UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DEPARTMENT OF INFORMATION ENGINEERING

BACHELOR'S DEGREE IN COMPUTER ENGINEERING

Fusing vision and inertial measurements for autonomous navigation in narrow channels

Supervisor:
Prof. Damiano Varagnolo

Candidate:
Thomas Sanavia

ACADEMIC YEAR: 2024/2025
Graduation date: – – —-

A ...

Quote
Author

Contents

Abstract

This thesis addresses the design, development, and validation of an automatic system for detecting roll, pitch, and yaw of an autonomous boat. The main objective is to create a low-cost, low-power, and easily transportable solution capable of accurately determining the boat's attitude. To achieve this, both computer vision and an onboard inertial sensor will be used. By fusing data from these two sources, a precise estimate of the boat's attitude will be made available to the entire system. For the computer vision part, two cameras will be used to exploit stereo vision. The attitude estimation will be processed on a Raspberry Pi 5, chosen for its low cost. For this purpose, ArUco Tags—two-dimensional fiducial markers commonly used in robotics and augmented reality for pose estimation—will be employed. Regarding the inertial sensor, the onboard device will provide velocities and accelerations along the three axes. Finally, an extended Kalman filter, which will utilize both inertial and computer vision data, will be applied to reduce drift and improve the accuracy of roll, pitch, and yaw estimation.

Introduction

In recent years, the field of **autonomous systems** has played an increasingly central role in robotics. All this has also been made possible thanks to solutions such as **Robot Operating System 2 (ROS2)**, which is one of the platforms for the development of robotic applications. This has been achieved thanks to its modular architecture, support for real-time communication, and its open-source nature, which has made it an excellent choice even for advanced robotic systems.

This thesis is part of the **Autodocking project**, focused on the ability of a boat to navigate and dock autonomously. Although the structure and the basic control systems were already in place, a computer vision–based system for autonomous navigation was missing. My contribution was related to the creation of a ROS2 node; more specifically, I dealt with the entire pipeline ranging from image acquisition, attitude estimation, and subsequent sensor fusion with the inertial sensor data. To achieve this, I needed a ROS2 node that handled the acquisition of images from the various cameras installed on the boat and their publication on a topic, so that my algorithm could access them.

Chapter 1

Introduction

1.1 Autonomous Surface Vehicles: challenges and applications

Unmanned Surface Vehicles (**USVs**) are particular surface vessels capable of navigating without the presence of a human crew to maneuver them. They can be remotely operated through radio or satellite communication, or be completely autonomous and controlled by a computer or artificial intelligence; in this case, they are referred to as **Autonomous Surface Vehicles (ASVs)**. ASVs aim to make journeys safer, reduce costs, increase operational continuity, and carry out prolonged missions in complex and even risky situations for humans.

All this has been made possible thanks to significant technological evolution in **sensors** (satellite, IMU, cameras, LIDAR, radar, etc.), major developments in robotics libraries, and onboard computational power that is

increasingly greater and more affordable. Despite this, the marine environment presents major challenges. For example, the plane of the sea surface is not a fixed reference but changes independently of the boat's movements; the surface creates reflections that can interfere with instrumentation; and wind and waves can put stability control systems to the test. In the marine context, therefore, attitude estimation (roll, pitch, and yaw) is essential, as it is decisive both for the success of the journey and for the precision required in maneuvers, such as docking.

The main challenges concern **sensors and algorithms**. The first challenge involves navigation with GNSS (Global Navigation Satellite System), which degrades in proximity to port infrastructures, bridges, or areas with many vertical obstacles, as part of the visible satellites are lost. This problem has been partly mitigated with inertial sensors (IMU), which, however, suffer from another issue: drift. The second challenge is the detection of possible obstacles at sea, such as buoys, debris, or even other vessels, often under conditions of reduced visibility or reflections. The third challenge is control and stability, made complex by wave motion that often causes very rapid variations. The last challenge is the ability to make real-time decisions, including very quick changes dictated by sudden shifts in the operational situation.

Despite the criticality of these challenges, ASVs are used in **various applications**. In the industrial field, they are employed for infrastructure inspection or short-range logistical support. In defense and security, they are used in special operations such as explosive ordnance disposal. In the environmental field, they are used for monitoring environmental parameters and collecting meteorological data over extended time horizons, thereby reducing

risks for crews.

