	Asset in the ARCore reference system

Table 4.1: Detailed explanation of the relevant terms in Eq. (4.3)

the angle misalignment along the azimuth direction. This is done by placing a QR code with its top-left finder pattern on the previously acquired point, and by over-imposing the red hologram on that spot using its GCS pose, as shown in Fig.4.9. This type of marker is usually used for the localization of robots mobile devices and other systems [56], [57], [58] and fits well also our scenario. Thus, we exploit this pattern to evaluate the accuracy of our location-based AR solution, using such markers as ground truth.



Figure 4.9: Error measurement process: the QR code top-left finder pattern is placed on the acquired spot, and the red holograms are placed using the acquired GCS pose.

Once the scenario is set up, the projections from the 3D world to the 2D camera plane are obtained for the top-left finder pattern and the red hologram. Next, the x -axis pixel offsets between the retrieved 2D positions and the center of the smartphone's screen are computed (pde_1 and pde_2). After that, the smartphone's focal length in pixels over the x -axis (fc_x) is used in Eq. (4.5) together with the offsets obtained above, allowing to

compute the angles α and β . Finally, the absolute difference between these two angles allows us to find the misalignment (E) between the resulting 2D position of the asset and the expected one, as shown in Fig.4.10.

$$E = \text{abs}(\alpha - \beta) = \text{abs}\left(\arctan\left(\frac{pde_1}{fc_x}\right) - \arctan\left(\frac{pde_2}{fc_x}\right)\right) \quad (4.5)$$

Using the proposed calibrated solution, we have evaluated the final error around the *yaw* angle, starting from the hypothesis that the smartphone during the process remained still, leading to a final average error of 0.449 degrees; the scenario of the acquisition is shown in Fig.4.9.

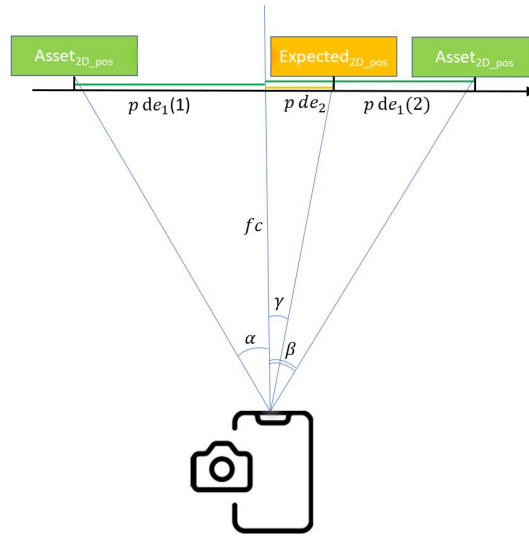


Figure 4.10: Graphical representation showing how the error along the azimuth direction is computed, in particular in this case we present two possible cases of the *Asset2D* position. In the first case the $pde_1(1)$ (number of pixel) will have a negative value, for the second case indeed, the value of $pde_1(2)$ will be positive.

At last, we have also evaluated qualitatively the contribution of the temporal calibration, enabling/disabling the roll-out filter and the low-pass filter. The implementation of the roll-out filter improved the user experience, leading to a better quality of the AR experience. The low-pass

filter, instead, contributes to a more stable experience, especially in the presence of the user’s motion.



Figure 4.11: Acquisition process: images are collected through the smartphone coupled with the RTK-GNSS receiver, and positioned in the environment, according to their camera pose.

In addition to the previous evaluations, the proposed solution is validated also through the collection of a small dataset composed by images and their corresponding poses in the GCS, as shown in Fig.4.11. The purpose of this last test is to evaluate our system for 3D reconstruction, helping the classic photogrammetry algorithms, as COLMAP [59], which can leverage GNSS information of the pictures as a starting point when attempting to align the frames, for an optimal outcome. Moreover, this information allows for the precise scaling of the 3D model calculated and, ultimately, its georeferencing. This enables to precise overlay of the 3D model on the real scene, blending the real world with the virtual content, as can be seen in Fig.4.12.

The results of the proposed solution are compared with the solutions currently available on the market, as shown in Tab. 4.2. The comparison is carried out according to parameters such as translation accuracy, rotation accuracy, covered area, and support for dynamic scenes (i.e., the possibility

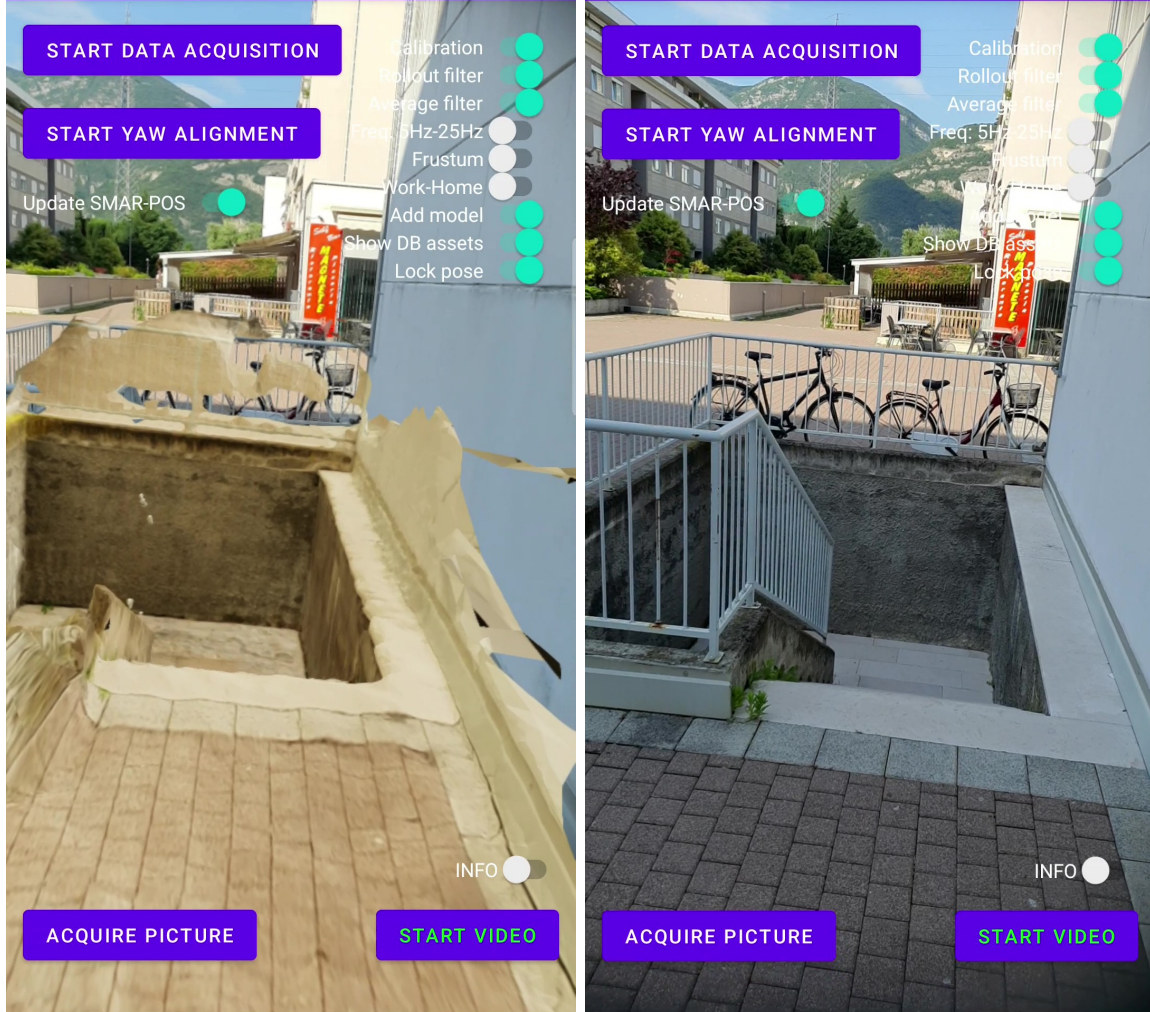


Figure 4.12: 3D digitalization and overlay of the virtual georeferenced information (left) with real-world content (right).

of deployment with limited or changing visual information).

Our solution defines translation and rotation accuracy values, see Table 4.2. It also notably excels compared to all visual positioning approaches in terms of the covered area and support for dynamic scenes.

Performing an explicit quantitative analysis is not straightforward. Indeed, all the providers of VPS solutions did not release any precise error metric, with only Niantic [61] giving some rough information about the translation accuracy, defining the error in the order of "centimeters". On

	Trans. acc.	Rot. acc.	Coverage	Dynamic
Sturfee VPS [60]	—	—	15 cities	X
Niantic Lightship [61]	centimeters	—	6 cities	X
Immersal VPS [62]	—	—	4 cities	X
Google VPS [63]	centimeters	—	Urban areas*	✓
BiodivAR [22]	—	—	Worldwide	✓
Project LARA [19]	< 20 cm	1°	Worldwide	✓
Our solution	1.6 cm @ 2 m	0.449°	Worldwide	✓

Table 4.2: Comparison of the proposed solution with other state-of-the-art approaches for location-based AR. Our translation accuracy is estimated at 2m distance. [KEYS: Trans. acc is the translation accuracy, Rot. acc. is the rotation accuracy, Coverage refers to the scalability of the solution, and Dynamic reports the robustness to environmental visual variations]

the other hand, also the works discussed in chapter 2 do not provide precise error metrics, and only Project LARA [19] defines a translation accuracy lower than 20cm, and a rotation accuracy of around 1 degree.

