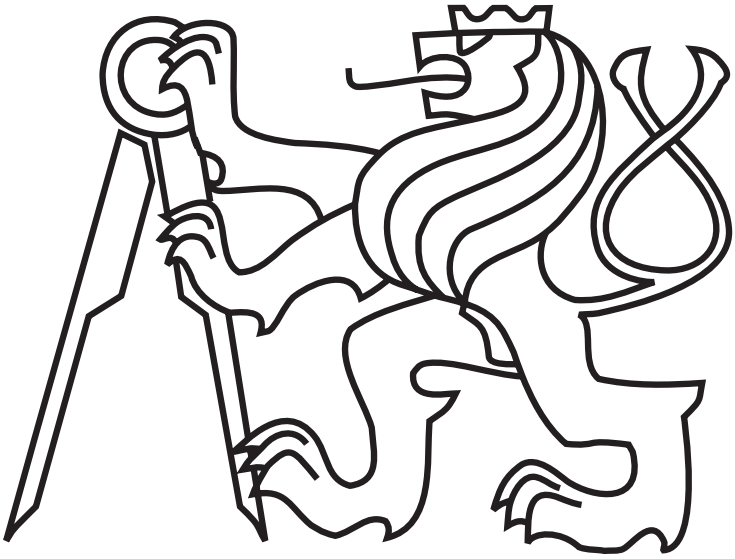
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering

BACHELOR THESIS



Jiří Ulrich

Fiducial Marker Detection for Vision-based Mobile Robot Localisation

Department of Cybernetics

Thesis supervisor: **doc. Ing. Tomáš Krajník, Ph.D.**

August, 2020

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Ulrich Jiří** Personal ID number: **474426**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer and Information Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Fiducial Marker Detection for Vision-Based Mobile Robot Localisation

Bachelor's thesis title in Czech:

Metody rozpoznávání referenčních značek pro vizuální lokalizaci mobilních robotů

Guidelines:

1) Learn about methods for detection and recognition of fiducial markers in images used in the mobile robotics domain.
2) Select the most popular methods suitable for ego-localisation in envrionments with known marker positions as well as external (reference) localisation of multiple robots.
3) Perform adjustments of the methods so that they can operate in a comparable manner.
4) Propose a set of performance measures relevant in both of the aforementioned scenarios.
5) Create a simulation environment capable to run repeatable experiments of fiducial-based visual localisation.
6) Design set of experiments to compare the performance of the selected methods in the two aforementioned scenarios.
7) Perform the experiments both in simulation and real data.
8) Compare the methods using the metrics specified in 4).

Bibliography / sources:

[1] P. Lightbody, T. Krajník et al.: A versatile high-performance visual fiducial marker detection system with scalable identity encoding. In Symposium on Applied Computing, 2017.
[2] Arvin et al: COS-phi: Artificial Pheromone System for Robotic Swarms Research. In IROS 2016.
[3] Wang, John, and Edwin Olson. "AprilTag 2: Efficient and robust fiducial detection." 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016.
[4] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. 2014. "Automatic generation and detection of highly reliable fiducial markers under occlusion". Pattern Recogn., June 2014.
[5] Salinas et al.: Mapping and Localization from Planar Markers. Pattern recognition 2018.
[6] Munoz-Salinas, Rafael, and Rafael Medina-Carnicer. "UcoSLAM: Simultaneous Localization and Mapping by Fusion of KeyPoints and Squared Planar Markers." arXiv 2019.

Name and workplace of bachelor's thesis supervisor:

doc. Ing. Tomáš Krajník, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **06.01.2020** Deadline for bachelor thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

doc. Ing. Tomáš Krajník, Ph.D. Supervisor's signature	doc. Ing. Tomáš Svoboda, Ph.D. Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date August 14th, 2020 Jiří Ulrich

Acknowledgement

I would like to thank to all the people that supported me during my work on the thesis. Especially, I would like to thank my supervisor Tomáš Krajník for the opportunity to choose this topic and learn a lot in many disciplines. I would like to thank Tomáš Vintr for moral support and encuragement to continue. Also, I would like to thank all the members of the Chronorobotics Labratory for creating such a unique environment where we can learn from each other. The work was partially supported by the Czech Science Foundation project 17-27006Y.