Within this framework, the present thesis focuses on one main issue: obtaining an **accurate real-time estimation** of the boat's attitude. To achieve this, computer vision techniques (stereo vision and ArUco markers), inertial measurements through the IMU, and an Extended Kalman Filter will be used to fuse the data, with the aim of improving the reliability of the estimation.

1.2 Attitude Estimation: roll, pitch, yaw

To describe the **attitude** of a vessel, the orientation of the rigid body with respect to a reference system is used. This is done through the three **Euler angles**[1]: roll (rotation on the longitudinal axis), pitch (rotation on the transverse axis), and yaw (rotation on the vertical axis). These three quantities are crucial as they influence stabilization and trajectory control and are fundamental in precision maneuvers.

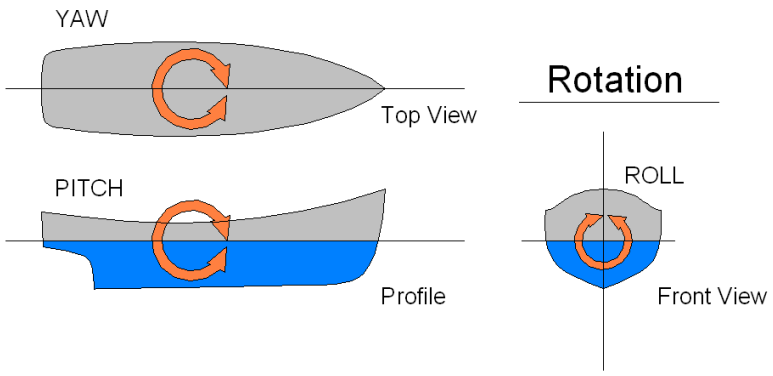

Figure 1.1: Three rotational degrees of freedom of a ASV

Attitude estimation can be obtained from different sources, each with its own advantages and disadvantages. Inertial sensors (IMU)[2], through gyroscopes and accelerometers, provide roll and pitch with a high update frequency. However, these estimates are affected by drift, caused by the accumulation of errors over time. For yaw, a magnetometer or an external observation is required, as the IMU does not provide a direct measurement of this angle. A visual sensor allows estimating the attitude through known structures (e.g., ArUco tags) or through the environment if it allows it (e.g., using the horizon line). This latter method is less affected by the drift problem but is highly sensitive to environmental conditions and also has a lower update frequency.

In the marine context, a robust pipeline is represented by the IMU, which provides a high-frequency prediction and, periodically, through computer vision, a measurement relative to the ArUco marker to reduce error accumulation. Through an EKF filter[3], the different contributions are integrated in order to obtain a coherent estimate even if the conditions are adverse or if one of the sources fails.

In summary, attitude estimation for an **ASV** requires a compromise between the speed of the readings and their accuracy. The integration between inertial and visual measurements provides a reliable estimate capable of supporting autonomous navigation. Subsequently, the different implementation choices to obtain the attitude estimation of the **Blue Boat** will be presented.

1.3 Translational Motions of the Boat Hull

Heave, **Sway**, and **Surge** are the three translational motions related to the hull of the boat. They describe the vertical, lateral, and longitudinal displacements, corresponding respectively to heave, sway, and surge in nautical terminology.

In a realistic scenario, they interact with roll, pitch, and yaw, affecting stability, control, and the quality of on-board attitude estimates. The precise definition of reference frames and axis conventions is necessary to correctly interpret the measurements, allowing the distinction between slow fluctuations, due to currents and average conditions, and rapid ones caused by waves and maneuvers.

In this project, these motions are mentioned only for terminological completeness, but they will neither be estimated nor computed. The objective is solely the calculation of angular attitude through visual and inertial observations.