Regards the the covered area, the proposed solution, as well as the ones discussed in chapter 2 do not require pre-mapped environments like the VPS technology, so our solution can be used anywhere, provided that the GNSS signal is available. Among the competitors in the VPS field, Google VPS stands out as the sole qualified competitor, thanks to the pre-mapped information from Google Street View. Still, the coverage is mostly limited to cities and streets in developed countries, and it does not cover, for example, off-road locations.

Finally, as highlighted in the last column of Table 4.2, the GNSS-based solutions are also the only ones suitable for dynamic scenes. Indeed, VPS solutions could suffer limitations in localizing the user when strong changes in illumination occur. This issue is even more evident in the case of changes in the scenario (for example due to: seasons, renovations of buildings, evolution of city design, and so on).

4.5 Discussion

We tested the system in many outdoor scenarios, one of the most interesting study cases being outdoor excavations: the goal is to superimpose the hologram after the road work is covered and validate the obtained results. In this particular case it is possible to appreciate the accuracy of the spatial calibration, as shown in Figs.4.13, 4.14. As can be seen from the pictures, the alignment is good, and this is possible when the system can rely on the differential correction, requiring the RTK-GNSS to attain an appropriate satellite coverage.

In fact, being the precision linked with the satellite coverage, partial occlusions might occur, such as when inside the excavations or near buildings/trees, thus preventing the visibility of the satellites. In such cases, the system performance could decrease.



Figure 4.13: (left) A scenario is shown, where the real scene is over-imposed with the virtual (pre-acquired) counterpart. Here, the specific case study is the excavation. The zoom of the red square area is also shown (right). The spatial accuracy of the over-imposition is very good.



Figure 4.14: In this example, the hologram of the excavation is overlaid on the scene (left) after the excavation has been covered (right).

As per our specifications, the devices are bound through a fixed hardware bracket. Although this does not pose issues per se, it is worth noting that if either of the two devices is moved or swapped, this generally requires a re-calibration. This could be seen as a potential limitation for widespread adoption of the proposed solution, which does not allow for an on-the-fly calibration.

However, to foresee a seamless swap between different devices, major attention should be paid to defining the geometry of the 3D support used for calibration, especially in accommodating different RTK-GNSS devices, each of them with a unique geometry and center of phase

4.6 Extended Reality

In this final section, I will briefly present an application we have developed using MR hardware, highlighting its potential but, more importantly, the current limitations for industrial scenarios. It is important to note that, at least regarding augmented reality, smartphones will eventually be replaced by smart glasses in the future, even though in the current solution presented

in this thesis, the smartphone is the device used. Before presenting the application, I will describe MR, and location-based and marker-based AR applications in more detail. In this context, I will emphasize the potential of using MR capabilities and finally show a prototype where I combine smart glasses and an RTK system, which I used to run my application.

In the realm of MR, there is a most known subset which is AR. There are two types of AR marker-based that use visual markers, such as QR codes or predefined images, to trigger digital content when scanned by a camera. This type of AR is particularly useful in museum settings, where it can provide additional information about artworks, or in marketing, to create interactive experiences around products. In contrast, location-based AR relies on GPS data, a compass, and other sensors to overlay digital information onto the real world based on the user's geographical location. This approach is ideal for applications like navigation, where it can guide users through cities, or in tourism, offering contextual information about landmarks as users explore an area. The key difference lies in how the AR content is triggered: by recognizing a specific marker or by detecting the user's physical location.

In the case of location-based AR, achieving high accuracy is crucial. This often requires using different devices for displaying the AR content and acquiring the GPS position. As a result, it's necessary to calculate the transformation matrix between these two devices to establish the relationship between their reference systems as we have seen. Additionally, Simultaneous Localization and Mapping (Vi-SLAM) algorithms are employed to maintain object tracking within the environment. These algorithms help ensure that virtual objects stay accurately positioned relative to the real world as the user moves, providing a seamless and immersive AR experience.

Our next goal is to move further toward virtuality, where interacting

with holograms becomes a tangible experience. A future where gestures are not just used to navigate digital information overlaid in the real world but become the means to directly manipulate three-dimensional holograms, creating an unprecedented immersive and interactive experience. This evolution will allow us to explore new frontiers in communication and interaction, transforming the way we perceive and engage with the virtual environment. This is also made possible through the use of new hardware, such as smart glasses. These devices enable users to interact with holograms directly through gestures, seamlessly blending the virtual and physical worlds. With smart glasses, the experience becomes even more immersive and intuitive, allowing for more natural and dynamic interactions with digital content in three-dimensional space. For this goal I decided to explore the development of a real application using MR devices, in this work I demonstrate the usage of gestures and head movement like in Fig. 4.15 as a way to interact with the application.

So, I present briefly the MR application developed, highlighting briefly the main limitations of the current hardware, despite the obvious advantage of using smart glasses for displaying AR content. This location-based MR solution, exploiting the Hololens and the Stonex. Combine the two devices with the support of the helmet, as you can see in Fig. 4.16.

Through a Bluetooth connection the devices communicate, so the Hololens acquired the GNSS position and rotation provided by the RTK-GNSS, the Stonex in this case. The final result is depicted in this Fig 4.17.

I tried this setup in many outdoor industrial scenarios, the goal of the application developed was to display the network underground, overlaid on the real world, with the possibility to interact with the objects to access all the needed information about the assets of the Utility. It was clear from the beginning the advantages, working hands-free with all the information overlaid on the field of view boosts efficiency and more



Figure 4.15: An example of interaction using gesture with HoloLens

importantly safety. The device has proven to be effective in gesture recognition, although it requires an initial training phase. Operators are used to interacting with a screen, so switching to interaction with smart glasses is not straightforward, especially for those less open to change. Even though the technology is very promising, the current hardware is not ready primarily because external light for smart glasses optical-see-through devices is interference that can even make holograms invisible. Since the hologram is projected onto the semi-transparent screen and is formed by light, external sunlight interferes. Another major challenge is definitely usability for long periods, due to the weight and bulk of the HoloLens, making it difficult to use.



Figure 4.16: The helmet as a support for the combination of the Stonex and the HoloLens

From my analysis these are the main limitations of the current hardware:

- The weight of the HoloLens added with the weight of the Stonex makes impossible the usability for the operators for a long period.
- HoloLens is designed for indoor scenarios, the holograms projected on the lens with the interference of the sunlight disappear.
- Getting all operators accustomed to using gestures, considering that they are used to the classic touch of a smartphone screen



Figure 4.17: The visualization of the holograms of the underground net from the Hololens

Chapter 5

3D Reconstruction

Neural radiance field-based methods have recently emerged as an alternative to traditional photogrammetry in the field of image-based 3D reconstruction. This innovation is especially significant for the challenging scenarios of excavation sites, where the accuracy and detail of 3D models are crucial. This research is motivated by the potential of NeRF to enhance the precision and reliability of reconstructions in such complex settings. By comparing NeRF with traditional photogrammetry across varied scenes, this study aims to rigorously assess their efficacy in capturing intricate details, surface textures, and overall geometric accuracy. Our goal is to evaluate the suitability of NeRF techniques to be adopted in real-world applications with a particular focus on excavation sites. This endeavor seeks to pave the way for more informed adoption of cutting-edge 3D reconstruction technologies working related fields. In recent years, the advancement of 3D reconstruction technologies has opened new avenues in the documentation and analysis of urban landscapes, such as working, industrial, and archaeological sites. Among these, photogrammetry has long been established as the baseline for precise, high-resolution mapping and modeling. However, the recent advent of Artificial Intelligence (AI) in the 3D field, thanks to the introduction of NeRF [9] and more recently 3D-

GS [10] techniques presents a novel paradigm, potentially overcoming some of the inherent limitations faced by traditional methods. This chapter aims to provide a comprehensive comparison between these cutting-edge techniques, focusing on their application in the industrial context of excavation sites. Excavation sites present unique challenges for 3D reconstruction due to their dynamic nature and intricate details. Operators' data collection methods must adapt to ensure fidelity in reconstructing occluded regions. Integrating geospatial data with 3D reconstructions aids utility companies in locating subsurface infrastructure accurately. This enhances worksite planning, and management, and reduces the risk of accidental damage during future excavation.

Moreover, the use of georeferenced data in 3D reconstruction enables the development of AR technologies, thus offering an additional layer of information, enhancing operational safety and efficiency. By overlaying digital models onto the physical world, operators can gain real-time insights, further preventing the accidental severing of critical infrastructure.

This chapter investigates the strengths and limitations of photogrammetry, NeRF, and 3D-GS in excavations, where geographical positioning data is essential. Utilizing datasets of images and precise coordinates, it aims to measure each method's efficacy, particularly where traditional photogrammetry is not accurate. Implicit methods like NeRF show promise in rendering complex scenes with diverse surface properties, while 3D-GS provides a very accurate estimation of the surfaces and fine structures, areas challenging for conventional methods.