Abstrakt

Tato práce představuje metody pro rozpoznávání referenčních značek vhodné pro externí a palubní lokalizaci založenou na obraze, a rovněž zavádí nový systém pro lokalizaci značek. Nově popsaná referenční značka je schopná určení plných 6 stupňů volnosti, unikátní identifikace a zpracování v reálném čase. Tento typ dostupných značek spolu s běžnou kamerou cílí k přijetí robotickou komunitou jako náhrada za současné drahé lokalizační systémy. V práci také porovnáváme představenou značku s nejpopulárnějšími současnými značkami z hlediska jejich kvality určení pozice a orientace. Jelikož kvalita metod závisí na schopnosti přesně určit pozici, orientaci a ID, navrhli jsme kritéria pro testování chyby jak pozice, tak i orientace založená na reálných aplikací v oborech mobilní a rojové robotiky. Za tímto účelem jsme vyvinuli umělé prostředí založené na simulaci s třemi různými scénáři společně se systémem automatického testování a vyhodnocení. Abychom získali povědomí, jak se metody chovají v reálných podmínkách, otestovali jsme je rovněž na datasetu s robotickým rojem.

Abstract

This thesis presents fiducial markers suitable for vision-based external and ego-localisation, as well as it introduces a new fiducial marker localisation system. The newly described marker is capable of the full 6 degrees of freedom estimation, unique identification and real-time evaluation. This kind of low-cost markers combined with an off-the-shelf camera aims to be accepted by the robotic community as a substitution of the current high-end expensive localisation systems. In the thesis, we also compare the fiducial marker with the most popular state-of-the-art methods in terms of pose estimation quality. As the quality of the methods depends on the ability to determine the pose and ID precisely, we proposed a set of criteria for testing both the position and orientation error based on real-world applications in fields of mobile and swarm robotics. For this purpose, an artificial simulation-based environment with three different scenarios was developed together with an automated testing and evaluation framework. To obtain knowledge about how the methods behave under real-world conditions, we also tested them on a swarm dataset.

Contents

List of Figures

List of Tables

1 Introduction

In the current world, the robotics has developed into a ubiquitous and well-established field. It happened mainly thanks to the cut in prices of resources, better accessibility, and also significant improvements in the manufacturing process and overall miniaturisation. The field ranges from heavy industrial manipulators over consumable retail robots to robots which are intended to mimic the behaviour of insects. As it might be concluded, each of the robotic areas requires robots to possess a particular set of skills depending on their purpose and the environment they operate. Although there are attempts to make robots fully sustainable and autonomous, we are not there yet. The robots usually operate in controlled environments, and the operators oversee them; thus, it does not support deployment straight out of the box or high robot acceptance by society.

Once robots are moving freely in an environment, they have to be able to navigate to their destination. However, for this purpose, they also have to know where they are within the operational space either if it is an office or outdoor field. The robots can get lost and fall into a malfunction or unpredictable state without having this kind of knowledge of their surrounding. So, they must have some sensors to perceive what is around them and based on that, try to determine the position and orientation in the space.

The camera is one of the most popular devices for accessing the necessary understanding of the environment. Therefore, there exist many methods of transforming the camera image frame into useful information about objects and their positions. One can either try to find any naturally occurring objects and estimate their positions, look for significant pixel patches in the image, or detect unique patterns. The *fiducial markers*, which this thesis is based on, are those unique patterns with specific properties, which are added to an environment to provide accessible and reliable sources for localisation. Their applications can be found in diverse fields like logistic warehouses, augmented reality, multi-robot and swarm robotics, and even evaluation of experiments where they can provide the position of the robot as ground truth. They are so popular because they are affordable and usually require only an off-the-shelf camera and a printer.

