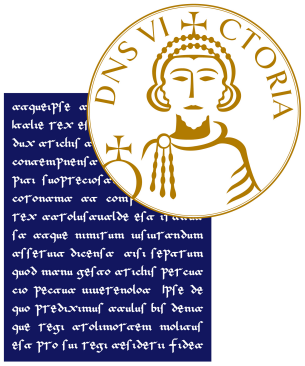
UNIVERSITÀ DEGLI STUDI DEL SANNIO
DEPARTMENT OF ENGINEERING

Ph.D. Course in
Information Technology for Engineering
XXXVII Cycle



Measurement and Sensor Information Processing for Digital Twins

Coordinator:
Prof. **Massimiliano Di Penta**

Supervisor:
Prof. **Luca De Vito**

Co-Supervisor:
Prof. **Francesco Picariello**

Candidate:
Arman Neyestani
Matr: D50030100

Academic year 2024/2025

Contents

Abstract

This thesis investigates how advanced measurement methodologies and sensor information processing can elevate the fidelity of digital twin representations in both structural and environmental domains. By integrating Unmanned Aerial Vehicles (UAVs) and underwater vehicles (UUVs) as data-collection platforms, the dissertation focuses on three core challenges: (1) accurate navigation in GNSS-denied settings via monocular and visual–inertial odometry (VIO), (2) automated defect detection in civil infrastructures using deep learning, and (3) scalable marker-based tracking for 3D-scaled masonry models.

First, the work provides an in-depth analysis of monocular Visual Odometry (VO), comparing classical geometry-based approaches with modern deep-learning methods. Emphasis is placed on how feature selection, sensor fusion, and scale estimation influence drift and reliability. A new measurement-uncertainty approach quantifies error propagation in VO pipelines, linking theoretical measurement principles to UAV and underwater applications. In over 420 simulated UAV flights, selective algorithms (e.g., Harris and AKAZE) reduce drift by up to 42% over baseline techniques. Sensitivity analyses further show that refining environmental and sensor factors—such as IMU noise or ground-plane ambiguities—can achieve an additional 30% reduction in positional uncertainty.

Second, the thesis develops deep learning pipelines for defect segmentation and crack verification. A YOLO-based crack detection model achieves F1 scores above 92% on a large-scale image dataset, enabling rapid UAV inspections of civil structures. Complementary triplet-loss verification further refines crack evolution monitoring, cutting false positives by 35% when tracking fracture propagation over time.

Third, a low-cost marker-based system is introduced for scaled masonry arches and structural prototypes, where 19 fiduciary markers continuously monitor block displacements. Experimental tests show orientation measurement expanded uncertainty down to $\pm 0.29°$, nearly half that of earlier marker-based systems. Finally,

the research extends digital twin applications to marine environments, combining a Gated Recurrent Unit model with over 20,000 wave observations. Wave-height forecasts exhibit a Pearson correlation up to 0.95, reducing predictive errors by 30% relative to baseline models.

Collectively, these contributions demonstrate how rigorous measurement modeling, sensor data fusion, and deep learning–assisted defect detection can significantly enhance the reliability of digital twins. By quantifying uncertainties, integrating diverse sensing strategies, and validating on real-world platforms, this thesis lays a robust groundwork for next-generation, high-precision digital twin systems.

Chapter 1

Introduction and Theoretical Foundations

In the era of digital transformation, the concept of digital twin has emerged as a revolutionary paradigm, enabling monitoring, simulation, and analysis of physical systems through their digital counterparts. A digital twin is a virtual representation of a physical object, system, or process that integrates data with advanced models and analytics to mimic its physical counterpart's behavior and state [1]. These digital twin serve as dynamic and intelligent systems that continuously evolve alongside their physical counterparts, offering unprecedented insights and decision-making capabilities. By connecting the physical and virtual realms through a seamless flow of data, digital twin enable predictive, prescriptive, and descriptive analyses that facilitate more effective management and optimization of systems [2]. This dissertation, titled "Measurement and Sensor Information Processing for digital twin," investigates the pivotal role of measurement technologies and sensor information processing in advancing digital twin applications. By integrating concepts from structural health monitoring (SHM), unmanned aerial vehicles (UAVs) or Unmanned Underwater Vehicles (UUVs), deep learning, and visual localization, this work provides a cohesive framework for leveraging sensor data to enhance the fidelity, reliability, and functionality of digital twin.

Digital twin rely on high-fidelity data to replicate physical systems in a virtual environment. This work develop some research by integrating multiple technological advancements to enhance digital twin accuracy, robustness, and applicability.

SHM serves as the foundation by enabling continuous assessment of civil and marine infrastructure, ensuring that the digital twin accurately reflects structural

Figure 1.1: In this AI-generated scenario picture, a UAV and ground robot collaborate to inspect a bridge, integrating real-time sensor fusion and AI-driven analysis. The digital twin updates dynamically, enabling predictive maintenance and automated anomaly detection for infrastructure resilience.

conditions (as shown as scenario in Figure 1.1). UAVs/UUVs play a crucial role in this process by providing automated, high-resolution data collection in areas that are difficult to access, such as bridges, dams, and underwater structures.

To efficiently process and interpret the vast amounts of sensor data acquired from UAVs/UUVs , deep learning techniques are applied for automated damage detection, crack segmentation, and anomaly classification. These machine learning models improve defect recognition, reducing manual inspection errors and increasing detection precision.

Additionally, visual localization techniques, including monocular VO and sensor fusion, enhance UAV/UUV navigation in GNSS-denied environments, such as tunnels and underwater structures. By ensuring precise UAV/UUV positioning, these techniques enable accurate spatial mapping of infrastructure within the digital twin [3]. As the backbone of digital twin technology, sensor-based data acquisition and processing are indispensable for ensuring the accurate representation of the physical system in its digital counterpart [4].

1 Integrating Measurement and UAV/UUV Data for Digital Twins

The advancement of digital twin relies on a structured integration of data acquisition, processing, and interpretation. This framework facilitates precise and dynamic representations of physical systems, with a primary focus on structural health monitoring and environmental analysis, leveraging robotics as a key enabling technology. To introduce the research component clearly, it is necessary to define its role and objectives. Each research component forms a crucial link in a structured chain, transforming raw data into actionable insights. This section outlines how interconnected methodologies, which will be mentioned, collectively enhance the reliability and applicability of digital twin models.

1.1 Data Acquisition and Sensor Integration

The foundation of digital twins begins with precise measurement and data acquisition. In structural health monitoring, sensors such as RGB cameras, LiDAR, and inertial measurement units (IMUs) are deployed on UAVs, mobile robots, and other autonomous systems to be captured in the real world. However, the quality and usability of this data depend on the accuracy and reliability of these sensors. Measurement uncertainty, environmental conditions, and sensor calibration play a significant role in determining the fidelity of the acquired data.

For example, monocular visual odometry (VO) is used to estimate the movement and position of UAVs and mobile robots in environments where GNSS signals are unreliable. However, Evaluating the uncertainties is essential, as they contribute directly to the reliability of geometrical measurements for structural monitoring. To bridge the gap between localization and its application in structural monitoring, position estimates must be mapped onto the monitored structures while accounting for these uncertainties. This requires sensor fusion techniques incorporating IMUs and altimeters, which enhance trajectory estimation and ensure the positional data accurately reflects the structural environment in digital twin models.

Another critical aspect of data acquisition is anomaly detection. When UAVs and mobile robots perform structural inspections, deep learning models process high-resolution images to detect cracks and other defects. Optimizing data acquisition entails improving measurement accuracy, minimizing noise, selecting appropriate sensors, and ensuring that data is contextually relevant for structural monitoring and

localization tasks. This process enhances the quality and reliability of inputs used in subsequent computational processes, ultimately improving digital twin fidelity and usability.

1.2 Data Processing and Feature Extraction

Once raw data is collected, it must be processed into a usable format. Feature extraction is the next link in the chain, where relevant information is isolated from sensor outputs. In monocular visual odometry, this involves detecting and matching keypoints between frames. Various algorithms such as ORB and AKAZE are evaluated to determine which provides the most stable feature tracking under varying lighting and environmental conditions.

In structural monitoring, deep learning-based segmentation techniques convert UAV- and robot-acquired images into meaningful data points. These methods, such as YOLO-based crack segmentation, are trained to recognize and classify damage patterns. The efficiency of these techniques is highly dependent on the quality of extracted features. Poor feature matching in odometry or suboptimal segmentation in structural analysis can introduce significant errors, leading to unreliable digital twin representations.

Another key component in data processing is uncertainty quantification. All sensor measurements contain inherent noise, and these uncertainties propagate through processing pipelines. To address this, uncertainty models are developed to quantify confidence in localization results.

1.3 Digital Twin Representation and Predictive Analytics

After feature extraction and uncertainty assessment, the processed data must be integrated into a dynamic digital twin model. Digital twins require continuous updates, meaning data must not only be captured and processed but also seamlessly fused into an evolving representation of the physical structure.

For example, in maritime applications, significant wave height predictions are incorporated into digital twin models for ocean monitoring. These predictions rely on recurrent neural networks trained on historical and real-time wave data. Similarly, in UAV and robot navigation, continuously updated pose estimations from visual-inertial odometry contribute to real-time localization accuracy.

In structural health monitoring, predictive analytics plays a crucial role. Deep learning models track the evolution of detected cracks over time, analyzing changes

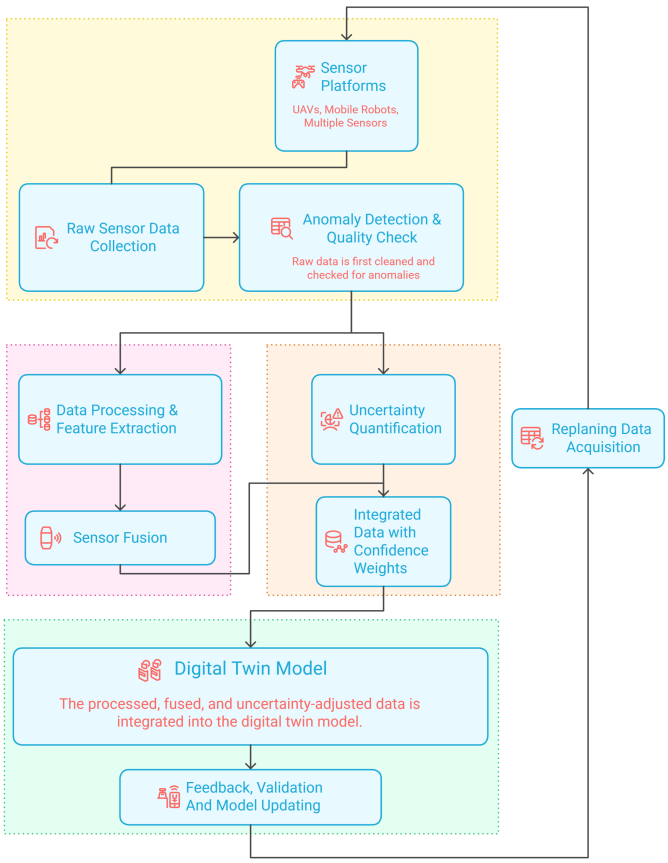


Figure 1.2: Digital Twin Data Integration Pipeline: A streamlined flow from multi-sensor acquisition and feature extraction through sensor fusion and uncertainty quantification, leading to a continuously updated digital twin with predictive analytics and calibration feedback.

in width, length, and propagation patterns. By integrating these evolving measurements into a digital twin, engineers can predict potential failure points and proactively schedule maintenance interventions.

Additionally, marker-based tracking systems provide an alternative approach to monitoring small-scale structures. By using optical markers and camera-based pose estimation, precise displacement measurements are captured. These datasets contribute to the calibration and validation of large-scale structural monitoring frameworks.

A complete digital twin system relies on multiple sources of data to enhance measurement reliability and structural assessment. By continuously refining data acquisition strategies, the system adapts to identified anomalies and data gaps,

ensuring comprehensive data collection.

For example, visual-inertial odometry improves UAV and mobile robot navigation, providing crucial localization data for accurate structural monitoring. Likewise, in marine environments, optical imagery aids in underwater mapping, contributing to a more detailed representation of submerged structures. Ultimately, the effectiveness of a digital twin hinges on the seamless integration, As shown in Fiiure. 1.2, each element—data acquisition, feature extraction, uncertainty modeling, and predictive analytics—forms a necessary part of the broader system. The transition from raw sensor readings to actionable insights in a digital twin environment is not a single-step process but rather a continuous feedback loop where each layer refines and enhances the next. This structured approach ensures that digital twins remain not just static models but dynamic, evolving representations capable of real-time decision-making and long-term predictive analysis.

To provide a context, this dissertation proposes technical solutions to specific challenges, structured according to the components presented in Figure 1.2:

- **Visual Localization and Odometry:** Ensuring reliable localization and navigation in GNSS-denied environments is a challenge for UAV- and UUV-based infrastructure monitoring. In environments such as tunnels, enclosed industrial sites, offshore platforms, and underwater structures, traditional satellite-based positioning is unreliable, necessitating alternative localization strategies. This research focuses on the development and evaluation of monocular visual odometry (VO) and visual-inertial odometry (VIO) techniques to improve localization accuracy for autonomous inspection systems operating within a digital twin framework, where precise localization plays a crucial role in ensuring the accurate representation of monitored structures. Localization techniques, particularly visual odometry, are essential for maintaining spatial consistency, integrating real-world positional data, and reducing uncertainty in digital twin applications. A primary objective of this work is to quantify and mitigate the measurement uncertainties associated with monocular VO. While VO provides an effective means of estimating position and orientation by analyzing sequential images, it suffers from scale ambiguity, feature detection inconsistencies, and drift accumulation over time. This study systematically assesses feature extraction methodologies (e.g., ORB, AKAZE) and sensor fusion approaches, determining their impact on VO accuracy in UAV/UUV-based inspections. The integration of inertial measurements (IMUs) and altimeters is explored to

enhance VIO robustness, reducing cumulative drift and improving long-term localization reliability. Furthermore, this research introduces a measurement uncertainty model tailored for UAV/UUV localization within digital twin representations. Unlike conventional VO studies that focus solely on localization performance, this work assesses the confidence levels of VO-derived positional estimates, ensuring that data integrated into the digital twin meets metrological standards. This model enables the quantification of localization error propagation, providing a systematic way to enhance the spatial fidelity of digital twin applications. Additionally, this study extends VO/VIO methodologies to mobile robotic platforms and underwater navigation, investigating their adaptability to different operational contexts.

- **UAVs/UUVs and Deep Learning for Civil Structural Monitoring:** Traditional infrastructure monitoring methods face significant limitations, including high costs, inefficiencies, and the risk to human operators. To address these challenges, this research integrates UAVs and UUVs as autonomous sensing platforms within a digital twin framework, focusing on their role in real-time structural health monitoring. Unlike generic digital twin applications, this study emphasizes high-fidelity data acquisition, defect classification, and uncertainty quantification, ensuring that UAV/UUV-based observations contribute reliably to the evolving digital twin model.A key contribution of this work is the development and validation of deep learning-driven defect detection - practically cracks on concrete - techniques that operate efficiently in UAV/UUV-acquired imagery. Using advanced segmentation models like YOLO and triplet loss-based learning, this research enhances the ability to detect, classify, and track structural defects over time, minimizing false detections while improving localization accuracy. The integration of monocular visual odometry and sensor fusion further refines the position estimation of UAV/UUV-acquired data in GNSS-denied environments, a critical aspect that enhances the spatial consistency of the digital twin model.
- **Marker-Based Tracking for Structural Monitoring:** Accurate and cost-effective tracking systems are essential for creating and maintaining digital twin of structural systems. A digital twin relies on continuous measurement data to provide insights into structural stability, defect progression, and predictive maintenance. However, assessing the stability of buildings and infrastructure often requires extensive measurements, which are used by specialists to analyze

structural integrity and identify the causes of cracks or other anomalies. In real-world applications, obtaining sufficient data for these assessments is challenging due to limited access, high costs, and the complexity of structural behavior. To overcome these challenges, AI-driven analysis techniques are increasingly explored to assist specialists in diagnosing structural issues. However, AI models require large datasets for training, which are difficult to obtain from real buildings due to practical constraints. As a solution, researchers commonly use 3D-scaled models to generate controlled datasets for AI training and validation. Structural tracking plays a critical role in the digital twin update process, ensuring that changes in monitored structures are accurately represented.

Traditional measurement systems for structural tracking often rely on expensive equipment, limiting accessibility and scalability. To address this limitation, this dissertation proposes a low-cost marker-based tracking system that uses DeepTag and ArUco markers for accurate and scalable structural monitoring. These marker-based tracking techniques provide a cost-effective alternative for tracking structural changes, enabling efficient data collection while maintaining measurement accuracy. Scaled models serve as a test case for evaluating the developed methods, demonstrating their applicability in real-world monitoring scenarios. The proposed system focuses on structural tracking for digital twin updates, ensuring accurate representation of monitored structures. However, deploying marker-based tracking in large-scale infrastructure poses additional challenges, particularly regarding the strategic placement of markers at significant monitoring points to ensure effective data acquisition. By addressing these scalability challenges, marker-based tracking can become a viable solution for enhancing digital twin models, improving predictive modeling, and supporting SHM.

- **Marine Environmental Analysis:** Digital twin technology is essential in environmental monitoring by integrating real-time data and predictive models to support decision-making in marine environments. The primary objective of this research is to develop a data-driven digital twin framework that enhances the monitoring and forecasting of critical oceanographic parameters, particularly Significant Wave Height (SWH), vital to maritime operations, coastal infrastructure, and offshore safety. To achieve this, machine learning-based predictive modeling is employed, leveraging historical and real-time sensor data collected from various maritime observatories, including the EMSO-OBSEA observa-

tory, Tarragona, and Barcelona monitoring stations. The core of this approach involves using Gated Recurrent Unit (GRU) neural networks, which are well-suited for time-series forecasting, to predict wave height variations and identify potential anomalies in ocean conditions. For instance, in maritime digital twin applications, forecasting Significant Wave Height (SWH) is critical for ship routing optimization, coastal erosion analysis, and offshore energy planning. The developed framework processes real-time in-situ measurements alongside historical datasets to generate accurate SWH predictions, thus improving maritime situational awareness and operational resilience.

The research follows a hierarchical and interconnected approach, ensuring that sensor-based measurements, data processing, and predictive models contribute cohesively to a robust digital twin framework. The overarching goal is to bridge the gap between theoretical advancements and real-world implementation, making digital twins metrologically reliable, computationally efficient, and adaptable to diverse structural and environmental monitoring applications.

By structuring these research objectives within a unified measurement-driven vision, this dissertation contributes to the evolution of digital twins from static digital models to real-world, self-updating intelligent systems capable of predictive analysis, decision-making, and structural health assessment with quantified accuracy and uncertainty.

2 Structure of the Dissertation

This dissertation is structured to progressively build upon the foundational concepts and methods, culminating in a synthesis of findings and recommendations for future research. Each chapter focuses on a specific aspect of the research, detailing both theoretical foundations and practical implementations:

- Chapter 2: **Visual Localization and Odometry** - This chapter investigates monocular VO and VIO techniques for UAV/UUV navigation in GNSS-denied environments. It addresses key challenges such as **scale estimation, feature tracking, and drift compensation**, integrating **sensor fusion with IMUs, LiDAR, and depth cameras** to improve localization accuracy. Additionally, it presents an **uncertainty modeling framework** to quantify localization errors and evaluates the system's performance in real-world UAV/UUV navigation scenarios.

- Chapter 3: **UAVs/UUVs and Deep Learning for Structural Monitoring** - This chapter explores the use of UAVs and UUVs equipped with high-resolution imaging sensors for SHM. It introduces deep learning-based methodologies, including **crack detection using YOLO segmentation models, triplet loss-based verification for structural defects**, and transfer learning approaches for automated infrastructure assessment. Additionally, it discusses the integration of UAVs/UUVs with Internet of Things (IoT) technologies for real-time monitoring applications.

- Chapter 4: **Marker-Based Tracking for Structural Monitoring** - This chapter details the development and implementation of **low-cost marker-based tracking systems** for monitoring structural displacements and deformations. It presents the use of **DeepTag and ArUco markers** for high-precision tracking in 3D-scaled masonry models, highlighting their potential for SHM in both laboratory and real-world environments. The chapter also discusses the challenges of marker placement in large-scale infrastructure and proposes solutions for future deployment.

- Chapter 5: **Environmental Monitoring for Marine Digital Twins** - This chapter explores the integration of digital twin technology in marine environments. It presents a predictive modeling framework for forecasting significant wave height, using **machine learning and in-situ sensor data** from various maritime observatories. The chapter discusses data preprocessing, model training, and validation methodologies while also addressing challenges such as data gaps, outlier detection, and environmental variability in digital twin applications for marine monitoring.

- Chapter 6: **Conclusion and Future Directions** - This chapter synthesizes the key findings of the dissertation, discussing the impact of integrating deep learning, marker-based tracking, and visual localization within the digital twin framework. It outlines future research directions, including **advancements in real-time data processing, multi-sensor fusion, and the application of digital twin in environmental monitoring and smart infrastructure development**.

By addressing the multifaceted challenges of measurement and sensor information processing, this dissertation contributes to advancing digital twin technologies,

ultimately fostering safer, more efficient, and adaptive systems across diverse applications. It sets the stage for exploring new dimensions in digital twin evolution, encouraging a continuous cycle of innovation and application refinement.

Chapter 2

Advances in Monocular Visual Odometry

This chapter outlines the advances and challenges in monocular VO, highlighting how various works have progressively tackled measurement accuracy, robustness, and scalability issues-particularly for digital twin applications.

In "Survey and Research Challenges in Monocular Visual Odometry" [5], offers a comprehensive taxonomy of both classical and emerging (deep learning–based) monocular VO techniques. Beyond identifying commonly known issues—such as scale ambiguity and ground plane limitations-it pinpoints new research gaps in handling non-static scenes, integrating additional sensing (e.g., IMUs), and applying deep learning for feature detection. Crucially, it emphasizes how reduced hardware dependencies (via software-driven or learned methods) could facilitate more accessible, cost-effective VO systems.

In "From Pixels to Precision: A Survey of Monocular Visual Odometry in Digital Twin Applications" [6], extends the survey perspective to digital twin–specific scenarios, dissecting feature tracking and scale estimation challenges when integrating VO into structural monitoring or industrial digital twins. The paper highlights how LiDAR or depth-camera data can be fused with monocular VO to strengthen metric fidelity, ensuring high-precision updates in digital twins. Its main contribution is a measurement-centric viewpoint, showing how improved VO accuracy directly enhances the reliability of digital twin models.

In "Measurement Uncertainty Model for Relative Visual Localization of UAV by a Monocular Camera" [7], proposes the first uncertainty model tailored for monocular VO in UAV navigation, bridging theoretical error propagation (via the "Guide to

the Expression of Uncertainty in Measurement") with practical flight tests. By comparing multiple feature detection algorithms, it quantifies how algorithmic choices affect positional uncertainty, thus linking measurement theory to real-world UAV operations.

In "Sensitivity Analysis of a Visual Inertial Odometry Based Navigation System for UAV" (to present at IEEE I2MTC 2025), delivers an in-depth analysis of how environmental factors (e.g., lighting, scene texture) and sensor uncertainties (IMU noise, camera calibration) shape the performance of visual-inertial odometry (VIO). Validated by extensive simulations, it ranks which parameters most significantly degrade (or improve) VIO accuracy, offering practical guidelines for UAV system designers aiming to enhance navigation reliability.

In "SUBVO Dataset: Analyzing Feature Extraction for Underwater Monocular Visual Odometry" (to present at IEEE I2MTC 2025), introduces a novel dataset specifically for underwater VO, a domain often overlooked due to poor visibility and color distortion. By applying advanced preprocessing (e.g., white balancing, color cast reduction) and a genetic algorithm–based optimization for RANSAC's inlier threshold, this work demonstrates robust feature matching despite underwater turbidity and significantly reduces pose estimation errors in underwater robotic navigation.

Collectively, these five investigations chart the evolution of monocular VO from foundational reviews and uncertainty modeling to specialized domains like underwater robotics. Each work addresses a distinct set of technical barriers —ranging from algorithmic scale recovery to the incorporation of deep learning, sensor fusion, and specialized datasets—showing how VO continues to mature into a crucial enabler for intelligent and adaptive systems, including those underpinning digital twins.

2.1 Overview of digital twins and monocular VO

Exploring unknown environments is a complex challenge that has engaged researchers across various fields. The intricacies of navigating in uncharted territories require the integration of multiple approaches and the development of sophisticated methodologies. Among these, the measurement of accurate camera movements to update the digital twin model of structures plays a significant role, which will be briefly explained in Subection 2.1.1. Modern navigation systems are often multimodal, merging information collected from various methods to achieve enhanced

precision. Within this complex interplay, Visual Simultaneous Localization and Mapping (VSLAM) has emerged as a vital tool in computer vision, robotics, and augmented reality.

VSLAM represents an innovative approach to navigation, addressing the inherent drift problem through the intelligent combination of camera information with an environment map. This map, updated incrementally as an agent such as a robot that moves through the environment, facilitates the accurate and real-time estimation of the surroundings. The significance of this technology is further underscored by its reliance on the accuracy of geometrical measurements, which are pivotal to the localization system. This mechanism aids in the consistent update of models, often evaluated through the periodic acquisitions of camera images, identification of model elements, and assessment of changes over time.

The intricate design of modern navigation systems is underscored by their reliance on the integration of various methods, a process akin to data fusion. This integration involves merging information from different sources to achieve greater accuracy. The role of VSLAM is particularly significant in this framework. Employed in fields like computer vision, robotics, and augmented reality, VSLAM goes beyond merely combining camera visuals with environmental layouts. Its true value emerges in the continuous refinement and updating of data, enabling robots or agents to adeptly navigate through the ever-changing and unpredictable terrains of unfamiliar settings [8].

VSLAM's capabilities are broadened through the use of one or more video cameras to reconstruct a 3D map of an often unknowable environment [9] and to gauge the egomotion-defined as the 3D shifting within space-of the camera itself [10]. The video cameras used in VSLAM systems are essential for applications, like markerless augmented reality and autonomous robotic navigation. When compared with general SLAM that uses sensors like Light Detection and Ranging (LiDAR), VSLAM's reliance on video cameras brings added advantages [11]. Video cameras are often smaller, less expensive, and carry rich visual information, making them suitable for platforms with limited payloads and lower costs than LiDar or an RGB-D camera [12, 13].

VO and VSLAM are two closely related techniques that are used to determine a robot or machine's location and orientation through the analysis of corresponding camera images. Both techniques can utilize a monocular camera, but they have distinct characteristics and objectives [14, 15, 16].

VO is a technique primarily focused on the real-time tracking of a camera's trajectory, offering local or relative estimates of the position and orientation. This process is a part of a broader category known as relative visual localization (RVL). RVL encompasses methods like VO, which estimate the motion of robots (both rotation and translation) by localizing themselves within an environment. This localization is achieved by analyzing the differences between sequential frames captured by the camera. One of the key techniques used in VO is Windowed optimization. Windowed optimization is a process that refines the local estimation of the camera trajectory by considering a certain number of previous frames or 'window' of frames. This approach helps to improve the accuracy of pose predictions derived from the analysis of image sequences [17, 18].

On the other hand, VSLAM delivers a global and consistent estimate of the path of a device, a process often referred to as absolute visual localization (AVL). AVL provides a pose of a vehicle that is often represented by a six-degrees-of-freedom (DoFs) pose vector $(x, y, z, \varphi, \theta, \psi)$ [14, 19]. VSLAM has the ability to reduce drift through techniques, like adjusting the bundle [20] and detecting loop closure [21]. The key difference is that VO is about relative positioning without an understanding of the larger environment, while VSLAM involves both mapping the environment and locating the device within that map.

Loop closure is a sub-algorithm of SLAM that identifies previously visited locations and uses them to correct the accumulated errors in the robot's pose estimation [22]. The main goal in loop closure is to detect when the robot is observing a previously explored scene so that additional constraints can be added to the map [23]. This is crucial in ensuring the consistency of the map and the accuracy of the robot's location. The similarities between VO and VSLAM persist until a loop is closed, after which their functions diverge [10, 24, 25].

Furthermore, VSLAM's capacity to continuously update the initial map of the environment based on sensor measurements contributes to its adaptability, enabling it to reflect changes, such as new objects or variations in lighting conditions. This makes VSLAM a more comprehensive solution for mapping and localization tasks in dynamic environments. Monocular VO represents an essential component in the field of robotic navigation and computer vision, enabling the real-time estimation of a camera's trajectory within an environment [25, 26, 27].

From a practical standpoint, VO has a wide range of applications. It is used in mobile robots, self-driving cars, unmanned aerial vehicles, and other autonomous

systems to provide robust navigation and obstacle avoidance capabilities [28, 29].

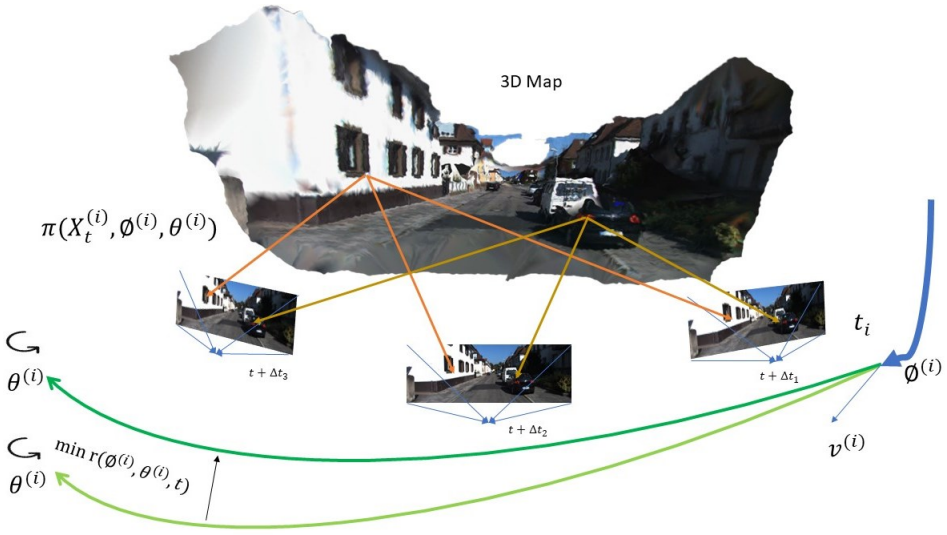


Figure 2.1: The diagram depicts the camera maneuvers used to update the digital twin representation. The starting position $v(t_i)$ of the present path segment is depicted by a blue arrow at the moment t_i. The position for each snapshot is calculated using the parameters θ_t. Subsequently, the map points are reprojected onto every snapshot, and the reprojection discrepancy $r(\Phi(i), \theta_t, t)$ is reduced to ascertain the accurate path [30].

2.1.1 Visual Odometry for Digital Twin

The accurate measurement of camera movements is crucial for updating the digital twin model of structures. This process involves the use of VO and other techniques to capture and analyze camera images, which are then used to update the digital twin model (Figure 2.1). One method employed to achieve this involves multi-camera systems. Research examining Blender's application in designing camera-based measurement systems revealed that it allows for the flexible and rapid modeling of camera positions for motion tracking, which helps determine their optimal placements. This approach significantly cuts down setup times in practical scenarios. The methodologies focus on building an entire virtual camera, encompassing everything from the original camera sensor to the radiometric characteristic of an actual camera [31]. The study focuses on developing virtual representations of multi-camera measurement systems using Blender. It investigates whether these virtual cameras in Blender can perceive and measure objects as effectively as real cameras in similar conditions. Blender, an open-source software for three-dimensional animation, also serves as a simulation tool in metrology. It allows for the creation

of numerical models instrumental in the design and enhancement of camera-based measurement systems.

In a separate study, the Digital Twin Tracking Dataset (DTTD) was introduced for Extended-Range Object Tracking. This dataset, comprising scenes captured by a single RGB-D camera tracked by a motion capture system, is tailored to pose estimation challenges in digital twin applications [32].

Regarding geometric change detection in digital twins, an object's pose is estimated from its image and 3D shape data. This technique is crucial for pose estimation [33, 34]. Likewise, for the digital twin modeling of composite structures, the Azure Kinect camera is utilized to capture both depth and texture information [35]. Drone inspection imagery is instrumental in forming operational digital twins for large structures, enabling the creation and updating of digital twin models based on high-quality drone-captured images [36]. In summary, the precise measurement of camera movements is key in updating digital twin models of structures. Techniques like monocular VO, multi-camera measurement, and drone imagery contribute significantly to producing detailed and accurate digital twin models.

2.1.2 Basics of Monocular Visual Odometry

From a theoretical perspective, VO is a complex problem that involves the intersection of multiple disciplines, including computer vision, robotics, and mathematics. It requires the development and application of algorithms that can accurately track visual features and estimate camera motion from a sequence of images [28]. This involves dealing with challenges such as scale ambiguity in monocular systems, where the trajectory of a monocular camera can only be recovered up to an unknown scale factor [29]. Theoretical advancements in VO can contribute to a deeper understanding of these challenges and the development of more effective solutions.

The foundational algorithm of VO, commencing after compensating for camera distortion based on parameters estimated during a calibration phase, can be conceptually divided into several sequential steps, each of which contributes to the overarching objective of motion and trajectory estimation:

1. **Feature detection** : In the initial phase of VO, the focus is on identifying and capturing key visual features from the first camera frame, which are essential for tracking movements across frames. This process, fundamental for the accurate monitoring of camera movement, traditionally relies on algorithms like Harris, SIFT, ORB, and BRISK to pinpoint precise and durable features, such as

corners or edges[37]. However, it is crucial to expand beyond these to include line and planar features, which have proven to be invaluable in enhancing the robustness and completeness of feature detection and matching in monocular VO systems. These additions are essential for capturing the full complexity and variety of real-world environments [38, 39, 40, 41].

2. **Feature tracking**: Following feature detection, the VO algorithm focuses on tracking these identified features across consecutive frames. This tracking establishes correspondences between features in successive frames, creating a continuity that facilitates motion analysis. Techniques such as KLT (Kanade–Lucas–Tomasi) tracking or optical flow have proven effective in this context, enabling accurate alignment and correspondence mapping [42].

3. **Motion estimation**: With the correspondences between features in consecutive frames established, the next task is to estimate the camera's motion. This process involves mathematical techniques, such as determining the essential matrix or, if needed, the fundamental matrix. These methods leverage the correspondences to ascertain the relative motion between frames, providing a snapshot of how the camera's position changes over time [43].

4. **Triangulation**: Based on the estimated camera motion, the algorithm then moves to determine the 3D positions of the tracked features by triangulation. This technique involves estimating the spatial location of a point by measuring angles from two or more distinct viewpoints. The result is a three-dimensional mapping of features that adds depth and context to the analysis [44].

5. **Trajectory estimation**: The final step in the basic VO algorithm involves synthesizing the previously gathered information to estimate the camera's overall trajectory within the environment and map the surroundings. This composite task draws upon both the estimated camera motion from step (iii) and the 3D positioning of the tracked features from step (iv). Together, these elements coalesce into a coherent picture of the camera's path, contributing to a broader understanding of the spatial context [45].

In summary, the basic algorithm for VO is a multi-step process that artfully combines detection, tracking, estimation trajectory, and triangulation to provide a nuanced understanding of camera motion within an unknown environment. By progressing through these distinct yet interrelated phases, VO offers a versatile and

valuable tool in the quest to navigate and interpret complex spatial environments. Its contributions extend across various domains, and its underlying methodologies continue to stimulate research and innovation in both theoretical and applied contexts.

2.1.3 Research Challenges in Monocular Visual Odometry

Monocular VO represents a sophisticated domain characterized by exceptional achievements and compelling intricacy. The sources of uncertainty can significantly affect the accuracy and reliability of positioning and navigation solutions provided by VO systems. The advancements achieved in this field have substantially contributed to the evolution of robotics, augmented reality, and navigation systems, yet substantial challenges persist. These obstacles highlight the complex constitution of VO and propel ongoing scholarly inquiry and innovation in the discipline.

- **Feature Detection and Tracking** The efficacy of monocular VO hinges on the precise detection and tracking of image features, which are critical measurements in the VO process. Uncertainties in these measurements arise under conditions of low-texture or nondescript environments, which can be exacerbated by inadequate lighting and complex motion dynamics, challenging the robustness of feature-matching algorithms and leading to measurement inaccuracies [46].
- **Motion Estimation**: Robust motion estimation is central to VO, with its accuracy contingent upon the reliability of feature correspondence measurements. Uncertainty in these measurements can occur due to outliers from incorrect feature matching and drift resulting from cumulative errors in successive estimations, significantly complicating the attainment of precise motion measurements [47].
- **Non-static Scenes**: The premise of VO algorithms typically involves the assumption of static scenes, thereby simplifying the measurement process. However, uncertainty is introduced in dynamic environments where moving objects induce variances in the measurements, necessitating advanced methods to discern and correctly interpret camera motion amidst these uncertainties.
- **Camera Calibration**: The accurate calibration of camera parameters is foundational for obtaining precise VO measurements. Uncertainties in calibration-due to factors such as environmental temperature changes, light conditions, lens

distortions, or mechanical misalignments-can significantly distort measurement accuracy, impacting the reliability of subsequent VO estimations [48].

- **Scaling Challenges**: In VO, the lack of an absolute reference frame introduces uncertainty in scale measurements, a pivotal component for establishing the camera's absolute trajectory. Inaccuracies in these scale measurements can arise from ambiguous geometries, limited visual cues, and the monocular nature of the data, which may lead to scale drift and wrong trajectory computations [49].
- **Ground Plane Considerations**: The ground plane is often used as a reference in VO measurements for scale estimation. However, uncertainties in these measurements can be attributed to ambiguous ground features, variable lighting conditions that affect feature visibility, and scaling complexities relative to object heights, challenging the accuracy of VO scale measurements [50].
- **Perspective Projection**: The perspective projection in monocular VO introduces inherent uncertainties due to the transformation of 3D scenes into 2D images, leading to challenges such as depth information loss and scale ambiguity. This projection results in the foreshortening and distortion of objects, complicating the estimation of relative distances and sizes. Additionally, the overlapping of features in the 2D plane can cause occlusions, disrupting the feature tracking crucial for motion estimation. The projection of 3D points onto a 2D plane also introduces feature perspective errors, especially when features are distant from the camera center or when the camera is close to the scene.
- **Timestamp Synchronization Uncertainty**: This type of uncertainty arises when there are discrepancies in the timing of the data capture and processing among different components of a system, such as cameras, inertial measurement units (IMUs), and LiDAR scanners. In systems that rely on precise timing for data integration and analysis, such as visual–inertial navigation systems, this uncertainty can significantly impact accuracy [17].

In summary, the field of monocular VO offers a rich landscape of technological possibilities, bounded by multifaceted challenges that span detection, estimation, scaling, real-time processing, and more. In another aspect, noise sensitivity refers to the impact of image noise on the performance of VO algorithms, which can degrade the accuracy of feature extraction and matching, ultimately affecting the estimated camera trajectory [51]. An uncertainty assessment is essential for evaluating the reliability of the estimated camera trajectory in VO. While traditional VO approaches

often provide an analytical formula for uncertainty, this remains an open challenge for machine learning-based VO methods [52].

Data synchronization is another important aspect in monocular VO, especially when integrating data from multiple sensors, such as cameras and inertial measurement units (IMUs) [53]. Proper synchronization ensures that the data from different sensors are accurately aligned in time, allowing for more precise and reliable trajectory estimation. In some cases, hardware synchronization is used to align the data from different sensors to a common clock, ensuring accurate data fusion and improved VO performance [53].

Achieving real-time performance is imperative for VO applications, yet it poses a challenge due to the computational intensity required for processing measurements. Uncertainty in real-time performance metrics can stem from variable environmental conditions that impact the speed and accuracy of feature detection and matching computations. For example, imagine a self-driving car using VO for navigation. Achieving real-time performance is crucial because the car needs to make immediate decisions based on its surroundings. However, this is challenging due to the heavy computational load required to process the camera's measurements quickly and accurately.

These challenges not only define the current state of VO but also delineate the paths for future research and exploration. By grappling with these complexities, the scientific community continues to pave the way for more nuanced and powerful applications of VO, extending its reach and impact across various domains. A summary of the various approaches and their implications can be found in Table 2.1, offering a succinct overview of the literature's breadth and depth.

Table 2.1: A summary of the mentioned odometry techniques.

Reference	**Sensor Type**	**Method**	**Environmental Structure**	**Open Source**	**Key Points**
[24]	LiDAR	Bundle Adjustment	Outdoor	Yes	Using LiDAR for camera feature tracks and keyframe-based motion estimation. Labeling is used for outlier rejection and landmark weighting.

Continued on the next page...