NeRF-based methods have recently proven to be a valuable alternative to traditional photogrammetry in the field of image-based 3D reconstruction. This innovation is especially significant for the challenging scenarios of excavation sites, where the accuracy and detail of 3D models are crucial. This research is motivated by the potential of NeRF to enhance the

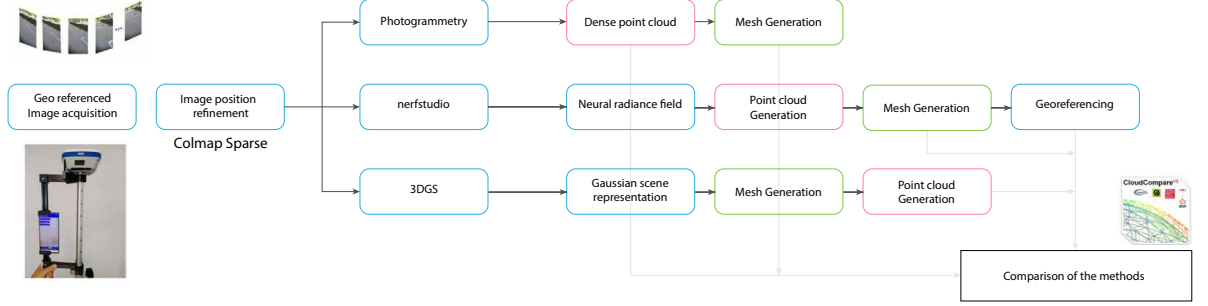


Figure 5.1: Overview of the proposed methodology

precision and reliability of reconstructions in such complex scenarios. By comparing NeRF with traditional photogrammetry across varied scenes, this study aims to comprehensively assess their performances in capturing intricate details, surface textures, and overall geometric accuracy. Our goal is to evaluate the suitability of NeRF techniques to be adopted in real-world applications with a particular focus on excavation sites.

5.1 Methodology

This study aims to evaluate the effectiveness and potential benefits of NeRF against traditional image-based reconstruction techniques, particularly photogrammetry, in the context of augmented/virtual reality applications. Our focus is on challenging outdoor scenarios. We include excavation sites and playground objects, which are characterized by unbounded environments and non-Lambertian surfaces. To facilitate a direct comparison, the same dataset of images, captured with georeferencing, is utilized across all reconstruction methods. This standardized approach ensures that differences in the reconstruction quality and efficiency can be attributed solely to the methodologies rather than due to a bad alignment.

5.1.1 Dataset acquisition

The datasets are collected using a system comprised of two devices: a smartphone and an RTK-GNSS spatially calibrated as we can see from the figure 5.2 in the paper [64]. These devices ensure highly accurate pose information for all the images we have collected. The study aims to analyze utility case studies, therefore, as scenarios, we have selected some simple playground games mixed with real excavation scenarios where the reconstruction is more challenging. Our datasets consist of 7 playground scenarios and 3 excavation scenarios.



Figure 5.2: System acquisition composed by Stonex and Smartphone for obtaining high accuracy georeferenced images

Acquisition Process: Our analysis focuses on handheld acquisitions performed by operators in the field. The trajectory involves rotating around the object when it is possible, maintaining a capture at eye level with the camera facing downward. The angle varies depending on whether the scenario is an excavation or a playground game. During acquisition, the frame rate is fixed at 5 frames per second with a resolution of 1280 x

	Number Images	Geo tras. acc.	Resolution
Parcogiochi1	153	< 2 cm	1280x720
Parcogiochi2	218	< 2 cm	1280x720
Parcogiochi3	159	< 2 cm	1280x720
Parcogiochi4	149	< 3 cm	1280x720
Parcogiochi5	151	< 3 cm	1280x720
Parcogiochi6	152	< 2 cm	1280x720
Parcogiochi7	245	< 3 cm	1280x720
Scavo5	565	< 3 cm	1280x720
Scavo7	468	< 3 cm	1280x720
Scavo13	745	< 3 cm	1280x720

Table 5.1: From this table, we describe the number of images and their resolution for all acquisitions, then we also include, for each scene, the accuracy of translation with respect to the GCS of the images.

720. The accuracy of the pose data is approximately 3 cm and less than 1 degree for each acquired image. We maintain a uniform velocity during acquisition, so the number of images for each scenario depends on the length of the trajectory. In table 5.1, we provide the main information about all the acquisition scenarios.

5.1.2 Methodologies employed

Three distinct reconstruction methodologies were applied to the captured datasets; an overview is shown in Figure 5.1:

1. **Photogrammetry:** The classical photogrammetric procedure involves estimating camera orientation parameters for sparse point cloud construction, generating a dense point cloud; mesh creation and texture extraction complete the reconstruction process. For this purpose, we used COLMAP, with all phases conducted in high-quality mode to ensure maximum detail and accuracy.

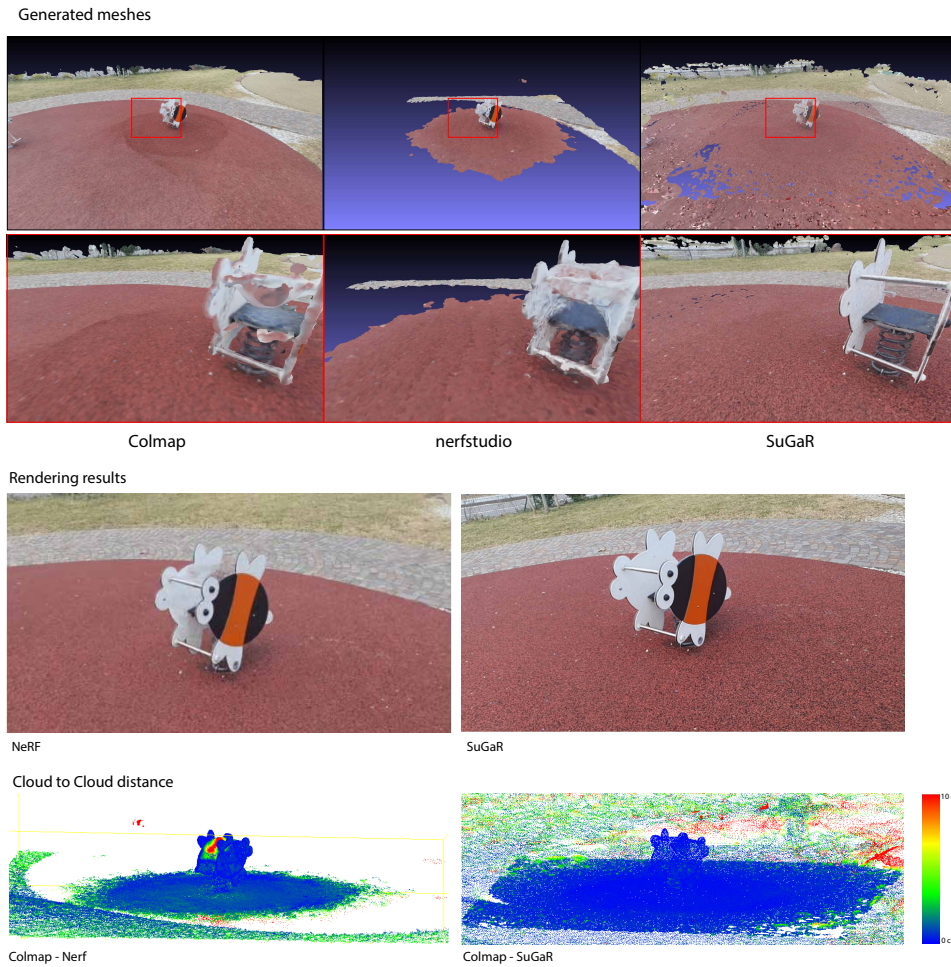


Figure 5.3: Comparison of the mesh obtained with the proposed methodologies on the playgrounds dataset. Other scenes can be found at: https://zenos4mbu.github.io/photogrammetry_nerf.github.io/