Unfortunately, several drawbacks are affecting the overall performance and use cases. Every camera has a naturally occurring noise and limited resolution; also every material, which the marker is made from, reflects the light differently. Thus, we have to ask how reliable the markers are and what is their precision because they may produce inconsistent results under varying illumination. Each marker localisation system addresses those in its specific manner and with different success; therefore, the motivation for this thesis is to evaluate those subjects.

The goal of this thesis is to improve an already existing fiducial marker positioning system, so it would also allow the full 6 degrees of freedom estimation of multiple markers. The marker should be capable of reliable and stable pattern detection, identification and pose estimation over a wide range of camera angles and lighting conditions. Current systems either support those aspects partially or not at all. Therefore, the marker and the algorithm

has to be analysed for additional information, which could introduce the missing pieces for a fully working localisation system. However, we have to ensure to maintain the performance and keep it comparable with the other commonly used fiducials. Thus, thorough testing and evaluation on clear and also noisy images are needed, in order to verify the marker's position between the most popular ones.

The thesis is composed of four main chapters. We start with an introduction to the state-of-the-art techniques used for localisation and mapping in robotics. The focus is mostly on vision-based localisation methods suitable for external and or ego-localisation, which we further expand on the systems that do not need any expensive pieces of equipment other than a printer and a camera. The second chapter is then elaborating on one selected fiducial, and we present several modifications which introduce a possibility of estimation the full six degrees of freedom and more stable marker identification. On top of that, we integrate the system with the Robot Operating System (ROS) [2], which involves a unified way of communication and a graphical interface for the users and developers. Following section then describes real-world and three simulated scenarios allowing automated and repeatable measurements generation of the most popular fiducial markers. In the last chapter, we evaluate the methods on those datasets and provide a comparison of their performance under various conditions.

2 State-of-the-art in localisation and mapping

The ability of mobile robots to autonomously move and operate requires particular knowledge and understanding of an environment. To reliably perform tasks in the operational space, the robots have to be able to navigate and or localise. The navigation problem is to plan and drive to given positions or along a given path. The robots also need to be provided with an abstract representation of surrounding, a map, to be able to plan. However, they also have to localise to know their position concerning the goal or along a given route. For localisation, they have to evaluate the incoming sensory data and estimate position and orientation within a specific frame. Once the localisation and mapping problems are addressed together, it is called the Simultaneous localisation and mapping (SLAM). As the robots operate, the sensors provide by nature uncertain and also inaccurate information, which raises a problem of correct data processing. Thus, it is beneficial to count with the imprecision and develop a way of noise cancellation; otherwise, it might lead to incorrect maps, localisation, or path planning.

2.1 Localisation

Localisation is a process of determining the current position and orientation of a robot from sensory data. It is necessary especially when the robot has to move not only along a given path, but its goal is changing on the fly. Localisation methods split into two groups based on the reference frame - relative or absolute [3, 4]. Further, robotics also differentiates between ego-localisation, when a sensor is on a robot, and external localisation where a sensor is off the robot.

2.1.1 Relative localisation

Relative localisation, also dead-reckoning, is estimating the robot's relative position and orientation to its starting pose. It relies on the previous pose assumption and the onboard sensory data change, which the robot uses for determining where and how much it moved. This approach implies the possibility of error accumulation over time, and so it is necessary to perform adjustments and error corrections along with the movement. As the sensors are on the robot, the ego-localisation is considered as part of the relative localisation. In general, relative localisation is more efficient than the absolut one in terms of memory and time requirements.

Odometry

Odometry employs the data from motion sensors, mainly magnetic or optical encoders, for the position estimation. Encoders signalise how much wheels, belts or other moving parts revolved in a specific fixed time. Thus, knowing the dimensions of them, the assumed

travelled distance by the wheel or belt can be calculated. Therefore, the position is then obtained through the integration of velocity from encoders. This integration results in drift. The error accumulates over time and a driven distance due to the incremental discrete estimation. Thus, it is not reliable nor precise from a long-term and long-distance perspective, particularly in uncertain environments. Nevertheless, it is accessible and widely spread thanks to the simplicity and low cost, mainly as their motors usually include encoders. Despite the erroneous characteristics in long-term, odometry provides a practical and precise localisation in the short-term perspective. In [5], the odometry is combined with camera-based position estimation, which corrects the position and orientation when a known object is observed.