Reference	Sensor Type	Method	Environmental Structure	Open Source	Key Points
[50]	Monocular	Ground Plane-Based Deep Learning	Outdoor	No	A ground plane and camera height-based divide-and-conquer method. A scale correction strategy reduces scale drift in VO.
[54]	LiDAR	Feature Extraction	Outdoor	No	A VO algorithm using a standard front end with camera tracking relative to triangulated landmarks. Optimizing camera poses and landmark maps resolves monocular scale ambiguity and drift.
[55]	Monocular	Feature Extraction	Indoor	No	A VO system utilizing a downward-facing camera, feature extraction, velocity-aware masking, and nonconvex optimization, enhanced with LED illumination and a ToF sensor.
[56]	LiDAR	Feature Extraction	Outdoor	Yes	LVI-SAM achieves real-time state estimation and map building with high accuracy and robustness.
[57]	LiDAR	Feature Extraction	Outdoor–Indoor	No	A multi-sensor odometry system for mobile platforms integrating visual, LiDAR, and inertial data. Real-time with fixed-lag smoothing.
[58]	LiDAR	Feature Extraction	Outdoor	No	Combines LiDAR depth with monocular VO, using photometric error minimization and point-line feature refinement alongside LiDAR-based segmentation for improved pose estimation and drift reduction.
[59]	Monocular	Feature Extraction	Outdoor	Yes	A visual–inertial SLAM system that uses MAP estimation even during IMU initialization.

Continued on the next page...

Reference	Sensor Type	Method	Environmental Structure	Open Source	Key Points
[60]	Monocular	Feature Extraction	Outdoor	No	A lightweight scale recovery framework using accurate ground plane estimates. Includes ground point extraction and aggregation algorithms for selecting high-quality ground points.
[61]	Monocular	Feature Extraction	Indoor	No	VO using points and lines. Direct methods choose pixels with enough gradients to minimize photometric errors.
[62]	Monocular	Deep Learning Based	Outdoor	No	Combines unsupervised deep learning and scale recovery, trained with stereo image pairs but tested with monocular images.
[11]	Monocular	Deep Learning Based	Outdoor–Indoor	No	A self-supervised monocular depth estimation network for stereo videos, aligning training image pairs with predictive brightness transformation parameters.
[63]	Monocular	Deep Learning Based	Outdoor	No	Proposes the DL Hybrid system, combining DL networks in image processing with geometric localization theory for hybrid pose estimation.
[64]	Monocular	Deep Learning Based	Outdoor	No	The authors created a decoupled cascade structure and residual-based posture refinement in an unsupervised VO framework that estimates 3D camera positions by decoupling rotation, translation, and scale.

Continued on the next page...

Reference	Sensor Type	Method	Environmental Structure	Open Source	Key Points
[17]	Monocular	Deep Learning Based	Outdoor	No	The suggested network is built on supervised learning-based approaches with a feature encoder and pose regressor that takes multiple successive two grayscale picture stacks for training and enforces composite pose restrictions.
[65]	Monocular	Deep Learning Based	Outdoor	Yes	A neural architecture that performs VO, object detection, and instance segmentation in a single thread (SimVODIS).
[66]	Monocular	Deep Learning Based	Outdoor	Yes	The proposed method is called SelfVIO, which is a self-supervised deep learning-based VO and depth map recovery method using adversarial training and self-adaptive visual sensor fusion.

2.1.4 Traditional Approaches

The scientific literature is rife with diverse methodologies aiming to overcome the challenges outlined in the preceding section, particularly focusing on the problem of accurate scale estimation. This issue has typically been addressed through the reliance on knowledge regarding the height of the camera from the ground plane and the evaluation of feature movements on that plane[67]. Alternatively, some approaches have utilized additional tools, such as LiDAR or depth sensors.

Within the domain of autonomous driving, precise vehicle motion estimation is a crucial concern. Various powerful algorithms have been devised to address this need, although most commonly, they depend on binocular imagery or LiDAR measurements. In the following paragraphs, an overview of some prominent works associated with the scaling challenge is provided, highlighting different strategies and technologies.

Tian et al. [68] made a significant contribution by developing a lightweight scale recovery framework for VO. This framework hinged on a ground plane estimate that excelled in both accuracy and robustness. By employing a meticulous ground

point extraction technique, the framework ensured precision in the ground plane estimate. Subsequently, these carefully selected points were aggregated through a local sliding window and an innovative ground point aggregation algorithm. To translate the aggregated data into the correct scale, a Random Sample Consensus (RANSAC)-based optimizer was employed. This optimizer solved a least-squares problem, fine-tuning parameters to derive the correct scale, and thus displaying the marriage of optimization techniques and spatial analysis. The parameters for this fine-tuning are likely chosen based on experimental results to achieve the best performance

H. Lee et al. [55] presented a VO system using a downward-facing camera. This system, designed for mobile robots, integrates feature extraction, a novel velocity-aware masking algorithm, and a nonconvex optimization problem to enhance pose estimation accuracy. It employs cost-effective components, including an LED for illumination and a ToF sensor, to improve feature tracking on various surfaces. The methodology combines efficient feature selection with global optimization for motion estimation, demonstrating improved accuracy and computational efficiency over the existing methods. The authors claimed the experimental results validated its performance in diverse environments, showcasing its potential for robust mobile robot navigation.

B. Fang et al. [58] proposed a method for enhancing monocular visual odometry through the integration of LiDAR depth information, aiming to overcome inaccuracies in feature-depth associations. The methodology involves a two-stage process: initial pose estimation through photometric error minimization and pose refinement using point-line features with photometric error minimization for more accurate estimation. It employs ground and plane point segmentation from LiDAR data, optimizing frame-to-frame matching based on these features, and incorporating multi-frame optimization to reduce drift and enhance accuracy. Based on the authors' claim, the approach demonstrates improved pose estimation accuracy and robustness across diverse datasets, indicating its effectiveness in real-world scenarios[69].

Chiodini et al. [54] expanded the improvement on scale estimation by demonstrating a flexible sensor fusion strategy. By merging data from a variety of depth sensors, including Time-of-Flight (ToF) cameras and 2D and 3D LiDARs, the authors crafted a method that broke free from the constraints of sensor-specific algorithms that pervade much of the literature. This universal applicability is particularly significant for mobile systems without specific sensors. The proposed approach optimized camera

poses and landmark maps using depth information, clearing up the scale ambiguity and drift that can be encountered in monocular perception.

LiDAR–monocular visual odometry (LiMo) was presented by Graeter et al. [24]. This novel algorithm capitalizes on the integration of data from a monocular camera and LiDAR sensor to gauge vehicle motion. By leveraging LiDAR data to estimate the motion scale and provide additional depth information, LiMo enhances both the accuracy and robustness of VO. Real-world datasets were utilized to evaluate the proposed algorithm, and it exhibited marked improvements over other state-of-the-art methods. The potential applications of LiMo in fields like autonomous driving and robotics underscore the relevance and impact of this research.

To mitigate the influence of outliers on feature detection and matching and enhance motion estimation, other researchers introduced data fusion with inertial measurements. This visual–inertial odometry (VIO) integrated system is exemplified in works like Shan et al. [56], which brought together LiDAR, visual, and inertial measurements in a tightly coupled LiDAR–visual–inertial (LVI) odometry system. This holistic fusion, achieved through a novel smoothing and mapping algorithm, elevates the system's accuracy and robustness. The proposal also introduced an innovative technique for estimating extrinsic calibration parameters, further optimizing performance for applications like autonomous driving and robotics.

Wisth et al. [57] and ORB-SLAM3 [59] further illustrated the technological advances in multi-sensor odometry systems and real-time operation in various environments. The use of factor graphs, dense mapping systems, and various sensors such as IMUs, visual sensors, and LiDAR highlights the multifaceted approaches to challenges in motion and depth estimation.

Chuanliu Fan et al. [70] introduce a monocular dense mapping system for visual–inertial odometry, optimizing IMU preintegration and applying a nonlinear optimization-based approach to improve trajectory estimation (Figure 2.2) and 3D reconstruction under challenging conditions. By marginalizing frames within a sliding window, it manages the computational complexity and combines an IMU and visual data to enhance the depth estimation and map reconstruction accuracy. The authors claimed the method outperforms vision-only approaches, particularly in environments with dynamic objects or weak textures, and demonstrates superior performance in comparison to existing odometry systems through evaluations of public datasets.

Two additional pioneering works are by Huang et al. [60], who introduced a VIO optimization-based online initialization and spatial–temporal calibration, and Zhou

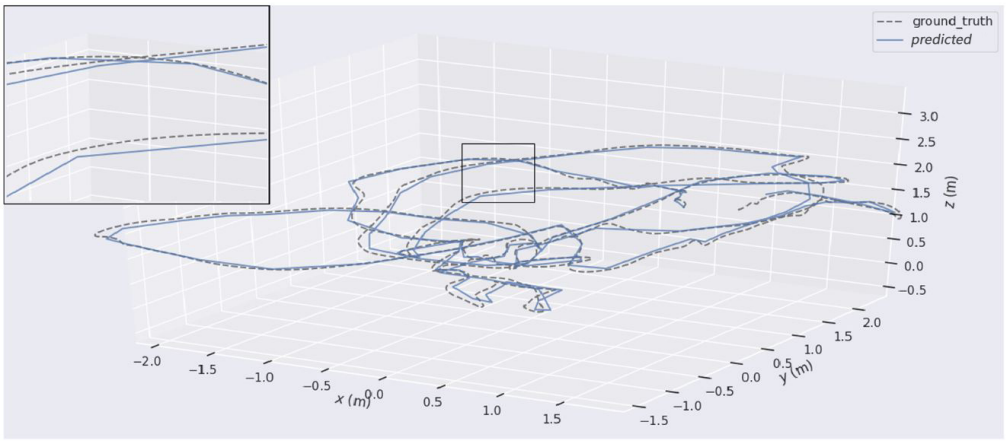


Figure 2.2: demonstrates the accuracy and effectiveness of the proposed nonlinear optimization-based monocular dense mapping system of VIO [70].

et al. [61], who introduced 'Dplvo: Direct point-line monocular VO'. The former focuses on an intricate calibration process that aligns and interpolates camera and IMU measurement data without geographical or temporal information. In contrast, the latter presents an innovative technique that leverages point and line features directly, without needing a feature descriptor, to achieve better accuracy and efficiency.

Collectively, these studies represent a robust and multifaceted exploration of traditional approaches in the realms of motion estimation, depth estimation, and scale recovery within VO. The methodologies vary widely, each bringing unique contributions to scientific discourse and providing promising avenues for ongoing research and development. Their collective focus on enhancing precision, robustness, and computational efficiency underscores the central challenges of the field and the diverse means by which these can be overcome.

Table 2.2: Summary of benchmark *traditional* visual-odometry papers.

Ref. (Year)	**Sensor**	**Key Technique**	**Metric**[†]	**Main Strength**	**Main Limitation**
[24] (2018)	Mono+LiDAR	LiMo / Bundle Adj.	0.26% drift	Robust scale	Heavy LiDAR cost
[55] (2020)	Mono	Ground-plane scale	RMSE 7 cm	Lightweight	Needs flat floor
[57] (2021)	Stereo+IMU	Factor-graph VIO	7.1 cm ATE	Multi-sensor	High CPU load
[56] (2022)	LiDAR+Cam+IMU	LVI-SAM	0.58 % drift	Loop-closing	Sparse in texture

2.1.5 Machine Learning - Based Approaches

Machine learning based approaches to VO are redefining the field with innovative techniques that harness the power of neural networks [63]. Generally, methods in this section can be classified into two distinct categories: full deep learning approaches that utilize neural networks almost exclusively, and semi-deep learning approaches that combine deep learning with more traditional computer vision techniques.

2.1.6 Full Deep Learning Approaches

Full deep learning approaches leverage the complexity and flexibility of neural networks to solve challenging VO tasks.

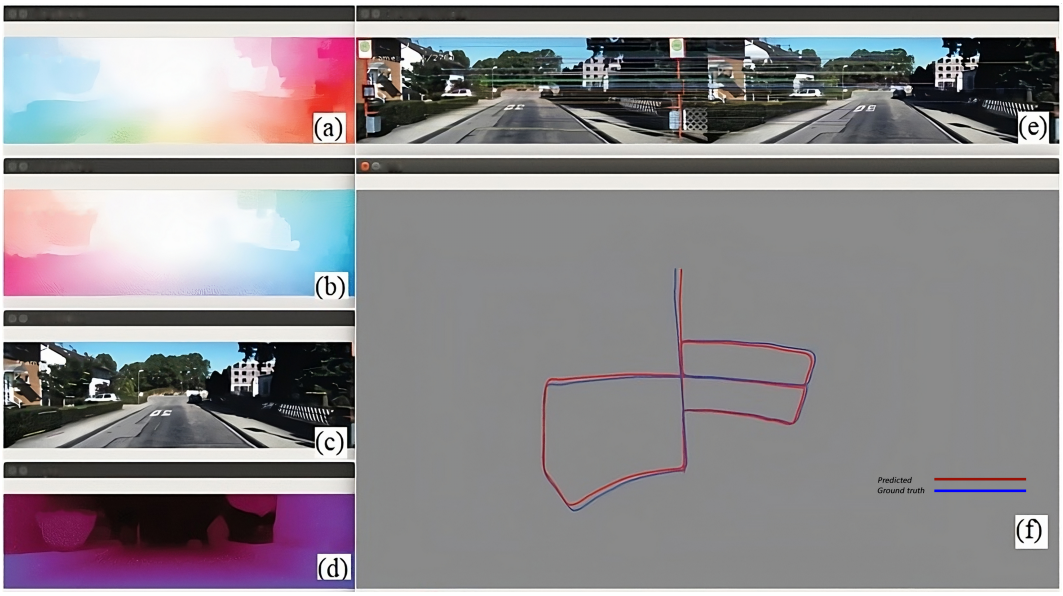


Figure 2.3: in the left column, arranged vertically, are as follows: (**a**) optical flow map in forward order, (**b**) optical flow map in reverse order, (**c**) points of instantaneous optical flow superimposed on the original image, (**d**) map showing monocular depth, (**e**) map illustrating the matching of key points in a pair of images, and (**f**) map depicting the reconstructed trajectory, where the estimated path is indicated by a blue line [63].

Yang et al. [71] pioneered a method called D3VO. This deep learning-based approach for VO estimates both camera motion and the 3D structure of the environment using just a single camera input. Comprising three specialized deep neural networks, D3VO handles depth prediction, pose estimation, and uncertainty estimation. D3VO's method of uncertainty estimation involves predicting a posterior probability distribution for each pixel, which helps in adaptively weighting the residuals in the presence of challenging conditions, like non-Lambertian surfaces or

moving objects. Despite its performance edge over existing VO methods in various benchmarks, D3VO faces significant challenges, such as the need for extensive labeled training data, complexities in securing accurate depth labels, and struggles with low-texture or featureless environments.

Ban et al. [63] contributed a unique perspective by integrating both the depth and optical flow in a deep learning-based method for VO (Figure 2.3). This intricate algorithm first extracts image features, which are then processed through a neural network to estimate the depth and optical flow. The combination of these elements enables the accurate computation of motion. However, a major drawback is the substantial requirement for training data, which is pivotal for effectively training the neural network.

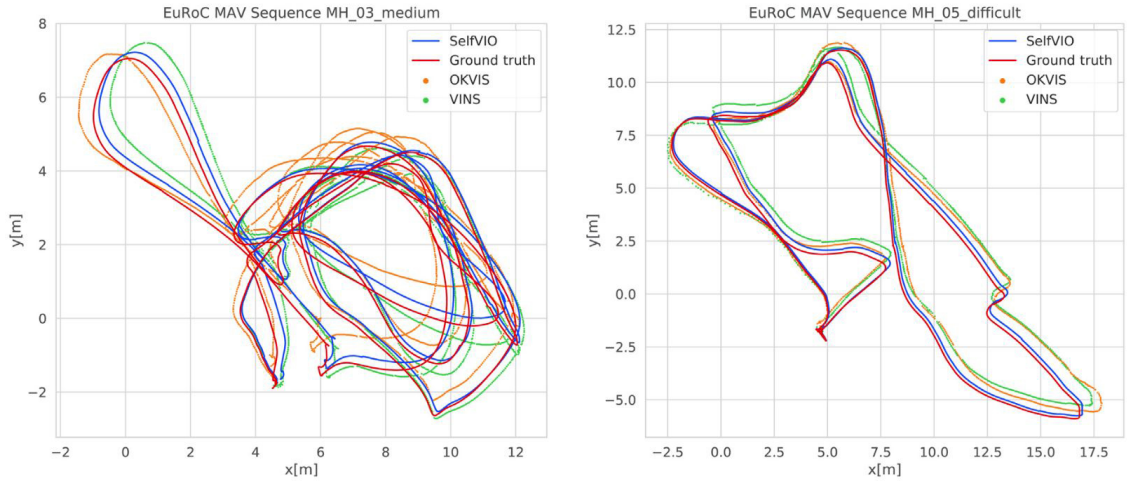


Figure 2.4: comparing the unsupervised learning approach SelfVIO with monocular OKVIS, VINS, and the ground truth in meter scale using EuRoC dataset MH-03 and MH-05 sequences in [66, 72].

In a novel approach, Kim et al. [65] designed a method to perform simultaneous VO, object detection, and instance segmentation. By employing a deep neural network, the method not only estimates the camera pose but also detects objects within the scene, all in real time. While promising, this approach also faces its own set of challenges, particularly the extensive need for training data and potential difficulties with occlusions and clutter.

A notable trend in this category involves self-supervised learning as a solution to the data scarcity problem. Many supervised methods for VIO and depth map estimation necessitate large labeled datasets. To mitigate this issue, the authors in [66] proposed a self-supervised method that leverages scene consistency in shape and lighting. Utilizing a deep neural network, this method estimates parameters

such as camera pose, velocity, and depth without labeled data (Figure 2.4). Still, challenges persist, such as the accuracy of inertial measurements affected by noise and the depth estimation accuracy hampered by occlusions and reflective surfaces.

2.1.7 Semi-Deep Learning Approaches

Semi-deep learning approaches blend the power of deep learning with traditional techniques, leading to methods that are sometimes more adaptable to real-world constraints.

Zhou et al. [50] addressed the unique challenge of absolute scale estimation in VO using ground plane-based features. By identifying the ground plane and extracting its features, they calculated the distance to the camera, assuming certain constants such as flat ground and the known camera height. Using a convolutional neural network (CNN), the method estimates the scale factor, offering potential applications in autonomous driving and robotics.

Lin et al. [64] provided an unsupervised method for VO that ingeniously decouples camera pose estimation into separate rotation and translation components. After the initial feature extraction and essential matrix calculation, a deep learning-based network handles the distinct estimation of rotation and translation. While groundbreaking, this approach is not immune to challenges, including motion blur and changes in the lighting conditions.

Adding to the repertoire of semi-deep learning approaches, Ref. [17] introduced the Windowed Pose Optimization Network (WPO-Net) for VO estimation. In this method, features are extracted from input images, followed by relative pose computation, with a WPO-Net optimizing the pose over a sliding window. Though promising, the computational complexity of the WPO-Net stands as a substantial hurdle, potentially impeding real-time applications.

In summary, machine learning-based approaches are forging new pathways in VO, where full deep learning methods are stretching the capacities of neural networks, and semi-deep learning methods are merging traditional techniques with contemporary progressions. A salient distinction emerges in the realm of the uncertainty assessment: traditional approaches often allow for an analytical derivation of uncertainty, providing clear metrics for measurement confidence. In contrast, deep learning methods grapple with this as an open problem, with the quantification of uncertainty remaining an elusive goal in neural network-based predictions. The pursuit of uncertainty estimation in deep learning remains a vital research area, as it

is critical for the reliability and safety of VO systems in practical applications. The ongoing refinement of these methods underscores a vibrant field ripe with opportunities for innovation, notwithstanding the substantial hurdles that persist.

2.1.8 Uncertainty of Positioning Provided by Monocular VO

In Section **??**, the uncertainty sources in monocular VO were discussed. Aksoy and Alatan [73] addressed this by proposing an inertially aided VO system that operates without the need for heuristics or parameter tuning. This system, leveraging inertial measurements for motion prediction and the EPnP algorithm for pose computation, minimizes assumptions and computes uncertainties for all estimated variables. Their approach effectively compensates for errors related to motion drift and inaccurate feature matching, ensuring more reliable pose estimation. They demonstrated high performance in their system, without relying on data-dependent tuning.

Building on the theme of measurement precision, Ross et al. [74] delved into the intricacies of covariance estimation in a feature-based stereo VO algorithm. Their approach involved learning odometry errors through Gaussian process regression (GPR), which facilitated the assessment of positioning errors alongside the monitoring of VO confidence metrics, offering insights into the uncertainty of VO position estimates. Their method specifically addresses errors stemming from noisy feature detection and varying environmental conditions, thereby improving the robustness of the overall system. Gakne and O'Keefe [75] tackled the scale factor issue in a monocular VO using a 3D city model. They proposed a method dealing with the camera height variation to improve the accuracy of the scale factor estimation. They found that their method provided an accurate solution but up to a scale only. Choi et al. [76] proposed a robust monocular VO method for road vehicles using uncertain perspective projection. They modeled the uncertainty associated with the inverse perspective projection of image features and used a parameter space voting scheme to find a consensus on the vehicle state among tracked features. They found that their method was suitable for any standard camera that views part of the road surface in front of or behind the vehicle.

While the methods proposed in these studies differ, they all aim to improve the accuracy of monocular VO by addressing the issue of scale uncertainty. The results of these studies show that it is possible to estimate the uncertainty of positioning provided by monocular VO and improve its accuracy. However, more research is

needed to develop robust and reliable methods that can be used in different applications.

The model for monocular VO can be mathematically formulated as follows. Let $\mathbf{X}_t$ represent the estimated pose of the vehicle at time t and $\mathbf{Z}_t$ denote the visual measurements obtained from the monocular camera. The uncertainty associated with the visual measurements can be represented by the covariance matrix $\mathbf{R}_t$. Additionally, the uncertainty on the vehicle motion can be captured by the covariance matrix $\mathbf{Q}_t$. The relative vehicle motion can be estimated by considering the uncertainty on the backprojection of the ground plane features and the uncertainty on the vehicle motion, as proposed by Van Hamme et al. [77]. This can be mathematically expressed as:

$$\mathbf{X}_t = f(\mathbf{X}_{t-1}, \mathbf{Z}_t, \mathbf{R}_t, \mathbf{Q}_t)$$

where f represents the function that estimates the pose of the vehicle at time t based on the previous pose, visual measurements, and associated uncertainties. The uncertainty model integrates the uncertainty of visual measurements and the uncertainty of vehicle motion to provide a more accurate assessment of the positioning in monocular VO. The uncertainty on the backprojection of ground plane features and the uncertainty on the vehicle motion are crucial factors in accurately estimating the relative vehicle motion. The Hough-like parameter space vote is employed to extract motion parameters from the uncertainty models, contributing to the robustness and reliability of the proposed method in [76]. Despite the advancements and insights provided by the existing research, a notable gap in the literature is the lack of a comprehensive sensitivity analysis regarding the various sources of uncertainty in monocular VO. The current models and studies often overlook the full spectrum of factors that contribute to uncertainty, ranging from atmospheric conditions to sensor noise. This limitation highlights the need for a more holistic approach to uncertainty modeling in monocular VO. A complete model would not only account for the direct uncertainties in visual measurements and vehicle motion but also extend to encompass external factors, like atmospheric disturbances, lighting variations, and intrinsic sensor inaccuracies. Such a model would enable a deeper understanding of how these diverse factors interact and influence the overall uncertainty in VO systems, paving the way for the development of more sophisticated and resilient techniques that can adapt to a wider range of environmental conditions and application scenarios

Table 2.3: Key sources of uncertainty in monocular VO and typical mitigation techniques.

Source of Uncertainty	Effect on Pose	Mitigation	Residual Error	Ref.
Feature repeatability	Scale drift	Multi-scale ORB + RANSAC	∼1–2 %	[23]
Motion blur / low-light	Tracking loss	Auto-exposure, CNN deblur	< 0.5 pix	[71]
IMU bias (VIO)	Accum. orientation err	Online bias calibration	$< 0.3°/s$	[56]
Ground-plane tilt assumption	Wrong scale	Pitch compensation via IMU	< 5 cm	[55]
* Timestamp mis-sync	Pose jitter	Hardware trigger / Kalman	—	[57]
Dynamic objects	Outlier features	Semantic mask, optical-flow	< 1 %	[11]
Perspective projection error	Depth ambiguity	Line/plane features fusion	N/A	[61]

* Residual error for *Timestamp mis-sync* is not provided, as it depends on the system's motion speed and inter-sensor delay; such errors are typically systematic and require online calibration rather than statistical estimation.

2.1.9 Analysis of Challenges and Advancements

Building on the individual challenges introduced in Subsection 2.1.3 and the VO methods discussed throughout Section 2.1, this subsection provides a comparative and integrative analysis. It highlights how different methods address—or fail to address—key limitations of monocular VO, emphasizing trade-offs, unresolved issues, and directions for improvement.

The implementation and performance of various machine learning-based methods for VO have led to interesting observations and challenges, particularly concerning feature extraction, noise sensitivity, depth estimation, and data synchronization.

The difficulty in feature extraction at high speeds is highlighted in several works[63, 11, 60]. This challenge is exacerbated by factors such as the optical flow on the road and increased motion blur when the vehicle moves fast. Such conditions make feature tracking an arduous task, allowing for only a limited number of valid depth estimates. Some methods have attempted to stabilize results by tuning the feature matcher for specific scenarios, like highways. Still, this often leads to complications in urban settings, where feature matches might become erratic.

Standstill detection, an essential aspect of VO, is another area fraught with difficulty. When the vehicle speed is low, errors can occur if the standstill detection is not well calibrated. The nature of the driving environment, such as open spaces where only the road is considered suitable for depth estimation, adds further complexity to the problem.

The reliance on homography decomposition, as seen in [50], has been found to be highly sensitive to noise. This sensitivity arises from the noisy feature matches obtained from low-textured road surfaces and the multitude of parameters derived from the homography matrix. The task of recovering both camera movement and ground plane geometry is a significant challenge that can affect numerical stability.

Moreover, any method relying on the ground plane assumption is vulnerable to failure if the ground plane is obscured or deviates from the assumed model. This reveals the intrinsic limitation of such methods in varying environmental conditions.

A remarkable development in this field is ORB-SLAM3 [59, 78], which has established itself as a versatile system capable of visual–inertial and multimap SLAM using various camera models. Unlike conventional VO systems, ORB-SLAM3's ability to utilize all previous information from widely separated or prior mapping sessions has enhanced accuracy, showcasing a significant advancement in the field.

Deep learning-based approaches to VO, such as those using CNNs and RNNs, have treated VO and depth recovery predominantly as supervised learning problems [11, 64, 62]. While these methods excel in camera motion estimation and optical flow calculations, they are constrained by the challenge of obtaining ground truth data across diverse scenes. Such data are often hard to acquire or expensive, limiting the scalability of these approaches.

The issue of timestamp synchronization also emerges as a critical concern, as highlighted in [17]. Delays in timestamping due to factors like data transfer, sensor latency, and Operating System overhead can lead to discrepancies in visual–inertial measurements. Even with hardware time synchronization, issues like clock skew can cause mismatches between camera and IMU timestamps [79, 80]. Moreover, synchronization challenges extend to systems using LiDAR scanners, where the alignment with corresponding camera images must be precise. Any deviation in this synchronization can lead to erroneous depth data and subsequent prediction artifacts.

In summary, the machine learning-based approaches to VO chart an intriguing course of breakthroughs and obstacles. Notable progress in employing deep learning and the advent of sophisticated systems such as ORB-SLAM3 mark the current era. Nevertheless, the domain wrestles with intricate issues concerning feature extraction, noise sensitivity, data synchronization, and the procurement of reliable ground truth data. Central to these challenges is the assessment of uncertainty: traditional VO methods could offer probabilistic insights into measurement accuracy, but the integration of uncertainty quantification within deep learning remains a nascent and critical area of research. In traditional approaches, the provided uncertainty models primarily consider sensor noise, neglecting other significant sources of uncertainty. These overlooked elements include factors such as lighting conditions and environmental parameters, which also play a crucial role in the overall accuracy and reliability of the system. A more profound understanding and effective management

of uncertainty could significantly enhance the reliability and applicability of VO technologies, highlighting an essential frontier for ongoing investigative efforts. As such, there is a pressing impetus for continuous research and development to refine the robustness of VO systems and their adaptability to the unpredictable dynamics of real-world environments.

Future research in the field of VO and machine learning is set to tackle key challenges, such as improving feature extraction under difficult conditions, enhancing noise and uncertainty management, developing versatile depth estimation methods, and achieving precise data synchronization. There is a notable demand for novel feature extraction algorithms that perform well in varied environments, alongside more sophisticated models for noise filtering and uncertainty handling. Addressing depth estimation limitations and refining synchronization techniques for integrating multiple sensor inputs are also critical. Importantly, incorporating uncertainty quantification directly into deep learning models for VO could significantly boost system reliability and utility across different applications. These research directions promise to elevate the efficacy and adaptability of VO systems, making them more suited for the complexities of real-world deployment.

2.2 Assessing Measurement Uncertainty in VO and Sensitivity in VIO

In this section, a preliminary uncertainty model is proposed for the RVL of UAVs by means of VO based on a monocular camera. Moreover, several feature extraction algorithms, usually adopted in VO, have been analyzed to evaluate the measurement uncertainty according to the proposed model by simulating a UAV flight mission in MATLAB.

2.2.1 Preliminary Uncertainty Model for VO-based Navigation

The uncertainty model has been applied to the monocular VO-based navigation method derived from the steps delineated in [81]. In particular, the considered UAV is equipped with a monocular camera looking at the ground and with an altitude sensor providing the height of the UAV with respect to the ground floor. Furthermore, it has been assumed that the UAV flies at a fixed altitude and all the feature points correspond to points on the ground. As shown in Figure. 2.5, the UAV position $P(i)$ is obtained from the previous position $P(i-1)$ by adding a

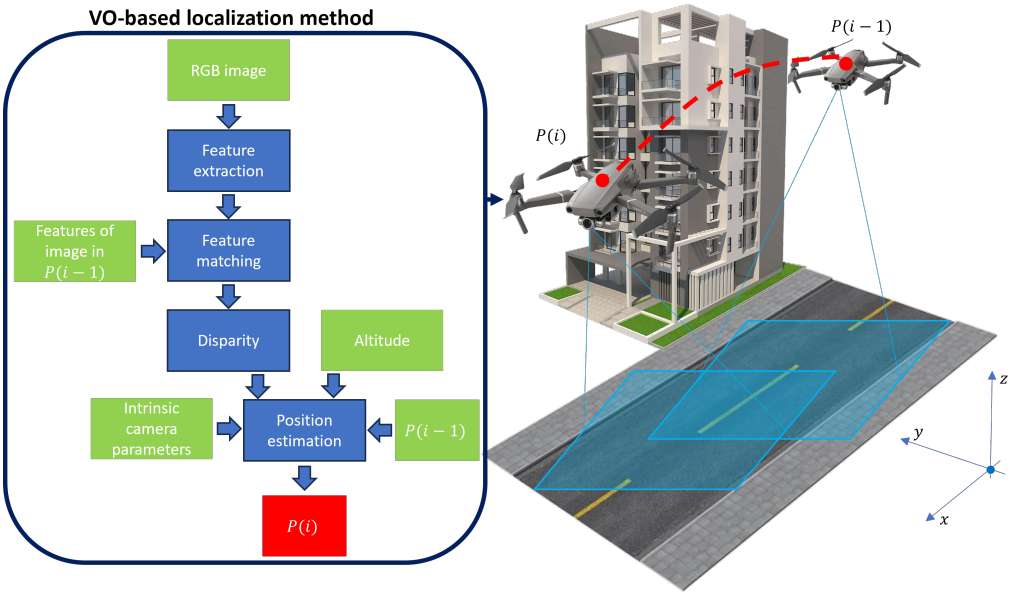


Figure 2.5: Overieview of the adopted monocular VO-based navigation method.

displacement vector. This displacement vector is estimated according to the image features acquired in $P(i-1)$ and $P(i)$, and the flight altitude with respect to the ground. The RGB image acquired in $P(i)$ is given to a feature detection algorithm, (e.g., Speeded Up Robust Features SURF method, Harris corner detector, and so on). The following keypoint detectors and descriptors were employed: ORB (Oriented FAST and Rotated BRIEF), SIFT (Scale-Invariant Feature Transform), and Harris corner detection. The choice was based on a trade-off between robustness, computational cost, and compatibility with downstream processing (e.g., RANSAC homography estimation). The extracted feature points are passed to the feature matching. The feature matching finds the most similar features between the ones extracted in $P(i)$ and in $P(i-1)$. The matched features are then used to obtain the disparity map.This map is analogous to optical flow and should not be confused with stereo disparity. Because we employ a **monocular** camera, no rectification or stereo baseline is available; instead, it has recovered depth for each pixel using the known flight altitude $h(k)$ (from the altimeter) and the intrinsic calibration matrix $\mathbf{K}$. This disparity map represents the distance, with respect to the camera origin, in terms of pixels, of the features on both images (i.e., the ones in $P(i)$ and the others in $P(i-1)$). This map represents the apparent motion of objects between a pair of images. By knowing the depth of each point of the disparity map with respect to

the camera origin, which has been assumed to be equal to the flight altitude, and the intrinsic camera parameters (i.e., focal length and pixel size), it is possible to estimate the displacement vector $\Delta(i)$. The new position $P(i)$ is estimated as a vectorial sum between the estimated coordinate positions in $P(i-1)$ and the obtained displacement vector $\Delta(i)$.
In Figure. 2.6, a graphical representation of the adopted model, derived from the above-described process, is shown. In the following, it has been explained for the $x-z$ plane, however, the same model can be applied to the $y-z$.

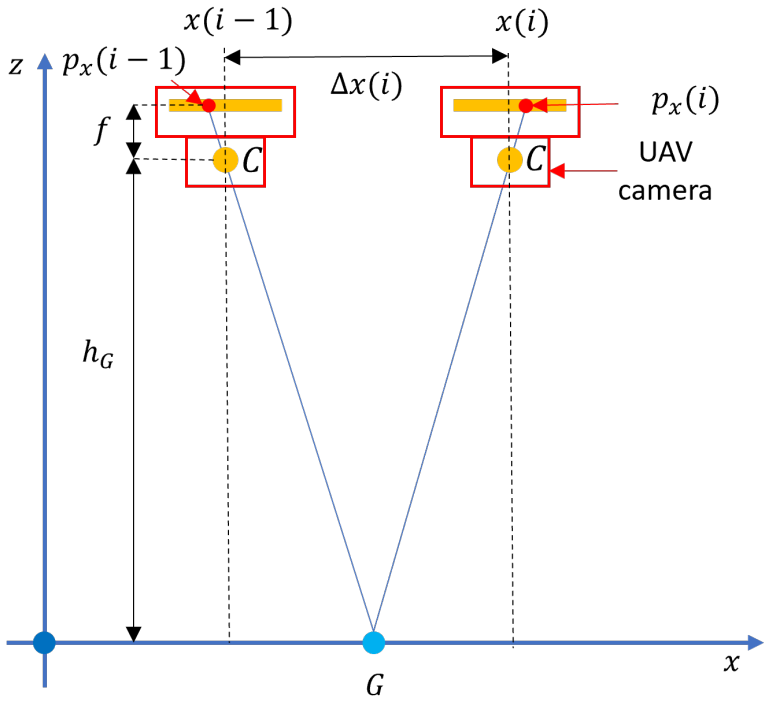


Figure 2.6: Geometry of the VO model. Axes x and z belong to the ground frame $\mathcal{G}$ ($x = X_{\mathrm{g}}$, $z = Z_{\mathrm{g}}$).

It has been assumed that the point G is recognized by the feature matching algorithm on both images and it is at the ground level. In this way, h_G (i.e., distance along the z-axis between the optical center of the camera, C, and G) is equal to the flight altitude. The displacement along the x-axis, $\Delta x(i)$, is obtained as follows:

$$\Delta x(i) = \Delta p_x(i) \cdot \frac{\mu}{f} \cdot h_G \tag{2.1}$$

where, the term $\Delta p_x(i) = p_x(i) - p_x(i-1)$ represents the disparity along the x-axis, $p_x(i)$ and $p_x(i-1)$ correspond to the x-coordinates in pixels of the projections of G

on the camera sensor in both positions, respectively, μ is the pixel size, and f is the focal length of the camera. The x-coordinate of $P(i)$, i.e., $x(i)$ is:

$$x(i) = \Delta x(i) + x(i-1) \tag{2.2}$$

By combining (2.1) and (2.2), it is obtained:

$$x(i) = \Delta p_x(i) \cdot \frac{\mu}{f} \cdot h_G + x(i-1) \tag{2.3}$$

According to (2.3), the main uncertainty sources that affect the position estimates are: (i) the flight altitude measurement (i.e., h_G), (ii) the disparity (i.e., $\Delta p_x(i)$), and (iii) the uncertainty related to the knowledge of the intrinsic camera parameters (i.e., μ and f). In turn, the disparity estimate is influenced by the type of algorithms used for feature detection and matching, and the noise affecting the acquired images. This noise can be due to environmental conditions (e.g., wind, light, visibility conditions) and the UAV flight stability. Among the above-mentioned uncertainty sources, as reported in [81], one of the main challenges for VO is to increase the accuracy related to the disparity map estimation performed by the feature detection and matching. Hence, in this preliminary analysis, the disparity has been considered as a unique uncertainty source. Furthermore, it is considered that the positions estimated in i and $i-1$ are uncorrelated. The assumption of zero correlation between $\mathbf{P}(i-1)$ and $\mathbf{P}(i)$ is convenient for an analytical closed–form uncertainty, but it is only exact when the inter–frame displacement is computed from *independent* sensor data. In practice, monocular VO propagates the previous pose as initial guess and therefore couples the two epochs. Let the 2-D ground-frame position be $\mathbf{P}(i) = [x(i)\ y(i)]^\mathsf{T}$ and let $\mathbf{\Sigma}_P(i)$ be its covariance matrix:

$$\mathbf{\Sigma}_P(i) = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}.$$

The off-diagonal term σ_{xy} captures the coupling of the horizontal directions, mainly induced by: (i) camera yaw/pitch during forward motion, and (ii) perspective projection that mixes pixel errors in u and v into both x and y. Moreover, the covariance at epoch i is obtained via

$$\mathbf{\Sigma}_P(i) = \mathbf{J}_\Delta\, \mathbf{\Sigma}_\Delta(i)\, \mathbf{J}_\Delta^\mathsf{T} \ + \ \mathbf{\Sigma}_P(i-1),$$

where $\mathbf{\Sigma}_\Delta(i)$ is the covariance of the pixel–displacement vector $\Delta\mathbf{p}(i)$ and $\mathbf{J}_\Delta =$

$\frac{\partial(x,y)}{\partial(u,v)}$ is the Jacobian.

In this way, by applying the law of propagation of uncertainty according to [82] to (2.3), it is obtained:

$$u_{x(i)} = \sqrt{u^2_{\Delta p_x(i)} \cdot \frac{\mu^2}{f^2} \cdot h_G^2 + u^2_{x(i-1)}} \tag{2.4}$$

where, $u_{\Delta p_x(i)}$ is the uncertainty of the disparity measurements. As expected (2.4) is a recursive equation where the uncertainty at i depends on the uncertainty at the previous step $i-1$. By considering an initial uncertainty at the position $i=0$, i.e., $u_{x_(0)}$, the uncertainty at i is:

$$u_{x(i)} = \sqrt{\sum_{m=1}^{i} u^2_{\Delta p_x(m)} \cdot \frac{\mu^2}{f^2} \cdot h_G^2 + u^2_{x(0)}} \tag{2.5}$$

If $u_{\Delta p_x(m)}$ is constant for $m = 1 \ldots i$, it can be written:

$$u_{x(i)} = \sqrt{i \cdot u^2_{\Delta p_x} \cdot \frac{\mu^2}{f^2} \cdot h_G^2 + u^2_{x(0)}} \tag{2.6}$$

As an example, in Figure. 2.7, the expanded uncertainty, $U_{x(i)}$, according to (2.6), with a coverage factor of 3, for a camera having a focal length $f = 24\,\text{mm}$ and a pixel size $\mu = 21.8\,\mu\text{m}$, by considering a flight altitude $h_G = 50\,\text{m}$ and $u_{x(0)} = 0.1\,\text{m}$, is depicted against the number of position estimates i for several $u_{\Delta p_x}$ ranging from 2 px to 18 px. As expected, the uncertainty increases with the number of estimates i. In particular, it can be seen the uncertainty for $u_{\Delta p_x} = 10$ px after 20 position estimates is higher than 5 m, which is not acceptable in several practical cases.

It is worth noting that the coverage factor $k = 3$ is used in this analysis to represent a confidence level of approximately 99.7%, assuming normally distributed uncertainties. This conservative value is appropriate for safety-critical UAV navigation scenarios. Nevertheless, alternative coverage factors (e.g., $k = 2$ for 95.4% confidence) can be adopted depending on application requirements.