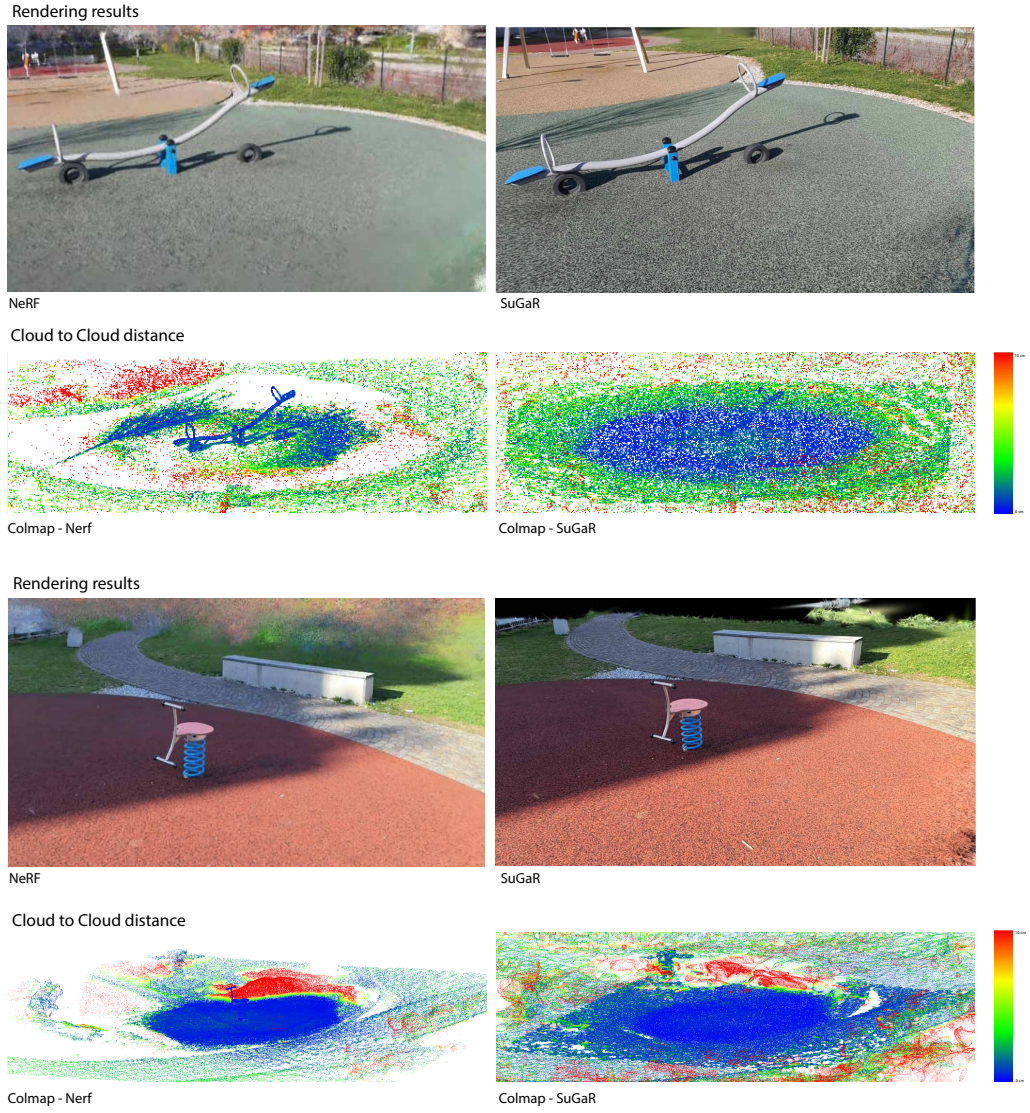


Figure 5.4: Comparison of the cloud-to-cloud distance of the proposed methodologies on the playground dataset.

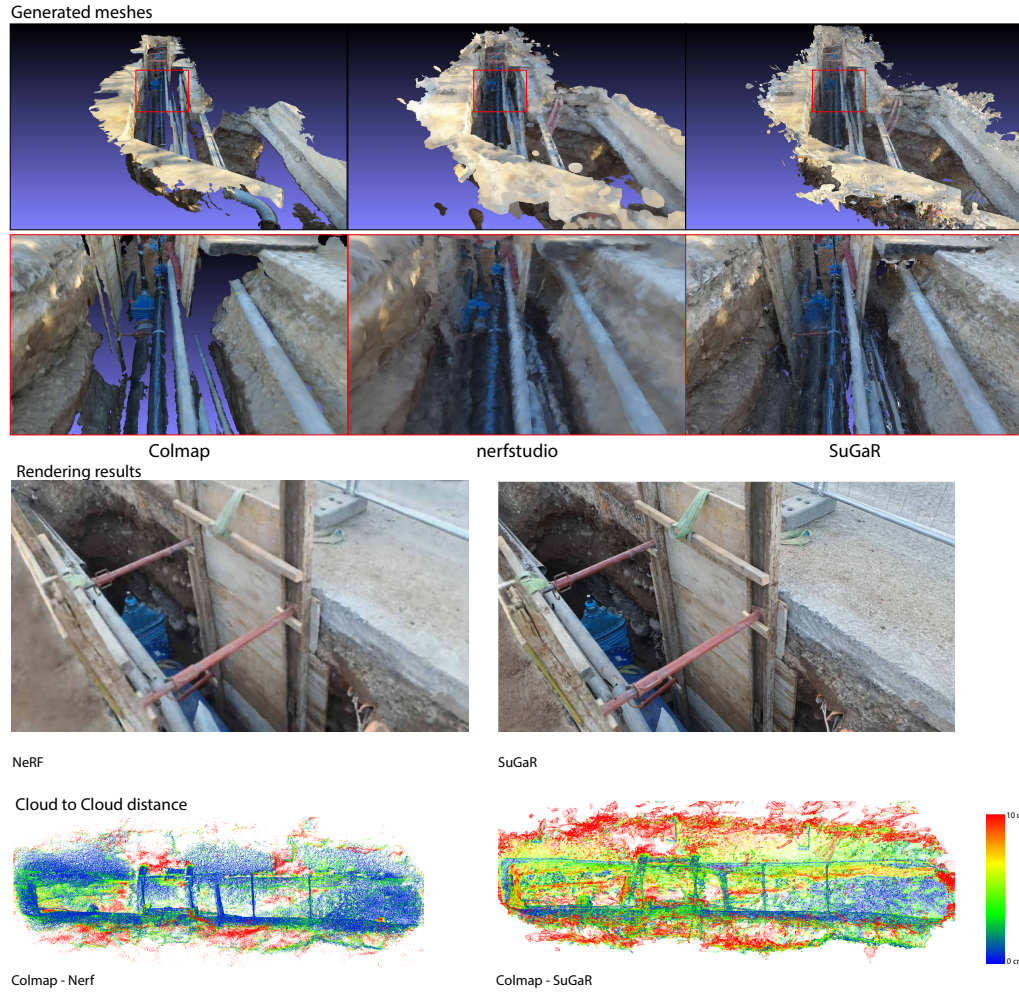


Figure 5.5: Comparison of the cloud-to-cloud distance of the proposed methodologies on the excavation sites dataset. Other scenes can be found at: https://zenos4mbu.github.io/photogrammetry_nerf.github.io/

2. **NeRF-Based Reconstruction:** The training of Neural Radiance Field reconstruction requires known camera poses as input. We use *nerfstudio* [65], and in particular "NeRFacto", a model strongly based on InstantNGP [27], used for its fast training and inference. We then extract the dense point clouds and textured mesh from NeRFstudio's API; in particular, for mesh extraction, we exploited Poisson reconstruction.
3. **SuGaR:** Similarly to NeRF, this method requires known camera poses as input. This explicit model is then trained to approximate the radiance field of the scene. The training of SuGaR involves more than one step. The training starts with 7k iterations of normal 3D-GS and 7k iterations of SuGaR finetuning to extract a more precise geometry.

The acquisition of our dataset incorporated georeferencing, to simplify the alignment process for the reconstructions. The only exception is NeRF, as an implicit framework this model normalizes its coordinates between -1 and 1. This aspect of NeRF requires an additional step to calibrate the model, incorporating scale and translation derived from the georeferenced input to ensure accurate alignment. For the dataset to be used in training, we first need to estimate the camera parameters from the input images. This estimation is necessary because the neural network requires knowledge of both the camera's positions and the corresponding images to accurately generate the scene representation. To achieve this, we utilized COLMAP, a known software for its application of Structure from Motion (SfM) techniques [33], for estimating three-dimensional structures from two-dimensional image sequences.

To facilitate comparison, given that outputs from photogrammetry are not directly comparable with those from NeRF or 3D-GS, we incorporate an additional conversion phase. NeRFstudio provides functionality to con-

vert NeRF outputs into point clouds and meshes. The point clouds are easily exported since the neural representation can be inspected at any 3D point. For the meshes, this conversion employs the *marching cubes* algorithm and the Poisson surface reconstruction method. In the SuGaR framework, the mesh extraction phase is also done through *marching cubes* or Poisson surface reconstruction. In this case, the reconstruction is enhanced thanks to the precise estimation of the normals of the sampled points. To obtain an accuracy metric we derive a cloud-to-cloud comparison using the CloudCompare software.

5.1.3 Comparative analysis framework

The comparative analysis between these methods focuses on the following key metrics: (i) **Accuracy and Detail Resolution**, to evaluate the fidelity of the reconstructed models to the original scenes, and (ii) **Processing Time**, to assess the efficiency of each methodology in terms of computational resources and time required for reconstruction. We propose using the point clouds generated by the studied methods to compare the fidelity of the reconstructed models. In this way, we can obtain a quantitative metric. To be more specific we measure the cloud-to-cloud deviation of the methods based on radiance fields concerning the reconstruction using classical photogrammetry. This measure is an absolute value, which does not tell which method performs better; it only informs about the deviation from one reconstruction to the other. Therefore, we also show the rendering results to see the performances in graphical terms, in Figure 5.5. In addition to this quantitative result, we also propose a qualitative comparison of the resulting meshes, comparing the proposed methodologies in Figure 5.3.

5.2 Discussion

We show a comparison of NeRF-based techniques against traditional photogrammetry utilizing COLMAP. All models are trained on an NVIDIA RTX 3090 GPU. The assessment focuses on their effectiveness in view synthesis and 3D reconstruction, particularly in expansive, unbounded environments. The results of our analysis highlight that the three methodologies produce high-quality point clouds, with very close results especially in the fine structures of the 3D scene, as illustrated in Figure 5.3. Notably, NeRF’s output shows a denser point cloud around high-frequency scene features but has gaps in smoother regions. The radiance field rendering results show that the quality of the reconstructed views is high and is very difficult to say if NeRFstudio or SuGaR presents the best result. However, the comparison illustrated in Figure 5.3 highlights a failure case of NeRFstudio, with a red area within the scene’s object of interest indicating a high cloud-to-cloud distance. This issue not only produces a discrepancy in the point cloud representation but also results in blurring within the targeted region of the neural reconstruction.