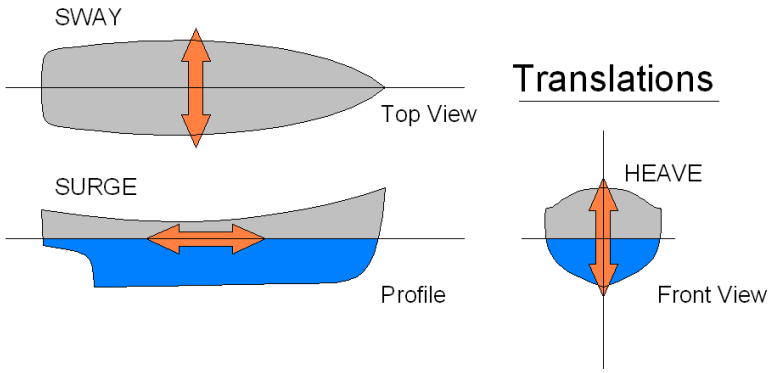

Figure 1.2: The three translations to which a floating body is subjected: surge (longitudinal), sway (lateral), and heave (vertical)

1.4 Computer Vision for Localization and Attitude Estimation

Computer vision is an essential source for estimating the attitude of the ASV. It provides roll, pitch, and yaw with respect to a known reference, namely the *ArUco marker*. In our case, it plays a complementary role with respect to the IMU, which provides an absolute measurement that is not obtained through temporal integration, but is instead sensitive to visual conditions (reflections, backlighting, fog, etc.). The goal of this chapter is to outline the principles and choices that allow for reliable attitude estimation using a stereo camera pair, leveraging *ArUco markers* and a geometric calibration of the two cameras.

The pipeline is structured into four main steps. First, a stereo camera calibration is performed to estimate the intrinsic parameters and distortions, followed by the calculation of rotation and translation between the two cameras. Then, marker detection is carried out through ArUco markers, which enable highly reliable identification provided they are properly scaled according to the distance and have a sufficient pixel resolution. Subsequently, attitude estimation is obtained by applying a pose estimation algorithm which, knowing the 2D corners and the 3D points of the pattern, can estimate the values of roll, pitch, and yaw. Finally, a measurement validation step is applied, where a decision filter determines whether to accept the measurement, discard it, or reduce its influence.

Triangulation, made more reliable by epipolar rectification, allows for the estimation of depth around the marker. These depth cues are used as

additional support to improve the attitude estimation.

Finally, temporal and synchronization aspects also play a crucial role. Synchronization is decisive to avoid disparities and inconsistent triangulations. The frame rate and the subsequent update time strongly influence the accuracy of the algorithm's estimation, since excessively low values reduce its quality.

This approach, entirely based on stereo vision and ArUco markers, enables precise attitude estimation. Its accuracy is determined by the quality of calibrations and the robustness of estimation algorithms, but it reaches its maximum effectiveness when integrated with IMU data, compensating for the weaknesses of each source.

1.5 Fiducial markers and ArUco tags

The ArUco marker is a square planar pattern with a high-contrast black border and an internal matrix encoding a unique identifier. This structure allows for fast localization and recognition of the four corners, providing a stable 2D–3D correspondence for attitude estimation. The use of a single marker reduces environmental and economic requirements. However, relying on just one marker requires greater care in localization, calibration, and validation of the measurements.

The recognition pipeline adopts a pair of calibrated cameras. After calibration of the individual cameras and subsequent stereo calibration (*rotation*, *translation*, and *epipolar rectification*), the images are acquired and rectified, thus aligning the epipolar lines. The detection of the marker involves several

steps: **adaptive thresholding**, contour extraction, quadrilateral selection, and ID decoding.

Knowing the marker size, the attitude is calculated through the **estimateAffine3D method**, obtaining the rotation and translation of the reference camera frame. If both cameras observe the marker, the two estimates can be fused, or the one with higher quality can be selected.

The use of a system based on **stereo vision** improves disparity estimation in the regions close to the marker and enables metric triangulation of depth, enhancing stability compared to a monocular approach. This becomes particularly evident when the marker is very small (i.e., far away) or when it is not perpendicular to the viewing angle, as both conditions reduce perspective information.

For each estimate, quality indicators are evaluated: reprojection error, effective presence of the marker (all four corners detected), and the angle between the camera and the marker, penalizing views that are too oblique. Furthermore, a comparison is made between the apparent size of the marker in pixels, in order to identify discrepancies due to triangulation or temporal misalignment. From a practical point of view, the use of a single ArUco marker[4] offers a simple and easily reproducible implementation path: a single print on a rigid surface with a matte finish (to reduce reflections) is sufficient. Under these conditions, a single ArUco marker used in a well-calibrated stereo vision pipeline provides roll, pitch, and yaw measurements while maintaining a good trade-off between precision and computational demand.