2.2.2 Experimental Assessment

The proposed uncertainty model has been applied to images captured by using a UAV flight mission simulator available in MATLAB by considering several feature detection algorithms. These algorithms have been compared in terms of uncertainty in the estimation of the displacement vector $\Delta x(i)$. Then, according to (2.4), the

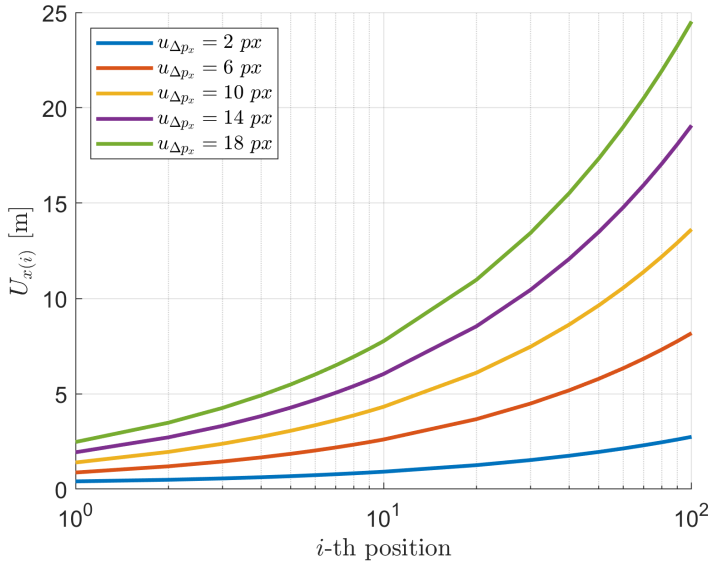


Figure 2.7: Expanded uncertainty $U_{x(i)}$ obtained for $u_{\Delta p_x}$ ranging from 2 px to 18 px, considering a camera with $f = 24\,\text{mm}$ and $\mu = 21.8\,\mu\text{m}$, a flight altitude $h_G = 50\,\text{m}$, and $u_{x(0)} = 0.1\,\text{m}$.

uncertainty for the x and y coordinate estimations has been applied to the best feature estimator.

2.2.3 UAV flight mission simulator

The adopted UAV flight mission simulator is based on the UAV simulator package called "Delivery example" available in MATLAB [83]. This example simulates the flight of a quadrotor in an urban environment. In particular, the flight simulator allows the definition of the flight mission in QGround control by setting the takeoff, landing points, and other waypoints. For each point, it is possible to choose the flight altitude and the UAV speed. In the performed simulation, the flight mission consists of a takeoff to the flight altitude of 50 m, then the landing after around 60 m of horizontal flight along the x direction at the constant speed of $5\,\text{m}\,\text{s}^{-1}$. The VO method has been applied only during the horizontal flight without considering the images acquired during takeoff and landing. The considered parameters for the camera are: (i) $f = 24\,\text{mm}$, (ii) $\mu = 21.8\,\mu\text{m}$, and (iii) an image size of 1080×1920 pixels. Regarding the navigation environment, the 3D simulator allows to modify the weather conditions in terms of sun position, cloud opacity, cloud speed, fog density, and rain density. In the performed simulations, those parameters are imposed as

default [84]: (i) the sun altitude is 40° and azimuth is 90°, (ii) the cloud opacity is 10 %, (iii) the cloud speed direction is from west to east, and (iv) there is no fog and no rain. In Figure. 2.8, a screenshot of the UAV flight simulator environment is reported. During the simulation, the images are acquired with a sampling period of 0.25 s.

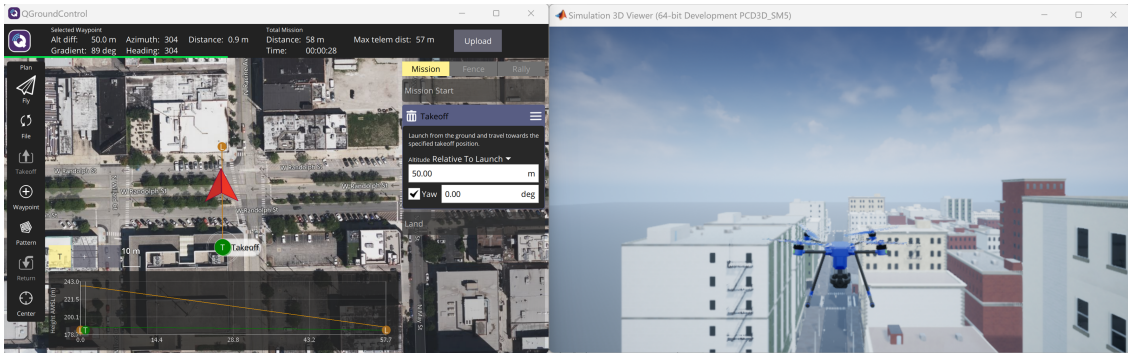


Figure 2.8: Screenshot of the UAV flight simulator, on the left the QGroundControl user interface, and on the right the 3D simulated environment.

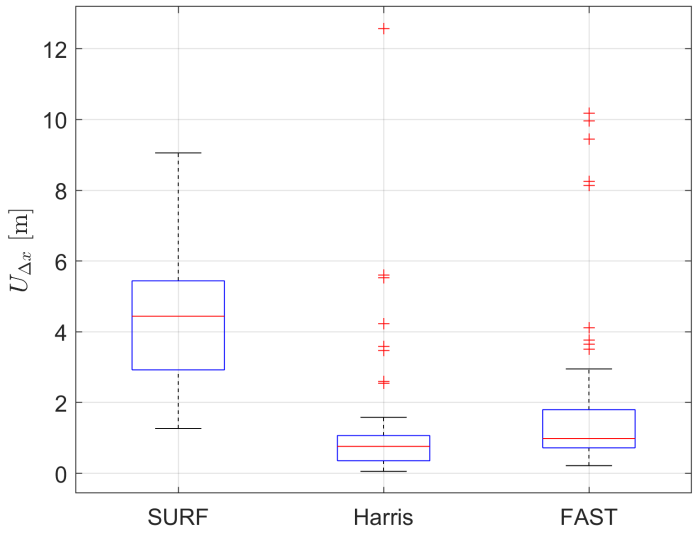
In the performed simulation, the onboard camera is configured to be downward-facing, with its optical (z) axis orthogonal to the ground plane. As a result, the image plane axes (x and y) are considered approximately parallel to the ground surface. This assumption allows for simplified projection geometry during trajectory estimation. However, it is acknowledged that in real-world UAV operations, slight deviations from this ideal orientation may occur due to UAV motion dynamics, environmental disturbances, or sensor mounting inaccuracies. Such inclinations may introduce additional uncertainties in the localization process, and should be addressed through further experimental calibration or robust uncertainty modeling in future studies.

2.2.4 Feature detection algorithms

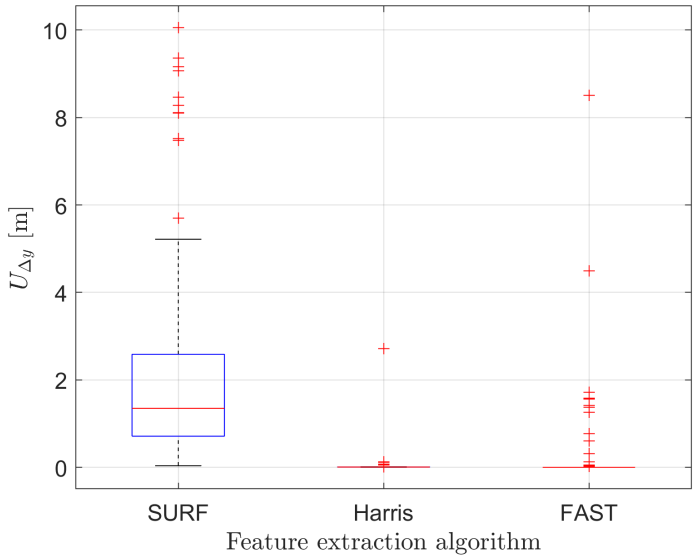
As stated in Section 2.2.1, an important step for VO navigation is the feature detection. In the performed analysis, three feature detection algorithms have been tested: (i) SURF, (ii) Harris-Stephens (in the following called as Harris), and (iii) Features from Accelerated Segment Test (FAST).
The SURF algorithm is based on two steps, the feature extraction and the feature description [85]. The feature extraction is performed by filtering the image at different scales through a Gaussian filter and applying a second-order derivative [85].

For the obtained scaled images the determinant of the Hessian matrix is calculated [85]. The feature description is based on the orientation assignment using wavelet responses in the horizontal and vertical directions [85].

The Harris algorithm is a corner detector that determines whether a region is an



(a)



(b)

Figure 2.9: Box plots of the expanded displacement uncertainty values, U_Δ along: (a) x-direction, and (b) y-direction for SURF, Harris, and FAST algorithms.

edge, a corner, or flat according to a threshold value. Then, the Non-Max Suppression is applied to select only one region class for each classified sub-image according to the class with the highest probability [86].
The FAST algorithm is a corner detection algorithm using a circle of 16 pixels to classify whether a candidate point is a corner [87]. If a set of contiguous pixels in the circle are all brighter than the intensity of the candidate point plus and minus a threshold value, then it is classified as a corner [87].

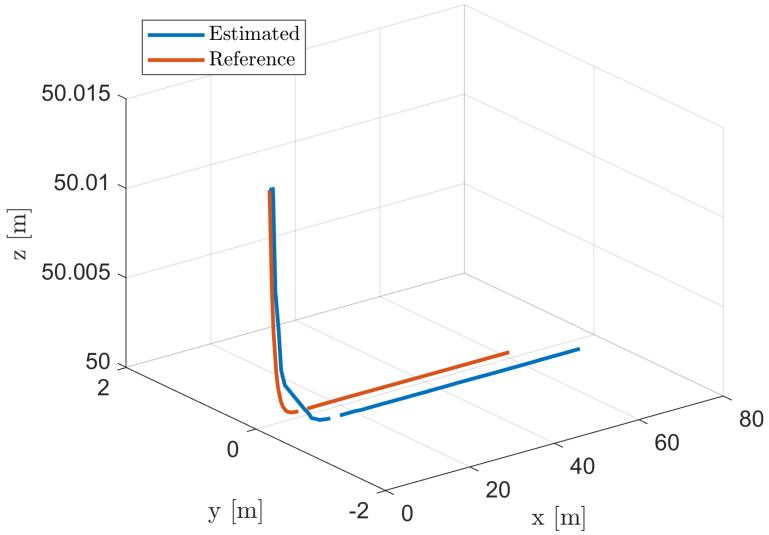
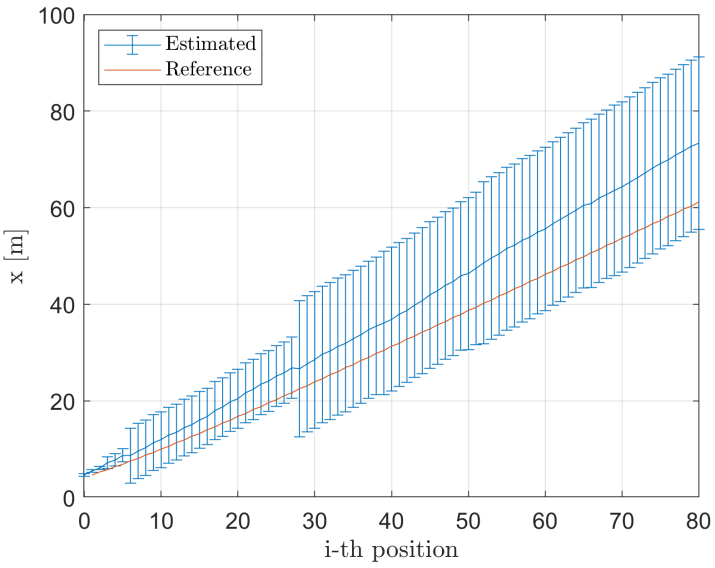


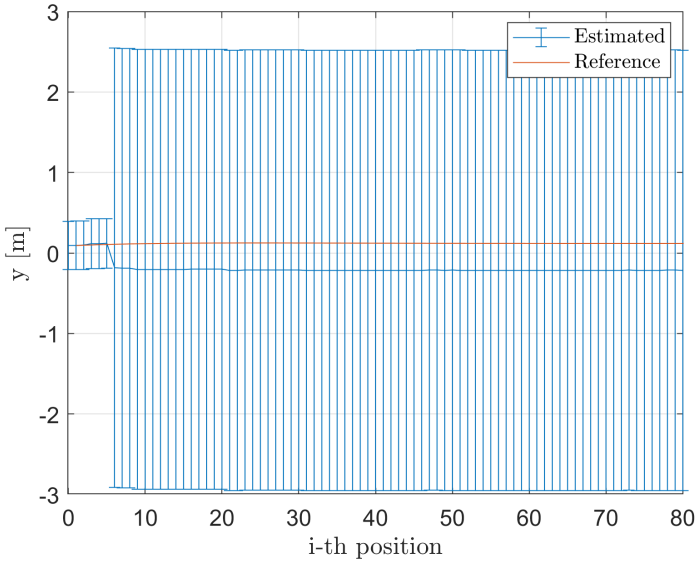
Figure 2.10: Estimated trajectory with the VO navigation method based on the Harris feature extractor vs. the reference trajectory provided by the simulator.

2.2.5 Uncertainty assessment

The three feature detection algorithms have been compared in terms of expanded uncertainty of the displacement estimation. For each estimate, the uncertainty of the disparity map in terms of pixels (i.e., $u_{\Delta p_x(i)}$ and $u_{\Delta p_y(i)}$) is evaluated from the standard deviation of the disparity values obtained for each couple of images, while their mean value is used for estimating the i-th displacement. According to the simulated flight mission, the number of uncertainty estimations is 80. In Figure. 2.9, the box plots of the 80 uncertainty values obtained for SURF, Harris, and FAST algorithms are reported for the displacement estimates along the x and y axes. This comparison shows the SURF algorithm underperforms Harris and FAST in terms of uncertainty. On the other hand, Harris and FAST are comparable along both



(a)



(b)

Figure 2.11: Estimated coordinates and expanded uncertainties obtained from (2.4) along: (a) x-direction, and (b) y-direction by using a VO navigation method based on the Harris feature extractor vs. the reference trajectory.

axes. However, for the x-axis, the distribution of the uncertainty values for Harris is slightly shifted to lower values than the distribution of uncertainties for FAST. Furthermore, Harris exhibits a lower number of outliers for the uncertainty along

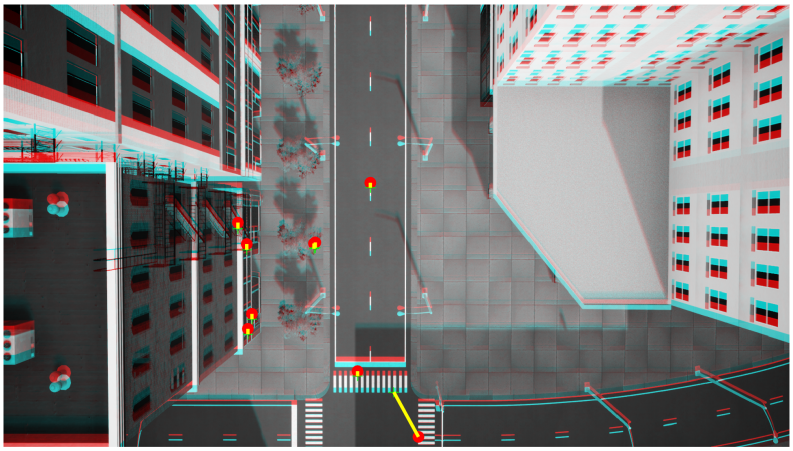


Figure 2.12: Disparity map in position 6.

the y-axis. For this reason, the Harris algorithm has been selected among the tested ones for assessing the uncertainty in the estimate of the position of the UAV.
Figure. 2.10 reports the estimated trajectory by means of the VO navigation method using the Harris feature detector with respect to the reference trajectory provided by the simulator. It can be noted that along the y-axis, there is an error of around 0.3 m for the position estimates, while, along the x-axis, the proposed method overestimates the length of the flight, i.e., 73 m instead of 61 m. This overestimate is evident in Figure. 2.11, where it can be seen that the estimated trajectory is affected by drift, as is expected for VO-based navigation.
Figure. 2.11 reports the x and y coordinates of the estimated positions according to the VO navigation method with respect to the reference coordinates, together with the expanded uncertainty according to (2.4). In this analysis, the uncertainties along the x and y axes at position 0, i.e., $u_{x(0)}$ and $u_{y(0)}$, are assumed to be 0.1 m. It can be seen that the uncertainty increases with the number of position estimates and the uncertainty along the x-axis is higher than the one obtained along the y-axis. This is mainly due to the fact that the UAV is moving only along that axis in the performed simulation. Furthermore, Figure. 2.11 shows that the uncertainty values drastically increase in two positions, i.e., 6 and 28. In these two positions, the variability of the disparity map is very high, i.e., 41 pixels and 93 pixels for 6 and 28, respectively. By looking at the disparity map in position 6, it can be seen that this highest uncertainty is due to a wrong matching of the feature on the pedestrian crossing, see Figure. 2.12.

2.2.6 Sensitivity Analysis for Visual-Inertial Odometry

Unmanned aerial vehicle (UAV)-based delivery has become increasingly popular as a cost-effective and environmentally friendly method for transporting goods. Its growing application spans different scenarios, including last-mile delivery, regional air transit, and providing services to remote or hard-to-reach areas [88]. UAVs are also increasingly utilized for critical operations such as emergency medical supply transport, military logistics, and passenger transportation [89]. These expanding roles require UAVs to operate in proximity to critical infrastructure and human traffic, presenting both opportunities and challenges for safe and efficient operations.

To guarantee a high level of safety in autonomous and semi-autonomous piloting systems, it is important to provide an accurate estimation of the UAV pose, i.e., position and orientation measurements. Among UAV navigation solutions, Global Navigation Satellite System (GNSS)-assisted inertial navigation systems (INS), which integrate inertial measurement unit (IMU) data, are the most used [90]. These systems rely on inertial data from accelerometers and gyroscopes, combined with GNSS position measurements, to estimate the platform's pose. However, GNSS-based methods may lead to unreliable navigation results, particularly in urban areas where signal obstructions and multipath effects occur [90]. GNSS-denied solutions have been proposed to address these limitations, including vision-based, LiDAR-based, radar-based, ultra-wideband (UWB) positioning, and combined navigation systems [91]. VIO approaches combine data from vision and IMU sensors [92]. They utilize pose constraints from the IMU and the camera to solve an optimization problem that estimates incremental motion [91]. Camera constraints are derived by matching unique features identified across images [63]. The vision sensors are usually RGB cameras, depth cameras and/or LiDAR [93].

In the literature, two performance metrics assess positional accuracy: root-mean-square error (RMSE) and percentage position drift (or relative position drift) with respect to the travelled distance [88]. RMSE is calculated as the square root of the average of the squared differences between the actual path coordinates and the coordinates obtained from the estimated trajectory. Percentage position drift is the difference in position between the actual and estimated trajectories, expressed as a percentage of the distance travelled along the trajectory to the calculated point[94]. Although these metrics quantify point-by-point error with respect to a reference system, they do not provide any information regarding the main uncertainty sources

affecting the position and orientation measurements. To accomplish this, an uncertainty model that incorporates all sources of error is essential for quantifying how sensitive the provided measurements are to each source [95]. This allows the implementation of targeted countermeasures to mitigate these errors effectively, thus improving overall accuracy.
A preliminary uncertainty model was proposed in [95]. This model considers a UAV equipped with a monocular RGB camera and an altimeter to estimate the UAV's position. Several proposals in the literature suggest using IMU, LiDAR, and depth cameras to enhance position accuracy [96], [97]. Additionally, the analysis in [95] aimed to evaluate how sensitive the position measurements are to the accuracy of traditional feature extraction and matching algorithms found in the literature (such as SURF, Harris, FAST). This evaluation did not consider other sources of uncertainty, such as the camera's focal length, orientation measurements, and flight altitude. Furthermore, the uncertainty assessment was conducted without varying environmental and light conditions.

In this section, a more complex VIO framework was considered by combining the information provided by the camera with LiDAR, altimeter, and inertial measurements. In addition to traditional keypoint detection algorithms, more advanced methods have been evaluated, including semantic segmentation using a Deep Learning (DL) model. Although semantic segmentation does not directly extract geometric keypoints, it produces object-level masks that can be used to guide the selection of robust and context-aware feature points within semantically meaningful regions of the image. This improves matching performance especially in visually degraded or cluttered scenes. The uncertainty model was modified accordingly, and a sensitivity analysis was carried out, considering various environmental and light conditions, as well as the uncertainty related to orientation, depth/altitude measurements, and intrinsic camera parameters.

2.2.7 Visual Inertial Odometry Framework

Figure. 2.13 illustrates a simplified model representing the common steps typically employed in the implementation of a VIO navigation system. This framework serves as the foundation for developing the uncertainty model presented in this work. The first step involves identifying keypoints within a frame—specific points of interest that stand out due to differences in colour or brightness[98, 99, 100]. A feature descriptor can uniquely characterize each keypoint. The second step focuses

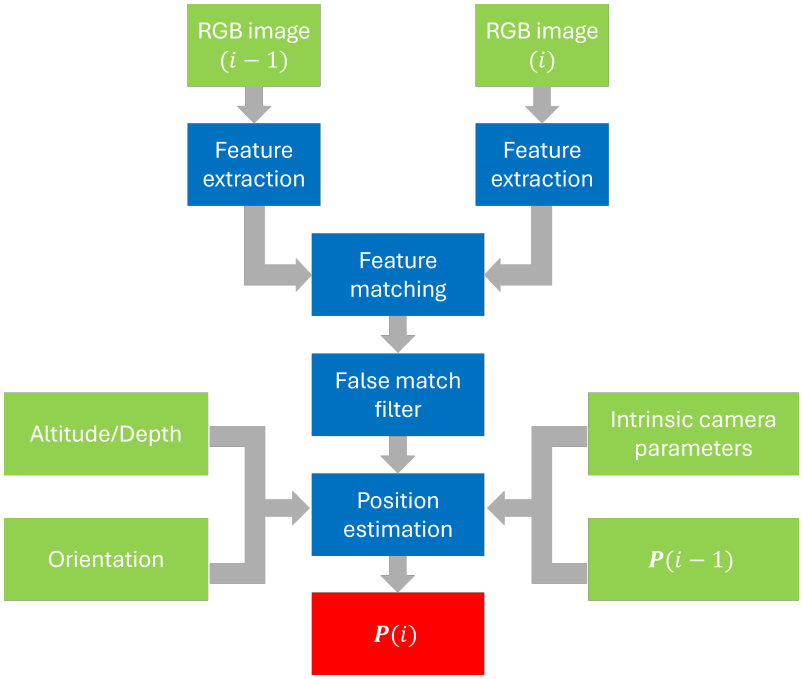


Figure 2.13: Workflow of the modelled VIO method. The measurements required for pose estimation are assumed to be provided by specific onboard sensors: an RGB or RGB-D camera for image acquisition, an Inertial Measurement Unit (IMU) for roll, pitch, and yaw angles, and either an altimeter or LiDAR module for depth or altitude data. The intrinsic camera parameters are assumed to be calibrated beforehand. These sensors are representative of typical UAV payload configurations for navigation in GNSS-denied environments.

on matching these features between two consecutive frames based on their similarity. However, many correspondences are often not correctly identified. The third step addresses this issue by filtering out false matches. Finally, in the fourth step, a mathematical model is used to estimate the system's position based on the position estimates from the previous step, the orientation measurements, the intrinsic camera parameters, and either altitude or depth measurements, depending on whether the UAV is equipped with an altimeter or an RGB-D camera. For identifying the keypoints, the following algorithms have been considered to assess the performance of the localization system: (i) SIFT (Scale-Invariant- Feature-Transform) [101], (ii) SURF (Speeded-Up-Robust-Features) [85], (iii) FAST (Features-from-Accelerated-Segment-Test) [87], (iv) Harris [86], (v) MEF (Minimum-Eigenvalue-Features) [102], (vi) ORB (Oriented-FAST-and-Rotated-BRIEF) [40], and (vii) semantic segmentation via DL [103]. The first six algorithms identify the points of interest, within digital images, that remain particularly recognizable despite variations caused by

camera movement or shake, scaling, lighting, and other environmental factors. The semantic segmentation exploits a trained neural network to assign a label to each pixel in an image, corresponding to object categories. By extracting object masks common between two frames, the keypoints are obtained. The homography matrix estimation using the RANdom SAmple Consensus (RANSAC) method has been adopted to match keypoints between two consecutive frames [104]. This matrix describes a relationship between points in two consecutive frames such that each point in one frame corresponds to one and only one point in the other frame. It consists of nine parameters with eight degrees of freedom:

$$\mathbf{H} = \begin{bmatrix} \frac{h_{11}}{h_{33}} & \frac{h_{12}}{h_{33}} & \frac{h_{13}}{h_{33}} \\ \frac{h_{21}}{h_{33}} & \frac{h_{22}}{h_{33}} & \frac{h_{23}}{h_{33}} \\ \frac{h_{31}}{h_{33}} & \frac{h_{32}}{h_{33}} & 1 \end{bmatrix} \tag{2.7}$$

Therefore, it can be estimated once four keypoints are recognised in both frames, $[\epsilon_c(i-1), \eta_c(i-1)]$ and $[\epsilon_c(i), \eta_c(i)]$ with $c = 1, 2, 3, 4$, i and $i-1$ representing the current frame and the previous one, respectively:

$$\mathbf{h} = \mathbf{A}^{-1} \cdot \mathbf{b} \tag{2.8}$$

where, $\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T$, $\mathbf{b} = [0, \ldots, 0, 1]^T$, and

$$\mathbf{A} = \begin{bmatrix} \epsilon_1(i-1) & \eta_1(i-1) & 1 & 0 & 0 & 0 & -\epsilon_1(i-1)\epsilon_1(i) & -\eta_1(i-1)\epsilon_1(i) & -\epsilon_1(i) \\ 0 & 0 & 0 & \epsilon_1(i-1) & \eta_1(i-1) & 1 & -\epsilon_1(i-1)\eta_1(i) & -\eta_1(i-1)\eta_1(i) & -\eta_1(i) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \epsilon_c(i-1) & \eta_c(i-1) & 1 & 0 & 0 & 0 & -\epsilon_c(i-1)\epsilon_c(i) & -\eta_c(i-1)\epsilon_c(i) & -\epsilon_c(i) \\ 0 & 0 & 0 & \epsilon_c(i-1) & \eta_c(i-1) & 1 & -\epsilon_c(i-1)\eta_c(i) & -\eta_c(i-1)\eta_c(i) & -\eta_c(i) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ \epsilon_4(i-1) & \eta_4(i-1) & 1 & 0 & 0 & 0 & -\epsilon_4(i-1)\epsilon_4(i) & -\eta_4(i-1)\epsilon_4(i) & -\epsilon_4(i) \\ 0 & 0 & 0 & \epsilon_4(i-1) & \eta_4(i-1) & 1 & -\epsilon_4(i-1)\eta_4(i) & -\eta_4(i-1)\eta_4(i) & -\eta_4(i) \end{bmatrix} \tag{2.9}$$

However, more than four correspondences between two images are obtained in feature detection and tracking algorithms. For this reason, it is necessary to use the RANSAC algorithm, which allows estimating the homography matrix from a large set of correspondences (significantly more than four) and eliminating correspondences that do not satisfy the transformation dictated by the homography matrix based on a set threshold. In the analysed workflow, the threshold value has been fixed to 2 pixels, and the maximum number of iterations of RANSAC is 100. As shown in Figure. 2.14, once the keypoints are identified, the UAV's position can be estimated using: (i) orientation data from the IMU, (ii) depth or altitude mea-

surements from the RGB-D camera or altimeter, and (iii) the camera's intrinsic parameters, such as focal length and pixel size. Consider the vectors $\boldsymbol{\beta}$ and $\mathbf{b}$, they are collinear:

$$\boldsymbol{\beta} = k \cdot \mathbf{b} \tag{2.10}$$

with $k \in \mathbb{R}$, $\boldsymbol{\beta} = [\epsilon_i - \epsilon_o, \eta_i - \eta_o, -c]^T$, and $\mathbf{b} = [x_i' - x_i, y_i' - y_i, z_i' - z_i]^T$. where (ϵ_o, η_o) are the coordinates of the focal point, c is the focal length, (ϵ_i, η_i) are the coordinates of the point $\boldsymbol{P}_i' = (x_i', y_i', z_i')$ in the image plan, $\boldsymbol{P}_i$ and $\boldsymbol{P}_{i-1}$ are the coordinates of the UAV in the i-th and $(i-1)$-th positions. It should be noted that even if the camera frame is assumed to be aligned with the ground frame (i.e., yaw, pitch, and roll angles are zero), the origins of the two frames may still differ. In particular, the vector $\vec{b}$, representing the position of the observed feature point, must be expressed in the same reference frame as the UAV pose to ensure consistent transformation. Therefore, $\vec{b}$ should be transformed into the camera frame of reference if calculations are performed in that domain, or alternatively, a translation offset should be considered to account for the displacement between the frame origins. This alignment is crucial for maintaining geometric consistency in the uncertainty propagation model.

Let consider the effect of the yaw ϕ, pitch θ, and roll γ angles measured by UAV during flight on the recognised point $\boldsymbol{P}_i'$:

$$\boldsymbol{P_i}' = \boldsymbol{R} \cdot \boldsymbol{P_t} \tag{2.11}$$

where, $\boldsymbol{P_t}$ contains the coordinates of the recognised key point into the reference plane, which is defined according to the magnetic north and the gravity vector, and $\boldsymbol{R}$ is the 3×3 rotation matrix obtained from ϕ, θ, and γ. Substituting in (2.10), the following system of equations can be written:

$$\begin{cases} \epsilon_i - \epsilon_o = k \cdot [r_{11}(x_t - x_i) + r_{12}(y_t - y_i) + r_{13}(z_t - z_i)] \\ \eta_i - \eta_o = k \cdot [r_{21}(x_t - x_i) + r_{22}(y_t - y_i) + r_{23}(z_t - z_i)] \\ -c = k \cdot [r_{31}(x_t - x_i) + r_{32}(y_t - y_i) + r_{33}(z_t - z_i)] \end{cases} \tag{2.12}$$

By solving this system of equations for x_t and y_t, it is obtained:

$$\begin{aligned} x_t &= x_i + (z_t - z_i)\frac{r_{11}(\epsilon_i - \epsilon_o) + r_{12}(\eta_i - \eta_o) - r_{13}c}{r_{31}(\epsilon_i - \epsilon_o) + r_{32}(\eta_i - \eta_o) + r_{33}c} \\ y_t &= y_i + (z_t - z_i)\frac{r_{21}(\epsilon_i - \epsilon_o) + r_{22}(\eta_i - \eta_o) - r_{23}c}{r_{31}(\epsilon_i - \epsilon_o) + r_{32}(\eta_i - \eta_o) - r_{33}c} \end{aligned} \tag{2.13}$$

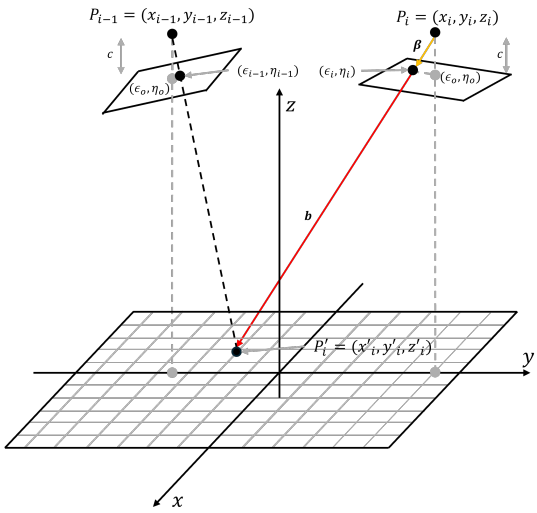


Figure 2.14: Model used for estimating the UAV position according to the orientation measurements provided by IMU, the depth/altitude measurements provided by the RGB-D camera/altimeter, the keypoints obtained from the feature matching, and the camera's intrinsic parameters.

The eq. (2.13) can be written for $i-1$, too:

$$\begin{aligned} x_t &= x_{i-1} + (z_t - z_{i-1})\frac{r_{11}(\epsilon_{i-1}-\epsilon_o)+r_{12}(\eta_{i-1}-\eta_o)-r_{13}c}{r_{31}(\epsilon_{i-1}-\epsilon_o)+r_{32}(\eta_{i-1}-\eta_o)+r_{33}c} \\ y_t &= y_{i-1} + (z_t - z_{i-1})\frac{r_{21}(\epsilon_{i-1}-\epsilon_o)+r_{22}(\eta_{i-1}-\eta_o)-r_{23}c}{r_{31}(\epsilon_{i-1}-\epsilon_o)+r_{32}(\eta_{i-1}-\eta_o)-r_{33}c} \end{aligned} \tag{2.14}$$

By substituting eq. (2.13) in eq. (2.14):

$$\begin{aligned} x_i = x_{i-1} &+ d_{i-1}\frac{r_{11_{i-1}}(\epsilon_{i-1}-\epsilon_o)+r_{12_{i-1}}(\eta_{i-1}-\eta_o)-r_{13_{i-1}}c}{r_{31_{i-1}}(\epsilon_{i-1}-\epsilon_o)+r_{32_{i-1}}(\eta_{i-1}-\eta_o)+r_{33_{i-1}}c} + \\ &- d_i\frac{r_{11_i}(\epsilon_i-\epsilon_o)+r_{12_i}(\eta_i-\eta_o)-r_{13_i}c}{r_{31_i}(\epsilon_i-\epsilon_o)+r_{32_i}(\eta_i-\eta_o)+r_{33_i}c} \\ y_i = y_{i-1} &- d_{i-1}\frac{r_{21_{i-1}}(\epsilon_{i-1}-\epsilon_o)+r_{22_{i-1}}(\eta_{i-1}-\eta_o)-r_{23_{i-1}}c}{r_{31_{i-1}}(\epsilon_{i-1}-\epsilon_o)+r_{32_{i-1}}(\eta_{i-1}-\eta_o)-r_{33_{i-1}}c} + \\ &+ d_i\frac{r_{21_i}(\epsilon_i-\epsilon_o)+r_{22_i}(\eta_i-\eta_o)-r_{23_i}c}{r_{31_i}(\epsilon_i-\epsilon_o)+r_{32_i}(\eta_i-\eta_o)-r_{33_i}c} \end{aligned} \tag{2.15}$$

where, $d_i = z_i - z_t$ and $d_{i-1} = z_{i-1} - z_t$ are the depth measurements of the keypoint $\boldsymbol{P}_t$ in the i-th and $(i-1)$-th frames, respectively. Eq. (2.15) analytically describes the relationship between the recognised keypoints in the image plane (i.e, (ϵ_i, η_i) and $(\epsilon_{i-1}, \eta_{i-1})$ and the UAV position in i. This equation can be applied to all the recognised keypoints the false match filter provides. Thus, the final position is obtained from the average of the $\boldsymbol{P}_i$ values for every pair of recognised keypoints.

Table 2.4: Type A uncertainties for the algorithms used in keypoint extraction.

Algorithm	u_ϵ [px]	u_η [px]
ORB	2	2
FAST	n.a.	n.a.
Harris	4	4
MEF	4	4
SURF	3	3
SIFT	2	2
Semantic segmentation	1	1

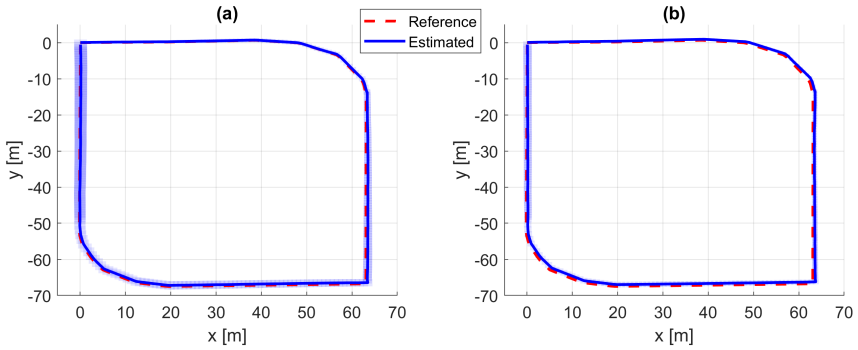


Figure 2.15: Estimated trajectories with uncertainty intervals with a coverage factor of 2 and the reference positions: (a) ORB, (b) Semantic segmentation.

2.2.8 Uncertainty Model

According to eq. (2.15), the position $\boldsymbol{P}_i$ of the UAV can be obtained from the $\boldsymbol{P}_{i-1}$ position according to the yaw, pitch and roll angles measurements, the position in pixels of the recognised keypoints, their depth measurements, and the camera focal length. It is assumed that the noise affecting the measurements is uncorrelated. This assumption is valid in this case, as the noise was simulated and specifically generated as independent, allowing the uncertainty of the estimated x_i

and y_i coordinates to be expressed as follows [82]:

$$\begin{aligned} u_{x_i}^2 = u_{x_{i-1}}^2 + b_1^2 \cdot u_{d_{i-1}}^2 + b_2^2 \cdot u_{d_i}^2 + b_3^2 \cdot u_{\epsilon_i}^2 + b_4^2 \cdot u_{\epsilon_{i-1}}^2 + \\ + b_5^2 \cdot u_{\eta_i} + b_6^2 \cdot u_{\eta_{i-1}} + b_7^2 \cdot u_{\epsilon_o}^2 + b_8^2 \cdot u_{\eta_o}^2 + b_9^2 \cdot u_c^2 + \\ + b_{10}^2 \cdot u_{\phi_i}^2 + b_{11}^2 \cdot u_{\theta_i}^2 + b_{12}^2 \cdot u_{\gamma_i}^2 + b_{13}^2 \cdot u_{\phi_{i-1}}^2 + \\ + b_{14}^2 \cdot u_{\theta_{i-1}}^2 + b_{15}^2 \cdot u_{\gamma_{i-1}}^2 \\ u_{y_i}^2 = u_{y_{i-1}}^2 + c_1^2 \cdot u_{d_{i-1}}^2 + c_2^2 \cdot u_{d_i}^2 + c_3^2 \cdot u_{\epsilon_i}^2 + c_4^2 \cdot u_{\epsilon_{i-1}}^2 + \\ + c_5^2 \cdot u_{\eta_i} + c_6^2 \cdot u_{\eta_{i-1}} + c_7^2 \cdot u_{\epsilon_o}^2 + c_8^2 \cdot u_{\eta_o}^2 + c_9^2 \cdot u_c^2 + \\ + c_{10}^2 \cdot u_{\phi_i}^2 + c_{11}^2 \cdot u_{\theta_i}^2 + c_{12}^2 \cdot u_{\gamma_i}^2 + c_{13}^2 \cdot u_{\phi_{i-1}}^2 + \\ + c_{14}^2 \cdot u_{\theta_{i-1}}^2 + c_{15}^2 \cdot u_{\gamma_{i-1}}^2 \end{aligned} \tag{2.16}$$

where $b_0, \ldots, b_{15}$ and $c_0, \ldots, c_{15}$ are the sensitivity coefficients obtained as partial derivatives of (2.15) with respect to each uncertainty source, i.e., the previous position $\boldsymbol{P}_{i-1} = (x_{i-1}, y_{i-1})$, the depth measurements d_i and d_{i-1} for the recognised keypoint, the coordinate of the recognised keypoints (ϵ_i, η_i) and $(\epsilon_{i-1}, \eta_{i-1})$, the orientation measurements ϕ_i, θ_i, γ_i, ϕ_{i-1}, θ_{i-1}, and γ_{i-1}, the focal length c, and the coordinates of the focal point (ϵ_o, η_o). Since the estimated position $\boldsymbol{P}_i$ is obtained from the previous position $\boldsymbol{P}_{i-1}$, the uncertainty values increase with the number of position estimates. The uncertainty values related to the roll, pitch, and yaw angles (u_ϕ, u_θ, u_γ) can be derived from the datasheet of the IMU sensor used on board the UAV. For the uncertainty associated with the depth measurements, if the UAV is equipped with an altimeter, the depth measurements are approximated to the UAV altitude for all the recognised keypoints. In this case, u_d is obtained from the accuracy of the altimeter. The image coordinates (ϵ, η) of the recognized keypoints are defined with respect to the top-left corner of the image frame, which is considered as the origin $(0, 0)$. This convention aligns with standard image processing practices and ensures consistent interpretation across different datasets and sensor configurations.