Considering the extensive usage of meshes in VR and AR applications, for their simplicity and low memory footprint, we present a comparison of the meshes produced with the three methodologies. In Figure 5.3 we show the obtained meshes also showing a detail of the reconstruction in the region of the 3D scene with finer details. As depicted in Figure 5.3, there’s a noticeable variance in detail and texture among the outputs. The COLMAP mesh, while being consistent, falls short on representing thin structures. In contrast, the NeRF mesh shows greater detail but presents some holes. The SuGaR mesh stands out for its superior detail, accurately capturing complex structures where others falter, thanks to its precise normal calculations. Another point to consider is the difference in accuracy between the

two scenarios we have examined. The playground scene is easier and has better results compared to the case of excavations. The complexity of the excavation scenario reduces the performance in reconstruction, especially with the SuGaR and NeRF methods. It is noticeable in figure 5.5 that there are many artifacts on the road surface visible on the Cloud to Cloud distance analysis, especially in the case of SuGaR, and there are also many holes, especially in the excavation bottom. Finally, we analyze the processing time for each method. Regarding this aspect, there is no difference between SuGaR and COLMAP. Instead, the best performance is observed with InstantNGP, which takes about a quarter of the time compared to the other methods. Additional materials regarding our analysis, can be accessed through this link ¹.

¹https://zenos4mbu.github.io/photogrammetry_nerf.github.io/

Chapter 6

Meta-Trainer

The application that I present in this chapter is called Trainreal; it is based on the project Meta-Trainer, which is the result of a collaboration of the MMLab (Multimedia Signal Processing and Understanding Lab) researchers and some students of the University of Trento. The team has participated in a challenge opened by Nreal (now formerly known as Xreal), called AR-jam [66]. Within the context of the challenge, we have developed an MR application, where it has been possible to experiment with gestures, and head tracking on the Xreal device smart glasses. As a final result, we were able to win the Honorable mention and our application has been added to the official store of Xreal [12]. In Fig. 6.1 it is possible to see the preview of the application.

Based on this application I present in the next section the work behind Trainreal [67].

The source code of the project is made available in our GitHub repository¹.

¹<https://github.com/Lorenzoarc/traiNreal>

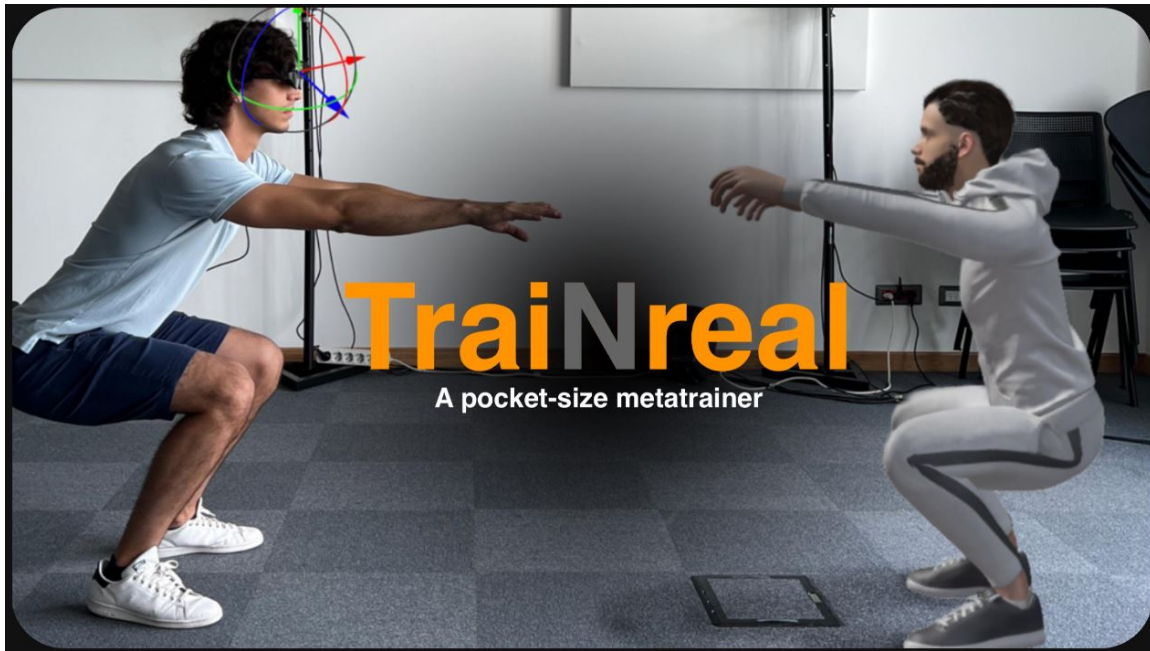


Figure 6.1: The preview of the application Trainreal on the Xreal store

6.1 Meta-Trainer

Meta-Trainer allows people to train on their own, supervised by a virtual trainer who provides exercise samples and real-time feedback. The system relies on a set of smart glasses for AR, to enable users to interact with the virtual trainer in a hybrid environment. Given a target exercise, the virtual trainer demonstrates the movements to be reproduced. Differently from video-based applications, the user can move around the virtual trainer to look at the exercise from different angles for a more immersive and realistic experience.

The AR glasses can also track head movements, which are then used to give feedback and track the training progress. This system offers an advancement compared to video-only trainers, not requiring the user to look at a screen while performing the exercise. This work is motivated by the necessity to inspire individuals to engage in effective training sessions from the comfort of their own homes. Training at home often presents chal-

lenges such as a lack of motivation and potential discomfort resulting from incorrect exercise execution. Moreover, individuals often miss the presence of a gym partner or personal trainer who typically guides workouts. In this context, the concept of having a personal trainer readily available becomes highly advantageous. In this work, I present a novel framework called Meta-Trainer which provides a novel interaction paradigm for training applications interacting by a virtual trainer in the Metaverse.

As we all have learned during the COVID pandemic, training is possible also from home. However, improvising can be ineffective or, even worse, dangerous [68]. This sums up to the fact that doing the workout alone can be alienating, and the motivation often turns to failure in the long term.

Available solutions often involve buying expensive equipment, like treadmills and exercise bikes, to have a complete system that can guide the user through the training while providing live feedback about the performances. The main limitations of these solutions are the initial investment for the hardware, its initial purchase and installation, and the space needed.

To cope with these issues, users usually rely on smartphone-based solutions, which require just a smartphone, where you generally watch and follow along with the training provided by a trainer.

However, such solutions, although very practical and accessible, still suffer some limitations: (1) a detailed demonstration of the movements to be replicated; (2) lack of real-time feedback on the execution performed by the athlete; (3) the necessity to keep looking at the phone while performing exercises [69]. One of the main limitations of smartphones and video-based solutions is the inability to provide feedback to the user about the exercise execution, which can potentially lead to injuries [70]. The scientific literature demonstrates [71] that the head's movement is an important descriptor of the quality of the exercise performed by the whole body. We overcome these limitations by proposing a system that relies on smart

glasses, connected to a smartphone where the main application is running.

In our solution, I propose a virtual personal trainer consisting of a high-quality avatar, whose role is to explain the execution of the workout exercises, provide active feedback on the quality of the athlete's performance, and actively support the user during the activity. The athlete can interact with the avatar through gestures as if it were a training mate. Our goal is to recreate an immersive experience, setting the user at ease and guiding him/her through a comfortable training experience. In this regard, using smart glasses allows the user to interact naturally with the environment, avoiding uncomfortable limitations like having to look at the screen while performing exercises.

To demonstrate the viability of our proposed solution, we focus our attention on three phases of the workout: warm-up, main exercise, and stretching. Following the advice of professional trainers, we have focused our attention on the *air squat*. In this exercise, the neck must maintain a perpendicular alignment with the ground while the gaze keeps pointing forward [71]. The smart glasses can track the trajectory of the head, so we exploit this feature to monitor the correctness of the movements during the squat and turn it into valuable real-time feedback.

The main contribution of our work can be summarised as follows:

- Immersivity of the experience: the personal trainer needs to mimic the behavior of a real human being.
- Multi-view perspective: exploring the three-dimensional environment, the user can learn the exercises by watching the execution from all the points of view.
- Monitoring exercises and live feedback: we utilize head posture information to monitor the exercise's quality.

- Gamification of the application: We provide users with a range of avatar options, including human-like or cartoon-like characters, tapping into the imaginative world of young individuals. Additionally, we offer the exciting prospect of rewarding users based on the quality of their movement during exercise.

6.2 Method

As far as the trainer component is concerned, our goal is to design near-lifelike human avatars that can accurately mimic the subtle movements and gestures commonly observed in a fitness environment. As for the application workflow, we aim to recreate the experience of a typical workout routine that a user might undertake either at the gym or in the comfort of their own home.

6.2.1 The Meta-Trainer avatar

Our study aims to develop an immersive virtual fitness trainer specifically designed for AR settings using smart glasses. The overall quality and realism of this experience rely heavily on the high-fidelity representation of the virtual trainer avatar. While available 3D characters provide satisfactory human body and mesh quality, they are mostly designed for gaming and entertainment purposes. In our scenario, instead, we require the avatar to reproduce accurate animations of gym exercises such as squats or push-ups. This turns out to be way more challenging as it implies the replication of accurate body articulations.

To address this issue, we utilized the precision of the Optitrack motion capture system [44], a cutting-edge tool known for its robust tracking capabilities and accurate data acquisition. The captured motion data is then applied to the chosen mesh, enabling realistic and accurate representations

of fitness exercises. The use of the motion capture system to acquire the template exercise ensures the realistic behavior of the avatar, providing a smooth and non-robotic experience that effectively engages and motivates users.

A key aspect of our project is the ability to visualize the trainer's movements from multiple perspectives within a 3D environment. For instance, the stretching exercises involve complex movements, which are not fully visible from a single viewpoint. The possibility to explore the model from different angles enhances the accurate reproduction of the exercise. Users can potentially interact with the virtual coach using gestures and voice commands, further enhancing the overall experience.