1.6 Sensor Fusion and Bayesian Approaches

Sensor fusion deals with solving the problem of combining heterogeneous and noisy measurements in order to obtain a state estimate that is more accurate than the one provided by individual sensors. In our case, we merge sensors with complementary properties. Stereo vision provides absolute measurements without drift but at a low frequency, while IMU provides angular velocity measurements at high frequency but suffers from drift.

In the Bayesian paradigm, the state is described by a probability distribution that evolves over time and is composed of two steps. The prediction step computes the next state based on the dynamic model and the noise statistics, while the update step uses the available measurement through the observation function and the measurement noise statistics. This alternation achieves an optimal trade-off between what is computed and what is effectively measured.

In our case, the state consists of **Euler angles** (roll, pitch, yaw), the dynamics are modeled by integrating angular velocities to predict the evolution of the angles using consecutive samples, and the observations come from orientation measurements derived from vision and inertial sensors, which are treated either as model input or as an observation depending on the architecture. It is also necessary to guarantee **numerical consistency** between prediction and update, managing, for example, the correct periodicity of measurements and possible discontinuities among angles.

Fusion provides several benefits. It reduces drift because vision-based observations remove the accumulation of error caused by inertial integration.

The temporal continuity of the IMU keeps the estimate stable due to its high update frequency.

Fusion also improves robustness, as weighting measurements based on their uncertainty allows discarding unreliable measurements from stereo vision (e.g., when the camera is at a steep angle relative to the ArUco marker or when the marker is too small) or from the IMU.

The quality of the fusion evaluation is assessed based on three main aspects: **angular accuracy**, **temporal stability**, and **behavior under varying visual quality**.

In summary, **sensor fusion with a Bayesian algorithm** allows combining stereo vision and IMU measurements to obtain the attitude estimate. The IMU provides a continuous and fast estimate, while stereo vision provides an estimate that does not suffer from drift since it is anchored to fixed references. Weighting the measurements according to their uncertainty allows operation even when the quality of observations is not uniform.

1.7 Kalman Filters as Bayesian Filters (Linear and Extended)

Kalman filters are part of the family of **Bayesian filters**; they recursively estimate the state of the system in a probabilistic way and update it through a dynamic model (prediction) with new observations (update) using Bayes' theorem. In linear cases with Gaussian noise of zero mean, the **Kalman Filter (KF)** provides an **optimal solution** since it maintains the state

described by mean and covariance, producing a Kalman gain that balances the model and the measurements. In nonlinear systems, typical in robotics and computer vision, the **Extended Kalman Filter (EKF)** is used, which applies a local linearization (Jacobian matrices) of the models to preserve the prediction structure.

In our case, to estimate the attitude, the minimum state includes the **Euler angles (roll, pitch, yaw)** and the **angular velocities**. The prediction integrates angular velocities over the sampling step, while the update incorporates an orientation observation coming from computer vision, with a covariance built from the quality indicators of the stereo vision pipeline (reprojection error, number and quality of corners, etc.). In this case, the filter avoids drift caused by the inertial system if the computer vision part is stable.

At the operational level, the linear KF, through known transition and observation matrices, calculates prediction and update. If the state is nonlinear, the **EKF** is used, which, through the calculation of Jacobian matrices, replaces constant matrices within the same recursive scheme. The success of the fusion depends on the consistency of the measurements, which in turn depends on temporal synchronization between computer vision and IMU measurements, as well as good performance conditions that allow everything to run in real time.

In summary, the **Kalman Filter** and its extended version, through a Bayesian approach, perform sensor fusion in a computationally efficient way: the prediction exploits the dynamic model and the IMU angular velocities to ensure continuity and responsiveness in any situation, while the update

also integrates computer vision, thus providing an accurate and temporally stable attitude estimate.