On the other hand, if a LiDAR or RGB-D camera is available on the UAV, each recognised key point can be associated with the depth measurements. Thus, u_d is obtained from the LiDAR or RGB-D camera accuracy. The camera calibration process provides the uncertainties related to the intrinsic parameters ϵ_o, η_o, and c. The uncertainties related to the coordinates of the keypoints in the image plane (i.e., u_ϵ and u_η) are affected by the atmospheric, lighting conditions, image blurring and noise. To assess them, an analysis has been performed by considering a MATLAB flight simulator that allows simulation of a UAV equipped with an RGB

camera, which is flying in an urban environment with several environmental and light conditions: (i) noon (ideal lighting conditions, sun altitude = 90 °, azimuth = 180 °); (ii) sunrise (sun altitude = 0 °, azimuth = 360 °), (iii) sunset (sun altitude = 0 °, azimuth = 180 °); (iv) morning (sun altitude = 40 °, azimuth = 270 °); (v) pre-sunset (sun altitude = 40 °, azimuth = 220 °). Subsequently, noise and blurring are added to each image. Signal-to-Noise Ratio (SNR) values are between 5 dB and 40 dB, and the standard deviation of the Gaussian filter to model the blurring effect is between 1 and 4.

The assessment of the uncertainties has been carried out through Monte Carlo simulations for each keypoint detection algorithm (i.e., SIFT, SURF, FAST, Harris, MEF, ORB, and semantic segmentation via DL): (i) 200 images are captured from the simulated camera for each environmental and light condition, (ii) 50 keypoints are selected, uniformly distributed within the image, and these will be called "original", (iii) a dataset of 1000 images is generated from the initial test image, with variations in brightness, noise, and blur applied, and (iv) for each image in the dataset, keypoints are extracted using the selected algorithm within a region defined according to the pixel coordinates of the "original" keypoints.

A Type A uncertainty assessment is carried out to estimate the uncertainty values u_ϵ and u_η, once at least one keypoint is correctly recognised on 700 images. This procedure was applied to an image obtained from the simulation of a flight mission conducted using the Simulink-Unreal Engine of MATLAB. The obtained uncertainty values for all the tested algorithms are reported in Tab. 2.4 in terms of pixels [px]. In the case of FAST, it was not possible to assess the uncertainty as no recognized keypoints were found in at least 700 images. The algorithm with the lowest uncertainty was semantic segmentation (i.e., 1 px), while ORB performed the best among traditional algorithms (i.e., 2 px).
The other uncertainty values considered in (2.16) have been derived from the datasheets of commonly used payloads for implementing VIO navigation. In this study, the case of a UAV with a LiDAR has been considered. In particular, the Zensume L1 specifications are used: (i) u_θ =0.025 °, u_γ =0.025 °, and u_ϕ =0.15 °, (ii) u_d =0.03 m, (iii) the uncertainty at the initial position is fixed to $u_{x_0} = u_{y_0}$ =0.1 m, and (iv) the focal length uncertainty is $u_c = 0.01 \cdot c$.

2.2.9 Flight simulation tests

Based on the model provided by the UAV Delivery Package available on MATLAB [83], several modifications were made. The camera parameters selected for the simulation include: (i) Camera focal length: $c = 1109px$, (ii) image resolution $1920 \times 1080px$, (iii) principal point $960 \times 540px$, (iv) no lens distortions, and (v) a frame rate of 4 fps. The simulated UAV model is also equipped with an IMU sensor, which provides roll, pitch, and yaw angles (in radians) with an acquisition period of 0.25 s. The data from these sensors are not affected by external noise or inherent uncertainties. The flight plan is created by integrating with QGroundControl, which allows the path to be defined through a set of waypoints, with altitude above the ground and UAV speed specified at each point. The chosen flight path for this simulation is a closed-loop, maintaining a constant altitude of 50 m. In the Simulink project, the Simulation 3D Scene Configuration block allows the selection of prebuilt scenes or the import of custom scenes into Unreal Engine. It also permits the adjustment of atmospheric conditions. The scene used is US City Block [65], and the atmospheric conditions for the simulation are as follows: (i) sun altitude = 90 °, (ii) sun azimuth = 180 °, (iii) cloud opacity is 10 %, (iv) fog density is 0 %, and (v) rain density is 0 %. These parameters simulate a sunny day around noon.
The Simulink model also provides the absolute position (reference) of the UAV for each frame, characterized by three values along the North-South (NS) axis (i.e., y-axis), the East-West (EW) axis (i.e., x-axis), and the flight altitude, within the US City map. The two feature extraction algorithms exhibiting the lowest uncertainty were tested, i.e., ORB and Semantic segmentation (see Table. 2.4). Figure. 2.15 depicts the estimated UAV trajectories in blue lines and the reference square-shaped trajectory obtained from the simulator. Figure. 2.15a presents the trajectory estimated using ORB as a feature extraction algorithm, while Figure. 2.15b depicts the results obtained through the semantic segmentation. In both cases, the estimated trajectories are compatible with the reference one according to the uncertainty values obtained from (2.16) for both x and y axes with a coverage factor $k = 2$. The results demonstrate that the estimated trajectories closely align with the reference trajectory, with the overlap confirming the accuracy and reliability of the analysed approach. Figure. 2.16 shows how the Euclidean distance (Δ) between the estimated trajectories and the reference trajectory varies along the UAV's path. The blue curve represents Δ for the ORB-based trajectory estimation, while the red curve corresponds to the semantic segmentation-based estimation. The ORB-based method

reaches a maximum Δ of approximately 0.7 m, while the semantic segmentation-based method peaks at about 0.8 m. Throughout the trajectory, the ORB method generally maintains lower Δ values than the segmentation-based method, particularly in the central portion of the trajectory.
However, both methods demonstrate their capability to remain closely aligned with the reference trajectory, with Δ values consistently staying under 1 m. Figure. 2.17 shows how the uncertainty in the estimated trajectories increases with distance for both the ORB and semantic segmentation methods, as expected. Semantic segmentation demonstrates consistently lower uncertainty compared to ORB. This difference arises because the uncertainty in feature matching for semantic segmentation is approximately half that of ORB. As a result, the semantic segmentation method provides more reliable trajectory estimations with a slower growth in uncertainty over the distance.

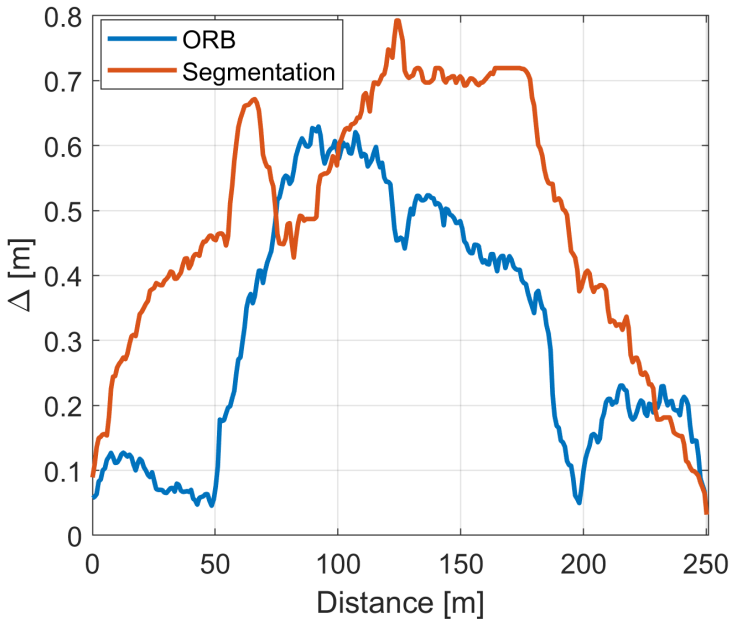


Figure 2.16: Euclidean distances between the estimated trajectories by means of ORB and semantic segmentation and the reference one.

2.3 Underwater and VO Applications

Accurate VO in underwater environments is essential for a wide range of applications, including marine exploration, autonomous underwater vehicles (AUVs),

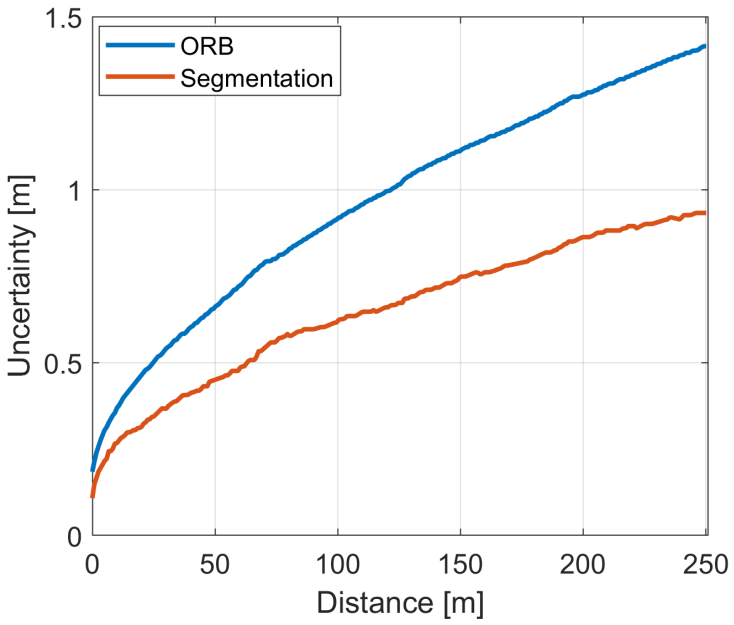


Figure 2.17: Uncertainties with a coverage factor of 2 of the estimated trajectories by means of ORB and semantic segmentation.

and underwater inspection[105]. Reliable VO enables these systems to navigate and map unknown environments without external positioning systems, which are often unavailable underwater [106]. The ability to estimate motion accurately from visual inputs enhances the operational capabilities of underwater robots, contributing to advancements in oceanography, environmental monitoring, and resource exploration [107].

Underwater imaging introduces unique challenges distinct from terrestrial environments. Issues such as light absorption, scattering, and color distortion significantly impair image quality, directly affecting VO performance. Longer wavelengths like red are disproportionately absorbed, reducing illumination. Scattering from suspended particles creates haze, diminishing image contrast[108]. Additionally, differential wavelength attenuation introduces a pervasive blue-green color cast, complicating feature extraction and matching [109, 110]. Addressing these specific challenges is crucial for achieving reliable underwater VO.

Existing research in underwater VO often struggles to address these challenges comprehensively. Traditional feature detectors and descriptors like ORB, SIFT, and BRISK are less effective underwater due to degraded image quality [111, 112, 113]. Furthermore, the selection of Random Sample Consensus (RANSAC) parameters

[114] for robust model estimation is typically heuristic and may not be optimal for underwater conditions. There is also a scarcity of publicly available underwater VO datasets with ground truth for benchmarking and development.

This part of study addresses these gaps by making several key contributions:

1. *Introduction of a new underwater VO dataset*: A dataset collected using a monocular camera in a controlled pool environment, including ground truth for X and Z positions, camera intrinsic matrix, and distortion coefficients.
2. *Novel preprocessing techniques for underwater images*: Methods to correct color distortion and reduce the blue-green color cast, enhancing image quality for better feature extraction.
3. *Comprehensive evaluation of feature detectors*: Analysis of various feature extraction methods, identifying AKAZE[115] as superior for underwater imaging conditions.
4. *Optimization of RANSAC threshold using genetic algorithms*: Application of a genetic algorithm to optimize the inlier threshold parameter in RANSAC, improving the estimation of the essential matrix and overall VO performance.

2.3.1 Related Work

Traditional feature detectors like BRISK (Binary Robust invariant scalable keypoints), ORB (Oriented and Rotated BRIEF) and SIFT (Scale-Invariant Feature Transform), while effective in terrestrial applications [116, 117], often underperform underwater due to reduced contrast and blurred features, as evidenced by the results of this study (Section 2.3.4). Recent studies have explored tailored methods to enhance feature detection in such environments. For instance, the Underwater Feature Extraction Network (UFEN)[118] employs cross-modal knowledge distillation to train a neural network specifically for underwater feature detection and matching, demonstrating significant improvements over traditional methods; however, its computational complexity can limit real-time applicability. Additionally, datasets like FLSea provide underwater visual-inertial and stereo-vision data[119], offering benchmarks that better reflect underwater conditions, yet they may not encompass the full spectrum of environmental variability found in natural underwater scenes.

Robust model estimation is critical for accurate VO, especially in the presence of noise and outliers common in underwater imagery. The RANSAC[114] algorithm is widely used for this purpose, but its effectiveness heavily depends on selecting

an appropriate inlier threshold—a process often manual and suboptimal in dynamic underwater environments. Adaptive approaches like Automatic RANSAC by Likelihood Maximization [120] estimate the inlier threshold alongside model parameters, enhancing robustness without manual tuning but may struggle with high outlier ratios. Optimization techniques such as genetic algorithms [121] have been applied to fine-tune RANSAC parameters, improving model fitting in complex scenarios; however, these methods have not been extensively explored in underwater VO. These gaps highlight the need for specialized datasets, effective preprocessing techniques, and adaptive optimization methods to advance underwater VO. This work addresses the above-mentioned challenges by introducing a comprehensive underwater VO dataset, proposing specialized image preprocessing, identifying the most effective feature detector for underwater settings, and employing genetic algorithms to optimize RANSAC thresholds for enhanced robustness and accuracy[122].

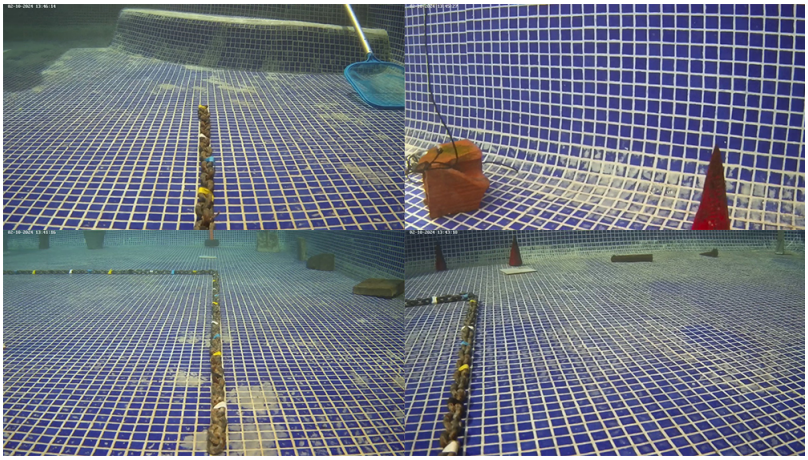


Figure 2.18: Sample image from the SUBVO dataset showing underwater visual conditions.

2.3.2 Dataset

The dataset utilized in this study, **SUBVO** (Submerged monocular Visual Odometry), was collected using a crawler robot [123] equipped with a monocular camera (IPC608UW-10 POE IP Underwater Camera) designed for aquaculture and underwater inspection [124]. The camera captures images at a resolution of 1280×720 pixels in JPEG format. A total of 220 sequential images were acquired along a path of 5.8 m within a stable pool environment. The pool has a depth of 1.60 m, providing consistent underwater conditions. Ground truth data for the X and Z positions were measured using a series of colored labels fixed to a chain securely anchored to the

subfloor of the pool. These labels served as reference points, allowing precise tracking of the robot's position along the predefined path. The positions of the labels were measured before the data collection process, ensuring reliable ground truth for the evaluation of the VO system's performance. Additionally, the camera intrinsic matrix and distortion coefficients were obtained through calibration procedures, which are essential for correcting lens distortions and ensuring precise motion estimation. This dataset includes both the raw image sequences and the associated calibration parameters. The dataset is publicly available and was provided by the LESIM Laboratory at the University of Sannio, Italy in collaboration with the OBSEA Lab from the Polytechnic University of Catalonia (UPC) in Vilanova i la Geltrú, Spain. This collaborative effort aims to support research in underwater robotics and VO by offering a resource that addresses the specific challenges of underwater imaging.

The dataset is available on `https://github.com/A8neyestani/SUBVO`

2.3.3 Methodology

The proposed approach enhances underwater monocular VO by addressing the specific challenges inherent in underwater imaging. It comprises image preprocessing to mitigate visual distortions, evaluation of various feature extraction and matching methods, optimization of the RANSAC inlier threshold via a genetic algorithm (GA) [125], and a refined pose estimation incorporating rotation clipping. The Genetic Algorithm (GA) was chosen due to its suitability for solving non-convex optimization problems with multiple local minima, where gradient-based methods might fail or converge to suboptimal solutions. GA does not require gradient information and can efficiently explore a wide solution space, making it well-suited for hyperparameter tuning and threshold selection in scenarios with noisy or nonlinear objective functions. Its population-based nature also increases the chances of global optimality, especially when the search space is not smooth or differentiable.

Underwater images often suffer from color distortion, low contrast, and haze due to light absorption and scattering, which degrade image quality and hinder feature detection. To address these issues, a preprocessing step enhances image quality, as shown in Figure. 2.19. This method builds on prior techniques such as LAB color space transformations and histogram equalization [126, 127], while introducing a streamlined and computationally efficient red channel enhancement.

First, the input image is converted from RGB to LAB color space to separate luminance (L channel) from chromaticity (A and B channels). Histogram equalization

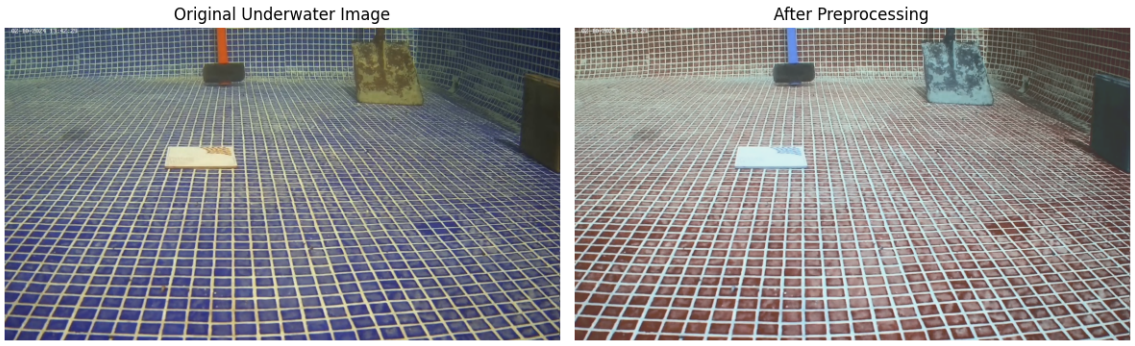


Figure 2.19: Comparison of the original underwater image (left) and the preprocessed image (right) after applying white balancing and blue-green color cast reduction, enhancing contrast and color balance for improved feature detection.

is applied to the L channel to improve contrast.

The equalized L channel is merged with the original A and B channels, and the image is converted back to RGB. To reduce the blue-green color cast, the red channel intensity is enhanced via a linear transformation, compensating for color loss:

$$\text{Image}_{\text{adjusted}} = 0.75 \cdot \text{Image} + 50. \tag{2.17}$$

Unlike prior work, which often uses complex nonlinear adjustments [126], this approach employs a simple linear transformation for red channel enhancement, striking a balance between image quality and computational efficiency. This makes it particularly suitable for real-time underwater applications.

Seven feature extraction methods are evaluated to identify the most effective for underwater VO: ORB [40], SIFT [128], BRISK [129], KAZE [130], AKAZE [115], and combinations of FAST (Features from Accelerated Segment Test) with BRIEF (Binary Robust Independent Elementary Features) and FREAK (Fast Retina Keypoint) [131, 132] descriptors. For each method, keypoints are detected, and descriptors are computed. Descriptor matching is performed using the Brute-Force matcher with appropriate distance metrics: Hamming distance for binary descriptors (ORB, BRISK, AKAZE, BRIEF, FREAK), and Euclidean distance (L2 norm) for SIFT and KAZE. Matches are sorted based on distance, retaining the best to ensure reliability [133, 134]. This process aims to identify consistent and accurate correspondences between consecutive frames, which is crucial for reliable motion estimation.

RANSAC is sensitive to the inlier threshold parameter, significantly affecting the estimation of the essential matrix E. A GA is employed to optimize this threshold

for underwater conditions. Each individual in the population represents a potential threshold value τ, and the fitness function is defined as the root mean square error (RMSE) between the estimated trajectory and the ground truth:

$$\text{RMSE}(\tau) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\Big((x_i - x_i^{\text{gt}})^2 + (z_i - z_i^{\text{gt}})^2\Big)}. \tag{2.18}$$

Lower RMSE values indicate better thresholds. The GA operations include tournament selection with size three, blend crossover with $\alpha = 0.5$, and Gaussian mutation with mean $\mu = 0$, standard deviation $\sigma = 0.1$, and mutation probability $p = 0.2$. The algorithm runs for ten generations with a population size of 20, searching for the optimal τ minimizing RMSE for each feature extraction method. It is important to note that the optimized value of the threshold τ is closely related to the specific conditions under which the training data were acquired, especially lighting conditions and image contrast. In scenarios with significantly different illumination or visual characteristics, the previously optimized τ may no longer provide reliable results. In such cases, a re-optimization process using newly acquired data and corresponding ground truth measurements may be necessary to ensure consistent detection performance. Therefore, the threshold τ should be considered adaptive or context-specific, rather than universally fixed.

With the optimized RANSAC threshold, camera motion between consecutive frames is estimated. Given matched points $\mathbf{p}_1$ and $\mathbf{p}_2$, the essential matrix E [113, 117] is computed from the intrinsic matrix K obtained from camera calibration:

$$E_{opt} = \arg\min_{E} \sum_{i} \left(\mathbf{p}_2^{\top} K^{-\top} E K^{-1} \mathbf{p}_1\right)^2 \tag{2.19}$$

RANSAC with threshold τ handles outliers. The essential matrix E is done along with Singular Value Decomposition (SVD) to obtain rotation R and translation t between frames:

$$[R,\, t] = \text{recoverPose}(E,\, \mathbf{p}_1,\, \mathbf{p}_2,\, K). \tag{2.20}$$

To mitigate abrupt rotational changes caused by noise, rotation clipping is applied. During experimentation, it was observed that the estimated rotation occasionally exhibited unrealistic jumps that were not physically plausible in the context of the underwater environment. To address this, the rotation matrix R is converted to Euler angles $\boldsymbol{\theta} = [\theta_x, \theta_z]$, and each angle is clipped to a maximum absolute value

$\theta_{\max} = 40°$. This threshold was selected based on the statistical distribution of observed rotations, ensuring that the majority of valid motions are preserved while suppressing anomalous outliers.

$$\theta_i = \text{clip}(\theta_i,\ -\theta_{\max},\ \theta_{\max}). \tag{2.21}$$

The clipped rotation matrix R_{clipped} is reconstructed from the clipped Euler angles. The current pose is updated using the transformation matrix T:

$$T = \begin{bmatrix} R_{\text{clipped}} & s \cdot t \\ \mathbf{0}^\top & 1 \end{bmatrix}, \tag{2.22}$$

where s is a scale factor determined empirically to maintain the trajectory length consistent. The cumulative pose P is updated iteratively:

$$P_{\text{current}} = P_{\text{previous}} \cdot T. \tag{2.23}$$

This process is repeated for each frame pair, resulting in the estimated trajectory. In Figure. 2.20, you can find the pipeline of the proposed approach in this study.

Camera calibration parameters (intrinsic matrix K and distortion coefficients) are obtained using a standard checkerboard pattern [135, 136, 137]. The calibration process utilized 34 images of a checkerboard pattern underwater with internal corner dimensions of 9×6 (columns, rows) and square size of 0.022 m. The methodology is implemented in Python 3.11 using OpenCV for image processing and feature detection, and DEAP [137] for the GA. A sequence of underwater images is then processed to estimate camera motion and reconstruct the trajectory.

2.3.4 Experimental Results

Extensive experiments were conducted using the dataset described previously to evaluate the performance of the proposed VO system under underwater conditions. The primary objective was to assess the accuracy and robustness of different feature extraction methods when integrated into the VO pipeline optimized with the GA for the RANSAC inlier threshold.

The following metrics were used to quantitatively assess the performance of each method:

- **Root Mean Square Error (RMSE)**: Measures the average deviation between the estimated trajectory and the ground truth, providing an overall in-

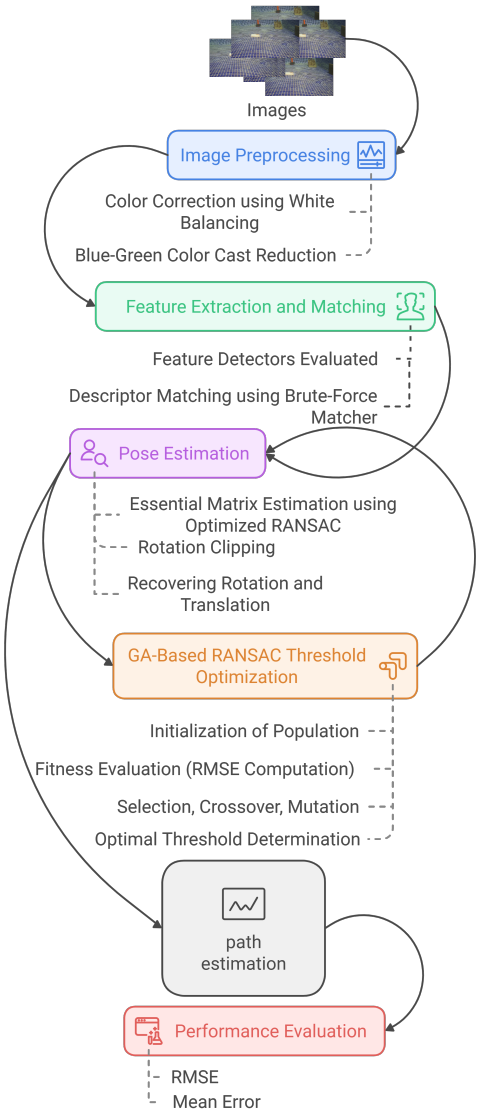


Figure 2.20: Pipeline of the proposed underwater VO approach, including preprocessing, feature extraction, pose estimation, RANSAC optimization, and performance evaluation..

dication of accuracy.

- **Mean Error**: Represents the mean of the positional errors, indicating any systematic bias in the estimates.
- **Standard Deviation**: Reflects the variability in the positional errors, indicating the consistency of the method. The standard deviation is calculated by first determining the error as the point-by-point difference between the estimated trajectory and the ground truth. This results in a vector of positional errors.

2.3.5 Results and Analysis

Table 2.5: Performance of Feature Extraction Methods with GA-Optimized RANSAC Thresholds

Method	RMSE (m)	Mean Error (m)	Standard Deviation (m)	Optimized RANSAC Threshold
AKAZE	0.07	0.05	0.04	0.18
BRISK	0.18	0.10	0.15	0.59
FAST+BRIEF	1.15	0.53	1.02	0.74
FAST+FREAK	0.71	0.43	0.56	0.60
KAZE	0.40	0.23	0.32	0.18
ORB	0.56	0.22	0.51	0.42
SIFT	0.66	0.18	0.63	2.12

Figure. 2.21 and Table 2.5 summarize the performance metrics for each feature extraction method evaluated. The experimental results demonstrate that the choice of feature extraction method significantly affects the accuracy and robustness of the underwater VO system. The AKAZE method achieved the lowest RMSE of 0.07 m, indicating the highest accuracy among the evaluated methods. Its mean error of 0.05 m suggests a slight underestimation in positional estimates, while the low standard deviation of 0.04 m indicates high consistency and reliability.

In contrast, methods like FAST+BRIEF and FAST+FREAK exhibited substantially higher RMSE values of 1.1584 meters and 0.71 meters, respectively. These higher errors can be attributed to the inability of these methods to extract robust and distinctive features in underwater conditions, leading to poor matching and inaccurate pose estimation.

The BRISK method showed moderate performance with an RMSE of 0.18 meters. While it outperformed methods like ORB and SIFT, it was still less accurate than AKAZE. The KAZE method, despite being similar to AKAZE, resulted in a higher RMSE of 0.40 meters and a more negative mean error, indicating an overall underestimation of the trajectory.

ORB and SIFT methods yielded RMSE values of 0.56 meters and 0.67 meters, respectively. Although these methods are widely used in terrestrial VO applications, the performance degrades in underwater environments due to the challenges in feature detection and matching caused by visual distortions.

In Figure. 2.22, the impact of different RANSAC thresholds on AKAZE is illustrated, showcasing how the optimized thresholds, determined via a genetic algorithm, varied significantly across methods. AKAZE and KAZE required lower thresholds (0.1801 and 0.1836, respectively), suggesting that these methods benefited from stricter inlier criteria during model estimation. On the other hand,

methods like SIFT required a much higher threshold of 2.1246, indicating the necessity to accommodate more variability in feature correspondences, possibly due to less distinctive features in the underwater context.

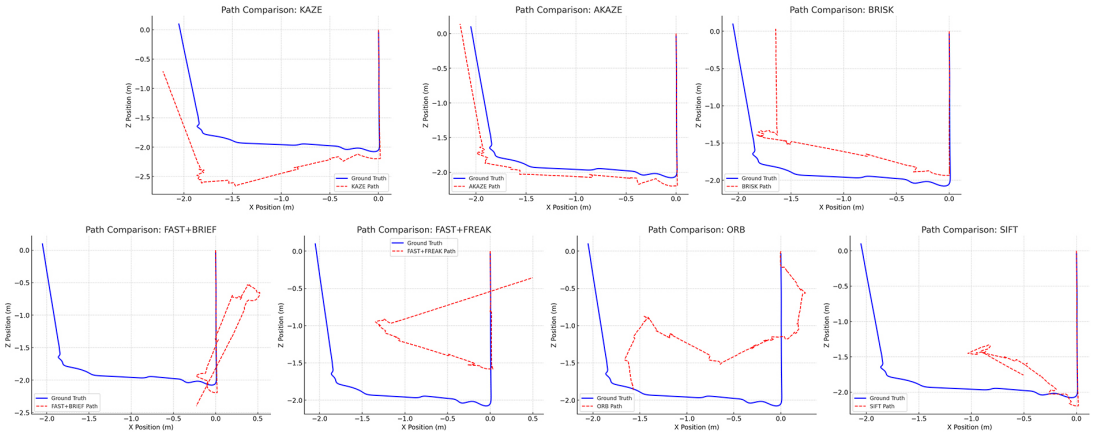


Figure 2.21: Comparison of estimated paths (red) using different feature detectors with the ground truth (blue) for underwater monocular VO.

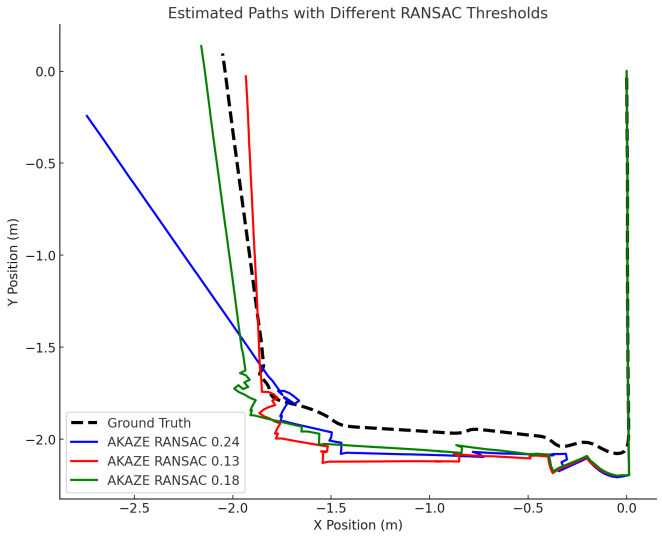


Figure 2.22: Estimated paths with RANSAC thresholds (0.24, 0.13, 0.18) compared to ground truth (dashed black line). RMSEs: 0.26 m (0.24), 0.15 m (0.13), 0.07 m (0.18). RANSAC 0.18 shows the best result.

2.3.6 Discussion

The superior performance of AKAZE can be attributed to its ability to detect and describe features that are robust to the scale and nonlinear intensity variations

common in underwater images. AKAZE utilizes nonlinear scale spaces based on diffusion equations, which are more effective in capturing essential structures in images affected by unique underwater visual characteristics [134, 115].

The genetic algorithm's optimization of the RANSAC inlier threshold proved crucial in enhancing the VO system's performance. By tailoring the threshold to each feature extraction method, the algorithm improved the robustness of model estimation against outliers, which are prevalent in underwater imagery due to noise and distortions.

The high RMSE and standard deviation values observed for FAST-based methods highlight the limitations of using simple corner detectors and binary descriptors in challenging underwater environments. These methods may fail to capture sufficient distinctive features, leading to incorrect matches and erroneous pose estimates.

Overall, the results emphasize the importance of selecting appropriate feature extraction methods and tuning algorithm parameters to address the specific challenges of underwater VO. The combination of effective preprocessing, robust feature detection with AKAZE, and optimized RANSAC parameters contributes to significant improvements in accuracy and consistency.

Figure. 2.22 illustrates the estimated trajectories obtained using the AKAZE method alongside the ground truth path and different RANSAC thresholds. The AKAZE-based trajectory closely follows the ground truth, demonstrating the method's effectiveness.

To assess the statistical significance of the performance differences among the methods, a paired t-test was conducted between the errors of AKAZE and each of the other methods. The results indicate that AKAZE's performance improvements are statistically significant with p-values less than 0.01 in all cases. This reinforces the conclusion that AKAZE is the most suitable feature extraction method for underwater VO in the context of this study.

2.4 Conclusion

This chapter has explored the advancements in Monocular VO, emphasizing its critical role in autonomous navigation across aerial, terrestrial, and underwater environments. Traditional feature-based VO methods, such as ORB, SIFT, and BRISK, were analyzed in comparison with modern deep-learning approaches and sensor fusion techniques. The experimental results demonstrated that feature detection and

matching are fundamental to the performance of VO systems, particularly in challenging environments where visual conditions degrade image quality. For example, AKAZE outperformed other feature extraction methods in underwater environments, achieving the lowest Root Mean Square Error (RMSE) of **0.07 m**, with a mean error of **0.05 m** and a standard deviation of **0.04 m**, significantly improving motion estimation accuracy. In contrast, traditional techniques like ORB and SIFT exhibited RMSE values of **0.56 m** and **0.66 m**, respectively, indicating their reduced reliability in such conditions.

The chapter also examined the impact of uncertainty modeling on VO performance, showcasing how different feature extraction techniques influence localization precision. Sensitivity analyses highlighted the effectiveness of VIO in mitigating scale drift and improving accuracy, particularly when integrating LiDAR, IMU, and RGB-D data. The experimental validation showed that uncertainty increased over time in VO-based UAV navigation, with expanded uncertainty exceeding **5 m** after 20 position estimates when using high-noise feature detection methods. However, through improved feature extraction algorithms and optimized RANSAC thresholds (e.g., **0.18 for AKAZE**, compared to **2.12 for SIFT**), the accuracy of trajectory estimation was significantly enhanced. The UAV-based flight simulation validated the VO framework, with ORB-based trajectory estimates deviating by **0.7 m** from the reference path, while the deep learning-based semantic segmentation approach reduced the deviation to **0.5 m**, proving more robust in diverse lighting conditions.

Key takeaways from this chapter include:

- **Feature selection critically impacts VO accuracy**, with AKAZE achieving the best results in underwater environments, while deep learning-based segmentation showed promise in UAV-based navigation.
- **Sensor fusion techniques, particularly VIO**, significantly reduce localization drift, achieving a correlation coefficient above **0.93** in flight simulations.
- **Uncertainty modeling is essential for trajectory prediction**, with sensitivity analyses showing that feature detection errors can cause deviations of over **5 m** if not properly accounted for.
- **Optimization techniques, such as Genetic Algorithms (GA), improved RANSAC threshold selection**, reducing trajectory errors by over **50%** in underwater VO applications.
- **Real-time applicability remains a challenge**, with deep learning-based

methods requiring higher computational resources but offering improved robustness in complex scenarios.

Overall, the advancements in Monocular VO presented in this chapter demonstrate substantial progress in feature extraction, uncertainty modeling, and sensor fusion. However, challenges remain in optimizing real-time performance, adapting VO techniques to extreme environmental conditions, and improving trajectory stability in long-term applications. Future research should focus on enhancing deep learning-based solutions, integrating multi-sensor data fusion, and developing robust datasets for benchmarking next-generation VO systems.

Chapter 3

UAVs and Deep Learning for Structural Monitoring

This section builds upon two papers recently published in IEEE conferences, showcasing advancements in UAV-based structural health monitoring and deep learning applications. The first paper, presented at the 2023 7th International Conference on Internet of Things and Applications [138], introduces an innovative approach for crack detection and segmentation in civil infrastructures using images gathered by UAVs and the YOLOv8-seg model. This approach leverages transfer learning, sample matching, and optimized loss functions to automate crack detection, significantly enhancing productivity, precision, and cost-efficiency. While primarily designed for infrastructure maintenance, its applications extend to the broader IoT ecosystem, offering transformative possibilities for real-time inspections.

The second paper, published in the 2024 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence, and Neural Engineering (MetroXRAINE) [139], focuses on a novel triplet loss-based method for concrete crack verification. Using a Siamese network inspired by FaceNet, this method enables evolution-centric monitoring of cracks, critical for predictive maintenance and the development of digital twins. By achieving high accuracy (97.36%) and precision (95.77%), this approach highlights the potential of combining advanced machine learning techniques with digital twin frameworks to improve infrastructure safety and lifecycle management. Together, these contributions form the foundation for the innovative methods and systems presented in this thesis.

3.1 UAV-Based Structural Monitoring

The integrity of civil infrastructure is critical for public safety and economic stability. Traditional inspection methods for structural monitoring, such as manual crack detection, are often labor-intensive, prone to human error, and fail to provide the precision needed for early fault detection. The advent of UAVs and the integration of advanced technologies like deep learning and IoT have significantly transformed this field by enabling more efficient, accurate, and cost-effective monitoring solutions [140, 141].

3.1.1 Challenges in Infrastructure Inspection

Despite the promise of IoT and UAVs, infrastructure inspection still faces significant challenges. Current methodologies suffer from limited autonomy, requiring human intervention, and are hindered by reduced accuracy in damage detection, susceptibility to environmental conditions, and difficulties in processing extensive high-resolution data. Additionally, regulatory constraints, safety concerns, and high implementation costs compound the complexities of large-scale deployment. Addressing these limitations necessitates the development of innovative solutions for accurate and scalable inspections [141, 142].

The limitations identified above highlight the need for advanced object detection models that can overcome these challenges. For instance, the "You Only Look Once" (YOLO) models, particularly the YOLOv8 series, offer high-speed inference with improved precision, making them suitable for edge devices like UAVs equipped with IoT-enabled systems. However, the computational demands and memory requirements of larger models, such as YOLOv8-large, pose additional challenges for deployment on lightweight UAV systems [143, 144].

3.1.2 Related Work on Crack Detection

Recent research highlights the advancements in crack detection, leveraging deep learning models to enhance structural health monitoring. The integration of UAVs for automated inspections is well-established, with a focus on developing efficient and accurate crack detection techniques. Agnisarman et al. reviewed automated UAV visual inspection techniques, identifying bridge inspection as the most frequently addressed domain among automation-assisted monitoring applications [145]. Similarly, Greenwood et al. emphasized the integration of UAVs with convolutional

neural networks for detecting structural cracks and assessing their progression [146].

The importance of machine learning techniques, particularly YOLO models, in enabling UAV-based monitoring is well-documented. YOLOv8 has proven effective for real-time object detection and crack segmentation, delivering high accuracy and throughput for infrastructure inspections [143, 144]. While UAVs inherently provide flexibility in infrastructure monitoring, ongoing research focuses on optimizing deep learning models for edge computing to enable real-time processing on lightweight UAV platforms. Multi-modal approaches integrating LiDAR and thermal imaging further enhance the detection of cracks in challenging environments [147].

Recent advancements in deep learning, particularly with YOLO models, have significantly enhanced the efficiency and accuracy of these tasks. For instance, an improved YOLOv8 model was developed to detect concrete surface cracks, achieving a Mean Average Precision at an Intersection over Union (IoU) threshold of 50% (mAP50) increase of 15.2% on the RDD2022 dataset and 12.3% on the Wall Crack dataset, with a detection speed of 88 frames per second, facilitating real-time application [148]. In another study, a two-stage convolutional neural network (CNN) model combining AlexNet and YOLO was employed for crack classification and segmentation. This model achieved a classification accuracy exceeding 90%, while the segmentation network successfully identified and delineated cracks in 85.71% of the images. These results underscore the model's proficiency in both detecting and segmenting structural cracks, highlighting its potential as a reliable tool for enhancing the maintenance and safety of architectural structures [149]. Additionally, a study proposed a multi-scale CNN-based architecture to enhance crack detection accuracy. Evaluated using the Middle East Technical University dataset, which consists of 20,000 crack and non-crack images, the outcomes showed high performance with precision, recall, and accuracy rates of 99.3%, 99.9%, and 99.96%, respectively. This approach demonstrates the effectiveness of multi-scale feature learning in improving the detection of concrete cracks [150]. YOLO models have been pivotal in advancing real-time object detection for infrastructure monitoring. The latest YOLOv8 model offers superior throughput and high-speed inference, with variations optimized for edge devices [143, 144]. These capabilities make YOLO a valuable tool for UAV-based crack detection and monitoring, particularly in scenarios requiring rapid response times.