6.2.2 The Application Workflow

The application aims at replicating the experience of a typical workout session, supervised by the virtual trainer. The hand gestures provide maximum freedom and easy interaction with the system. The smart glasses can track hand movements and allow seamless interaction without the need to constantly look at the smartphone screen.

As shown in Figure 6.2, we now break down the implemented architecture, providing the details of the main components of the proposed solution, consisting of a prototype app, and the related workflow:

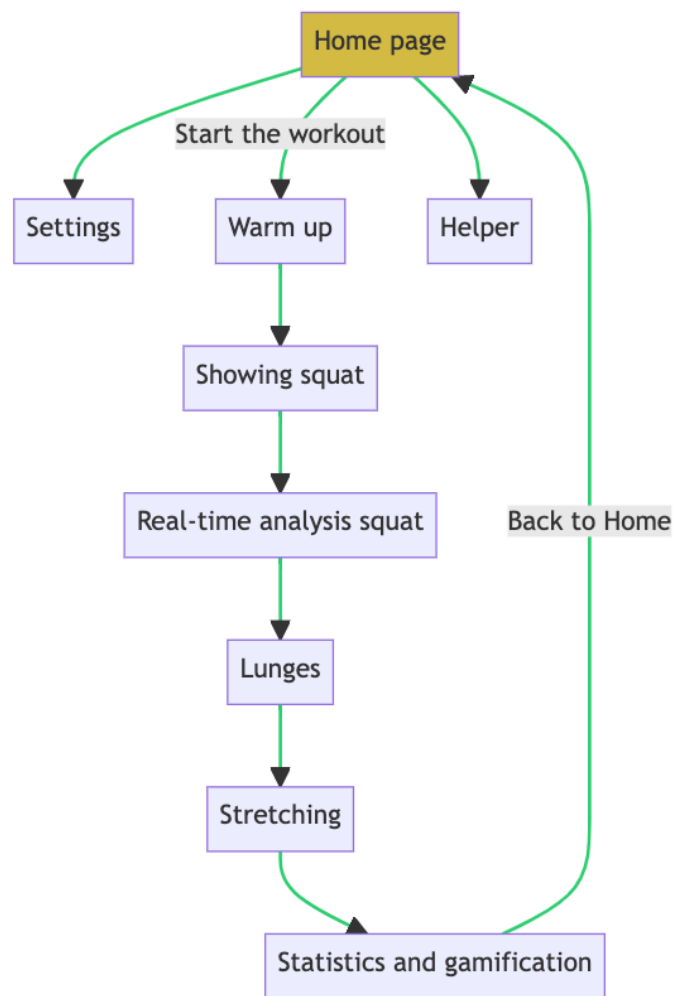


Figure 6.2: Workflow of the application.



(a) Hardware setup of the application. (b) Observe the trainer. (c) Real-time feedback. (d) Movement analysis.

Figure 6.3: The hardware utilized consists of Nreal Light smart glasses, which are paired with a smartphone. With this setup, the user can visually observe the trainer during the exercise (b) and receive real-time feedback (c). This feedback enables the user to make immediate adjustments to their movement, ensuring proper form. The analysis of movement includes tracking the user's head movement (d) as a key factor.

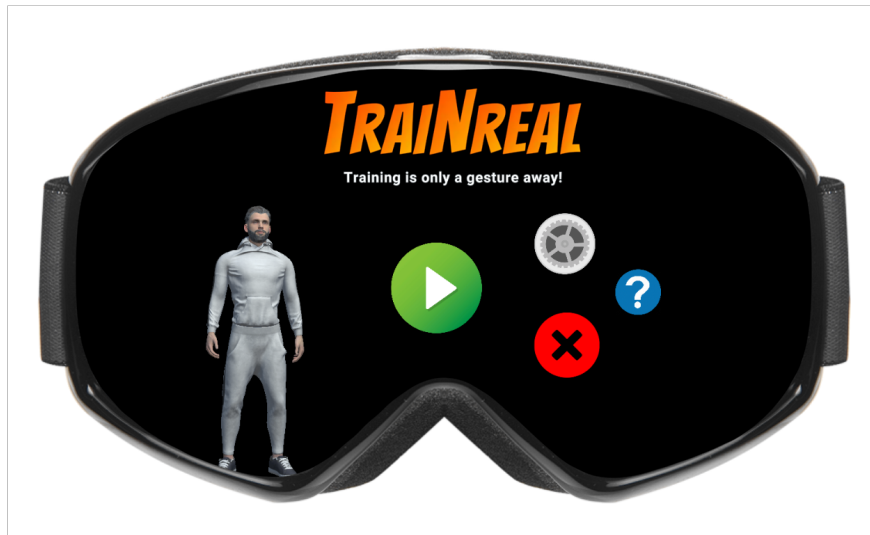


Figure 6.4: Starting interface when wearing the smart glasses.

1. *Initialization*: the home page of the app allows the user to choose the activity to be conducted, including a brief tutorial, as well as the app

settings and initialization of the training session.

2. *Settings*: users can customize the trainer's appearance and input personal details such as their name, weight, and height for a more appropriate training session. Additionally, they can choose their dominant hand for interacting with the app.
3. *Helper*: this section presents users with disclaimers and information about the app. The user can also enable or disable the quick tutorial.
4. *Play*: when the user virtually presses the Play button, the actual workout is started.

6.2.3 Workout Session

1. *Warm Up*: The workout begins with a quick warm-up, including jumping jacks, high skips, and back skips. Then, users can move, stop, or turn around the avatar to get a better view of the exercises;
2. *Squat*: as of now, the main part of the workout includes squats and lunges. At first, the user learns the right way to do a squat by watching the movements performed by the Meta-Trainer. To better understand the movements, users can pause the avatar and closely observe the exercise. Seeing the exercise from different views is helpful for people who are new to working out (Figure 6.3b).
3. *Real-time analysis of the squat*: once the observation phase is complete, the user can start the workout. In this scene, we exploit the integrated sensors of the smart glasses to track users' head movements, which are a meaningful descriptor of the overall performance [71]. Real-time feedback shows the overall performance through graphical representations, as shown in Fig. 6.3d. Users get two graphs as feedback in this scene: one for the top view and one for the front view of

the exercise. These graphs show how well they are performing. For instance, when the user performs a squat, they will see a circle with color scales, and the objective is to keep the position as close as possible to the circle. A black point moving away from the green zone indicates an incorrect execution. The user can then correct their exercise to avoid injuries. After the user completes the squat descent and returns to the initial position, the app displays the vertical trajectory of the head, providing immediate feedback;

4. *Lunges*: after the squat part, we move on to lunges. This part shows the user the right way to do the lunge. As already seen in the previous scenes, the user can stop, observe, and move around the trainer;
5. *Stretching*: the workout experience ends with the stretching session. Watching the trainer from different angles is particularly useful in this scene, since some body parts may be hidden from a single viewpoint.

Statistics and gamification: in this final scene, we provide the user with a summary of their workout, showing them key statistics. When previous workout data are present, we compare their results to see progress.

6.3 Scores and Feedback

The gamification aspect of our application plays a crucial role in enhancing user engagement through the earning of rewards based on the quality of the exercises. To evaluate the quality of the exercise, we track the athlete's head trajectory in real-time, and, using a virtualized plot, we display the position of the head overlapped with the reference position. Scores are calculated based on the accuracy of the movement, promoting the gamification aspects, in the form of digital trophies/badges. To enhance user experience and create an emotional bond between the application and the user, we

also implement realistic gestures of the trainer avatar when the exercise is performed successfully, as shown in Fig 6.5. To provide a comprehensive training experience, we incorporate familiar tools commonly used during workouts. This includes a stopwatch, tracking of calories burned, hydration alerts, and a repeat counter. Additionally, at the end of each session, users can record short clips of their training sessions, and share them on social media. Our application is a reliable tool for athletes at all levels, facilitating self-improvement and offering a gateway to sports and fitness for beginners. However, its use extends beyond domestic training and can be valuable also for professional athletes seeking to improve their exercise posture and self-supervision.

6.4 Results

6.4.1 Hardware Setup

The hardware used for this application consists of the smart glasses Nreal Light, an optical see-through MR device, coupled with a smartphone. When the application is launched on the smart glasses, users can place their smartphones in their pockets and use hand gestures to interact with the application (Fig. 6.3a). The interaction with the application is facilitated through hand gestures, eliminating the need for physical contact with the smartphone. This gesture-based control system enhances user comfort and allows for uninterrupted and immersive workout sessions.