Chapter 2

Attitude Estimation with Computer Vision

2.1 Vision Hardware and Experimental Setup

Component	Model/Version	Role	Cost
Raspberry Pi 5	8GB RAM	Frame acquisition and attitude computation	85€
Right Pi Camera	V3	Capture right frame	27€
Left Pi Camera	V3	Capture left frame	27€
Heatsink	Active	CPU cooling	11€
SanDisk Extreme	64GB	OS storage	15€
Total			**165€**

Table 2.1: Components used for attitude estimation

The computer vision part consists of a pair of rigidly mounted and calibrated cameras. The baseline is 12 cm, also measured during calibration. The cameras operate at a resolution of 1280×720 px with a frame rate of 30

fps, providing an excellent compromise between visual quality and required resources.

The calibration procedure is performed live by framing the calibration checkerboard. Calibrations are carried out for each camera (focal length, principal point, and distortion) and subsequently for the stereo pair (rotation and translation). At the end of the calibration, the files are saved and later loaded by the attitude estimation program.

The visual reference for estimation is a single **ArUco marker** (in our case, a 6×6 dictionary) with a known size. The marker is printed on a rigid and opaque support to reduce potential problems caused by reflections. The marker size must be such that it remains visible even at a distance, since the acquisition resolution is limited.

Each cycle for the estimation calculation begins with the rectification of the cameras using the previously acquired calibration files. Subsequently, if the marker is detected, the algorithm proceeds to the next step. If present, the four vertices are triangulated to reconstruct the 3D coordinates. The reconstructed points are compared with the theoretical model of the marker, thus allowing the estimation of the **complete pose** (rotation and translation) using the *estimateAffine3D* method. The rotation is converted to quaternions to avoid 180° jumps in the measurements. Finally, the video stream is displayed with the real-time attitude estimation, showing roll, pitch, and yaw on the screen.

The measurement quality is ensured by checking, in each frame, that all marker corners are visible for correct identification. In this case, the combination of stereo vision, ArUco marker, and pose estimation using the

estimateAffine3D method provides stable and reliable attitude estimates.

2.2 Stereo Camera Calibration

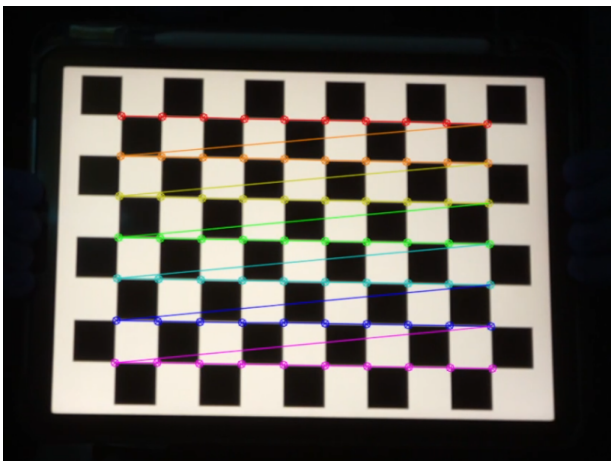

Figure 2.1: Checkerboard pattern with detected feature points during camera calibration.

The calibration of the **stereo cameras** was implemented through an algorithm that performs a *live acquisition* of a checkerboard of known size, used as a reference. The two cameras are rigidly mounted on a **bracket** to avoid movements and acquire the images of the calibration pattern. For each acquired frame, the inner corners of the checkerboard are detected in both frames, subsequently the coordinates are made more accurate through **sub-pixel precision**. This procedure is repeated for each acquisition frame (in our case 50 frames), in order to collect a sufficiently high number of samples to make our calibration more *stable*.

Once the acquisition of the frames is completed, the calibration of the **individual cameras** is performed. For each camera, the fundamental opti-

cal parameters are calculated, represented by: focal length, principal point and lens distortion. These parameters are essential to correct the radial and tangential distortions created by the optics, thus ensuring a faithful representation of **reality**.

The next part is **stereo calibration**, which allows estimating the geometric relationship between the two cameras. Thanks to the common points of the pattern, the algorithm calculates the rotation matrix and the translation vector of the right camera with respect to the left one. To ensure consistency with the physical setup, the translation vector is scaled so that the **baseline** is exactly 12 cm, that is the real distance between the two cameras.

Finally, the matrices necessary to realign the images and ensure the subsequent triangulation of the points in space are calculated. All the parameters are saved in different files, so that they can be used by the program that estimates the *attitude*.