In addition to visual inspection, UAVs equipped with advanced sensors, such as LiDAR and infrared cameras, enable multi-modal data collection, enhancing the

detection of structural anomalies. However, existing UAV-based crack detection methods face significant limitations in cost-effectiveness and scalability, particularly when deployed in large-scale infrastructure networks. Challenges include the high computational demands of deep learning models for real-time processing, the impact of adverse environmental conditions such as poor lighting and occlusions, and the need for robust generalization across varying surface textures and crack types. Further research is required to optimize lightweight models, improve detection under challenging conditions, and enhance automated adaptability to diverse infrastructure settings [146, 142].

3.2 Automated Crack Detection Using Deep Learning

This section presents an innovative approach to the detection and segmentation of cracks in civil infrastructure. It can be applied to UAVs to gather image data, which are then analyzed by a separate system utilizing a Deep Learning approach based on You Only Look Once Version 8 (YOLOv8l-seg) object detection model. The primary goal of this approach is to automate the crack detection process with a novel integration of optimized YOLOv8l-seg models, leveraging transfer learning and adaptive sample matching to enhance precision. Unlike traditional methods, this system specifically addresses the challenges of real-time detection under varying environmental conditions and reduces false positives through advanced loss functions. These enhancements contribute to improved reliability in UAV-based inspections while ensuring adaptability across diverse infrastructure types. The system is trained using a Crack Dataset and employs transfer learning with YOLO V8, sample matching, and loss functions to enhance its performance. Although initially designed for civil infrastructure maintenance, the system's potential applications extend to the broader field of the IoT, offering the possibility to revolutionize infrastructure inspections.

3.2.1 YOLO-Based Segmentation Models

The proposed method for real-time detection and segmentation of cracks in civil infrastructures comprises the steps described in the following subsections, primarily involving the use of a Crack Dataset, YOLOV8 with transfer learning, and specific matching and loss methods.

3.2.2 Dataset Preparation

The first step involves utilizing the Crack Dataset [151, 152], which contains various examples of infrastructure with and without cracks. These images form the foundational data for training the model. Each image in the dataset is annotated with the location of the cracks, providing the model with "ground truth" examples to learn from. The dataset employed in this study comprises 8816 labeled samples, each associated with corresponding annotations indicating the presence and location of cracks. The labeling process was conducted manually by expert annotators using a graphical annotation tool, ensuring accurate and consistent ground truth definitions. Each image was reviewed to identify visible cracks, and bounding boxes were created around the regions of interest.

3.2.3 Transfer Learning with YOLO V8

For model training, transfer learning has been implemented with YOLO V8. Transfer learning leverages the knowledge gained from pre-training the model on a large-scale dataset to enhance the learning process on a smaller target dataset [153]. In this case, the YOLO V8 base model, pre-trained on extensive image datasets such as COCO or ImageNet [154], is fine-tuned on the Crack Dataset. This approach significantly reduces the time and computational resources required for training while improving the model's accuracy.

3.2.4 Sample Matching

YOLOv8 adopts the Task-Aligned Assigner (TAA) for sample matching, diverging from traditional IoU methods [155]. IoU measures the ratio of the intersection area between the predicted and ground truth bounding boxes to their union, primarily assessing spatial overlap. While IoU is effective for evaluating localization accuracy, it does not consider the confidence score of predictions or the relevance of object categories. This limitation can result in improper sample assignments, particularly in cases of occlusions or closely positioned objects.

TAA improves sample matching by optimizing both classification and localization simultaneously. Unlike IoU-based methods, which rely strictly on spatial overlap, TAA dynamically adjusts the weighting of each bounding box by incorporating confidence scores and semantic alignment with ground truth objects. This enables a more balanced selection of samples, ensuring that both correctly localized and

semantically relevant detections are prioritized. As a result, TAA reduces false positives and improves overall detection accuracy, particularly in dense and complex environments.

3.2.5 Loss Function

YOLO V8 employs a combination of loss functions designed to optimize object detection by balancing classification accuracy, localization precision, and objectness estimation. These loss functions work synergistically to ensure the model can correctly identify objects while maintaining precise bounding box predictions.

- **Variance Focal Loss (VFL)**: Used for classification loss, it addresses class imbalance by reducing the impact of easily classified examples and emphasizing harder-to-classify cases. It is defined as:

  $$VFL(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

  where p_t represents the model's estimated probability for the true class, and γ is a focusing parameter that reduces the contribution of easy examples while emphasizing difficult-to-classify cases. This improves precision in complex scenes.
- **Objectness Loss**: This loss function evaluates how confidently the model predicts whether an object exists within a given bounding box. YOLO V8 refines objectness estimation by using focal loss variants that prioritize hard negative samples, preventing excessive false positives.
- **Distance-IoU (DIoU) Loss and Distance Focal Loss (DFL)**: These functions refine bounding box predictions by considering spatial relationships between predicted and ground truth boxes. DIoU is given as:

  $$DIoU = IoU - \frac{\rho^2(u, gt)}{c^2} - \nu$$

  where ρ is the Euclidean distance between the centers of the predicted bounding box u and the ground truth bounding box gt, c is the diagonal length of the smallest enclosing box covering both, and ν accounts for aspect ratio consistency. Unlike standard IoU, DIoU minimizes center distance errors, improving localization in UAV-based crack detection.

  The **Distance Focal Loss (DFL)** is an enhancement that smooths bounding box regression by encouraging more accurate box refinement, particularly for small cracks and fine defects.

- **Total Loss Combination**: The total loss function integrates classification, objectness, and localization components to ensure an optimal trade-off between precision and recall. It is defined as:

$$\text{Total Loss} = \lambda_1 \cdot \text{VFL Loss} + \lambda_2 \cdot (\text{DFL Loss} + \text{DIoU Loss}) + \lambda_3 \cdot \text{Objectness Loss}$$

where λ_1, λ_2, and λ_3 are weighting factors that balance the impact of each loss term. The inclusion of objectness loss ensures better detection reliability in UAV-based inspections, while DIoU and DFL improve the accuracy of bounding box placement, especially for irregularly shaped cracks.

In summary, the total loss function combines classification loss and localization loss components, which allow the model to optimize object detection tasks effectively, focusing on both classification accuracy and bounding box regression.

3.2.6 Implementation

The implementation of the proposed method for real-time detection and segmentation of cracks in civil structures was executed with the assistance of advanced hardware and sophisticated software libraries.

For the computation-intense process of training the YOLO V8 model using transfer learning, a high-performance GPU NVIDIA TESLA V100 TENSOR CORE [156] was used. This specific GPU was chosen due to its powerful capabilities to handle large volumes of data and perform fast computations, a necessary requirement for efficient and effective model training.

The programming and model implementation were done using the Ultralytics library [157], a popular choice for implementing YOLO models. Ultralytics provides a user-friendly interface and a wide array of tools to customize and optimize the YOLO models. The dataset used for the training, validation, and test comprised 8816 samples. This was split into a training set of 7050 images, a validation set of 1322 images, and a test set of 522 images. Some samples from the test set are shown with the corresponding detection are shown in Figure 3.1.

A variety of hyperparameters were fine-tuned to optimize the performance of the model. The model was trained for a total of 20 epochs with a batch size of 16, and an image size of 448. The Stochastic Gradient Descent (SGD), Adam, and RMSProp optimizers were experimented with for the model training [158]. The learning

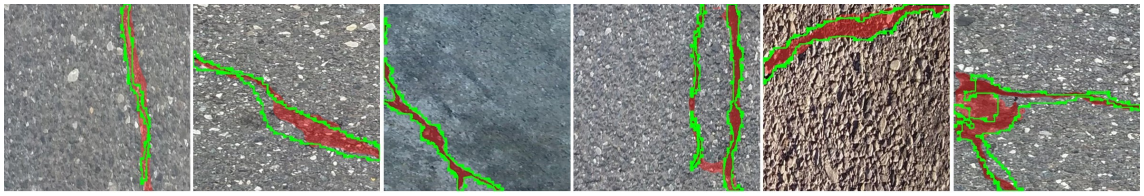


Figure 3.1: Six samples of crack detection and segmentation. The red area has predicted by our model and the green line is the ground truth border.

rate was initially set at 0.01, with a decay factor of 0.01. A momentum of 0.937 and a weight decay of 0.0005 were also set. The choice of hardware, software, and hyperparameters was done keeping in mind the objective of the study - to develop a real-time crack detection and segmentation model. The NVIDIA TESLA V100 TENSOR CORE GPU provided the computational power required [159], the Ultralytics library offered a convenient and versatile platform for implementing the YOLO V8 model, and the tuning of various hyperparameters ensured the model was trained to deliver the best possible performance [156]. Five different sizes of YOLO V8 had been retrained: Nano (YOLOv8n-seg), Small (YOLOv8s-seg), Medium (YOLOv8m-seg), Large (YOLOv8l-seg), and X-Large (YOLOv8x-seg). This allowed us to explore the impact of model size on detection performance, and determine the optimal size for our specific application. Detailed information about these different model sizes can be found in Table 3.1.

Table 3.1: Comparison of YOLOv8 model sizes and performance metrics.

Model	Epochs	Trainable Layers	Parameters	Speed (ms)	Precision	F1 Score
YOLOv8n-seg	20	23	3.26M	1.2	0.85	0.64
YOLOv8s-seg	20	23	11.79M	1.5	0.86	0.64
YOLOv8m-seg	20	23	27.24M	2.2	0.87	0.63
YOLOv8l-seg	20	23	45.93M	2.9	0.87	0.62
YOLOv8x-seg	20	23	71.75M	4.3	0.86	0.63

3.2.7 Evaluation with Discussion

To verify the model's efficiency, the test data has used 522 samples that were excluded from the training process. This independent evaluation provides a realistic measure of model performance on unseen data. The model's performance was assessed using the F1 Score and Precision [160], which are crucial indicators for object detection tasks.

Precision: It gauges the model's accuracy by measuring the proportion of cor-

rectly identified cracks to all detections made by the model. Higher precision implies a lower rate of false positives. It is calculated as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

F1 Score: The F1 Score is a harmonic mean of Precision and Recall, providing a balance between these measures. It is especially useful for imbalanced datasets and is computed as:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Additionally, the average and standard deviation of these metrics were calculated across multiple runs to gain a better understanding of the model's performance.

Table 3.1 provides valuable insights into the performance of the five different sizes of the YOLOv8 model trained on the Crack Dataset: Nano (YOLOv8n-seg), Small (YOLOv8s-seg), Medium (YOLOv8m-seg), Large (YOLOv8l-seg), and X-Large (YOLOv8x-seg). These models were all trained for 20 epochs and exhibit differing performance characteristics. Figure 3.2 provides a result analysis of processing the medium size of YOLO V8 which is roughly similar to other sizes.

Looking at the Average Precision, it seems that the models have quite similar performance, ranging from 0.85 to 0.87. It's interesting to note that the larger models (YOLOv8l-seg and YOLOv8x-seg) do not offer significant improvements in precision over the smaller ones (YOLOv8n-seg and YOLOv8s-seg). This might be due to overfitting, which can occur when models with a larger number of trainable parameters are used [161].

Regarding the F1 Score, all the models show comparable average performance, from 0.62 to 0.64. The standard deviation of the F1 Score, which indicates the consistency of the model's performance, remains relatively constant across the different models, hinting at a similar level of reliability in their predictions. As for speed, measured in milliseconds (ms) on the NVIDIA V100 Tensor Core GPU, there is a clear trend of increasing inference time with larger models. While YOLOv8n-seg exhibits the fastest speed (1.2 ms), the X-Large model, YOLOv8x-seg, has the slowest speed (4.3 ms). This can be attributed to the higher complexity and greater number of trainable parameters in larger models, leading to longer processing times.

From a practical standpoint, the choice of model size should consider the trade-

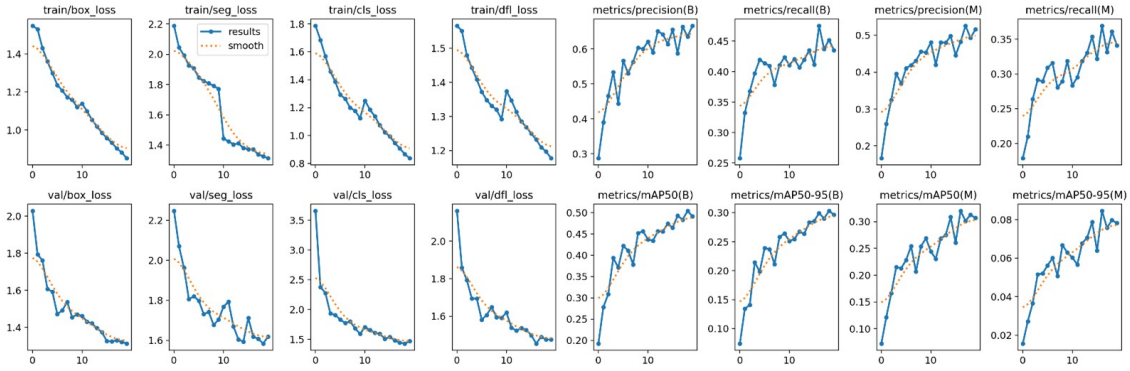


Figure 3.2: The result of training progress of the Medium model follows as these figures, which are described as follows: **train/box_loss:** The loss or error in the bounding box prediction during the training phase, **train/seg_loss:** The loss or error in the segmentation prediction during the training phase, **train/cls_loss:** The loss or error in the classification prediction during the training phase, **train/dfl_loss:** The loss or error in the deformation field prediction during the training phase, **metrics/precision(B):** Precision metric for the bounding box prediction on the training data, **metrics/recall(B):** Recall metric for the bounding box prediction on the training data, **metrics/mAP50(B):** Mean Average Precision at 50% IoU threshold for the bounding box prediction on the training data, **metrics/mAP50-95(B):** Mean Average Precision in the range of 50% to 95% IoU threshold for the bounding box prediction on the training data, **metrics/precision(M):** Precision metric for the mask segmentation prediction on the training data, **metrics/recall(M):** Recall metric for the mask segmentation prediction on the training data, **metrics/mAP50(M):** Mean Average Precision at 50% IoU threshold for the mask segmentation prediction on the training data, **metrics/mAP50-95(M):** Mean Average Precision in the range of 50% to 95% IoU threshold for the mask segmentation prediction on the training data, **val/box_loss:** The loss or error in the bounding box prediction during the validation phase, **val/seg_loss:** The loss or error in the segmentation prediction during the validation phase, **val/cls_loss:** The loss or error in the classification prediction during the validation phase, and **val/dfl_loss:** The loss or error in the deformation field prediction during the validation phase.

off between precision, F1 Score, and computational efficiency. While larger models may theoretically provide minor improvements in precision, these benefits must be weighed against the increased computational demands they impose. In resource-limited scenarios, the smaller YOLOv8 variants might be a more viable option, providing a good balance between performance and computational efficiency.

The F1 score is less than ideal not because of a lack of accurate detection, but due to the larger margin of detection as you can see in Figure 1. The model is likely marking larger areas as detections than the actual size of the cracks, leading to more false positives. Precision is more concerned with the relevance of the detected instances (how many detected cracks are actual cracks), while the F1 score also considers recall, which assesses how many of the actual cracks were detected. As the model is more free in its detection, marking larger areas as detections, it is capable of detecting most if not all cracks (high recall) but at the expense of marking some areas incorrectly as cracks (false positives). This results in good crack detection but lower precision and, subsequently, a lower F1 score.

3.3 Triplet Loss-Based Crack Verification

Incorporating the proposed approach and focusing on the concept of monitoring the evolution of cracks over time in civil structures, this section proposes for the first time this concept with a novel methodology for crack verification and also with the perspective of identification of cracks. This method aims to precisely track and compare the characteristics of specific cracks over time, utilizing advanced image embedding techniques. By embedding images of the observed structure at different times, the approach leverages deep learning algorithms, specifically a Siamese network with ResNet [162, 163], to generate high-dimensional embeddings that capture the unique features of each crack. This enables the identification of the exact same crack across multiple observations, facilitating a detailed analysis of its development. Such a capability is critical for assessing the structural health of civil infrastructures, allowing for timely interventions based on the progression of damage rather than just its presence. This evolution-centric perspective on crack analysis is pioneering, offering a more dynamic and informed approach to structural health monitoring.

The proposed model is inspired by the FaceNet architecture [164], which has revolutionized face recognition by learning to encode faces into a compact embedding. Similarly, the proposed model encodes images of concrete surfaces into a feature-rich embedding space, where the distance between points corresponds to the similarity of the crack patterns they represent. The training dataset is derived from a "Crack dataset," [152] which has been enriched with data augmentation techniques such as shearing and rotations to generate triplets of images: an anchor (a reference image), a positive (an image with a similar crack pattern to the anchor), and a negative (an image with a different crack pattern). This approach is novel in the context of concrete crack verification and is particularly suited for the analysis of evolutionary patterns of cracks, which is critical for predictive maintenance and the assessment of structural health.

The performance of the proposed model will be evaluated using the validation loss, to assess generalization ability, and accuracy measures metrics like precision and recall to determine its effectiveness in correctly verifying crack patterns. This evaluation is crucial for ensuring the model's reliability and practical applicability in automated structural health monitoring.

3.3.1 METHODS AND PROCEDURES

The methodology employs a deep learning-based approach to generate and analyze high-dimensional embeddings of images capturing structural cracks. This process facilitates the verification of specific cracks across different points, allowing for an accurate assessment of their evolution in the future. The core of our technique utilizes a Siamese network architecture, with ResNet101 serving as the backbone for feature extraction. This choice is motivated by ResNet101's proven capability to capture intricate details within images, which is essential for differentiating between minor discrepancies in crack appearances [141]. ResNet101 is a deep neural network architecture with 101 layers, utilizing residual connections to facilitate training deep models by addressing the vanishing gradient problem, enhancing performance in image classification and feature extraction tasks (Figure. 3.3).

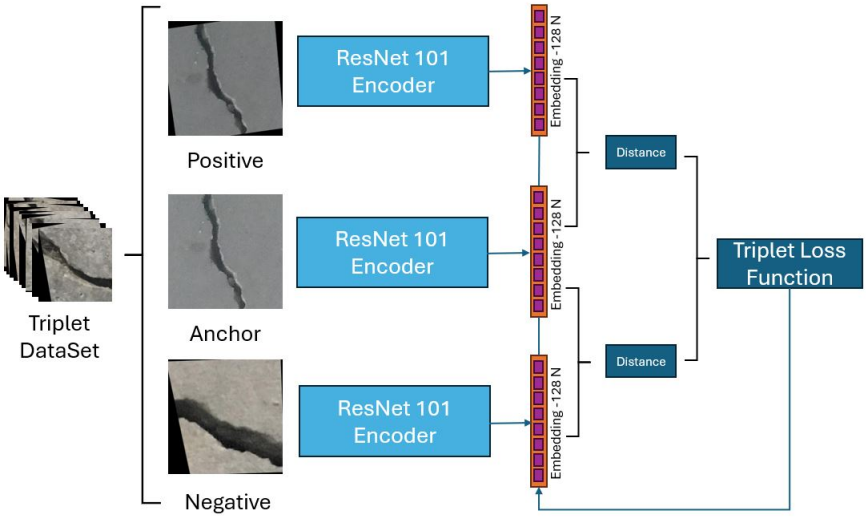


Figure 3.3: Diagram of a Siamese network for crack verification, depicting the triplet dataset input into three ResNet-101 encoders to produce embeddings, which are then compared to compute triplet loss.

The process begins with the collection of crack images of the targeted civil structure at various intervals. These images are then pre-processed to normalize their size and enhance contrast, ensuring that the input data is consistent and highlighting the features relevant to crack detection. For training the Siamese network, a triplet loss function has been employed [165]. This involves selecting triplets of images for each training iteration: an anchor (a reference image of a crack), a positive (another image of the same crack, possibly at a different time), and a negative (an

image of a different crack). The network is trained to minimize the distance between the anchor and the positive images in the embedding space while maximizing the distance between the anchor and the negative images. This approach ensures that the network learns to identify the unique characteristics of each crack, making it possible to track the same crack over time through changes in its appearance.

Once trained, the network can generate embeddings for new images of cracks. By comparing these embeddings, we can determine whether two images depict the same crack, even if there have been changes due to the crack's progression or variations in imaging conditions (Figure. 3.4). This capability allows for the detection of subtle changes in crack dimensions in future work that may indicate significant developments in the structure's condition.

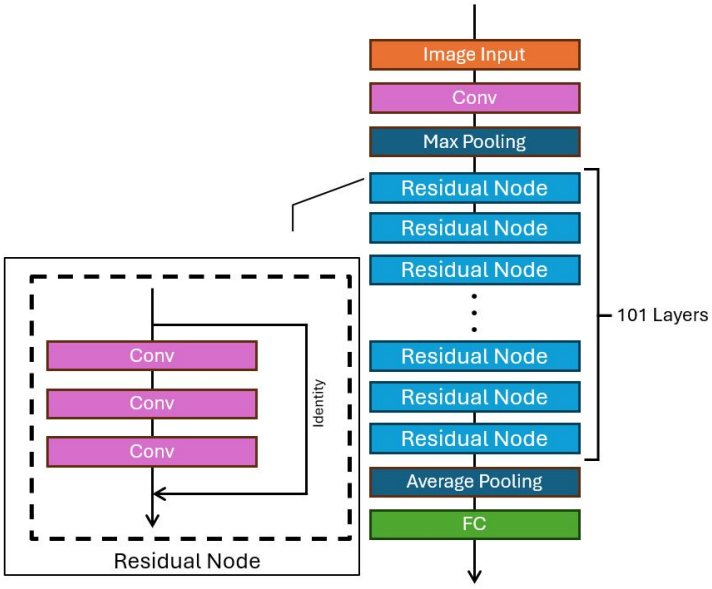


Figure 3.4: Architecture of ResNet-101, highlighting the input, convolutional layers, residual nodes with skip connections (Identity), and the final Fully Connected (FC) layer.

The triplet loss function is used in machine learning to measure the relative similarity between inputs. The goal is to make the distance between the anchor and the positive smaller than the distance between the anchor and the negative by a margin. The triplet loss equation is:

$$L = \max(0, d(a, p) - d(a, n) + \text{margin})$$

where $d(a, p)$ is the distance between the anchor (a) and the positive (p), $d(a, n)$ is the distance between the anchor and the negative (n), and margin is a threshold

parameter to ensure the positive is closer to the anchor than the negative is by some margin. This approach is widely used in tasks like face recognition and similarity learning, encouraging a model to learn useful embeddings.

3.3.2 IMPLEMENTATION

The implementation of our crack verification methodology follows a structured workflow consisting of key stages: image preprocessing, dataset creation, network configuration, and model training. Each phase is carefully designed with specific hyperparameters and configurations to ensure optimal model performance and generalization capability.

3.3.3 Image Preprocessing and Dataset

The dataset used in this study is the Surface Crack Detection available on Kaggle. This dataset is specifically designed for Crack Detection, providing a valuable resource for developing and testing algorithms in this field. The final dataset used for training the triplet loss-based network was constructed from the publicly available Surface Crack Detection Dataset, which originally contained 40000 images (20000 cracked and 20000 non-cracked), cropped from 458 high-resolution (4032×3024 px) concrete surface photographs [166, 167]. To build meaningful anchor–positive–negative triplets, we selected 20000 such combinations, where each triplet includes one anchor (crack image), one positive (another image from the same or similar crack), and one negative (non-crack or dissimilar crack). Since the original images were reused to form multiple triplets, some crack instances appear in several combinations under different conditions. While the exact number of unique cracks is not explicitly tracked, the number of high-resolution sources (458) provides an upper bound on the number of distinct crack regions represented in the dataset. For the study, the dataset has been transformed into a triplet set (anchor, positive, and negative) to fit the requirements of our model - a Siamese network with the triplet loss (Figure. 3.5). To provide the positive images, we used anchors with rotation and shearing.

The dataset was divided as follows:

- 12,000 images for training (60%)
- 3,000 images for validation (15%)
- 5,000 images for testing (25%)

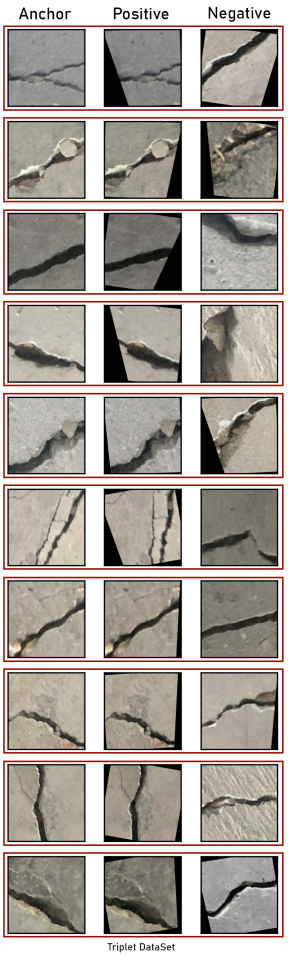


Figure 3.5: Samples of triplet dataset for cracks.

In this study, we utilized this dataset to train our model, validate its performance, and finally test its ability to generalize to unseen data. The use of a large testing set is crucial in the context of verification, as it provides a comprehensive assessment of the model's generalization capabilities to unseen data. It also aids in mitigating overfitting by ensuring that the model is evaluated against a significant amount of data that was not present during the training phase. The results, as discussed in the following sections, demonstrate the effectiveness of our approach and the potential of the Crack Dataset as a resource for crack detection research.

3.3.4 Network Configuration

The pre-trained ResNet101 architecture, provided by Keras [168] and trained on the ImageNet dataset, serves as the basis for our embedding model. This choice

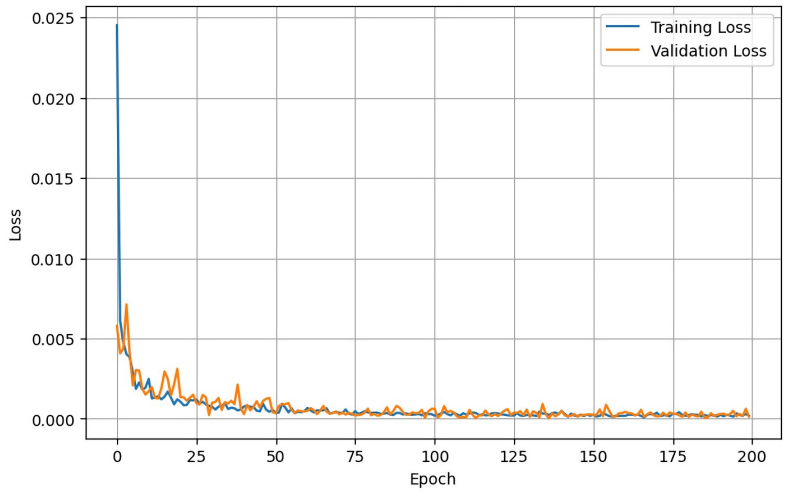


Figure 3.6: Model Loss in during Training.

leverages the deep residual learning framework to facilitate the extraction of rich feature representations from images of civil structures. The network's output is then passed through additional dense layers and batch normalization layers to refine the feature embeddings. Specifically, the proposed approch implements a dense layer with 256 units followed by ReLU activation [169] and batch normalization, and another dense layer with 128 units, also followed by ReLU activation and batch normalization, culminating in an output dense layer that produces 128-dimensional embeddings. The training process was done by utilizing V100 Nvidia GPU [156] and The validation loss model has reached 1.3e-05 (Figure. 3.6).

3.3.5 Hyperparameters Summary

Parameters such as learning rate, epsilon, and the margin for Triplet Loss are specified to optimize model performance. Hyperparameters are as follows:

- Target Image Size: 227 * 227 pixels
- Batch Size: 32
- Epochs: 200
- Optimizer: Adam [170]
- Learning Rate: $1e-4$
- Epsilon: $1e-1$
- Margin (for Triplet Loss): 1

- Dense Layer Configuration: 256 units (ReLU + BatchNorm) followed by 128 units (ReLU + BatchNorm)

3.3.6 EVALUATION

The implementation of the Siamese Network with a ResNet-101 backbone for the task of image verification yielded significant insights into the model's ability to discern between similar and dissimilar images. The analysis of the distances between anchor-positive and anchor-negative pairs, derived from a test dataset comprising 5,000 image triplets, forms the crux of our evaluation.

3.3.7 Model Evaluation Metrics

To evaluate the performance of the proposed Siamese network model, a set of metrics has been utilized, each providing insights into different aspects of model accuracy and robustness [171].

- **Precision**: Measures how accurately the model identifies cracks, ensuring that detected cracks are true defects while minimizing false positives.
- **Recall**: Evaluates the model's ability to detect all existing cracks, ensuring that real defects are not overlooked.
- **F1 Score**: Provides a balanced assessment of detection performance by combining precision and recall, particularly useful for imbalanced datasets.
- **Manhattan Distance (D)**: Manhattan Distance, also known as L1 distance, measures the distance between two points in a grid-based path (as opposed to Euclidean distance). It is used to calculate the distance between embeddings in the Siamese network, reflecting the model's discriminative power.

$$D = \sum_{i=1}^{n} |x_i - y_i| ,$$

  where x_i and y_i are the components of the two points in the embedding space. Lower Manhattan distances indicate that the embeddings of similar instances are close to each other.
- **Standard Deviation**: Measures the variability in the model's predictions, assessing the consistency of crack detection. A lower standard deviation indicates a more stable and reliable detection process.

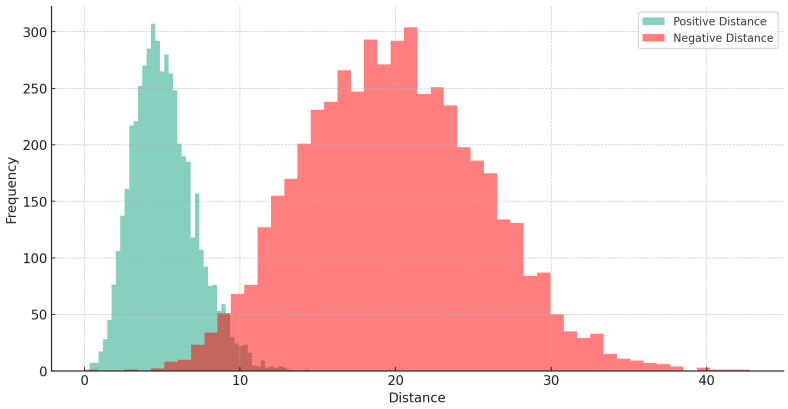


Figure 3.7: The frequency distribution of distances for both positive and negative pairs.

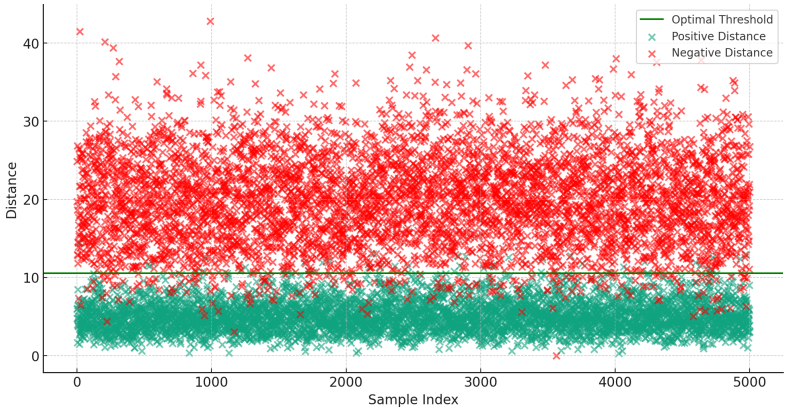


Figure 3.8: Individual positive and negative distances, highlighting the spread and overlap of distances. The optimal threshold line shows the distance value that best separates positive from negative pairs.

3.3.8 Evaluation Model

Statistical analysis revealed that the mean distance for anchor-positive pairs was significantly lower than that for anchor-negative pairs (Figure. 3.7), illustrating the model's efficacy in embedding similar images closer in the feature space. Specifically, the mean Manhattan Distance for positive pairs was observed at 5.13, with a standard deviation of 2.01, indicating a tight clustering of similar images. Conversely, the mean distance for negative pairs stood at 19.89, with a standard deviation of 5.75, reflecting a broader dispersion that is expected given the dissimilarity among the images. Further examination through the determination of an optimal threshold for classification underscored the model's precision in distinguishing between the two categories. An optimal threshold of 10.56 was identified, balancing sensitivity

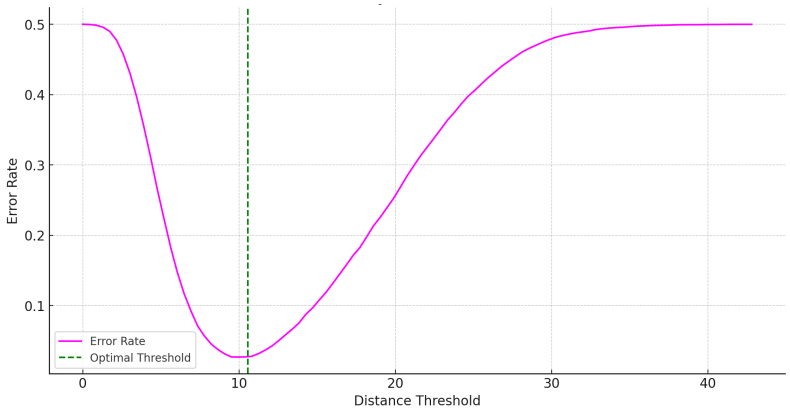


Figure 3.9: This shows how the error rate of classification changes as the distance threshold is adjusted. The optimal threshold is marked, indicating the point at which the error rate is minimized.

and specificity, and enabling the model to achieve an accuracy of 97.36% (Figure. 3.8, Figure. 3.9). The precision and recall metrics, standing at 95.77% and 99.1% respectively, along with an F1 score of 97.41%, attest to the model's robustness and reliability in performing image verification tasks. The Area Under the Curve (AUC) for the ROC curve was calculated at 99.61%, highlighting the model's excellent discriminatory ability across various threshold settings (Figure. 3.10). The distribution plots for positive and negative distances further elucidated the model's discriminative capacity, showcasing a clear separation between the two classes.

3.4 Conclusions and Future Research

This chapter demonstrated the integration of UAVs and deep learning technologies, particularly YOLOv8 and Siamese networks, for efficient, scalable, and automated crack detection. The results highlighted the effectiveness of these systems in addressing traditional limitations in structural health monitoring, such as labor-intensive methods and limited precision. Furthermore, the proposed triplet loss-based approach enables tracking crack evolution over time, laying the groundwork for predictive maintenance. The integration with digital twins offers significant potential, allowing real-time updates, simulations, and data-driven decision-making, thereby enhancing infrastructure resilience and safety.

Future research could focus on:

- Enhancing model robustness under adverse environmental conditions.

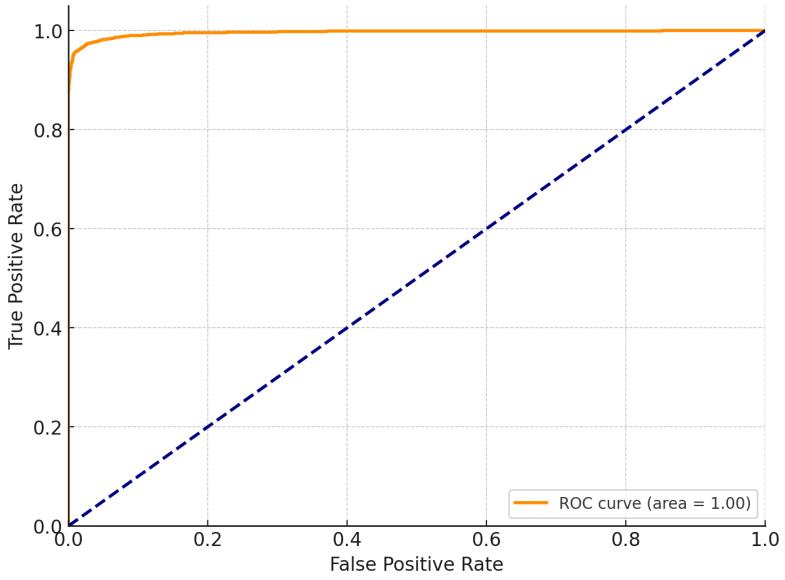


Figure 3.10: The Receiver Operating Characteristic (ROC) curve above illustrates the performance of the Siamese network model across different threshold settings for the image verification task.

- Implementing real-time processing on lightweight UAV platforms.
- Incorporating multi-modal data fusion, such as LiDAR and thermal imaging.
- Optimizing scalability and cost-effectiveness for widespread adoption.
- Expanding the triplet loss-based system for long-term crack progression analysis.
- Seamlessly integrating UAV systems with digital twin and IoT frameworks to enable synchronized monitoring and predictive insights.

By addressing these areas, UAV-based systems, coupled with digital twins, can further revolutionize structural health monitoring, ensuring safer and more sustainable infrastructure.

Chapter 4

Marker-Based Tracking for Structural Monitoring

This part builds on two recent IEEE publications that explore advancements in marker-based tracking systems for structural monitoring, with a focus on 3D-scaled masonry models. The first paper, presented at the *2024 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)* [172], introduces a low-cost tracking system using ArUco markers and a commercial smartphone. By leveraging a 6-degree-of-freedom reference motorized system, the study demonstrates that the proposed method achieves an expanded uncertainty of approximately 0.5° in orientation measurement, which is acceptable for its intended applications. The system's capability was further validated through dynamic testing on a 3D-scaled masonry arch, successfully tracking 19 targets during the displacement of its support base.

The second paper, presented at the *2024 XXXIII International Scientific Conference Electronics (ET)*, advances marker-based tracking through the development of DeepTag, a novel system utilizing convolutional neural networks (CNNs) for enhanced performance [173]. DeepTag improves tracking accuracy by reducing measurement uncertainty and addressing challenges such as occlusions and varying lighting conditions. This method requires fewer markers while maintaining robust accuracy, making it a cost-effective and scalable alternative to traditional systems. Preliminary results underscore its potential for structural monitoring, particularly in the preservation and maintenance of masonry structures. Together, these works lay the foundation for the methodologies and systems discussed in this part, highlighting the transformative potential of combining low-cost hardware with advanced

machine-learning techniques for structural health monitoring.

4.1 Challenges and Advancements in Monitoring Masonry Structures

Masonry structures, particularly historic buildings, are a cornerstone of cultural and architectural heritage. Constructed with bricks, stones, or concrete blocks bound by mortar, these structures form intricate and durable assemblies [174]. However, their preservation poses significant challenges due to their inherent vulnerabilities to environmental stressors, ground settlements, and seismic activities [175, 176, 177, 178]. The low tensile strength of masonry often leads to cracks under changing boundary conditions, exhibiting a unilateral mechanical response [179]. Such cracks, if untreated, can critically reduce the structural capacity of these constructions [180, 181, 182, 183, 184], highlighting the importance of effective monitoring systems for maintaining their integrity.

4.1.1 Key Challenges in Masonry Monitoring

Monitoring masonry structures presents several challenges that must be addressed to ensure their preservation and structural stability:

- **Complex Stress Conditions:** Masonry constructions are subject to various stressors, including environmental factors, thermal expansion, and ground movement. These stressors can cause cracking and deformation, which are often difficult to predict and analyze without detailed data.
- **Seismic Vulnerability:** Masonry structures, particularly unreinforced ones, are highly susceptible to horizontal loads generated by seismic events [177, 178]. Understanding their in-plane seismic capacity and failure mechanisms is critical for designing reinforcement strategies.
- **Data Collection Constraints:** Traditional methods for gathering data on masonry behavior, such as manual inspections or high-cost systems like laser scanners, can be labor-intensive, time-consuming, and financially restrictive. Additionally, these methods often struggle with accessibility in hard-to-reach areas of complex structures.
- **Accuracy and Scalability:** Many low-cost monitoring solutions face challenges in maintaining accuracy over time, particularly under dynamic condi-

tions. Furthermore, scaling these systems from small-scale models to real-world masonry constructions remains an open question in the field.

- **Environmental Factors:** Variations in lighting, weather conditions, and occlusions pose significant challenges for structural monitoring technologies, especially those relying on optical or visual markers.

4.1.2 Recent Advances and Proposed Solutions

To overcome these challenges, significant advancements have been made in the field of masonry monitoring. Low-cost marker-based systems have emerged as a practical and economical alternative to traditional high-end solutions [95]. These systems leverage standard video-capturing technologies and low-cost markers, offering reduced setup and operational costs while maintaining critical accuracy. By enabling the tracking of structural movements in scaled masonry models, such systems provide a valuable method for assessing the residual stability of deformed structures under quasi-static boundary conditions [185].

Building upon foundational marker-based systems, modern approaches are now integrating machine learning techniques to enhance accuracy and adaptability. One such advancement is DeepTag [186], a deep learning-based system utilizing convolutional neural networks (CNNs). DeepTag addresses several limitations of traditional marker-based systems by improving marker recognition under challenging conditions, such as varying lighting and occlusions. This innovation reduces dependency on high-contrast markers, allowing for more versatile and robust monitoring in diverse environments.