6.4.2 Application

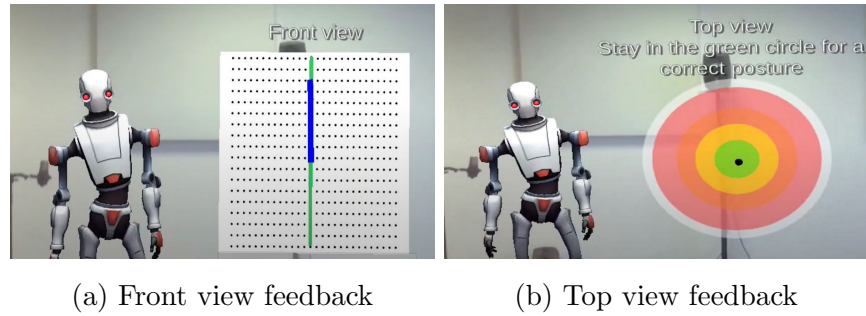
The proposed solution enables users to check their correct posture during activities and fosters an engaging experience between the Meta-Trainer and the user. Upon opening the app, users can select their preferred Meta-

Trainer’s appearance from the settings. The training workflow begins, consisting of two types of exercises. Firstly, the Meta-Trainer demonstrates the exercises and encourages user participation. Secondly, and most importantly, the Meta-Trainer provides feedback on the quality of the exercises. The interaction between the user and the Meta-Trainer is designed to be as human-like as possible, thanks to realistic animations, voice prompts, and movements. The feedback section also includes an evaluation of the activities by monitoring the user’s head postures during the exercises. We have selected the back squat as an exercise to analyze and as is stated in the paper [71], the pose of the head is a fundamental parameter for a correct execution of the exercise. Thanks to the application, we can now monitor the user’s head posture during exercises. We provide two views: a top view and a front view, allowing real-time assessment of the exercise from a head posture perspective. This enables users to instantly check if they are performing the exercise correctly in terms of their head posture. In Figure 6.5, for the front view (Figure 6.5a) the goal is to match the blue line of the user trajectory with the green line which is the correct one. This feedback is presented during the exercise so the user can correct in real-time his posture directly in the next repetition. In Figure 6.5b, we see the top view feedback interface where the black point represents the user position; during the back-squat movement, he/she has to remain within the green circle. When the user completes the exercise, the software elaborates a score indicating the quality of the exercise execution.

We compute the final reward as a score $\mathcal{L}(d)$, analyzing the quality of the squat concerning the target template, as follows:

$$\mathcal{L}(d) = \begin{cases} (\frac{1}{d+0.5} - 1) * 100, & \text{if } d \in [0, 0.5) \\ 0, & \text{if } d \geq 0.5 \end{cases} \quad (6.1)$$

where d is the distance between the center of the circle presented in



(a) Front view feedback

(b) Top view feedback

Figure 6.5: Interface feedback of the back squat

Figure 6.5b and the current position of the user's head. This distance is computed every half second and the values are collected throughout the squat exercise. The final score, which indicates the performance, falls within a range (higher is better). It is calculated by taking the mean of all the data collected during the exercise.

Chapter 7

Conclusion

7.1 Capturing 3D scene

In this thesis, we have proven the effectiveness of the proposed solution to perform the spatial calibration between two devices. The adoption of the MoCap system and an ad-hoc printed 3D bracket in our procedure allowed us to meet our requirements. Furthermore, we have also addressed the temporal inconsistency between the coupled devices, by statistically modeling the delay and building a roll-out filter. Using also a low-pass filter to increase even more the hologram stability in the user experience.

The proposed spatial and temporal calibration procedures need to be performed just once for a specific couple of devices since the results are then reproducible on many other equal setups. The adoption of such a system is validated through many tests for AR purposes, showing improvements in rotation accuracy and in the delays present inside the AR experience. This enables the user to obtain an accurate and real-time overlaying of the holograms on the real world.

To evaluate the accuracy of the proposed solution, a technique to measure the angular error around the altitude axis has been designed and implemented. The results highlight an angular error of 0.449° for the proposed solution, which translates into 1.6 cm of translation error at a distance of

2 meters. Unfortunately, the approaches of other authors do not give such precise metrics, not allowing a direct comparison. Furthermore, the codes of the solutions proposed in chapter 2 are not open-source and rely on ad-hoc hardware. Therefore, it has not been possible for us to apply our error evaluation metrics on their solutions.

As the final part, I tested the applicability of the smart glasses as hardware for the industrial world. I tried to adopt the device in our use cases, developing real prototypes and applications. For example the visualization of the electricity network overlay on the real world in Fig. 4.17.

7.2 3D Reconstruction

I provide a comparative analysis of neural radiance field-based reconstruction methods and classical photogrammetry for unbounded scenarios. I show results in playgrounds and excavations sites, to access the performances in easy and complex scenarios. In our setup, photogrammetry has proven to provide superior reliability in complex scenes, especially on excavation sites. Proving also better results in modeling completely flat areas which in the NeRF methods presents some artifacts. Although training/reconstruction times are generally not the main concern in the reconstruction of working areas, some applications might benefit from fast reconstruction times. In this aspect NeRFstudio provided the best speed in the reconstruction, requiring just 15 minutes for the training of a scene. An important aspect that needs to be analyzed is the reliance of the current rendering pipelines for virtual and augmented reality purposes, which rely on meshes representations. This need is advantage for the classical photogrammetry since its final goal is to obtain a mesh representation. In contrast, neural rendering technologies focus primarily on view synthesis, offering an alternative that eliminates the need for mesh generation.

SuGaR and more in general 3D-GS techniques produce an explicit representation that allows for the splatting of Gaussians in the same way traditional methods splat triangles. This feature enables SuGaR to render the scene in real-time, making it possible to use it in existing pipelines. In the future, I see 3D-GS to be a potential replacement for mesh representations, especially in scenarios requiring the realistic reconstruction of complex environments.

7.3 Meta-Trainer

To enrich the experience of smart glasses I participated with other colleagues in the project Meta-Trainer. The project consists of an AR application that exploits the smart glasses Xreal [11] which allows users to train from home in an immersive environment with real-time feedback on their performances. The application allows for safe and interactive training sessions suitable for newbies and expert athletes. From this experience, I can say that despite the great evolution of the devices in the last years and the big advantages that they bring, like the possibility to work hands-free, the smart glasses are not ready yet. On the other hand, I believe that the transition from smartphones to smart glasses is not that far away, at least in very specific contexts, and I think that solutions like Xreal or Snapdragon ODM, that combine the smart glasses with the smartphone, could be a promising and challenging area that is worth being explored.

7.4 Future research and consideration

As a prospect for the company's research, the idea is to take another step forward. The goal is to understand georeferenced 3D models by leveraging artificial intelligence algorithms to automatically segment and classify the

contents within these models.

This approach represents a significant innovation, as it will enable operators, after collecting the 3D scene, to automatically have all the information classified and integrated into their geographic information system layers on the application's map.

As a final reflection on my Ph.D. experience in industrial innovation, the goal of the program was to bridge the gap between the research world and industry. From my experience, despite the challenges of balancing both activities, the experience has been a success.

I had the opportunity to introduce MR smart glasses into the academic environment, sparking interest among students and within the research group itself. At the University and within the MMLab, I was able to consult with valid researchers on my topic to improve my work and had also the opportunity to use laboratories, especially the Multisensory lab [72].

I also had the opportunity to work with talented students, bringing them into my company for internships and thesis work, giving them the chance to experience the corporate world. A significant achievement was being able to create a final product for the company the 3D Photo Survey [73], thanks in part to the research and relationships developed during my Ph.D. period. In conclusion, despite having a part-time experience at the University, I believe the overall experience was a success.

Bibliography

- [1] Web resources. continuum virtuality reality. https://www.researchgate.net/figure/The-reality-virtuality-continuum-described-by-Milgram-et-al-1_fig1_361590992. Accessed: 2024-07-29.
- [2] Web resources. leicascanner. <https://leica-geosystems.com/it-it/products/laser-scanners/scanners>. Accessed: 2024-07-29.
- [3] Web resources. Iphonedidar. <https://www.howtogeek.com/695823/what-is-lidar-and-how-will-it-work-on-the-iphone/>. Accessed: 2024-07-29.
- [4] Web resources. Drone acquisition. <https://reconaerialmedia.com/professional-drone-company/>. Accessed: 2024-07-29.
- [5] Web resources. Smartphone ar. <https://telset.id/ngehits/6-teknologi-anyar-yang-wajib-ada-di-smartphone-baru/>. Accessed: 2024-07-29.
- [6] Web resources. Rtk scheme. <https://theconstructor.org/surveying/real-time-kinematic-survey/560661/>. Accessed: 2024-07-29.
- [7] Web resources. Arcoda s.r.l. <https://www.arcoda.it/ws/it/index.html>.

-
- [8] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.
 - [9] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.
 - [10] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.
 - [11] Web resources. Xrealair. <https://www.xreal.com/it/air2ultra>. Accessed: 2024-07-29.
 - [12] Web resources. Trainreal. <https://www.xreal.com/it/arlab/detail/?id=13&appname=TrainReal>. Accessed: 2024-07-29.
 - [13] Web resources. blk360. <https://shop.leica-geosystems.com/it-it-IT/leica-blk/blk360/overview>. Accessed: 2024-07-29.
 - [14] Web resources. rtc360. <https://leica-geosystems.com/it-it/products/laser-scanners/scanners/leica-rtc360>. Accessed: 2024-07-29.
 - [15] Web resources. Dronertk. <https://www.dji.com/it/phantom-4-rtk/info>. Accessed: 2024-07-29.
 - [16] Web resources. snapdragonodm. <https://spaces.qualcomm.com/platform-2/>. Accessed: 2024-07-29.
 - [17] Web resources. hololens2. <https://www.microsoft.com/it-it/hololens/hardware#documenta-le-esperienze>. Accessed: 2024-07-29.