2.3 Detection of ArUco markers

Detection of ArUco markers is fundamental for our **computer vision pipeline**, as it represents a reliable visual reference to be used in the subsequent phases of attitude estimation. ArUco markers are black squares containing a unique binary pattern, *easily distinguishable* even under difficult lighting conditions. They are designed to minimize ambiguities that may occur between different markers, ensuring stable and fast recognition.
Each frame acquired by the cameras is processed by the function

`detectMarkers`, which is responsible for detecting the markers present within the frames. The algorithm uses a 6x6 marker dictionary. Once detected, both the coordinates of the vertices and the marker ID are returned; all this information will be used in the subsequent parts of the algorithm.

To improve the accuracy of the algorithm, the OpenCV library provides the `DetectorParameters` structure, which allows adjusting some detection parameters. In our case, we exclusively use `CORNER_REFINE_SUBPIX`, which allows achieving **sub-pixel precision** in the detection of the vertices. This is particularly useful when the marker is observed at long distances.

An important aspect concerns candidate markers that are subsequently discarded. During frame analysis, there may be regions that resemble the searched marker but do not contain a valid code. These are discarded but can still be visualized and analyzed through the function `refineDetectedMarkers`, which is especially useful when aiming to develop a particularly stable and robust system.

In conclusion, the detection process provides a **fast and reliable response** in identifying the markers present in the scene, returning the necessary information such as their ID and vertex positions. These data, once validated, are the essential starting point for the subsequent phases of triangulation and attitude estimation, thus ensuring *consistency and reliability* throughout the process.

2.4 Pose Estimation Algorithms

The estimation of the boat's attitude represents the **most important phase** of the entire pipeline, as it precisely determines the position of the boat with respect to the marker in three-dimensional space, in reference to the stereo camera system.

The process begins with the detection of the marker in the two rectified frames. In both frames, the markers are identified and subsequently the vertices corresponding to the corners of the square are extracted. These points, once extracted, are used to perform triangulation through the *triangulatePoints* function, which reconstructs the 3D coordinates of the vertices in a common space. The result is a set of 3D points in space that describe the geometry of the marker. In parallel, an **ideal marker model** is defined in the reference system. This model has known dimensions, specified in the program (in our case 12 cm), whose vertices are described in the plane $z = 0$. The goal is to find the rigid transformation that maps the ideal model to coincide with the 3D points previously reconstructed.

To obtain the transformation, we use the *estimateAffine3D* function, which returns an affine matrix A, consisting of the rotation matrix R and the translation vector T. The rotation describes the orientation of the marker, while the translation its position with respect to the stereo system.

The rotation matrix is then converted into quaternions using the **SciPy** library, in order to eliminate the ambiguities of the Euler angles. The quaternions are subsequently transformed into Euler angles (*roll*, *pitch*, and *yaw*) for a quicker interpretation.

Finally, to guarantee the **temporal continuity** of the estimation and avoid sign changes, a consistency check between the current pose and the pose of the previous frame has been implemented.

The implemented algorithm, as described, makes it possible to obtain a **robust and fast estimation** with respect to a marker, thus providing a basis for the kinematic models of the system.

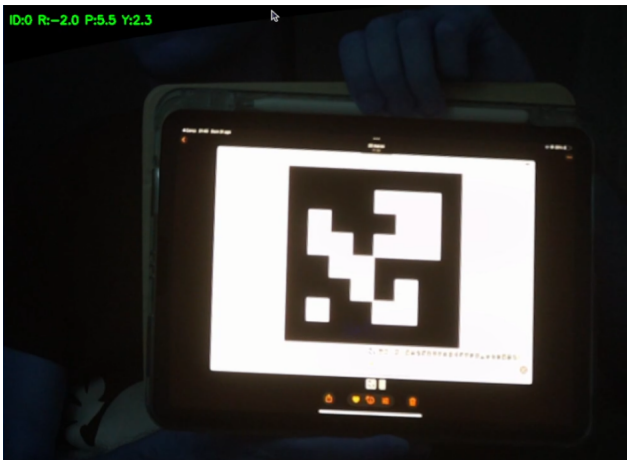

Figure 2.2: Detection of an ArUco marker with estimated roll, pitch, and yaw values displayed.