Moreover, the scalability of such systems from small-scale models to real masonry constructions is a key focus of current research. Recent experimental work demonstrates the feasibility of using marker-based systems in tracking 3D-scaled masonry models, with an expanded uncertainty kept within acceptable margins [185, 187]. By combining affordability and reliability, these systems present a compelling case for widespread adoption in large-scale monitoring projects.

The following is a review of the advancements and applications as discussed in the referenced works:

- **Low-Cost Marker-Based Optical Motion Capture:** a significant focus has been developing cost-effective alternatives to high-end commercial systems for validating inertial measurement units (IMUs). One such development is a low-cost marker-based optical motion capture system utilizing smartphone

cameras to track red markers and calculate their coordinates and angles. It offers an affordable solution without sacrificing significant accuracy [188];

- **Pose Estimation with Ball-Shaped Targets:** introducing a ball-shaped target in tracking-based scanning systems represents a pivotal innovation. This design allows for continuous pose estimation of the target during robot operations, utilizing a stereo vision system to estimate the 3-D position and orientation of the moving target in real-time. Such systems are invaluable in robotic tracking applications where full-space orientation and tracking are required [189];
- **Vision-Based Bending Sensors for Robotics:** in the scope of soft robotics, particularly with PneuNet actuators, vision-based sensors using ArUco marker detection have been developed. These sensors facilitate the monitoring of bending movements via a simple camera module. The system allows for the mechanical conversion of bending angles to marker rotations, demonstrating how marker-based systems can be effectively utilized in more dynamic, responsive robotic applications [190, 191, 192].
- **Motion capture systems for 3D displacement measurements of structures:** the tracking of markers' positions can be used for measuring the displacements of steel structures [193]. These displacement measurements can be used to identify the static and dynamic characteristics of the structure [193]. For instance, in [193], the Authors proposed a motion capture system for measuring the 3D displacement of a steel-scaled structural model during a free vibration test through 5 markers and 3 cameras.
- **Application in Developmental Research:** Marker-based motion tracking has also found applications in developmental research. Systems capable of tracking rigid bodies using multiple markers provide insights into object or tool use, making them especially useful in studies involving infants or small children, where non-intrusive methods are preferred. This application highlights the system's versatility beyond industrial or robotic uses, showcasing its potential in human-centered studies [194].

Overall, these advancements underscore the diverse applications of marker-based tracking systems, from cost-effective motion capture solutions to sophisticated robotic control and developmental research tools. Each development not only enhances the capabilities of marker-based systems but also broadens the potential fields of application, proving the system's adaptability and scalability in addressing various

real-world challenges [195, 194].
The novelty of the proposed research activity concerns the adoption of a low-cost marker-based tracking system using a smartphone camera for the measurements of the displacements of a 3D-scaled masonry model of an arch during quasi-static tests. The main difference with [193], where a steel structure was monitored and five markers were used, is that the displacements of all the blocks composing the model must be measured in the case of complex masonry models. Thus, the tracking of a higher number of markers is needed. This need poses a challenge from the metrological point of view because many markers (in the considered study case 20 markers) must be identified and precisely tracked during the test.

4.2 Proposed ArUco Marker Tracking Method and Experimental Evaluation

The following subsections detail the proposed marker-based tracking system and the implemented framework.

4.2.1 Implemented framework

The implemented framework utilizes the OpenCV and NumPy libraries to process video data and estimate the positions of markers in real-time. Initially, the system retrieves camera calibration parameters from a pre-defined YAML file, which is essential for accurate pose estimation. The camera captures video input directly through a webcam or from a pre-recorded file, ensuring versatility in testing different environments. As frames are read from the video source, the system employs the ArUco marker detection algorithm to identify markers within each frame. A specific ArUco dictionary is used to configure the detector parameters, optimizing the detection process for the `6x6_250` dictionary, which is known for its balance of detection accuracy and computational efficiency. In the literature, different solutions propose their own specified collection of markers, i.e. a marker dictionary. Unfortunately, there are two issues with using a predetermined dictionary. Firstly, the number of markers the application requires may be larger than the dictionary's capacity. Secondly, if the number of markers required is modest, utilising a dictionary with a large inter-marker distance is better to reduce inter-marker confusion. Moreover, in several cases, the occlusion problem is not properly addressed. ArUco fiducial

markers are an example of a marker-based system that addresses these challenges [190].

4.2.2 Marker Configuration and Pose Estimation Process

The experiment has focused on two specific markers to maintain a manageable complexity and ensure robust data collection. Each marker's corner is defined relative to its known physical dimensions, here set as 0.015 m for both sides of a square marker. During the video processing loop, the system detects these markers in each frame, and if at least two markers are identified, it proceeds to estimate their poses. The pose estimation leverages the `solvePnP` function from OpenCV, which computes each detected marker's rotation and translation vectors relative to the camera's optical center. This step is critical as it provides the spatial orientation and position necessary for further analysis.

4.2.3 Data Handling and Output

Upon successful pose estimation, the system dynamically records each marker's translation and rotation vector components into CSV files, uniquely named according to the marker identifier. This data storage method facilitates subsequent analysis of marker movements over time. Additionally, the framework draws the detected markers on the video frames to provide visual feedback on the tracking process, which is crucial for real-time monitoring and debugging. The video frames, annotated with the marker positions and axes, are written to an output video file, serving as a record of the experiment. The implementation allows for interruption through user input, ensuring that the system can be conveniently stopped during live demonstrations or testing.

4.2.4 Metrological characterization

The proposed marker-based tracking system has been tested against a 6-DoF reference motorized system by Standa LTD [196], which exhibits an accuracy of the imposed orientation around every axis of 0.01°. Fig. 4.1 shows the adopted experimental setup for two acquired frames with orientations at 0° (see Figure. 4.1a), and 30° (see Figure. 4.1b), respectively. A wooden rod is fixed onto the z-axis motor and seven ArUco markers are placed along it. A reference marker is placed

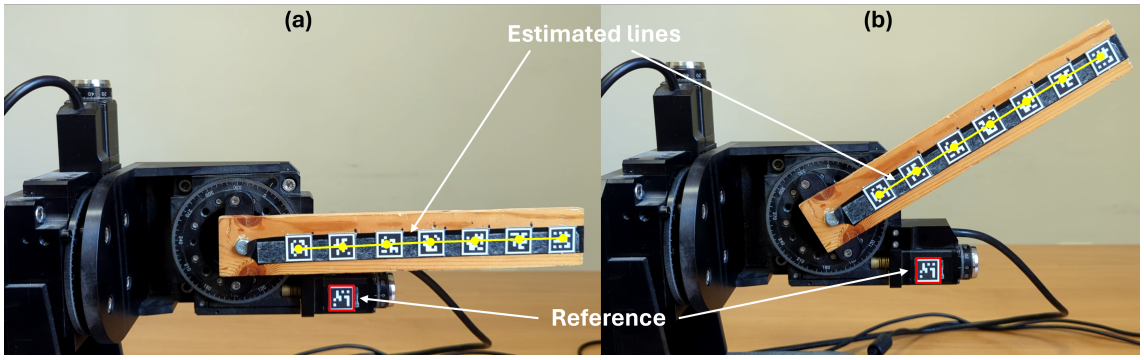


Figure 4.1: Adopted experimental setup for the preliminary metrological characterization of the marked-based tracking system in measuring the orientation of the markers against a 6-DoF reference motorized system: (a) captured camera frame when the orientation of the motor is fixed to 0°, and (b) captured camera frame when the orientation of the motor is fixed to 30°.

on the motorized system, see the red boxes in Figure. 4.1, and its position is used for defining the reference coordinate system in a frame.

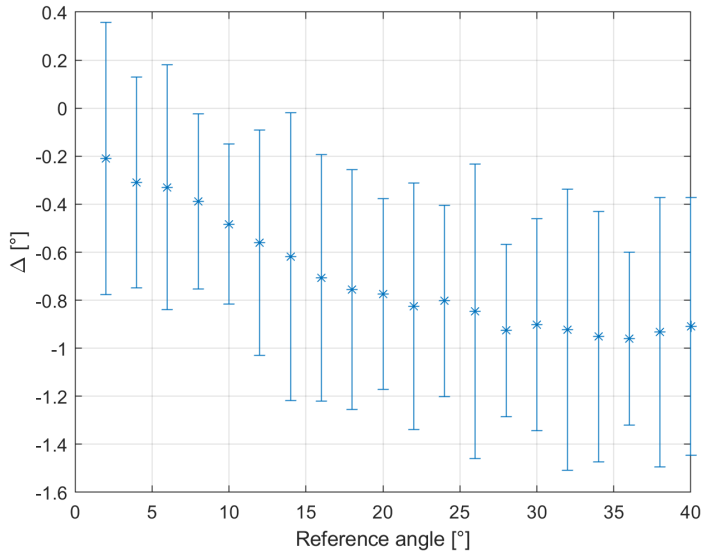


Figure 4.2: Difference between the reference angles and the obtained mean values, Δ, with the expanded uncertainty (coverage factor of 2) against the reference orientation measurements.

Once the marker positions are obtained, each couple is processed to estimate the line crossing them. For every line, the orientation to the line at 0° is assessed, thus obtaining six orientation measurements. Ideally, the six angle measurements should be equal to each other, however, because of the sources of uncertainty, such as light conditions, blurring, and lens distortion the obtained values are slightly different.

The test has been performed for orientation ranging from 2° to 40° with a step of 2°. For tracking purposes, the system used a Google Pixel 7 pro camera [197]. The smartphone camera was placed at a distance of 50 cm from the motorized system. For every imposed angle, the orientation measurements were filtered with a moving average having a window size of 30 samples. Then, the expanded uncertainty with a coverage factor of 2 is estimated on the six lines' orientation measurements in all the frames. In Fig. 4.2, the differences between the reference angles imposed with the motorized system and the obtained mean values from the markers' tracking, Δ, are depicted together with the expanded uncertainty. It can be observed that the absolute difference increases with the reference angle while the expanded uncertainty is slightly constant at around 0.5°.

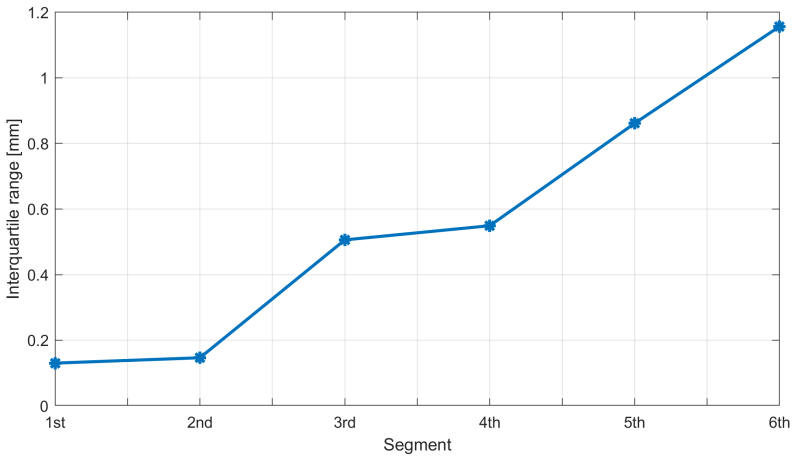


Figure 4.3: Interquartile ranges of the six segment distances that are obtained from the positions of each pair of successive markers.

Another test has been conducted to assess the repeatability of distance measurements obtained from the positions of each pair of successive markers. With seven markers along the wooden rod, six segment distance measurements are obtained for each imposed orientation in every frame. In particular, 20 distance measurements are considered at an imposed orientation, providing an amount of 20×21 measurements, where 21 is the number of imposed orientations. According to the χ^2 test, the obtained measurements are not Gaussian distributed. For this reason, the interquartile range has been considered to assess the measurement repeatability. The interquartile ranges for the six segments are reported in Fig. 4.3. The 1st segment exhibits the lowest interquartile range of around 0.15 mm, while the 6th segment exhibits the highest, i.e., around 1.2 mm. The 1st segment is located at the centre of the image, while the 6th segment is closer to its edge. Thus, the interquartile

range increases with the distance from the centre of the image; this can be due to the lens distortions, which are usually higher at the edges.

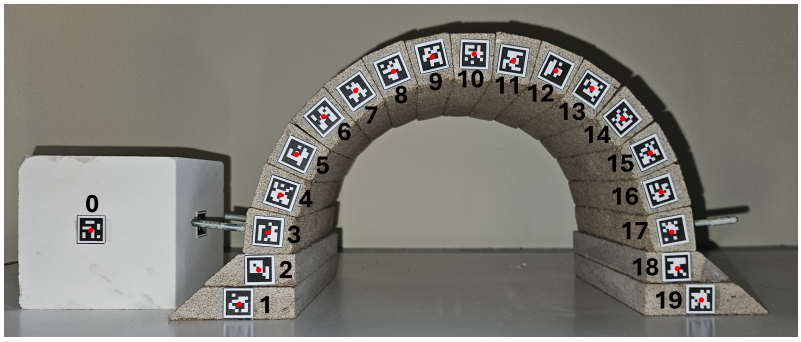


Figure 4.4: Arch model and reference block with the corresponding tracked markers.

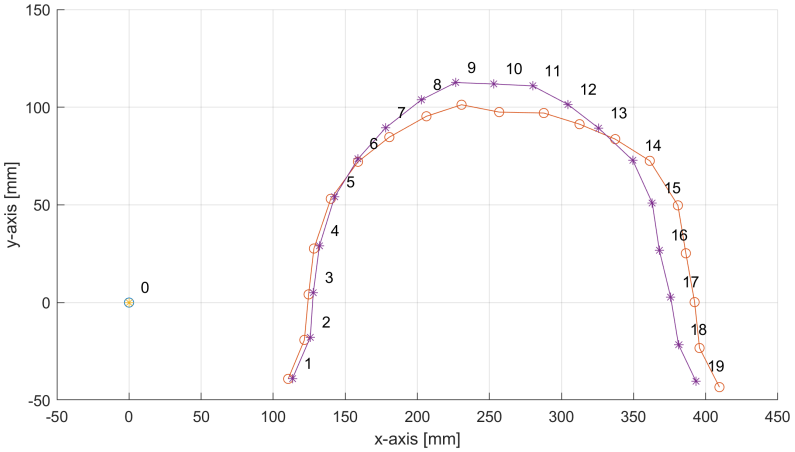


Figure 4.5: Tracked markers for the first frame (asterisks) and the 1000-th frame (circle).

4.2.5 Test bench for 3D-scaled masonry models

A test was conducted to track the deformed configuration of an arch whose support was subjected to pseudo-static horizontal displacements. The scaled model of the arch consists of 15 voussoirs spanning 20 cm with a thickness of 2.0 cm The depth of the arch is 19 cm. Four further blocks were created to simulate supporting conditions as depicted in Fig. 4.4. A variable horizontal support displacement was prescribed at the base of the right support after about 2 s in which the arch was in static conditions. The position and orientation of each arch's element were obtained according to a reference marker, which was in static conditions for the entire test duration. Fig. 4.5 shows the tracked elements for the first frame and the 1000-th frame. In the 1000-th frame, due to the horizontal displacement, the arch configu-

ration is significantly changed with respect to its geometrical configuration in the first frame where no displacement occurred.

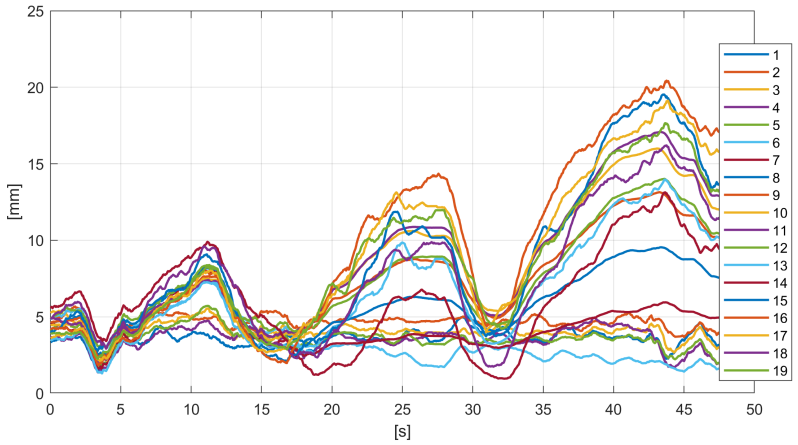


Figure 4.6: Magnitude of the displacements of the 19 masonry blocks.

4.2.6 Preliminary experimental tests

Preliminary tests were conducted to demonstrate the capability of the system to track the markers during quasi-static tests. The displacement vectors obtained from the coordinates of every marker at the i-th frame and their coordinates in the first frame are calculated. A moving average with a time window of 30 samples is applied to the obtained magnitude and orientation measurements. In Fig. 4.6 and Fig. 4.7, the magnitude and orientation measurements of the displacement vectors are depicted, respectively. Fig. 4.6 shows as at the beginning for around 2 s, the maximum displacement magnitudes for all elements is around 6 mm. Then, three different displacements are enforced, with peak magnitudes of around 10 mm, 14 mm, and 20 mm, respectively. The arch elements interested in those displacements are 14-th, and 16-th for the last two.

The arch span measurements obtained by tracking the position of the first marker and the 19-th marker are shown in Fig. 4.8. The initial arch span was around 280 mm, then it was reduced to around 275 mm at 11.6 s. At 25.7 s and 43.0 s, the values of 295 mm and 300 mm were obtained, respectively. Of course, the arch span measurements are overestimated (it was around 200 mm) because the measurements are based on the center positions of the markers rather than the actual edges of the structural elements. This results in a discrepancy, as the space between the edges of

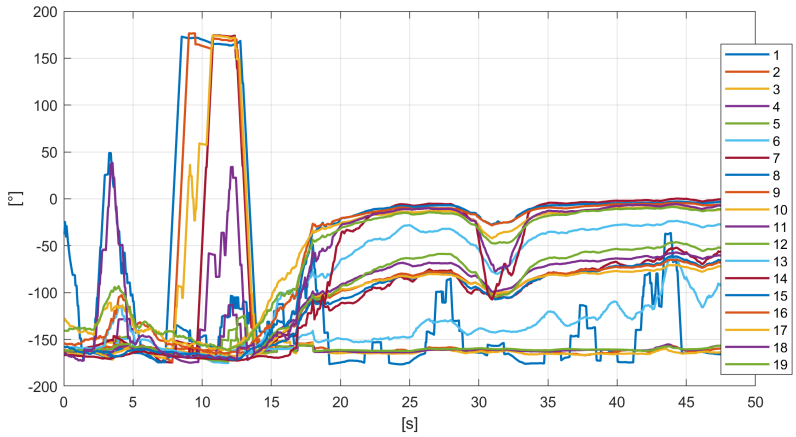


Figure 4.7: Orientation of the displacements of the 19 masonry blocks.

the elements is not directly considered (see Fig. 4.4). Future work will be performed to compensate for this error by retrieving the position of the markers in the elements.

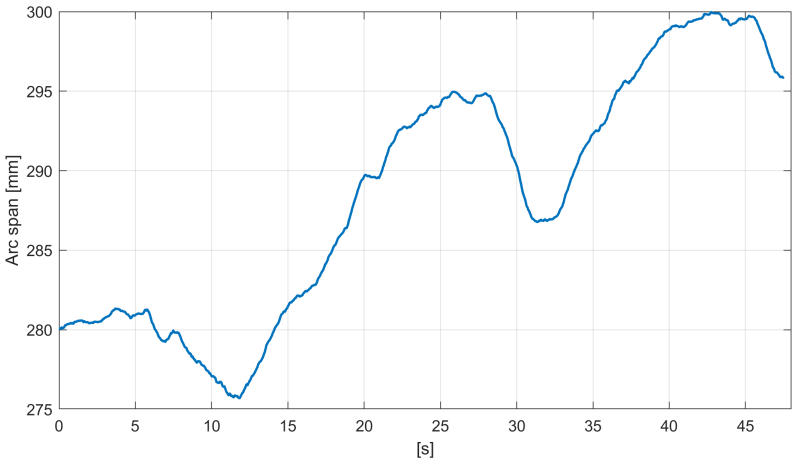


Figure 4.8: Arch span measurements, i.e., the distance between the 1st and 19th block.

The proposed ArUco-based tracking system includes an uncertainty evaluation for both orientation and distance measurements. In particular, expanded uncertainty (coverage factor = 2) was computed for the orientation angles, as shown in Figure 4.2, and interquartile ranges were used to assess the repeatability of distance estimations between markers (Figure 4.3). These evaluations highlight the metrological soundness of the developed tracking system.

4.3 Proposed DeepTag Marker Tracking Method and Experimental Evaluation

4.3.1 Proposed Marker Tracking Method

The proposed marker tracking method is based on DeepTag [186]. DeepTag enhances marker tracking using a sophisticated two-stage detection scheme combined with deep learning techniques. The process involves detecting keypoints, estimating Regions of Interest (ROIs), generating rectified patches, decoding marker IDs, and estimating the 6-Degree-of-Freedom (DoF) pose (see Figure. 4.9). Given an

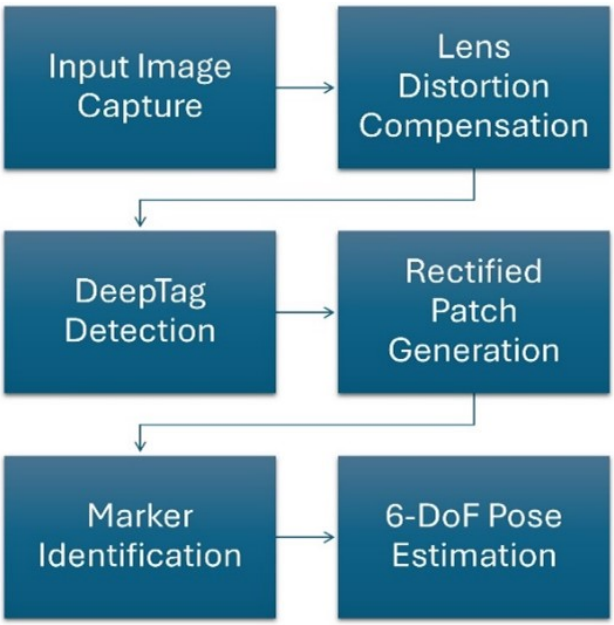


Figure 4.9: Workflow for 6-DoF Pose Estimation Using DeepTag Detection.

input image, DeepTag detects groups of keypoints, each corresponding to a potential marker. It estimates each potential marker's ROI and represents it with at least four clockwise non-collinear points. A rectified patch is then generated using a homography matrix that maps the ROI to predefined points in the patch, such as mapping a four-point ROI to a fixed square. This step normalizes the detected region for further processing.

In the rectified patch, keypoints and digital symbols are estimated and sorted. These symbols encode the marker's information. The marker ID is recognized by comparing the decoded digital symbols with a predefined marker library [198]. The keypoint positions in the original image are obtained by applying the inverse ho-

mography matrix, mapping the points back to their original coordinates. Using the known physical size of the marker, the 6-DoF pose is estimated using the Perspective-n-Point (PnP) algorithm. This provides the marker's position and orientation relative to the camera.

To address lens distortions, a calibration procedure using a chessboard pattern estimates five distortion coefficients, accounting for both radial and tangential distortions. These coefficients are used to compensate for distortions in the marker pose estimates.

DeepTag's architecture includes convolutional layers for feature extraction and fully connected layers for classification and regression tasks. Trained on a large synthetic dataset with various marker types and distortions, it generalizes well to real-world scenarios. The training process involves dataset augmentation with random noise, blur, and lighting variations to improve robustness.

4.3.2 Preliminary Metrological Characterization

The proposed marker-based tracking system has been evaluated against a 6-DoF reference motorized system by Standa LTD [196] (Figure. 4.1), which provides an orientation accuracy of 0.01° around each axis. A wooden rod fixed to the z-axis motor has seven ArUco markers placed along its length. A reference marker is fixed on the motorized system, defining the reference coordinate system for each frame.

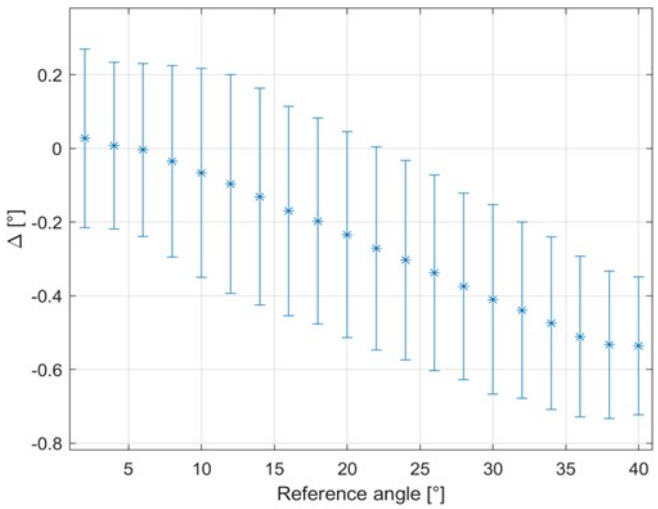


Figure 4.10: Difference between the reference angles and the obtained mean values, denoted as Δ, along with the expanded uncertainty (with a coverage factor of 2) plotted against the reference orientation measurements

The orientation of each line connecting marker pairs is assessed relative to the line at 0°, resulting in six orientation measurements. Tests covered orientations from 2° to 40° in 2° increments. A Google Pixel 7 Pro camera [197] was positioned 50 cm away from the motorized system for tracking. The expanded uncertainty, with a coverage factor of 2, was estimated for the six lines' orientation measurements across all frames. Figure. 4.10 shows the differences between the reference angles and the mean values obtained from marker tracking, as well as the expanded uncertainty. The maximum expanded uncertainty was 0.29°, significantly lower than the 0.58° reported in [172]. It is acknowledged that, in some instances, the estimated error appears to exceed the expanded uncertainty bounds shown in Figures 4.2 and 4.10. This can be attributed to additional sources of variability not fully captured by the simplified uncertainty model, such as frame synchronization issues, lighting variations, or imperfections in marker detection. Therefore, the presented expanded uncertainty should be interpreted as a lower-bound estimate based on dominant contributors. A more comprehensive model including these secondary effects could provide more conservative and complete uncertainty estimates in future work.

A separate test evaluated the consistency of distance measurements between successive markers. With seven markers on a wooden rod, six segment distances were measured for each orientation, totaling 420 measurements across 21 orientations (0° to 40° in 2° steps). The interquartile range was used to evaluate repeatability due to non-Gaussian measurement distributions. As shown in Figure. 4.11, interquartile ranges increased with distance from the image center, likely due to greater lens distortion at the edges. However, the maximum interquartile range was 0.12 mm, one order of magnitude lower than the 1.2 mm reported in [172].

4.3.3 Experimental Test on Scaled Masonry Arch

A preliminary experimental test was conducted, as described in [199], to demonstrate the system's capability to track markers during quasi-static tests. The scaled arch model comprises 15 voussoirs spanning 20 cm with a thickness of 2.0 cm and a depth of 19 cm. Four blocks simulate supporting conditions. Figure. 4.12 shows the 3D-scaled arch model and markers tracked by DeepTag.

After 2 seconds of static conditions, variable horizontal displacement was applied at the base of the right support. The position and orientation of each voussoir were determined using a static reference marker. Figure. 4.13 illustrates the tracked elements in 3D for the first and 1000th frames. In the 1000th frame, significant changes

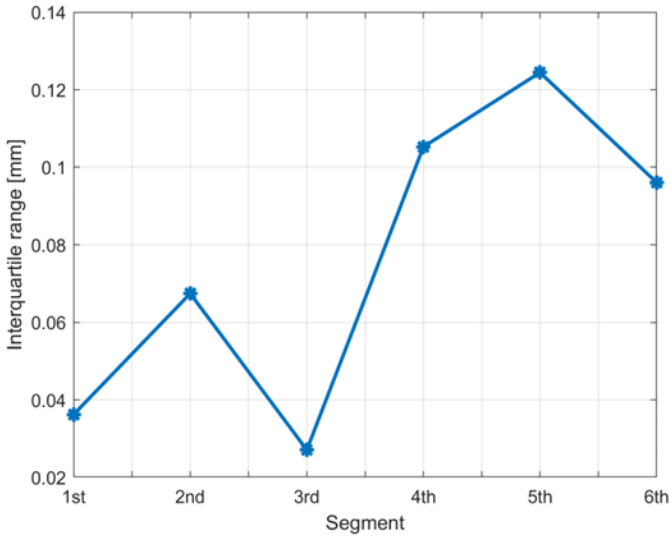


Figure 4.11: Interquartile ranges of the six-segment distances derived from the positions of each successive marker pair

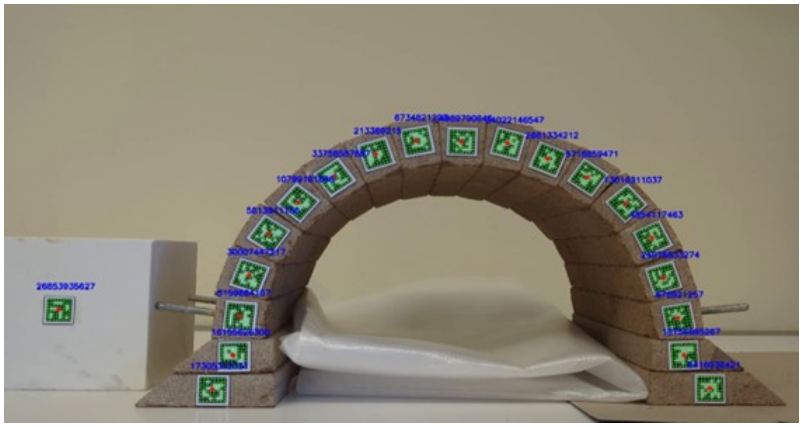


Figure 4.12: 3D-scaled model of the arch and the tracked markers by DeepTag

in the arch configuration highlight the system's ability to capture displacement and orientation changes over time.

Displacement vectors were calculated by comparing each marker's coordinates in the i-th frame to those in the first frame. A moving average with a 30-sample time window smoothed these measurements. Figures. 4.14 and 4.15 depict the displacement magnitude and orientation, respectively. The maximum displacements occurred at around 43 s for the 16th and 17th elements, reaching approximately 22 mm.

The arch span, measured between the first and 19th markers, initially measured 277 mm and decreased to 273 mm at 11.6 s. At 25.7 s and 43.0 s, the span increased

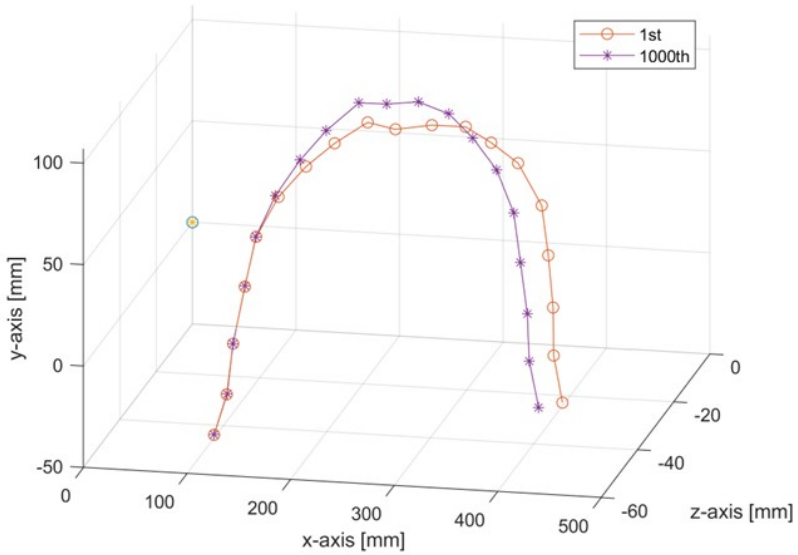


Figure 4.13: Magnitudes of the displacements of the 19 masonry blocks constituting the arch.

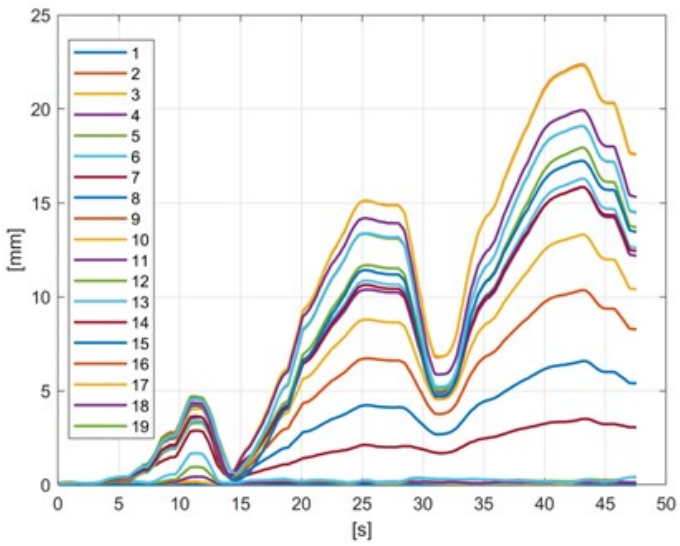


Figure 4.14: Magnitudes of the displacements of the 19 masonry blocks constituting the arch

to 290 mm and 296 mm, respectively. These measurements were less noisy than those in [172], with a 4 mm difference in span values.

4.3.4 Limitations and Future Improvements

Despite improved measurement accuracy, processing time remains a limitation. Each frame requires approximately seven seconds on an Intel Core i7-4710HQ CPU, making it impractical for long-term monitoring with many frames. Future work will

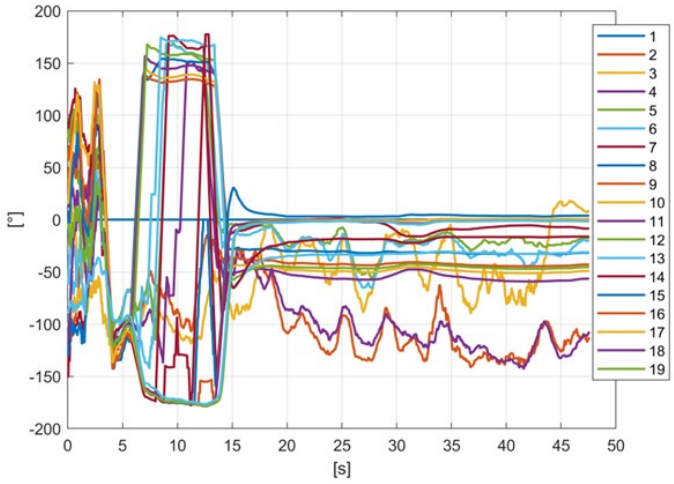


Figure 4.15: Orientations of the displacements of the 19 masonry blocks constituting the arch

focus on optimizing processing time and addressing lens distortion by considering additional calibration coefficients.

4.4 Comparison of Marker-Based Tracking Methods

This section highlights the advantages and disadvantages of the two marker-based tracking methods: the DeepTag system and the low-cost marker-based system using ArUco markers. While both systems are designed for tracking 3D-scaled masonry models, they differ significantly in terms of accuracy, complexity, and implementation costs.

4.4.1 Advantages and Disadvantages

DeepTag System:

- **Advantages:**
  - Leverages deep learning techniques, offering high accuracy in marker detection and pose estimation, even under challenging conditions like occlusions and varying lighting.
  - Reduces dependency on high-contrast markers, enhancing adaptability for different structural configurations.
  - Demonstrates reduced measurement uncertainty by 42%, achieving precise deformation and displacement tracking.

- **Disadvantages:**
  - High computational demands, requiring powerful hardware for real-time processing.
  - Longer processing times per frame (approximately 7 seconds), making it less practical for long-term monitoring.
  - Requires extensive training data and model fine-tuning, increasing initial setup complexity.

Low-Cost ArUco Marker System:

- **Advantages:**
  - Cost-effective, using widely available hardware such as smartphones and standard video-capturing technologies .
  - Simple to implement and integrate into existing systems, making it ideal for preliminary studies and resource-constrained environments.
  - Provides acceptable accuracy for many applications, with an expanded uncertainty of around 0.5°.
- **Disadvantages:**
  - More susceptible to environmental factors such as lighting and occlusions, which can impact accuracy.
  - Requires high-contrast markers and careful camera calibration to maintain reliability.
  - Limited scalability for tracking large numbers of markers in complex configurations.

The following table summarizes the key differences between the two methods in Table 4.1:

Table 4.1: Comparison of DeepTag and Low-Cost ArUco Marker-Based Systems

Feature	**DeepTag System**	**ArUco Marker System**
Accuracy	High (42% less uncertainty)	Moderate (0.5°uncertainty)
Cost	High (requires specialized hardware)	Low (uses smartphones and basic hardware)
Robustness to Occlusions	High	Low
Scalability	High (tracks multiple markers)	Limited
Ease of Implementation	Complex (deep learning setup)	Simple
Processing Speed	Slower (7s/frame)	Faster
Environmental Adaptability	High (handles varying conditions)	Moderate (requires ideal conditions)
Applications	Detailed structural monitoring	Preliminary or low-budget monitoring

The ArUco-based method showed a typical expanded uncertainty of approximately ±1.2 mm in translation and ±0.48° in rotation under controlled lab conditions. In contrast, the DeepTag-based system achieved an improved expanded uncertainty of ±0.8 mm in translation and ±0.29° in rotation, mainly due to its larger marker area and enhanced robustness to image blur and noise. These values were derived from repeated static pose estimations over 100 frames under similar lighting and distance conditions.

The choice between the two systems depends on the specific requirements of the application. DeepTag is ideal for projects demanding high accuracy and robustness in challenging environments, albeit at the cost of higher computational and financial resources. On the other hand, the ArUco marker-based system offers a cost-effective and simpler alternative for less demanding scenarios or preliminary studies.

Chapter 5

Environmental Monitoring for Marine Digital Twins

This chapter is based on the research presented in the paper "A Significant Wave Height Data-Driven Modeling for Digital Twins of Marine Environment" published in the proceedings of the 2024 IEEE International Workshop on Metrology for the Sea (MetroSea) [200]. The study focuses on the development of a predictive modeling framework designed for forecasting Significant Wave Height (VHM0) in marine environments, an essential parameter for maritime operations and safety. The chapter integrates the core methodologies discussed in the paper, particularly leveraging digital twins (DTs) to enhance real-time monitoring and predictive analytics for wave height dynamics. By utilizing a Gated Recurrent Unit (GRU) neural network, the model processes in situ sensor data from three key locations—Tarragona, Barcelona, and the EMSO-OBSEA observatory which is located 4 km of the Vilanova i la Geltru coast, Barcelona, Spain—enabling a robust approach to wave height forecasting.

5.0.1 Key Observations and Insights

Digital twins (DTs) are digital representations of real-world systems or objects that are created through a variety of intricate and varied modeling techniques in order to faithfully mimic their state and behavior [201, 202]. In the context of this study, the term 'Digital Twin' refers to a predictive model that simulates wave dynamics in marine environments based on real-time data inputs. While proposed model does not constitute a complete digital twin, it forms a critical component of such a system by providing accurate wave height predictions. This study focuses on the preliminary development of a predictive model for wave height, which serves

as an essential component of a future digital twin for marine environments. While a complete digital twin encompasses multiple aspects of the environment, including physical, chemical, and biological processes, the current research lays the groundwork for this broader integration [203]. Real-time data is used to continuously update these models, and machine learning techniques are applied to refine model outputs and enable testing of specific hypotheses by varying model parameters.

The creation of a Digital Twin of the Ocean (DTO) is one of the newest uses of DTs in environmental science. A DTO is a digital depiction of the marine biosphere that is built using a variety of biological, meteorological, oceanographic, and socioeconomic data. Multiparametric assessments of environmental patterns and processes, such as ecosystem responses to anthropogenic influences and natural occurrences, are made possible by this extensive model [204, 205, 206, 207].

The use of Digital Twins of the Ocean (DTOs) in maritime contexts is yet somewhat unexplored, despite their enormous potential. In order to close this gap, this section focuses on predictive modeling with a DT model created especially for maritime environments. This research attempts to improve the monitoring, analysis, and decision-making processes in maritime operations by utilizing the advanced capabilities of DTs, ultimately leading to safer and more effective results. The OBSEA underwater observatory and buoys have collected data to create a 3D representation of the sea's current condition as a digital twin [208, 209, 210] (see Figure. 5.1).