- [18] Dinar Sharafutdinov, Mark Griguletskii, Pavel Kopanev, Mikhail Kurenkov, Gonzalo Ferrer, Aleksey Burkov, Aleksei Gonnochenko, and Dzmitry Tsetserukou. Comparison of modern open-source visual slam approaches. *Journal of Intelligent & Robotic Systems*, 107(3):43, 2023.
- [19] E. Stylianidis, E. Valari, K. Smagas, A. Pagani, J. Henriques, A. Garca, E. Jimeno, I. Carrillo, P. Patias, Ch. Georgiadis, A. Kounoudes, and K. Michail. Lara: A location-based and augmented reality assistive system for underground utilities' networks through gnss. In *2016 22nd International Conference on Virtual System & Multimedia (VSMM)*, pages 1–9, 2016.
- [20] E. Valari, A. Pagani, I. Carrillo, A. Kounoudes, K. Michail, and K. Smagas. Augmented reality geovisualisation for underground utilities. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88:173–185, 202.
- [21] P. Patias, V. Tsioukas, C. Pikridas, F. Patonis, and Ch. Georgiadis. Robust pose estimation through visual/gnss mixing. In *2016 22nd International Conference on Virtual System & Multimedia (VSMM)*, pages 1–8, 2016.
- [22] Julien Mercier, Nicolas Chabloz, Gregory Dozot, Olivier Ertz, Erwan Bocher, and Daniel Rappo. Biodivar: A cartographic authoring tool for the visualization of geolocated media in augmented reality. *ISPRS International Journal of Geo-Information*, 12(2), 2023.
- [23] Prashant Ganesh, Kyle Volle, Paul Buzaud, Kevin Brink, and Andrew Willis. Extrinsic calibration of camera and motion capture systems. In *SoutheastCon 2021*, pages 01–08, 2021.

- [24] Gwon Hwan An, Siyeong Lee, Min-Woo Seo, Kugjin Yun, Won-Sik Cheong, and Suk-Ju Kang. Charuco board-based omnidirectional camera calibration method. *Electronics*, 7(12):421, 2018.
- [25] Marco Stranner, Clemens Arth, Dieter Schmalstieg, and Philipp Fleck. A high-precision localization device for outdoor augmented reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 37–41, 2019.
- [26] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.
- [27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.
- [28] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.
- [29] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022.
- [30] Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18049–18060, 2024.

-
- [31] Web resources. Stonex850a. <https://www.stonex.it/it/project/s850a-ricevitore-gnss/>. Accessed: 2024-07-29.
 - [32] Tao Zhang Xiang Gao. *Introduction to Visual SLAM: From Theory to Practice*. Springer Singapore, 2021.
 - [33] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
 - [34] Jihyung Kim, Jonghyeon Ka, Yelin Lee, Younghak Lee, Sangyeon Park, and Wooksung Kim. Mixed reality-based outdoor training system to improve football player performance. In *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–3. IEEE, 2022.
 - [35] Jeffrey Kim, Darren Olsen, and Jack Renfro. Construction workforce training assisted with augmented reality. In *2022 8th International Conference of the Immersive Learning Research Network (iLRN)*, pages 1–6. IEEE, 2022.
 - [36] Lúcia Pombo and Margarida Morais Marques. Marker-based augmented reality application for mobile learning in an urban park: Steps to make it real under the edupark project. In *2017 International symposium on computers in education (SIIE)*, pages 1–5. IEEE, 2017.
 - [37] Jan Čejka, Marino Mangeruga, Fabio Bruno, Dimitrios Skarlatos, and Fotis Liarokapis. Evaluating the potential of augmented reality interfaces for exploring underwater historical sites. *IEEE Access*, 9:45017–45031, 2021.
 - [38] Jorge D Camba and Manuel Contero. From reality to augmented reality: Rapid strategies for developing marker-based ar content using

- image capturing and authoring tools. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–6. IEEE, 2015.
- [39] Jiacheng Shang, Si Chen, Jie Wu, and Shu Yin. Arspy: Breaking location-based multi-player augmented reality application for user location tracking. *IEEE Transactions on Mobile Computing*, 21(2):433–447, 2020.
- [40] Ram Kumar Vankadara, Bodanki Jaya Naga Sarath, Kurapati Sandeep Viswaas, Nalabolu Hari Babu, and Sampad Kumar Panda. Performance analysis of various ionospheric delay corrections in single-frequency gps positioning solution at a low latitude indian location. In *2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–5. IEEE, 2022.
- [41] Alexander Abdulov and Alexander Abramnikov. Research of visual positioning system for mobile robots remote automatic control. In *2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pages 1–6. IEEE, 2020.
- [42] Hriday Bavle, Paloma De La Puente, Jonathan P How, and Pascual Campoy. Vps-slam: Visual planar semantic slam for aerial robotic systems. *IEEE Access*, 8:60704–60718, 2020.
- [43] Jiacheng Liu and Ziyang Meng. Visual slam with drift-free rotation estimation in manhattan world. *IEEE Robotics and Automation Letters*, 5(4):6512–6519, 2020.
- [44] OptiTrack - Motion Capture Systems. <https://optitrack.com/>. Accessed: 2024-05-29.

- [45] Demoz Gebre-egiabher Scott Gleason. *GNSS Applications and Methods*. Artech House, 2009.
- [46] Edward J. Krakiwsky Petr Vaníček. *Geodesy: The Concepts*. Elsevier, 2011.
- [47] Mohinder S Grewal, Lawrence R Weill, and Angus P Andrews. *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. John Wiley & Sons, 2013.
- [48] Elliott D. Kaplan. *Understanding GPS/GNSS: Principles and Applications*. Artech House, 2017.
- [49] Web resources. 3dprinter. <https://3dpadvisor.com/blog/creality-ender-3-v3-review/>. Accessed: 2024-07-29.
- [50] Web resources. blender. <https://www.blender.org/>. Accessed: 2024-07-29.
- [51] Web resources. solidedge. https://www.dex.siemens.com/plm/solid-edge?cclcl=it_IT?stc=usdf201336. Accessed: 2024-07-29.
- [52] Web resources. cura. <https://ultimaker.com/software/ultimaker-cura/>. Accessed: 2024-07-29.
- [53] Web resources. Arcore - official documentation. <https://developers.google.com/ar/develop>.
- [54] Zainab Oufqir, Abdellatif El Abderrahmani, and Khalid Satori. Arkit and arcore in serve to augmented reality. In *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–7. IEEE, 2020.
- [55] Web resources. Motive 3.0. (version 3.0). <https://www.naturalpoint.com/optitrack/products/motive/>.

- [56] Nike Suryawati, Sritrusta Sukaridhoto, Hestiasari Rante, Evianita Dewi Fajrianti, and Oktafian Sultan Hakim. Indoor navigation using augmented reality for pens postgraduate building. In *2023 International Electronics Symposium (IES)*, pages 593–598. IEEE, 2023.
- [57] David Jurado-Rodríguez, Rafael Muñoz-Salinas, Sergio Garrido-Jurado, and Rafael Medina-Carnicer. Design, detection, and tracking of customized fiducial markers. *IEEE Access*, 9:140066–140078, 2021.
- [58] Ayyaj S Maner, Dilip Devasthale, Vikas Sonar, and Rajamani Krishnamurti. Mobile ar system using qr code as marker for ehv substation operation management. In *2018 20th National Power Systems Conference (NPSC)*, pages 1–5. IEEE, 2018.
- [59] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017.
- [60] Web resources. stufvps. <https://sturftee.com/>. Accessed: 2024-07-29.
- [61] Web resources. lightship. <https://lightship.dev/products/vps/>.
- [62] Web resources. Immersalvps. <https://immersal.com/>. Accessed: 2024-07-29.
- [63] Web resources. Geospatial. <https://developers.google.com/ar/develop/geospatial?hl=it>. Accessed: 2024-07-29.

- [64] Lorenzo Orlandi, Kevin Depedri, and Nicola Conci. Spatial-temporal calibration for outdoor location-based augmented reality. *IEEE Sensors Journal*, 2024.
- [65] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023.
- [66] Web resources. Nrealarjam. https://www.reddit.com/r/AR_MR_XR/comments/vdldov/nreal_ar_jam_dev_contest_with_100k_in_prizes/. Accessed: 2024-07-29.
- [67] Lorenzo Orlandi, Giulia Martinelli, Francesco Laiti, Davide Lobba, Niccoló Bisagno, and Nicola Conci. Meta-trainer: An augmented reality trainer for home fitness with real-time feedback. In *2023 IEEE International Workshop on Sport, Technology and Research (STAR)*, pages 90–93. IEEE, 2023.
- [68] Mohitsinh Parmar, Shailesh Khant, and Atul Patel. Covifit mobile application for fitness improvement during third wave of covid19. In *2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM)*, volume 2, pages 253–258. IEEE, 2022.
- [69] Yanxiang Yang and Joerg Koenigstorfer. Determinants of fitness app usage and moderating impacts of education-, motivation-, and gamification-related app features on physical activity intentions: cross-sectional survey study. *Journal of Medical Internet Research*, 23(7):e26063, 2021.

-
- [70] Wesam Saleh A Al Attar and Mohamed A Husain. How did athletes train and avoid injuries during the covid-19 quarantine period? 2021.
- [71] Gregory D Myer, Adam M Kushner, Jensen L Brent, Brad J Schoenfeld, Jason Hugentobler, Rhodri S Lloyd, Al Vermeil, Donald A Chu, Jason Harbin, and Stuart M McGill. The back squat: A proposed assessment of functional deficits and technical factors that limit performance. *Strength and conditioning journal*, 36(6):4, 2014.
- [72] Web resources. 3dphotosurvey. <https://www.cimil.disi.unitn.it/>. Accessed: 2024-07-29.
- [73] Web resources. 3dphotosurvey. <https://pressroom.arcoda.it/3d-photo-survey/>. Accessed: 2024-07-29.