2.5 Limitations and Critical Issues of the Visual Method

The stereo camera approach to the problem of attitude estimation, using **ArUco markers**, offers several advantages in terms of *simplicity*, cost, and ease of implementation. Nevertheless, there are several **limitations and challenges** that must be taken into account to ensure reliable use in a marine scenario.

The first aspect to consider concerns the dependence on **lighting conditions**. The accuracy with which the marker is detected may deteriorate in the presence of strong light, pronounced shadows, or reflections. Although ArUco markers are designed to provide stable recognition, certain conditions may lead to *false negatives* or errors in the detection of corners.

The second aspect to be considered is the **distance and angle of observation**. If the marker is observed from a great distance or with a considerable tilt, the edges appear less defined and the detection function may return points that are not very accurate. This results in a deterioration of triangulation and the subsequent pose estimation, especially regarding the accuracy of quaternions and thus the calculation of Euler angles.

Another critical aspect concerns **stereo triangulation**. For the algorithm to work, it relies on the correspondence of points in the two rectified frames. If the frames are misaligned due to calibration issues or uncompensated distortion, the three-dimensional reconstruction may be inaccurate, negatively affecting the transformation estimated by the function `estimateAffine3D`.

The issue of **false positives** must also be considered. In certain conditions, visual patterns may resemble a marker and therefore be mistakenly identified as such. Although functions like `refineDetectedMarkers` help to mitigate the problem, in complex systems this risk cannot be entirely eliminated.

The last critical aspect is **temporal continuity**. Although controls are in place to avoid sign flips in the quaternions, if tracking is lost or interruptions occur in the marker detection, discontinuities in the estimation may arise,

leading to oscillations in roll, pitch, and yaw values.

In summary, the method based on ArUco markers and stereo vision provides an efficient and low-cost solution, but it presents limitations due to *environmental conditions* and the sensitivity of the cameras. For an application like ours, these limitations must be considered and, if necessary, compensated by additional sensors that complement the estimation.

Figure 2.3: Image caption

2.5.1 Sub-section title

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas (see Listing 2.1). Suspendisse arcu magna, faucibus ut tincidunt non, ultrices ut turpis. Nullam tristique vehicula massa, id commodo orci sollicitudin vel. Donec nibh ante, ultrices non facilisis sed, mattis id ligula. Sed sed orci sit amet nulla egestas gravida. Suspendisse laoreet, massa vel sagittis gravida, lectus ligula feugiat risus, a aliquam dolor eros ac orci. Nulla egestas tortor quis nunc scelerisque sed tincidunt massa scelerisque. Pellentesque vulputate pharetra lectus, vitae ultricies nisi luctus eu. Nam congue dui eu quam euismod vitae fermentum sem vehicula. Etiam ac leo id nisi placerat posuere. Curabitur mattis augue eget dolor tempus accumsan consequat diam imperdiet. Sed tristique orci id lacus vulputate rhoncus. Morbi tincidunt ante sed turpis luctus tincidunt et sit amet augue. Cum sociis

natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nunc viverra urna non libero sodales euismod et eleifend sapien. Donec aliquet risus non massa dignissim sollicitudin. Integer a ligula eros. Morbi et lacinia augue [**?**].

```
<p>
Pellentesque ac tortor eget eros iaculis euismod
vitae vitae augue.
</p>
<!-- comment -->
```

Listing 2.1: caption text

Inserimento bibliografico [2] [1] [3]

Bibliography

[1] E. W. Weisstein, "Euler angles," *https://mathworld. wolfram. com/*, 2009.

[2] Q. Zhou, Z. Li, G. Yu, H. Li, and N. Zhang, "A novel adaptive kalman filter for euler-angle-based mems imu/magnetometer attitude estimation," vol. 32, no. 4, p. 045104.

[3] M. B. Alatise and G. P. Hancke, "Pose estimation of a mobile robot based on fusion of imu data and vision data using an extended kalman filter," *Sensors*, vol. 17, no. 10, p. 2164, 2017.

[4] H. C. Kam, Y. K. Yu, and K. H. Wong, "An improvement on aruco marker for pose tracking using kalman filter," in *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, pp. 65–69, IEEE, 2018.