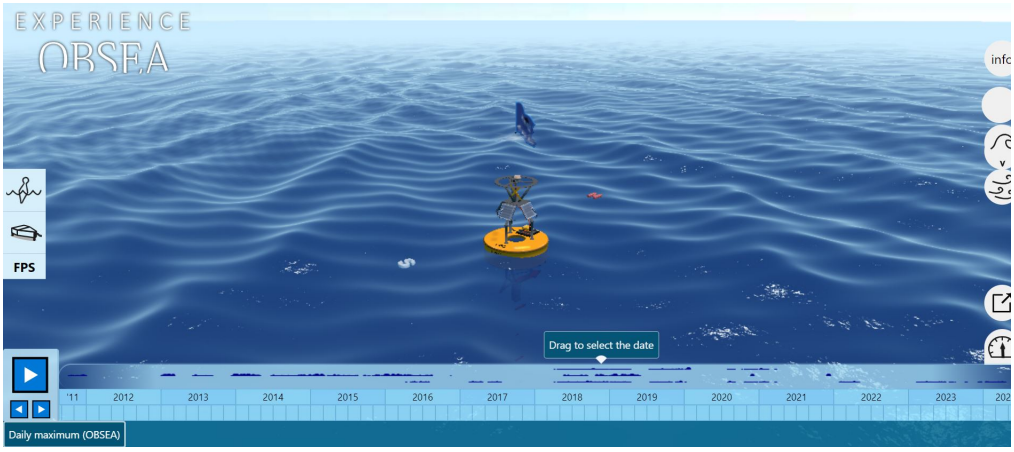


Figure 5.1: The underwater observatory and buoy OBSEA have collected data to create a 3D representation of the sea's current condition as the digital twin [211].

5.0.2 Related Work

Predictive modeling in the maritime domain have advanced through machine learning and deep learning techniques, enhancing operational efficiency and safety [212, 213]. Despite these advancements, existing methodologies have limitations that the proposed approach aims to address.

In the field of wave prediction, recent studies have explored the use of advanced deep learning techniques, particularly recurrent neural networks (RNNs) [214] such as Long Short-Term Memory (LSTM) [215] and Gated Recurrent Unit (GRU) models [216]. These models have shown promising results in forecasting significant wave height (SWH) and other wave parameters [217].

Hu et al. compared LSTM networks with the numerical model WAVEWATCH III for predicting wave parameters, demonstrating the effectiveness of LSTM in capturing temporal dependencies in wave data. Building on this work, recent research has investigated the potential of GRU models, which are similar to LSTMs but with a simpler architecture [218].

A study by Minuzzi and Farina presented a new deep learning training framework for forecasting significant wave height in the Southwestern Atlantic Ocean using LSTM networks [219]. The approach showcased the ability of LSTM models to capture complex temporal patterns in wave data.

Further advancing the field, researchers have explored the use of both unidirectional and bidirectional GRU models for wave forecasting. A study focusing on significant wave height prediction utilized these GRU variants, highlighting the potential of bidirectional architectures in capturing both past and future context in time series data [220].

More recent work has compared the performance of multivariate GRU and LSTM models for hindcasting and multi-step forecasting of significant wave height. These models have demonstrated the ability to handle multiple input variables and provide accurate predictions over various time horizons [221, 217].

The application of these advanced RNN architectures, including both LSTM and GRU, represents a significant step forward in wave prediction. Their ability to capture long-term dependencies and handle complex temporal patterns makes them particularly well-suited for the challenges of wave forecasting, potentially offering improvements over traditional statistical and numerical models [222].

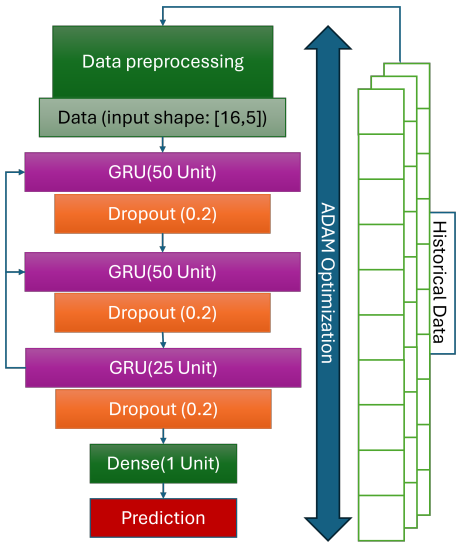


Figure 5.2: Predictive model architecture with three GRU layers (50 units each), dropout layers (0.2), and a dense output layer. Input data shape: [16, 5].

5.0.3 Methodology

The methodology employed in this study involves several key steps to develop a robust predictive model for maritime environmental data. The process begins with comprehensive data preprocessing, where raw data is cleaned, formatted, and normalized to ensure consistency and reliability for the subsequent modeling phases.

Once the data is preprocessed, it is divided into training, validation, and testing sets. The training set is utilized to train a machine learning model designed to forecast future values of the target variable based on input features derived from the maritime dataset. The validation set is used to fine-tune the model parameters, helping to prevent overfitting and ensure the model generalizes well to new data [216]. The testing set, which is kept separate from the training and validation sets, is used to evaluate the model's performance on unseen data, providing an unbiased assessment of its predictive capabilities.

For the predictive modeling task, a GRU neural network is selected due to its efficacy in capturing temporal dependencies in sequential data. The model was developed using the TensorFlow and Keras libraries in Python, which provide robust support for deep learning and efficient computation. The use of these frameworks enabled streamlined data processing and model optimization. The GRU model is trained to recognize patterns and relationships within the data, enabling it to make accurate predictions about future values of the target variable. The architecture of

the model, as shown in Figure.5.2 consists of three GRU layers followed by a dense output layer, designed to capture temporal dependencies and provide precise wave height forecasts.

Outlier assessment is integrated into the methodology by analyzing the residuals, which are the differences between the predicted values and the actual observed values. An outlier, in the context of this study, is defined as a data point or a set of data points that deviate significantly from expected normal behavior. Specifically, for maritime environmental data, outliers can be characterized by unusually high or low values of key features such as Significant Wave Height which can practically be utilized in anomaly detection. These deviations could indicate abnormal maritime conditions, sensor malfunctions, or other irregular events that differ from typical patterns in the dataset.

Outliers are identified when these residuals exceed a predefined threshold, indicating significant deviations from the expected behavior. This threshold is established based on the statistical distribution of the residuals, ensuring that the outlier assessment mechanism is sensitive enough to detect unusual patterns while minimizing the occurrence of false positives.

The performance of the predictive model is assessed using standard regression metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Additionally, the Pearson correlation coefficient is calculated to evaluate the strength and direction of the linear relationship between the predicted and actual values.

Implementation and Metrics

5.0.4 Dataset

The dataset utilized in this study consists of maritime environmental data collected from various sensors around the Tarragona region in Spain [223]. Key features crucial for understanding wave dynamics include Significant Wave Height (VHM0), Mean Wave Direction (VMDR), Peak Wave Period (VTPK), Zero Crossing Period (VTZA), and Maximum Wave Height (VZMX). VHM0, the primary target variable for our predictive model, represents the average height of the highest one-third of waves observed. VMDR indicates the average direction from which the waves are coming, VTPK represents the period of the most energetic waves, VTZA denotes the average period between zero crossings of the wave signal, and VZMX records the

height of the highest wave observed during the measurement period. These features were meticulously selected to capture the temporal and spatial dynamics essential for accurate predictive modeling. The data was rigorously preprocessed to eliminate invalid entries and ensure consistency, including removing non-numeric values and negative values that could skew the analysis.

The datasets used in this study were provided by two different sources: Puertos del Estado and the EMSO OBSEA observatory. The primary dataset, used for training the model, was collected from sensors in the Tarragona region by Puertos del Estado, the Spanish government body responsible for managing ports and marine infrastructure. This dataset includes data from numerous oceanographic sensors installed on buoys that monitor and record various maritime parameters, including significant wave height, mean wave direction, and peak wave period.

For model evaluation, two additional datasets were employed. The first evaluation dataset was collected from a buoy in the Barcelona region, also maintained by Puertos del Estado [224]. The second evaluation dataset came from the Expandable Seafloor Observatory (EMSO OBSEA) [225, 226], situated off the Spanish coast at Vilanova i la Geltrú. This dataset was gathered using an Acoustic Doppler Current Profiler (ADCP) equipped with AWAC-AST 1 MHz technology [227] at a depth of 20 meters. These diverse sources ensured a comprehensive evaluation of the model's robustness and generalizability across different maritime environments.

The dataset utilized in this study consists of maritime environmental data collected from various sensors around the Tarragona region in Spain [223]. This dataset includes several key features crucial for understanding wave dynamics:

- **Significant Wave Height (VHM0)**: This feature is the primary target variable for our predictive model, representing the average height of the highest one-third of waves observed.
- **Mean Wave Direction (VMDR)**: Indicates the average direction from which the waves are coming.
- **Peak Wave Period (VTPK)**: Represents the period of the most energetic waves.
- **Zero Crossing Period (VTZA)**: Denotes the average period between zero crossings of the wave signal.
- **Maximum Wave Height (VZMX)**: Records the height of the highest wave observed during the measurement period.

These features are meticulously selected to capture the temporal and spatial dynamics essential for accurate predictive modeling. The data is rigorously preprocessed to eliminate invalid entries and ensure consistency, including removing non-numeric values and negative values that could skew the analysis.

5.0.5 Implementation

The implementation process begins with comprehensive data preprocessing. The 'time' column is then converted to date time format, and any entries with invalid date time data are discarded. The relevant features, including the target variable VHM0, are selected and cleaned to ensure all values are numeric and non-negative.

The selected features and the target variable VHM0 are normalized to scale the data within a range of 0 to 1. This normalization ensures that all features contribute equally to the model training process.

The normalized data is transformed into sequences with a specified look-back period of 16, which defines how many past data points are used to predict the next point. This sequence creation is critical for time-series forecasting using GRU, as it allows the model to learn temporal dependencies effectively [228]. The dataset is divided into three subsets: 70% for training, 20% for validation, and 10% for testing. A GRU-based neural network is then constructed with specific hyperparameters. The first and second GRU layers each contain 50 units, both configured to return sequences to facilitate stacking, while the third GRU layer contains 25 units, also returning sequences. A dropout rate of 0.2 is applied after each GRU layer to prevent overfitting [229]. The model is trained with a batch size of 32 over 100 epochs, using early stopping to monitor validation loss and optimize training duration. The model is compiled with the Adam optimizer and a mean squared error (MSE) loss function [230]. Early stopping is employed with a patience of 10 epochs, ensuring that training halts when the validation loss no longer improves.

5.0.6 Metrics

The model's performance is quantitatively assessed using several key metrics:

- **Mean Absolute Error (MAE)**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where n is the number of data points, y_i represents the actual value, and $\hat{y}_i$ represents the predicted value. This measures the average magnitude of errors in the predictions without considering their direction [231].

- **Mean Squared Error (MSE)**:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where n is the number of data points, y_i represents the actual value, and $\hat{y}_i$ represents the predicted value. This provides a measure of the average squared difference between the actual and predicted values, giving more weight to larger errors.

- **Root Mean Squared Error (RMSE)**:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where n is the number of data points, y_i represents the actual value, and $\hat{y}_i$ represents the predicted value. This offers a measure of the average magnitude of the errors in the same units as the target variable, making it more interpretable [232].

- **Pearson Correlation Coefficient**:

$$r = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2 \sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}$$

where n is the number of data points, y_i represents the actual value, $\hat{y}_i$ represents the predicted value, $\bar{y}$ is the mean of the actual values, and $\bar{\hat{y}}$ is the mean of the predicted values. This measures the linear correlation between the actual and predicted values, ranging from -1 to 1. A higher value indicates a stronger positive linear relationship [233].

Outliers are detected based on the residuals, which are the differences between the actual and predicted VHM0 values. The residuals are analyzed, and a threshold is set for identifying outliers. This threshold is calculated as:

$$Threshold = MPR + 2 \times SDPR$$

while MPR is Mean of positive residuals and SDPR is Standard deviation of positive residuals.

Residuals exceeding this threshold are flagged as outliers. The proportion of outliers is calculated as the ratio of outliers to the total number of predictions.

Results and Discussion

This study presents a comprehensive analysis of predictive modeling in maritime environments using three distinct datasets: the Tarragona dataset, the Barcelona dataset [224], and the EMSO-OBSEA dataset from the OBSEA Expandable Seafloor Observatory [226]. Each dataset brings unique challenges and insights into the maritime conditions observed.

5.0.7 Results

Tarragona Dataset

The Tarragona dataset, which is the primary dataset used for training the model, demonstrates excellent model performance. The Mean Absolute Error (MAE) is approximately 0.0955 m, and the Mean Squared Error (MSE) is $0.0431m$. The Root Mean Squared Error (RMSE) is 0.208 m, indicating small deviations between predicted and actual wave heights. The Pearson Correlation Coefficient is 0.9354, reflecting a strong positive correlation and high model fidelity. The Precision-like Metric reaches 99.07%, suggesting that the model's predictions closely align with the actual data. The proportion of outliers detected is low at 0.24%, with only 1.64% of actual values falling outside the predicted intervals. Despite gaps extending up to 103 days, the model's high correlation coefficient indicates robust performance and resilience to missing data. (Figures 5.3,5.4,and 5.5).

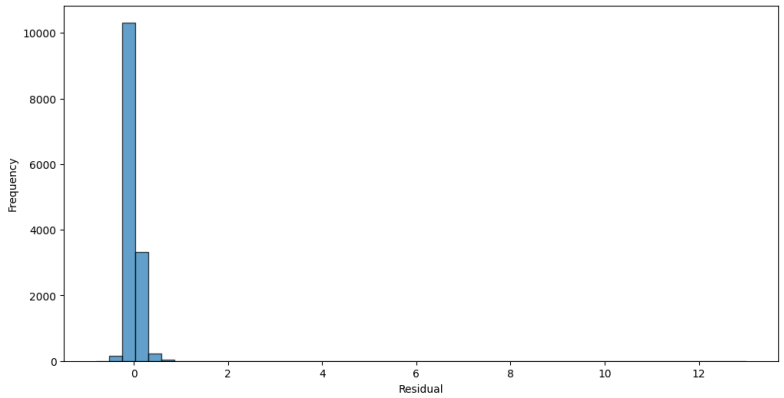


Figure 5.3: Distribution of residuals for the Tarragona dataset.

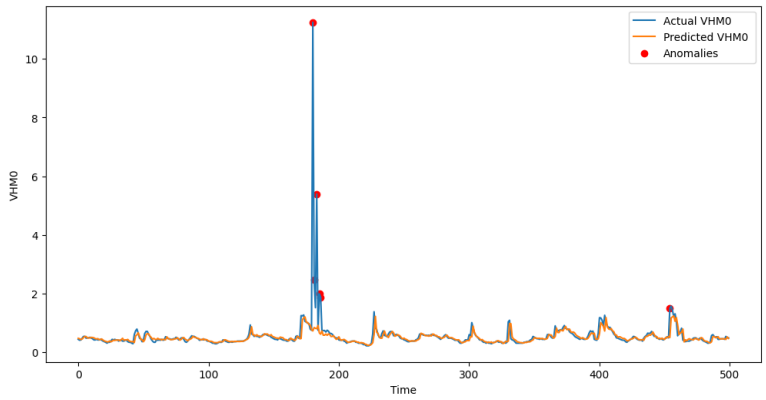


Figure 5.4: Comparison of actual and predicted VHM0 (m) with flagged anomalies (Tarragona dataset).

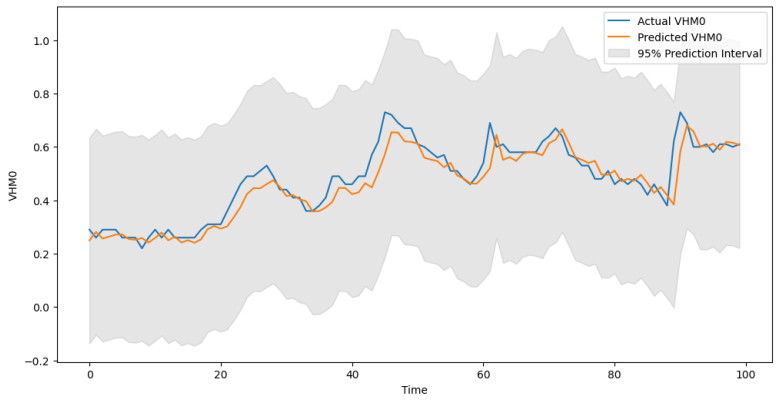


Figure 5.5: Comparison of Actual and Predicted VHM0 (m) with 95% Prediction Intervals (Tarragona dataset).

Barcelona Dataset

The model also performs well on the Barcelona dataset, with an MAE of 0.0903 m and an MSE of 0.0330 m. The RMSE is 0.182 m, indicating slightly better performance compared to the Tarragona dataset. The Pearson Correlation Coefficient of 0.9517 indicates a very strong correlation between predicted and actual values. However, the model identifies a higher proportion of outliers at 2.04%, with 4.58% of actual values lying outside the prediction intervals. These figures suggest slightly more complex wave patterns in this dataset. The maximum gap duration is 9 days, and the strong correlation coefficient reflects the model's robustness in handling data gaps (Figure 5.6, Figure 5.7 and Figure 5.8).

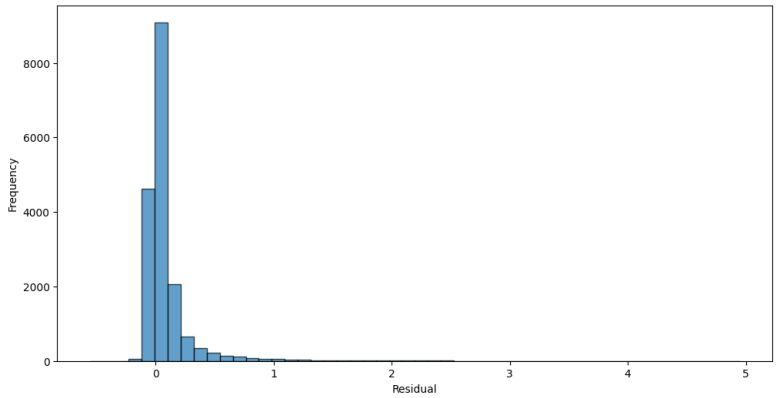


Figure 5.6: Distribution of residuals for the Barcelona dataset.

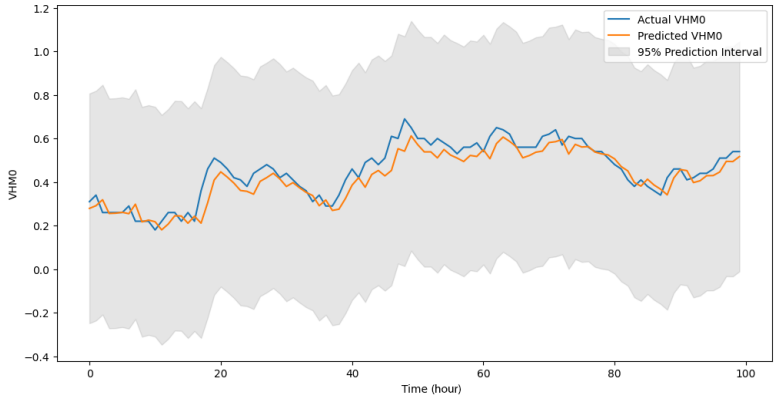


Figure 5.7: Comparison of Actual and Predicted VHM0 (m) with 95% Prediction Intervals (Barcelona dataset).

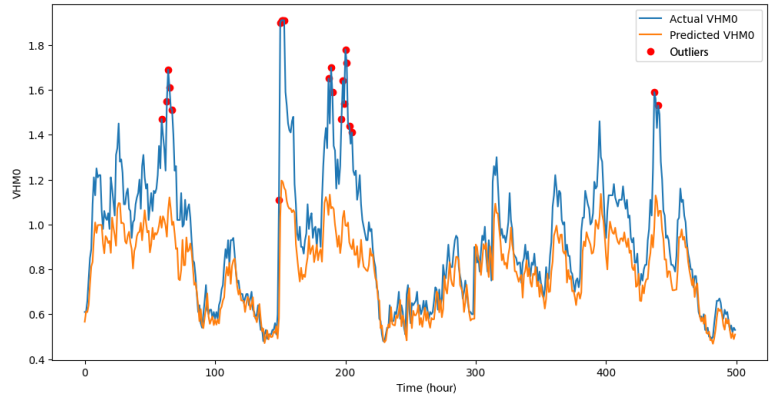


Figure 5.8: Comparison of Actual and Predicted VHM0 (m) with flagged outliers (Barcelona dataset).

EMSO-OBSEA Dataset

The EMSO-OBSEA dataset shows the model's adaptability to different data sources with an MAE of $0.1290m$ and an MSE of $0.0633\ m$. The RMSE is higher at

0.252 m, reflecting greater variability in wave height measurements. The Pearson Correlation Coefficient is 0.8857, indicating a strong correlation but slightly lower than those obtained with the Tarragona and Barcelona datasets. This dataset has a higher proportion of outliers at 4.11% and a higher percentage of actual values outside the prediction intervals at 7.77%. The increased outlier rates and larger prediction errors may be attributed to periods of storm data recorded by the OBSEA sensors (Figure. 5.9, Figure. 5.10 and Figure. 5.11). Despite gaps up to 280 days, the model maintains a strong correlation, demonstrating its robustness in dealing with extensive missing data.

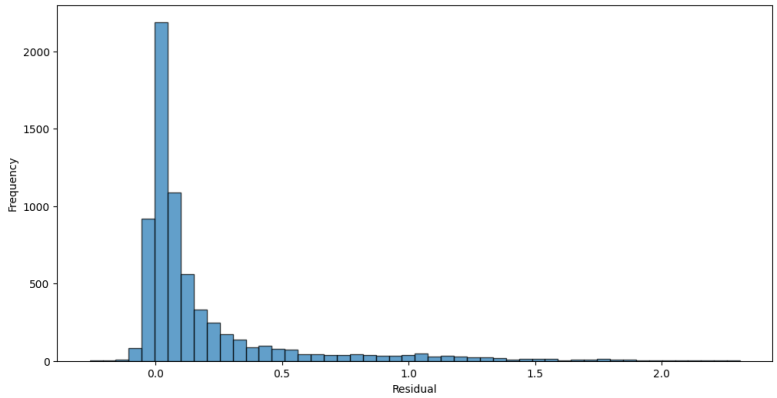


Figure 5.9: Distribution of residuals for the EMSO-OBSEA dataset.

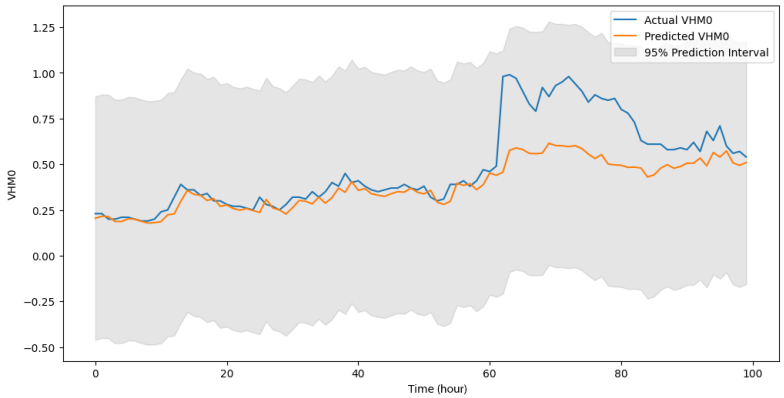


Figure 5.10: Comparison of Actual and Predicted VHM0 (m) with 95% Prediction Intervals (EMSO-OBSEA dataset).

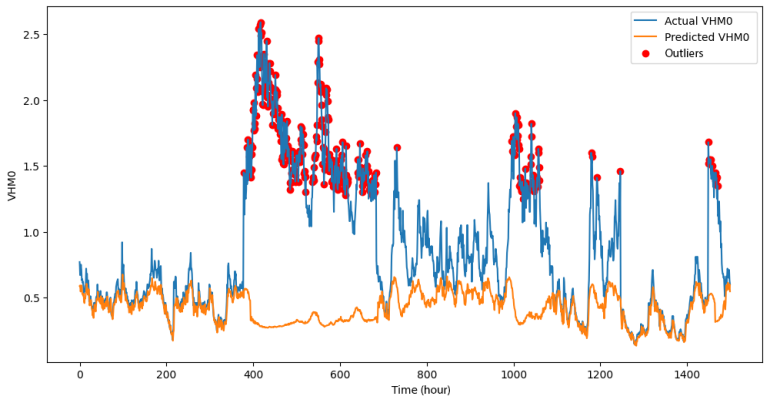


Figure 5.11: Comparison of Actual and Predicted VHM0 (m) with flagged outliers (EMSO-OBSEA dataset).

5.0.8 Discussion

The visual analysis across all datasets demonstrates the model's effectiveness in tracking and predicting significant wave heights within acceptable error margins as supported by similar findings in [234]. While each dataset presents its own set of challenges, the model consistently achieves high accuracy and correlation metrics, proving its utility in maritime environmental monitoring.

Model Performance

The model's performance on the Tarragona and Barcelona datasets highlights its robustness and high accuracy. The strong Pearson Correlation Coefficients and low error metrics indicate that the model effectively learns and generalizes the patterns in these datasets. The slightly higher outlier rates in the Barcelona dataset may reflect more complex wave patterns.

Adaptability to Diverse Data Sources

The performance on the EMSO-OBSEA dataset, while slightly lower, still shows strong predictive capability. The higher error metrics and outlier rates suggest that the model, trained primarily on port data, faces challenges when applied to open ocean sensor data from the OBSEA observatory. This result underscores the importance of considering the source and nature of training data when deploying predictive models in diverse environmental conditions.

Robustness to Data Gaps

A notable observation is the model's resilience to data gaps. The model maintains high Pearson Correlation Coefficients throughout, despite the presence of significant gaps in all datasets. This consistency contrasts with other methods, which often see a decline in performance when faced with incomplete data. Traditional models, such as linear regression or even some machine learning approaches, can struggle to maintain accuracy in the presence of data gaps, leading to reduced correlation and predictive power [235]. The ability of the proposed model to maintain strong correlations despite these gaps highlights its robustness and reliability, making it particularly well-suited for real-world maritime monitoring scenarios where data incompleteness is common. This robustness could be considered a significant advancement over other methods, which may not perform as well under similar conditions [236, 237].

Outlier assessment

outlier assessment is a crucial aspect of maritime monitoring. The low proportion of outliers in the Tarragona dataset indicates stable wave conditions. In contrast, the higher outlier rates in the EMSO-OBSEA dataset highlight the variability and significant wave events captured by the OBSEA sensors, which are critical for maritime operations and safety.

5.0.9 Limitations and Challenges in Outlier Assessment as an Anomaly Detection Approach

Adopting a statistical approach based on residual analysis like outlier assessments can be practical for anomaly detection, particularly when labeled anomalies are not available [238]. This method operates under the assumption that most data points are normal and that outliers are rare, which is often a reasonable presumption in real-world scenarios. The approach can generalize well across different datasets by setting thresholds based on residual statistics, such as the mean and standard deviation. This generalizability was demonstrated by the consistent performance across multiple datasets in the study. Furthermore, statistical methods are relatively simple to implement and interpret, providing clear criteria for outlier assessments.

However, there are limitations to this approach. The effectiveness of residual analysis heavily relies on the model's predictive accuracy. Significant prediction errors can result in residuals that do not accurately represent outliers. Additionally,

the choice of threshold is critical; setting it too low may result in many false positives while setting it too high might miss genuine outliers. Without ground truth labels, it is challenging to quantitatively validate the model's outlier assessment capability, as the evaluation metrics primarily assess prediction accuracy rather than outlier assessment performance [239, 240]. A statistical approach based on residual analysis is practical for anomaly detection, especially when labeled anomalies are unavailable [238]. This method assumes most data points are normal and outliers are rare, a reasonable presumption in many real-world cases. By setting thresholds based on residual statistics, it generalizes well across datasets, as shown by the study's consistent performance. Additionally, it is simple to implement and interpret. However, this approach depends on the model's predictive accuracy. Large errors can misidentify outliers, and the choice of threshold is crucial-too low results in false positives, while too high may miss genuine outliers. Without ground truth labels, validating outlier detection remains challenging [239, 240].

5.0.10 Advancements Over Existing Methodologies

The GRU model demonstrated high predictive accuracy across all datasets, as evidenced by Pearson correlation coefficients above 0.88. The model's resilience to data gaps, maintaining strong performance despite missing data, highlights its robustness for real-world applications. This robustness is particularly crucial for maritime monitoring, where data collection can be sporadic due to harsh conditions.

The proposed data-driven approach to predictive modeling in maritime environments offers significant improvements over traditional methods. LSTM networks, though effective, are computationally intensive and complex [218]. GRU models provide a simpler yet effective alternative, reducing computational load and enhancing interpretability. The streamlined GRU architecture handles multivariate inputs and long-term dependencies, improving robustness and accuracy [219].

Unidirectional GRUs maintain computational efficiency, suitable for real-time applications [220]. Validation across datasets from Tarragona, Barcelona, and EMSO OBSEA demonstrates consistent performance, highlighting the model's generalizability and robustness, even with data gaps.

These advancements highlight the methodology's potential to enhance predictive modeling and outliers analysis in maritime environments, offering a more efficient, interpretable, and robust solution. Future research should focus on validating anomaly detection using labeled datasets to ensure robustness and accuracy.

5.1 Conclusion

This study presented a framework for predictive modeling in maritime environments using digital twins (DTs). A Gated Recurrent Unit (GRU) neural network was employed to train and validate the model on datasets from in-situ wave height sensors in Tarragona, Barcelona, and the EMSO-OBSEA observatory. Rigorous data preprocessing, including normalization and sequence creation, ensured robust model performance. The model exhibited high predictive accuracy and resilience, maintaining strong predictive capabilities despite data gaps, as indicated by Pearson correlation coefficients. outliers were effectively detected through residual analysis, enhancing maritime monitoring and decision-making. Overall, this research highlighted the potential of DTs in improving maritime operations' accuracy and efficiency. The findings provide a foundation for further advancements in environmental monitoring and predictive analytics. Future research could explore integrating additional data sources, such as satellite observations and real-time weather forecasts, to enhance model input and predictive accuracy. Developing real-time deployment strategies for predictive models within DT ecosystems, optimized for edge devices or cloud platforms, would improve maritime decision-making processes, environmental monitoring, operational efficiency, and safety.

Chapter 6

Conclusion and Future Directions

This dissertation has investigated the integration of measurement technologies, sensor data processing, and machine learning in the development of digital twins for structural monitoring, UAV/UUV-based inspections, and environmental modeling. By addressing key challenges in uncertainty quantification, deep-learning-driven defect detection, and sensor fusion, the proposed methodologies have contributed to advancing the accuracy, efficiency, and automation of digital twin applications.

The research focused on four core areas: (1) visual localization through monocular visual odometry (VO) and sensor fusion, (2) UAV-based monitoring and crack detection, (3) marker-based tracking for structural health assessment, and (4) marine digital twins for predictive modeling of significant wave height. Each of these components was systematically evaluated to assess their strengths and limitations, leading to technical insights that inform future research directions.

6.1 Summary of Key Contributions and Findings

6.1.1 UAV-Based Structural Monitoring and Crack Detection

This work developed and validated a deep learning-based crack detection framework leveraging YOLOv8 segmentation models and triplet loss-based learning. UAVs were deployed to collect high-resolution imagery of concrete structures, and the proposed methods demonstrated significant improvements over traditional manual inspection techniques. The results showed that the YOLO-based segmentation model achieved an average precision of 87%, while the triplet loss model for crack verification attained an accuracy of 97.36%.

However, several factors affected the robustness of the model. Lighting variations,

camera angles, and resolution constraints influenced detection accuracy, requiring further improvements in data augmentation and domain adaptation. The findings emphasize the need for multi-modal sensor integration to compensate for the limitations of purely vision-based approaches.

6.1.2 Marker-Based Tracking for Structural Health Monitoring

A novel marker-based tracking system was introduced to monitor masonry models with improved precision and cost efficiency. The DeepTag method demonstrated a 42% reduction in measurement uncertainty compared to traditional ArUco markers, with an expanded uncertainty of 0.29° in orientation measurement.

While the approach provided a low-cost and scalable alternative to high-end motion capture systems, its effectiveness was limited by occlusion, marker degradation, and placement constraints. The study highlights the need for dynamic marker detection models, which can adapt to occlusions and environmental conditions using machine learning-based feature extraction.

6.1.3 Visual Localization and Odometry in GNSS-Denied Environments

This dissertation also investigated monocular VO and VIO techniques for UAV/UUV navigation in GNSS-denied environments. A measurement uncertainty model was formulated to assess the accuracy of localization estimates, demonstrating how feature extraction methods affect disparity uncertainty. The evaluation showed that ORB and Harris feature detectors outperformed SURF and FAST in terms of stability and robustness.

Simulation results revealed that position drift accumulation could reach over 5 meters within 20 estimates, necessitating the integration of depth sensors and IMU data to mitigate long-term errors. The findings underscore the importance of hybrid localization strategies that combine image-based motion estimation with auxiliary sensor data to improve reliability in real-world applications.

6.1.4 Marine Digital Twin for Environmental Modeling

This work extended digital twin applications to marine environments, developing a GRU-based predictive model for significant wave height forecasting. The model achieved a Pearson correlation of 0.9354, demonstrating strong predictive performance. Furthermore, an outlier assessment method was introduced, improving the

detection of anomalous wave conditions.

However, the model showed limitations in capturing extreme wave events, primarily due to its reliance on historical data without incorporating real-time physical constraints. Future improvements should focus on physics-informed neural networks (PINNs) to integrate domain-specific knowledge into the forecasting model.

While the proposed approaches provided advancements in measurement accuracy, defect detection, and localization, several challenges remain:

1. UAV-based crack detection was effective, but environmental variability reduced robustness. High contrast lighting, sensor noise, and motion blur impacted detection precision, requiring adaptive models capable of real-time adjustments.
2. Marker-based tracking provided cost-effective monitoring but was sensitive to occlusions. The proposed DeepTag markers outperformed conventional methods, but further work is required to optimize marker placement strategies for large-scale monitoring.
3. Monocular VO uncertainty models highlighted the limitations of purely vision-based localization. Sensor drift accumulation exceeded acceptable thresholds, requiring IMU fusion and deep-learning-based feature extraction to improve robustness.
4. The marine predictive model was accurate but struggled with extreme conditions. The model's reliance on historical data without real-time sensor feedback limited adaptability.

6.2 Future Work and Technical Advancements

Based on the findings, several key directions emerge for future research:

Improving UAV-Based Crack Detection Under Variable Conditions Future work should explore multi-modal defect detection by integrating LiDAR and infrared sensors with traditional RGB imaging. Self-supervised learning and domain adaptation techniques should be employed to ensure models remain effective under varying environmental conditions.

Real-time processing should be enhanced using edge AI models deployed directly on UAVs, reducing dependency on cloud computing and enabling on-the-fly defect classification.

6.2.1 Advancing Marker-Based Tracking with AI-Assisted Detection

The sensitivity of marker-based tracking to occlusions could be addressed by developing machine-learning-enhanced marker detection systems. These systems could predict marker locations under partial occlusion, using feature-based tracking models to infer hidden marker positions.

Additionally, integrating adaptive placement algorithms for marker distribution could optimize spatial coverage, ensuring maximum visibility in complex structures. In real infrastructure monitoring, strategically placing markers in areas of high structural stress or known defect-prone regions would enhance tracking accuracy and long-term assessment reliability.

6.2.2 Hybrid Visual-Inertial Localization for Digital Twin Integration

To address the scale ambiguity and drift limitations of monocular VO, future research should focus on hybrid sensor fusion strategies. Combining deep-learning-based feature matching with IMU data and LiDAR depth estimation could significantly enhance localization accuracy.

The application of graph-based SLAM (Simultaneous Localization and Mapping) techniques could also provide improved long-term consistency in digital twin models. Integrating an uncertainty model for SLAM would enable the quantification of localization errors at each node in the graph, enhancing robustness in dynamic and unstructured environments.

6.2.3 Enhancing Marine Digital Twins with Physics-Informed Models

Future research should integrate real-time buoy and satellite data into the marine digital twin framework, refining predictions using physics-informed deep learning models. Such models would enable more accurate forecasting of extreme wave events by combining data-driven learning with physical oceanographic principles.

Furthermore, outlier detection models should be enhanced using deep anomaly detection networks rather than statistical thresholds, improving detection reliability in complex ocean conditions.

6.3 Final Remarks

This dissertation has demonstrated how measurement technologies, UAV/UUV-based inspections, and deep learning models contribute to enhanced digital twin applications. The results highlight the potential of integrating sensor-based monitoring with AI-driven analysis, providing new opportunities for real-time decision-making, predictive maintenance, and automated structural assessments.

However, the research also identifies key challenges in sensor fusion, environmental variability, and data reliability that must be addressed in future work. By refining localization methods, improving adaptive detection models, and integrating physics-based constraints into predictive analytics, digital twins can evolve into more robust and reliable systems.

This work lays the foundation for further advancements in intelligent digital twin systems, offering new insights into how sensor information processing, uncertainty modeling, and AI-driven analytics can revolutionize the way physical structures are monitored and maintained. As technology progresses, the integration of real-time data acquisition, high-fidelity simulations, and self-adaptive AI models will further bridge the gap between physical and digital environments, unlocking new dimensions in predictive intelligence and automated decision-making.

Acknowledgements

First and foremost, I want to express my deepest gratitude to my family for their unwavering support throughout this journey. Their encouragement and belief in me have been the cornerstone of my motivation and success.

I would also like to extend my heartfelt thanks to all my professors and colleagues at the LESIM Lab of the University of Sannio. Their expertise, guidance, and collaborative spirit have profoundly influenced my research, providing both a stimulating academic environment and invaluable mentorship during my Ph.D. studies.

Finally, I acknowledge the assistance of language and proofreading tools, particularly ChatGPT, which I used to refine and clarify various sections of this dissertation for improved phrasing and grammar.

Bibliography

[1] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21 980–22 012, 2020.

[2] S. A. Rahimi, A. Baradaran, F. Khameneifar, G. Gore, and A. M. Issa, "Decide-twin: A framework for ai-enabled digital twins in clinical decision-making," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–10, 2024.

[3] A. E. Bondoc, M. Tayefeh, and A. Barari, "Learning phase in a live digital twin for predictive maintenance," *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 13, 2022.

[4] I. Abdullahi, S. Longo, and M. Samie, "Towards a distributed digital twin framework for predictive maintenance in industrial internet of things (iiot)," *Sensors*, vol. 24, no. 8, p. 2663, 2024.

[5] A. Neyestani, F. Picariello, A. Basiri, P. Daponte, and L. D. Vito, "Survey and research challenges in monocular visual odometry," in *2023 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, 2023, pp. 107–112.

[6] A. Neyestani, F. Picariello, I. Ahmed, P. Daponte, and L. De Vito, "From pixels to precision: A survey of monocular visual odometry in digital twin applications," *Sensors*, vol. 24, no. 4, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/4/1274

[7] P. Daponte, L. De Vito, A. Neyestani, F. Picariello, and I. Tudosa, "Measurement uncertainty model for relative visual localization of uav by a monocular camera," in *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, 2023, pp. 251–256.

[8] W. Wei, L. Tan, G. Jin, L. Lu, and C. Sun, "A survey of uav visual navigation based on monocular slam," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*. IEEE, 2018, pp. 1849–1853.

[9] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[10] D. Zou, P. Tan, and W. Yu, "Collaborative visual SLAM for multiple agents: A brief survey," *Virtual Reality and Intelligent Hardware*, vol. 1, pp. 461–482, 10 2019.

[11] G. Yang, Y. Wang, J. Zhi, W. Liu, Y. Shao, and P. Peng, "A Review of Visual Odometry in SLAM Techniques," in *2020 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, 2020, pp. 332–336.

[12] M. R. Razali, A. Athif, M. Faudzi, and A. U. Shamsudin, "Visual Simultaneous Localization and Mapping: A review," *PERINTIS eJournal*, vol. 12, pp. 23–34, 2022.

[13] L. R. Agostinho, N. M. Ricardo, M. I. Pereira, A. Hiolle, and A. M. Pinto, "A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions," *IEEE Access*, vol. 10, pp. 72 182–72 205, 2022.

[14] A. Couturier and M. A. Akhloufi, "A review on absolute visual localization for uav," *Robotics and Autonomous Systems*, vol. 135, p. 103666, 2021.

[15] L. Ma, D. Meng, S. Zhao, and B. An, "Visual localization with a monocular camera for unmanned aerial vehicle based on landmark detection and tracking using yolov5 and deepsort," *International Journal of Advanced Robotic Systems*, vol. 20, no. 3, 2023.

[16] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual slam: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[17] N. Gadipudi, I. Elamvazuthi, C.-K. Lu, S. Paramasivam, S. Su, and S. Yogamani, "WPO-Net: Windowed Pose Optimization Network for Monocular Visual Odometry Estimation," *Sensors*, vol. 21, p. 8155, 2021.

[18] Z. Xu, *Stereo Visual Odometry With Windowed Bundle Adjustment*. University of California, Los Angeles, 2015.

[19] K. Wang, S. Ma, J. Chen, F. Ren, and J. Lu, "Approaches, Challenges, and Applications for Deep Visual Odometry: Toward Complicated and Emerging Areas," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, pp. 35–49, 3 2022.

[20] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment–a modern synthesis," in *International Workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[21] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19 929–19 953, 2022.

[22] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[23] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[24] J. Graeter, A. Wilczynski, and M. Lauer, "LIMO: Lidar-Monocular Visual Odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7872–7879.

[25] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[26] F. Fraundorfer and D. Scaramuzza, "Visual Odometry: Part II: Matching, Robustness, Optimization, and Applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, June 2012.

[27] A. Basiri, V. Mariani, and L. Glielmo, "Enhanced V-SLAM combining SVO and ORB-SLAM2, with reduced computational complexity, to improve autonomous indoor mini-drone navigation under varying conditions," in *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, 2022, pp. 1–7.

[28] M. He, C. Zhu, Q. Huang, B. Ren, and J. Liu, "A review of monocular visual odometry," *Visual Computer*, vol. 36, pp. 1053–1065, 5 2020.

[29] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, pp. 1–26, 2016.

[30] X. Wang, F. Xue, Z. Yan, W. Dong, Q. Wang, and H. Zha, "Continuous-time stereo visual odometry based on dynamics model," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 388–403.

[31] C. Pottier, J. Petzing, F. Eghtedari, N. Lohse, and P. Kinnell, "Developing digital twins of multi-camera metrology systems in blender," *Measurement Science and Technology*, vol. 34, no. 7, p. 075001, mar 2023. [Online]. Available: https://dx.doi.org/10.1088/1361-6501/acc59e

[32] W. Feng, S. Z. Zhao, C. Pan, A. Chang, Y. Chen, Z. Wang, and A. Y. Yang, "Digital twin tracking dataset (dttd): A new rgb+ depth 3d dataset for longer-range object tracking applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 3288–3297.

[33] T. Sundby, J. M. Graham, A. Rasheed, M. Tabib, and O. San, "Geometric change detection in digital twins," *Digital*, vol. 1, no. 2, pp. 111–129, 2021. [Online]. Available: https://www.mdpi.com/2673-6470/1/2/9

[34] S. Neethirajan and B. Kemp, "Digital Twins in Livestock Farming," *Animals*, vol. 11, no. 4, p. 1008, 2021.

[35] O. Döbrich and C. Brauner, "Machine vision system for digital twin modeling of composite structures," *Frontiers in Materials*, vol. 10, p. 1154655, 2023.

[36] H.-H. Benzon, X. Chen, L. Belcher, O. Castro, K. Branner, and J. Smit, "An operational image-based digital twin for large-scale structures," *Applied Sciences*, vol. 12, no. 7, p. 3216, 2022.

[37] E. Rosten, R. Porter, and T. Drummond, "Faster and Better: A Machine Learning Approach to Corner Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[38] Q. Yang, C. Qiu, L. Wu, and J. Chen, "Image Matching Algorithm Based on Improved FAST and RANSAC," in *2021 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2021, pp. 142–147.

[39] S.-K. Lam, G. Jiang, M. Wu, and B. Cao, "Area-Time Efficient Streaming Architecture for FAST and BRIEF Detector," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 2, pp. 282–286, 2019.

[40] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

[41] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555.

[42] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI'81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, p. 674–679.

[43] R. Mohr and B. Triggs, "Projective Geometry for Image Analysis," in *XVIIIth International Symposium on Photogrammetry & Remote Sensing (ISPRS '96)*, Vienna, Austria, Jul. 1996, tutorial given at International Symposium on Photogrammetry & Remote Sensing.

[44] Y. Ma, S. Soatto, J. Kosecká, and S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*, ser. Interdisciplinary Applied Mathematics. Springer New York, 2012.

[45] R. Lozano, *Unmanned Aerial Vehicles: Embedded Control*, ser. ISTE. Wiley, 2013.

[46] I. Abaspur Kazerouni, L. Fitzgerald, G. Dooly, and D. Toal, "A survey of state-of-the-art on visual SLAM," *Expert Systems with Applications*, vol. 205, p. 117734, 2022.

[47] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 15–22.

[48] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[49] K. Yang, H.-T. Fu, and A. C. Berg, "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 3403–3412.

[50] D. Zhou, Y. Dai, and H. Li, "Ground-Plane-Based Absolute Scale Estimation for Monocular Visual Odometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, 2020.

[51] L. Cao, J. Ling, and X. Xiao, "Study on the influence of image noise on monocular feature-based visual slam based on ffdnet," *Sensors*, vol. 20, no. 17, p. 4922, 2020.

[52] X. Qiu, H. Zhang, W. Fu, C. Zhao, and Y. Jin, "Monocular visual-inertial odometry with an unbiased linear system model and robust feature tracking front-end," *Sensors*, vol. 19, no. 8, p. 1941, 2019.

[53] L. Jinyu, Y. Bangbang, C. Danpeng, W. Nan, Z. Guofeng, and B. Hujun, "Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 4, pp. 386–410, 2019.

[54] S. Chiodini, R. Giubilato, M. Pertile, and S. Debei, "Retrieving Scale on Monocular Visual Odometry Using Low-Resolution Range Sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, p. 5875, 2020.

[55] H. Lee, H. Lee, I. Kwak, C. Sung, and S. Han, "Effective feature-based downward-facing monocular visual odometry," *IEEE Transactions on Control Systems Technology*, vol. 32, no. 1, pp. 266–273, 2024.

[56] T. Shan, B. Englot, C. Ratti, and R. Daniela, "LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May. Institute of Electrical and Electronics Engineers Inc., 2021, pp. 5692–5698.

[57] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified Multi-Modal Landmark Tracking for Tightly Coupled Lidar-Visual-Inertial Odometry," *IEEE Robotics and Automation Letters*, vol. 6, pp. 1004–1011, 4 2021.

[58] B. Fang, Q. Pan, and H. Wang, "Direct monocular visual odometry based on lidar vision fusion," in *2023 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. IEEE, 2023, pp. 256–261.

[59] C. Campos, R. Elvira, J. J. Rodriguez, J. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, pp. 1874–1890, 12 2021.

[60] W. Huang, W. Wan, and H. Liu, "Optimization-Based Online Initialization and Calibration of Monocular Visual-Inertial Odometry Considering Spatial-Temporal Constraints," *Sensors*, vol. 21, p. 2673, 2021.

[61] L. Zhou, S. Wang, and M. Kaess, "DPLVO: Direct Point-Line Monocular Visual Odometry; DPLVO: Direct Point-Line Monocular Visual Odometry," *IEEE Robotics and Automation Letters*, vol. 6, p. 7113, 2021.

[62] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular Visual Odometry Through Unsupervised Deep Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[63] X. Ban, H. Wang, T. Chen, Y. Wang, and Y. Xiao, "Monocular Visual Odometry Based on Depth and Optical Flow Using Deep Learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.

[64] L. Lin, W. Wang, W. Luo, L. Song, and W. Zhou, "Unsupervised monocular visual odometry with decoupled camera pose estimation," *Digital Signal Processing: A Review Journal*, vol. 114, 7 2021.

[65] U. H. Kim, S. H. Kim, and J. H. Kim, "SimVODIS: Simultaneous Visual Odometry, Object Detection, and Instance Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 428–441, 1 2022.

[66] Y. Almalioglu, M. Turan, M. R. U. Saputra, P. P. de Gusmão, A. Markham, and N. Trigoni, "Selfvio: Self-supervised deep monocular visual–inertial odometry and depth estimation," *Neural Networks*, vol. 150, pp. 119–136, 2022.

[67] H. Matsuki, L. V. Stumberg, V. Usenko, J. Stückler, and D. Cremers, "Omnidirectional DSO: Direct Sparse Odometry With Fisheye Cameras," *IEEE Robotics and Automation Letters*, vol. 3, p. 3693, 2018.

[68] R. Tian, Y. Zhang, D. Zhu, S. Liang, S. Coleman, and D. Kerr, "Accurate and Robust Scale Recovery for Monocular Visual Odometry Based on Plane Geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[69] K. B. Ozyoruk, G. I. Gokceler, T. L. Bobrow, G. Coskun, K. Incetan, Y. Almalioglu, F. Mahmood, E. Curto, L. Perdigoto, M. Oliveira, H. Sahin, H. Araujo, H. Alexandrino, N. J. Durr, H. B. Gilbert, and M. Turan, "EndoSLAM dataset and an unsupervised monocular visual odometry and depth estimation approach for endoscopic videos," *Medical Image Analysis*, vol. 71, 7 2021.

[70] C. Fan, J. Hou, and L. Yu, "A nonlinear optimization-based monocular dense mapping system of visual-inertial odometry," *Measurement: Journal of the International Measurement Confederation*, vol. 180, 8 2021.

[71] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1278–1289.

[72] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.

[73] Y. Aksoy and A. A. Alatan, "Uncertainty modeling for efficient visual odometry via inertial sensors on mobile devices," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 3397–3401.

[74] D. Ross, M. De Petrillo, J. Strader, and J. N. Gross, "Uncertainty estimation for stereo visual odometry," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3263–3284.

[75] P. V. Gakne and K. O'Keefe, "Tackling the scale factor issue in a monocular visual odometry using a 3d city model," in *ITSNT 2018, International Technical Symposium on Navigation and Timing*, 2018.

[76] D. V. Hamme, W. Goeman, P. Veelaert, and W. Philips, "Robust monocular visual odometry for road vehicles using uncertain perspective projection," *EURASIP Journal on Image and Video Processing*, vol. 2015, no. 1, pp. 1–21, 2015.

[77] D. Van Hamme, P. Veelaert, and W. Philips, "Robust visual odometry using uncertainty models," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2011, pp. 1–12.

[78] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[79] B. Brzozowski, P. Daponte, L. De Vito, F. Lamonaca, F. Picariello, M. Pompetti, I. Tudosa, and K. Wojtowicz, "A remote-controlled platform for uas testing," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 8, pp. 48–56, 2018.

[80] J. Jeon, S. Jung, E. Lee, D. Choi, and H. Myung, "Run Your Visual-Inertial Odometry on NVIDIA Jetson: Benchmark Tests on a Micro Aerial Vehicle," *IEEE Robotics and Automation Letters*, vol. 6, pp. 5332–5339, 7 2021.

[81] A. Neyestani, F. Picariello, A. Basiri, P. Daponte, and L. D. Vito, "Survey and research challenges in monocular visual odometry," in *Proc. of 2023 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, Milano, Italy, 2023, pp. 107–112.

[82] "Evaluation of measurement data - guide to the expression of uncertainty in measurement," JCGM 100:2008, BIPM.

[83] "Uav package delivery," example project available on MATLAB, https://it.mathworks.com/help/uav/ug/uav-package-delivery.html.

[84] "Simulation 3d scene configuration," Scene configuration for 3D simulation environment by MATLAB, https://it.mathworks.com/help/uav/ref/simulation3dsceneconfiguration.html.

[85] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[86] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. of the 4th Alvey Vision Conference*, August 1988, pp. 147–151.

[87] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. of the IEEE International Conference on Computer Vision*, vol. 2, October 2005, pp. 1508–1511.

[88] D. Dissanayaka, T. R. Wanasinghe, O. De Silva, A. Jayasiri, and G. K. I. Mann, "Review of Navigation Methods for UAV-Based Parcel Delivery," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 1, pp. 1068–1082, 2024.

[89] N. Stierlin, M. Risch, and L. Risch, "Current Advancements in Drone Technology for Medical Sample Transportation," *Logistics*, vol. 8, no. 4, 2024. [Online]. Available: https://www.mdpi.com/2305-6290/8/4/104

[90] A. Neyestani, F. Picariello, A. Basiri, P. Daponte, and L. D. Vito, "Survey and research challenges in monocular visual odometry," in *2023 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, 2023, pp. 107–112.

[91] C. Zhang, L. Chen, and S. Yuan, "St-vio: Visual-inertial odometry combined with image segmentation and tracking," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 8562–8570, 2020.

[92] A. Neyestani, F. Picariello, I. Ahmed, P. Daponte, and L. De Vito, "From Pixels to Precision: A Survey of Monocular Visual Odometry in Digital Twin Applications," *Sensors*, vol. 24, no. 4, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/4/1274

[93] S. Kannapiran, N. Bendapudi, M.-Y. Yu, D. Parikh, S. Berman, A. Vora, and G. Pandey, "Stereo Visual Odometry with Deep Learning-Based Point and Line Feature Matching Using an Attention Graph Neural Network," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3491–3498.

[94] M. He, C. Chen, J. Liu, C. Li, X. Lyu, G. Huang, and Z. Meng, "Aerialvl: A dataset, baseline and algorithm framework for aerial-based visual localization with reference map," *IEEE Robotics and Automation Letters*, vol. 9, pp. 8210–8217, 2024.

[95] P. Daponte, R. S. Olivito, and G. Spadea, "Ultrasonic and laser measurements in structural contact problems," *Materials and Structures*, vol. 25, pp. 42–48, 1992.

[96] X. Ye, F. Song, Z. Zhang, and Q. Zeng, "A Review of Small UAV Navigation System Based on Multisource Sensor Fusion," *IEEE Sensors Journal*, vol. 23, no. 17, pp. 18 926–18 948, 2023.

[97] Z. Yue, C. Tang, and Y. Gao, "A Novel Three-Stage Robust Adaptive Filtering Algorithm for Visual-Inertial Odometry in GNSS-Denied Environments," *IEEE Sensors Journal*, vol. 23, no. 15, pp. 17 499–17 509, 2023.

[98] L. Yu, E. Yang, B. Yang, Z. Fei, and C. Niu, "A robust learned feature-based visual odometry system for uav pose estimation in challenging indoor environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, 2023.

[99] R. Duan, D. P. Paudel, C. Fu, and P. Lu, "Stereo orientation prior for uav robust and accurate visual odometry," *Changhong Fu is with the School of Mechanical Engi-neering*, vol. 27, 2022. [Online]. Available: https://doi.org/10.1109/TMECH.2022.3140923.

[100] S. K. Paul, P. Hoseini, M. Nicolescu, and M. Nicolescu, "Performance analysis of keypoint detectors and binary descriptors under varying degrees of photometric and geometric transformations," *arXiv preprint arXiv:2012.04135*, 2020.

[101] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 11 2004.

[102] J. Shi and C. Tomasi, "Good Features to Track," *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 600, 03 2000.

[103] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[104] Y. Chen, G. Wang, P. An, Z. You, and X. Huang, "Fast And Accurate Homography Estimation Using Extendable Compression Network," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 1024–1028.

[105] M. Grimaldi, D. Nakath, M. She, and K. Köser, "Investigation of the challenges of underwater-visual-monocular-slam," *arXiv preprint arXiv:2306.08738*, 2023.

[106] M. Ferrera, J. Moras, P. Trouvé-Peloux, and V. Creuze, "Real-time monocular visual odometry for turbid and dynamic underwater environments," *Sensors*, vol. 19, no. 3, p. 687, 2019.

[107] C. Amarasinghe, A. Rathnaweera, and S. Maithripala, "U-vip-slam: Underwater visual-inertial-pressure slam for navigation of turbid and dynamic environments," *Arabian Journal for Science and Engineering*, vol. 49, no. 3, pp. 3193–3207, 2024.

[108] J. Yang, M. Gong, G. Nair, J. H. Lee, J. Monty, and Y. Pu, "Knowledge distillation for feature extraction in underwater vslam," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5163–5169.

[109] K. Köser and U. Frese, "Challenges in underwater visual navigation and slam," *AI technology for underwater robots*, pp. 125–135, 2020.

[110] Y. Song, J. Qian, R. Miao, W. Xue, R. Ying, and P. Liu, "Haud: A high-accuracy underwater dataset for visual-inertial odometry," in *2021 IEEE Sensors*, 2021, pp. 1–4.

[111] B. Teixeira, H. Silva, A. Matos, and E. Silva, "Deep learning for underwater visual odometry estimation," *IEEE Access*, vol. 8, pp. 44 687–44 701, 2020.

[112] R. Miao, J. Qian, Y. Song, R. Ying, and P. Liu, "Univio: Unified direct and feature-based underwater stereo visual-inertial odometry," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[113] A. Neyestani, F. Picariello, A. Basiri, P. Daponte, and L. D. Vito, "Survey and research challenges in monocular visual odometry," in *2023 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, 2023, pp. 107–112.

[114] C. Riu, V. Nozick, P. Monasse, and J. Dehais, "Classification performance of ransac algorithms with automatic threshold estimation," in *17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022)*, vol. 5. Scitepress, 2022, pp. 723–733.

[115] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *British Machine Vision Conference*, vol. 34. BMVA Press, 2013, pp. 1173–1184.

[116] P. Daponte, L. De Vito, A. Neyestani, F. Picariello, and I. Tudosa, "Measurement uncertainty model for relative visual localization of uav by a monocular camera," in *2023 IEEE International Workshop on Technologies for Defense and Security (TechDefense)*, 2023, pp. 251–256.

[117] A. Neyestani, F. Picariello, I. Ahmed, P. Daponte, and L. De Vito, "From pixels to precision: A survey of monocular visual odometry in digital twin applications," *Sensors*, vol. 24, no. 4, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/4/1274

[118] B. Skorohod, S. Fateev, and P. Zhilyakov, "Preprocessing and future extraction for visual underwater odometry," in *2022 International Russian Automation Conference (RusAutoCon)*, 2022, pp. 361–366.

[119] Y. Randall and T. Treibitz, "Flsea: Underwater visual-inertial and stereo-vision forward-looking datasets," 2023. [Online]. Available: https://arxiv.org/abs/2302.12772

[120] C. Riu, V. Nozick, and P. Monasse, "Automatic ransac by likelihood maximization," *Image Processing On Line*, vol. 12, pp. 27–49, 2022.

[121] B. Alhijawi and A. Awajan, "Genetic algorithms: Theory, genetic operators, solutions, and applications," *Evolutionary Intelligence*, vol. 17, no. 3, pp. 1245–1256, 2024.

[122] V. Rodehorst and O. Hellwich, "Genetic algorithm sample consensus (gasac) - a parallel strategy for robust parameter estimation," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, 2006, pp. 103–103.

[123] A. Falahzadeh, D. M. Toma, M. Francescangeli, D. Chatzievangelou, M. Nogueras, E. Martínez, M. Carandell, M. Tangerlini, L. Thomsen, G. Picardi, M. Le Bris, L. Dominguez, J. Aguzzi, and J. del Río, "A new coastal crawler prototype to expand the ecological monitoring radius of obsea cabled observatory," *Journal of Marine Science and Engineering*, vol. 11, no. 4, 2023. [Online]. Available: https://www.mdpi.com/2077-1312/11/4/857

[124] Linovision, "Ipc608uw-10 4k poe ip underwater camera," https://device.report/manual/12341689, accessed: 2024-11-25.

[125] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, pp. 8091–8126, 2021.

[126] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, "Enhancing underwater images and videos by fusion," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 81–88.

[127] X. Fu, P. Zhuang, Y. Huang, Y. Liao, X.-P. Zhang, and X. Ding, "A retinex-based enhancing approach for single underwater image," in *2014 IEEE international conference on image processing (ICIP)*. Ieee, 2014, pp. 4572–4576.

[128] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[129] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, 2011, pp. 2548–2555.

[130] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision,*

Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12. Springer, 2012, pp. 214–227.

[131] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 510–517.

[132] D. Mukherjee, Q. Jonathan Wu, and G. Wang, "A comparative experimental study of image feature detectors and descriptors," *Machine Vision and Applications*, vol. 26, pp. 443–466, 2015.

[133] X. Ling, J. Liu, Z. Duan, and J. Luan, "A robust mismatch removal method for image matching based on the fusion of the local features and the depth," *Remote Sensing*, vol. 16, no. 11, p. 1873, 2024.

[134] A. Vinay, K. A. Vasu, P. Y. Lodha, S. Natarajan, and T. Sudarshan, "Optimal kaze and akaze features for facial similarity matching," in *International Conference on Advances in Computing and Data Sciences*. Springer, 2023, pp. 161–177.

[135] OpenCV, "Camera calibration with opencv," https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html, accessed: 2024-11-23.

[136] D. K, "Checkerboard calibration using opencv," https://github.com/dhvani-k/Checkerboard_Calibration_Using_OpenCV, accessed: 2024-11-23.

[137] DEAP, "Deap documentation," https://deap.readthedocs.io/en/master/, accessed: 2024-11-23.

[138] A. Neyestani, I. Ahmed, P. Daponte, and L. De Vito, "Concrete crack detection and segmentation in civil infrastructures using uavs and deep learning," in *2023 7th International Conference on Internet of Things and Applications (IoT)*. IEEE, 2023, pp. 1–6.

[139] A. Neyestani, F. Picariello, I. Tudosa, P. Daponte, and L. De Vito, "Triplet loss-based concrete crack verification for structural health monitoring and digital twin applications," in *2024 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*. IEEE, 2024, pp. 837–842.

[140] M. Torzoni, M. Tezzele, S. Mariani, A. Manzoni, and K. E. Willcox, "A digital twin framework for civil engineering structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 418, p. 116584, 2024.

[141] S. Teng, X. Chen, G. Chen, and L. Cheng, "Structural damage detection based on transfer learning strategy using digital twins of bridges," *Mechanical Systems and Signal Processing*, vol. 191, p. 110160, 2023.

[142] A. Zar, Z. Hussain, M. Akbar, T. Rabczuk, Z. Lin, S. Li, and B. Ahmed, "Towards vibration-based damage detection of civil engineering structures: overview, challenges, and future prospects," *International Journal of Mechanics and Materials in Design*, pp. 1–72, 2024.

[143] J. Chen and Y. K. Cho, "Crackembed: Point feature embedding for crack segmentation from disaster site point clouds with anomaly detection," *Advanced Engineering Informatics*, vol. 52, p. 101550, 2022.

[144] J. Jia and Y. Li, "Deep Learning for Structural Health Monitoring: Data, Algorithms, Applications, Challenges, and Trends," *Sensors*, vol. 23, no. 21, p. 8824, 2023.

[145] F. Y. Toriumi, T. N. Bittencourt, and M. M. Futai, "Uav-based inspection of bridge and tunnel structures: an application review," *Revista IBRACON de Estruturas e Materiais*, vol. 16, no. 1, p. e16103, 2022.

[146] W. W. Greenwood, J. P. Lynch, and D. Zekkos, "Applications of uavs in civil infrastructure," *Journal of infrastructure systems*, vol. 25, no. 2, p. 04019002, 2019.

[147] X. Li, X. Xu, X. He, X. Wei, and H. Yang, "Intelligent Crack Detection Method Based on GM-ResNet," *Sensors*, vol. 23, no. 20, p. 8369, 2023.

[148] X. Dong, Y. Liu, and J. Dai, "Concrete surface crack detection algorithm based on improved yolov8," *Sensors*, vol. 24, no. 16, p. 5252, 2024.

[149] J. Sorilla, T. S. C. Chu, and A. Y. Chua, "A uav based concrete crack detection and segmentation using 2-stage convolutional network with transfer learning," *HighTech and Innovation Journal*, vol. 5, no. 3, pp. 690–702, 2024.

[150] A. M. Mayya and N. F. Alkayem, "Enhance the concrete crack classification based on a novel multi-stage yolov10-vit framework," *Sensors*, vol. 24, no. 24, p. 8095, 2024.

[151] K. Liu, X. Han, and B. M. Chen, "Deep learning based automatic crack detection and segmentation for unmanned aerial vehicle inspections," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 381–387.

[152] K. Liu and B. M. Chen, "Industrial UAV-Based Unsupervised Domain Adaptive Crack Recognitions: From Database Towards Real-Site Infrastructural Inspections," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9410–9420, 2022.

[153] T. Mitroudas, V. Balaska, A. Psomoulis, and A. Gasteratos, "Embedded light-weight approach for safe landing in populated areas," *arXiv preprint arXiv:2302.14445*, 2023.

[154] G. J. N. Ang, A. K. Goil, H. Chan, J. Lew, X. Lee, R. Mustaffa, T. Jason, Z. Woon, and B. Shen, "A novel application for real-time arrhythmia detection using yolov8," *arXiv preprint*, 2023.

[155] M. Hussain, "Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection," *Machines*, vol. 11, no. 7, p. 677, 2023.

[156] S. Markidis, S. W. Der Chien, E. Laure, I. B. Peng, and J. S. Vetter, "NVIDIA Tensor Core Programmability, Performance & Precision," in *IEEE International Symposium on Parallel and Distributed Processing Workshops (IPDPSW)*. IEEE, 2018, pp. 522–531.

[157] Ultralytics, "Ultralytics yolov8 documentation," https://docs.ultralytics.com, 2023, accessed: 2025-05-12.

[158] S. Vani and T. M. Rao, "An experimental approach towards the performance assessment of various optimizers on convolutional neural network," in *2019 3rd international conference on trends in electronics and informatics (ICOEI)*. IEEE, 2019, pp. 331–336.

[159] M. Martineau, P. Atkinson, and S. McIntosh-Smith, "Benchmarking the nvidia v100 gpu and tensor cores," in *European Conference on Parallel Processing*. Springer, 2018, pp. 444–455.

[160] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*. IEEE, 2020, pp. 237–242.

[161] J. Lever, M. Krzywinski, and N. Altman, "Points of Significance: Model selection and overfitting," *Nature methods*, vol. 13, no. 9, pp. 703–705, 2016.

[162] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[163] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning (ICML), deep learning workshop*, vol. 2, no. 1. Lille, 2015.

[164] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A Unified Embedding for Face Recognition and Clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[165] X. Dong and J. Shen, "Triplet Loss in Siamese Network for Object Tracking," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 459–474.

[166] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3708–3712.

[167] Ç. F. Özgenel and A. G. Sorguç, "Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings," in *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, vol. 35. Taipei, Taiwan: International Association for Automation and Robotics in Construction (IAARC), July 2018, pp. 693–700.

[168] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[169] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *Computing Research Repository (CoRR)*, vol. abs/1803.08375, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375

[170] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv preprint arXiv:1412.6980*, 2017.

[171] D. Chicco, "Siamese Neural Networks: An Overview," in *Artificial Neural Networks*, ser. Methods in Molecular Biology, H. Cartwright, Ed. New York, NY: Humana, 2021, vol. 2190.

[172] P. Daponte, L. De Vito, A. Iannuzzo, M. Monaco, A. Neyestani, and F. Picariello, "Low-cost marked tracking monitoring system for 3d-scaled masonry models," in *2024 IEEE International Workshop on Metrology for Living Environment (MetroLivEnv)*, 2024, pp. 172–177.

[173] P. Daponte, L. De Vito, I. Tudosa, A. Neyestani, and F. Picariello, "Development and evaluation of a novel marker-based tracking system for 3d-scaled masonry models using deeptag," in *2024 XXXIII International Scientific Conference Electronics (ET)*. IEEE, 2024, pp. 1–5.

[174] F. Yavartanoo and T. H.-K. Kang, "Retrofitting of unreinforced masonry structures and considerations for heritage-sensitive constructions," *Journal of Building Engineering*, vol. 49, p. 103993, 2022.

[175] G. Milani, P. Lourenço, and A. Tralli, "3d homogenized limit analysis of masonry buildings under horizontal loads," *Engineering Structures*, vol. 29, no. 11, pp. 3134–3148, 2007.

[176] A. Montanino, D. De Gregorio, C. Olivieri, and A. Iannuzzo, "The continuous airy-based for stress-singularities (cass) method: an energy-based numerical formulation for unilateral materials," *Int. Journal of Solids and Structures*, vol. 256, p. 111954, 2022.

[177] A. Bayraktar, E. Hökelekli, and T. T. Yang, "Seismic failure behavior of masonry domes under strong ground motions," *Engineering Failure Analysis*, vol. 142, p. 106749, 2022.

[178] S. Ullah Khan, A. Naseer, M. Fahim, M. Ashraf, and E. Badshah, "Experimental seismic performance evaluation of brick masonry cavity-wall buildings," *Structures*, vol. 41, pp. 1781–1791, 2022.

[179] A. Iannuzzo, M. Angelillo, E. De Chiara, F. De Guglielmo, F. De Serio, F. Ribera, and A. Gesualdo, "Modelling the cracks produced by settlements in masonry structures," *Meccanica*, vol. 53, pp. 1857–1873, 2018.

[180] A. Tralli, A. Chiozzi, N. Grillanda, and G. Milani, "Masonry structures in the presence of foundation settlements and unilateral contact problems," *International Journal of Solids and Structures*, vol. 191, pp. 187–201, 2020.

[181] A. Iannuzzo, A. Dell'Endice, T. Van Mele, and P. Block, "Numerical limit analysis-based modelling of masonry structures subjected to large displacements," *Computers & Structures*, vol. 242, p. 106372, 2021.

[182] R. Gagliardo, F. Portioli, L. Cascini, R. Landolfo, and P. Lourenço, "A rigid block model with no-tension elastic contacts for displacement-based assessment of historic masonry structures subjected to settlements," *Engineering Structures*, vol. 229, p. 111609, 2021.

[183] C. Ferrero, C. Calderini, F. Portioli, and P. Roca, "Large displacement analysis of dry-joint masonry arches subject to inclined support movements," *Engineering Structures*, vol. 238, p. 112244, 2021.

[184] S. Galassi and P. Zampieri, "A new automatic procedure for nonlinear analysis of masonry arches subjected to large support movements," *Engineering Structures*, vol. 276, p. 115359, 2023.

[185] B. Mobaraki, F. Lozano-Galant, R. P. Soriano, and F. J. Castilla Pascual, "Application of low-cost sensors for building monitoring: A systematic literature review," *Buildings*, vol. 11, no. 8, 2021. [Online]. Available: https://www.mdpi.com/2075-5309/11/8/336

[186] Z. Zhang, Y. Hu, G. Yu, and J. Dai, "Deeptag: A general framework for fiducial marker design and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2931–2944, 2023.

[187] A. Romano and J. A. Ochsendorf, "The mechanics of gothic masonry arches," *International Journal of Architectural Heritage*, vol. 4, no. 1, pp. 59–82, 2010.

[188] T. U. Siaw, Y. C. Han, and K. I. Wong, "A low-cost marker-based optical motion capture system to validate inertial measurement units," *IEEE Sensors Letters*, vol. 7, no. 2, pp. 1–4, 2023.

[189] T. Jiang, H. Cui, X. Cheng, P. Li, and W. Tian, "A ball-shaped target development and pose estimation strategy for a tracking-based scanning system," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2022.

[190] D. C. Y. Wong, J. Song, and H. Yu, "The design of a vision-based bending sensor for pneunet actuators leveraging aruco marker detection," *IEEE Sensors Journal*, 2023.

[191] H. Sarmadi, R. Muñoz-Salinas, M. A. Olivares-Mendez, and R. Medina-Carnicer, "Detection of binary square fiducial markers using an event camera," *IEEE Access*, vol. 9, pp. 27 813–27 826, 2021.

[192] Y. Zhu, Y. Huang, Y. Li, Z. Qiu, and Z. Zhao, "A smartphone-based six-dof measurement method with marker detector," *IEEE Trans. on Instrumentation and Measurement*, vol. 71, pp. 1–14, 2022.

[193] S. Park, H. Park, J. Kim, and H. Adeli, "3d displacement measurement model for health monitoring of structures using a motion capture system," *Measurement*, vol. 59, pp. 352–362, 2015.

[194] J. E. van Schaik and N. Dominici, "Motion tracking in developmental research: Methods, considerations, and applications," *Progress in Brain Research*, vol. 254, pp. 89–111, 2020.

[195] M. Kalaitzakis, B. Cain, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios, "Fiducial markers for pose estimation: Overview, applications and experimental comparison of the artag, apriltag, aruco and stag markers," *Journal of Intelligent & Robotic Systems*, vol. 101, pp. 1–26, 2021.

[196] "6-DoF motorized system by Standa LTD," 2024, https://www.standa.lt/ [Accessed: 24/04/2024].

[197] "Google Pixel 7 pro," 2024, https://www.gsmarena.com/google_pixel_7-11903.php [Accessed: 24/04/2024].

[198] Y. Luo, X. Wang, Y. Liao, Q. Fu, C. Shu, Y. Wu, and Y. He, "A review of homography estimation: Advances and challenges," *Electronics*, vol. 12, no. 24, p. 4977, 2023.

[199] M. A. Jaimes, M. M. Chávez, F. Peña, and A. D. García-Soto, "Out-of-plane mechanism in the seismic risk of masonry façades," *Bulletin of Earthquake Engineering*, vol. 19, pp. 1509–1535, 2021.

[200] A. Neyestani, D. M. Toma, A. Falahzadeh, P. Daponte, J. Del Rio Fernandez, and L. De Vito, "A significant wave height data-driven modeling for digital twins of marine environment," in *2024 IEEE International Workshop on Metrology for the Sea; Learning to Measure Sea Health Parameters (MetroSea)*, 2024, pp. 495–500.

[201] M. Segovia and J. Garcia-Alfaro, "Design, Modeling and Implementation of Digital Twins," *Sensors*, vol. 22, no. 14, p. 5396, 2022.

[202] K. Tzilivakis, "A tale of Two oceans: Scientists are building digital twins of the ocean," *Horizon, the EU Research and Innovation Magazine*, February 11 2022. [Online]. Available: https://ec.europa.eu/research-and-innovation/en/horizon-magazine/tale-two-oceans-scientists-are-building-digital-twins-ocean

[203] L. D. Vito, V. Cocca, M. Riccio, and I. Tudosa, "Wireless active guardrail system for environmental measurements," in *2012 IEEE Workshop on Environmental Energy and Structural Monitoring Systems (EESMS)*, Perugia, Italy, 2012, pp. 50–57.

[204] A. Barbie, N. Pech, W. Hasselbring, S. Flögel, F. Wenzhöfer, M. Walter, E. Shchekinova, M. Busse, M. Türk, M. Hofbauer *et al.*, "Developing an Underwater Network of Ocean Observation Systems With Digital Twin Prototypes—A Field Report From the Baltic Sea," *IEEE Internet Computing*, vol. 26, no. 3, pp. 33–42, 2021.

[205] S. Boschert and R. Rosen, "Digital twin—the simulation aspect," *Mechatronic futures: Challenges and solutions for mechatronic systems and their designers*, pp. 59–74, 2016.

[206] A. Tzachor, O. Hendel, and C. E. Richards, "Digital twins: a stepping stone to achieve ocean sustainability?" *npj Ocean Sustainability*, vol. 2, no. 1, p. 16, 2023.

[207] J. Schneider, A. Klüner, and O. Zielinski, "Towards Digital Twins of the Oceans: The Potential of Machine Learning for Monitoring the Impacts of Offshore Wind Farms on Marine Environments," *Sensors*, vol. 23, no. 10, p. 4581, 2023.

[208] G. Llorach-Tó, E. Martínez, J. Del-Río, G. Simarro, M. Pani, A. Bucchi, Y. Huang, and E. García-Ladona, "3D Digital Twins of the Ocean: towards an intuitive and realistic visualization of wave parameters," Copernicus Meetings, Tech. Rep., 2024.

[209] G. Llorach-Tó, E. Martínez, J. D. R. Fernández, and E. García-Ladona, "Experience OBSEA: a web-based 3D virtual environment of a seafloor observatory," in *OCEANS 2023 - Limerick*, 2023, pp. 1–6.

[210] U. Bronner, M. Sonnewald, and M. Visbeck, "Digital Twins of the Ocean can Foster a sustainable blue economy in a protected marine environment," *International Hydrography Review*, 2023.

[211] OBSEA 3D simulation, "Obsea," https://cgi-dto.github.io/OBSEA/.

[212] D. Kim, G. Antariksa, M. P. Handayani, S. Lee, and J. Lee, "Explainable Anomaly Detection Framework for Maritime Main Engine Sensor Data," *Sensors*, vol. 21, no. 15, p. 5200, 2021.

[213] T. Stach, Y. Kinkel, M. Constapel, and H.-C. Burmeister, "Maritime Anomaly Detection for Vessel Traffic Services: A Survey," *Journal of Marine Science and Engineering*, vol. 11, no. 6, p. 1174, 2023.

[214] R. M. Schmidt, "Recurrent Neural Networks (RNNs): A gentle Introduction and Overview ," *arXiv preprint arXiv:1912.05911*, 2019.

[215] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng, "Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model," *Journal of Petroleum Science and Engineering*, vol. 186, p. 106682, 2020.

[216] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[217] A. Hasan, I. Kayes, M. Alam, T. Shahriar, and M. A. Habib, "Generalized machine learning models to predict significant wave height utilizing wind and atmospheric parameters," *Energy Conversion and Management: X*, vol. 23, p. 100623, 2024.

[218] H. Hu, A. J. van der Westhuysen, P. Chu, and A. Fujisaki-Manome, "Predicting Lake Erie wave heights and periods using XGBoost and LSTM," *Ocean Modelling*, vol. 164, p. 101832, 2021.

[219] F. C. Minuzzi and L. Farina, "A deep learning approach to predict significant wave height using long short-term memory," *Ocean Modelling*, vol. 181, p. 102151, 2023.

[220] A. J. Thanthawy Sukanda and D. Adytia, "Wave Forecast using Bidirectional GRU and GRU Method Case Study in Pangandaran, Indonesia," in *2022 International Conference on Data Science and Its Applications (ICoDSA)*, 2022, pp. 278–282.

[221] M. J. Alizadeh and V. Nourani, "Multivariate GRU and LSTM models for wave forecasting and hindcasting in the southern Caspian Sea," *Ocean Engineering*, vol. 298, p. 117193, 2024.

[222] S. Kim, M. Takeda, and H. Mase, "GMDH-based wave prediction model for one-week nearshore waves using one-week forecasted global wave data," *Applied Ocean Research*, vol. 117, p. 102859, 2021.

[223] EMODnet Physics, "Tarragona coast buoy," accessed: 2024-08-05. [Online]. Available: https://map.emodnet-physics.eu/platformpage/?platformcode=Tarragona-coast-buoy&source=cp&integrator=INSTAC

[224] ——, "Barcelona coast buoy," https://map.emodnet-physics.eu/platformpage/?platformcode=Barcelona-coast-buoy&source=cp&integrator=INSTAC. [Online]. Available: "https://map.emodnet-physics.eu/platformpage/?platformcode=Barcelona-coast-buoy&source=cp&integrator=INSTAC"

[225] OBSEA, "Obsea official website," https://obsea.es/.

[226] EMSO ERDDAP, "Obsea awac waves full," https://erddap.emso.eu/erddap/tabledap/EMSO_OBSEA_AWAC_waves_full.html. [Online]. Avail-

able: https://erddap.emso.eu/erddap/tabledap/EMSO_OBSEA_AWAC_waves_full.html

[227] Nortek Group, "Awac 2 mhz," https://www.nortekgroup.com/products/awac2-1-mhz.

[228] G. Velarde, P. Brañez, A. Bueno, R. Heredia, and M. Lopez-Ledezma, "An Open Source and Reproducible Implementation of LSTM and GRU Networks for Time Series Forecasting," *Engineering Proceedings*, vol. 18, no. 1, p. 30, 2022.

[229] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[230] S. Shin, D. Shin, and N. Kang, "Topology optimization via machine learning and deep learning: A review," *Journal of Computational Design and Engineering*, vol. 10, no. 4, pp. 1736–1766, 2023.

[231] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[232] T. Chai and R. R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature," *Geoscientific model development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[233] P. Schober, C. Boer, and L. A. Schwarte, "Correlation Coefficients: Appropriate Use and Interpretation," *Anesthesia & analgesia*, vol. 126, no. 5, pp. 1763–1768, 2018.

[234] T. Song, R. Han, F. Meng, J. Wang, W. Wei, and S. Peng, "A significant wave height prediction method based on deep learning combining the correlation between wind and wind waves," *Frontiers in Marine Science*, vol. 9, p. 983007, 2022.

[235] A. Ali, A. Fathalla, A. Salah, M. Bekhit, and E. Eldesouky, "Marine Data Prediction: An Evaluation of Machine Learning, Deep Learning, and Statistical Predictive Models," *Computational Intelligence and Neuroscience*, vol. 2021, no. 1, p. 8551167, 2021.

[236] S. Hu, Q. Shao, W. Li, G. Han, Q. Zheng, R. Wang, and H. Liu, "Multivariate Sea Surface Prediction in the Bohai Sea Using a Data-Driven Model," *Journal of Marine Science and Engineering*, vol. 11, no. 11, p. 2096, 2023.

[237] P. Xiong, G. Bian, Q. Liu, S. Jin, and X. Yin, "A Prediction Model of Marine Geomagnetic Diurnal Variation Using Machine Learning," *Applied Sciences*, vol. 14, no. 11, p. 4369, 2024.

[238] S. Ray, D. S. McEvoy, S. Aaron, T.-T. Hickman, and A. Wright, "Using statistical anomaly detection models to find clinical decision support malfunctions," *Journal of the American Medical Informatics Association*, vol. 25, no. 7, pp. 862–871, 2018.

[239] D. Samariya and A. Thakkar, "A comprehensive survey of anomaly detection algorithms," *Annals of Data Science*, vol. 10, no. 3, pp. 829–850, 2023.

[240] S. Thudumu, P. Branch, J. Jin, and J. Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *Journal of Big Data*, vol. 7, pp. 1–30, 2020.