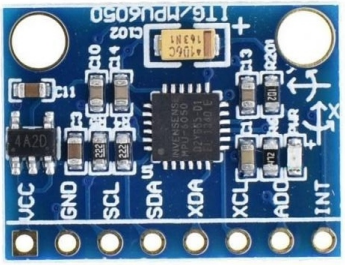


Figure 1: Combined kit of accelerometer and gyroscope

Inertial localisation

Inertial localisation involves gyroscopes and accelerometer sensors which provide an acceleration contrary to the velocity in odometry. Inertial sensors are better in orientation estimation than the odometry sensors because they directly measure a rotation contrary to the estimation from the driven path of individual wheels. The low-cost localisation system based only on inertial sensors is presented in [6]. However, there is a drift growing with time caused by two integrations to calculate the position which produces the quadratic error, the main disadvantage of those. Both sensors are nowadays produced as a single chip as in Figure 1.

Accelerometers measure the linear acceleration of a robot in multiple directions. They are sensitive to any disruption in the horizon, which results in false measurement as of the influence of the gravitational force. Thus, in a rugged environment, the sensor has to be isolated from unexpected bounces and tilt changes. Also, if the robot moves or turns slowly, it is hard to separate the data from noise.

Gyroscopes provide the angular acceleration which integrates to the angle estimation. They are often combined with the odometry to enrich the position estimation with reliable orientation. The downsides are larger dimensions, which ensure reliable and noise-isolated data; thus, it might be a restriction for smaller robots. Gyroscopes are also affected by the centrifugal and Coriolis forces, and therefore at fast turning, the sensor loses its precision.

2.1.2 Absolute localisation

Absolute localisation, also the kidnapped-robot problem, is directly determining the position and orientation of the robot within a global reference frame. It establishes the pose without any information about its previous state. Thus, there is no error accumulation over time as the pose estimation is established newly whenever the robot localises; therefore, occasional errors in the past do not have any influence on the current estimation. Absolute localisation can be either external or ego-localisation. Ego-localisation, in this case, requires a new understanding of the reference frame to localise correctly, contrary to the external localisation when sensors can monitor the whole environment.

Figure 2: Personal magnetic compass

Figure 3: One of the several GPS satellites

Compasses

Compasses are ego-localisation sensors which measure the heading of the robot. It is particularly significant for localisation and navigation because a slight offset in the angle can result in a sizeable lateral error. Therefore, they are often used in mobile robotics to support orientation estimation as in [7]. However, they cannot be used for position estimation as they measure only the angle off the pole direction. The common disadvantage of magnetic compasses is the magnetic field itself as it can deform by steel constructions, power lines, or natural magnetic anomalies. Thus, it is not suitable for indoor applications. In mobile robotics, the variation of a magnetic compass, the fluxgate compass, is popular because of its low price and the lack of moving parts which makes it less fragile.

Global positioning system

Global positioning system (GPS) consists of several satellites on the orbit emitting high-frequency signal with encoded information of their current position. The standard precision is around 15 meters depending on the number of visible satellites. It can be improved to the accuracy of centimetres using differential GPS (DGPS), which provides local location correction through fixed reference stations. The benefit is the lack of additional equipment

other than a sensor on a robot, so it is convenient for portable outdoor platforms. The GPS application is limited to exteriors only as the signal is not strong enough to be received inside buildings or underground [8].

Landmark localisation

Landmarks are significant and detectable objects which easily distinguish from the others. Usually, they have known characteristics as a geometric shape, emitted signal, or high contrast colours and patterns. Landmarks are recognisable through the robot sensors, and their relative position to the robot is often possible to estimate. They can be divided based on the origin, natural or artificial, or based on energy activity, active or passive.

Natural landmarks are objects, not necessarily with origin in nature, which were initially not meant to support localisation. However, they form such features that are easily noticeable like the window edges or in general any corners. These features have specific contrast and spatial disposition that the robot can rely on and localise by them. Landmarks extracted from a camera image are typical passive natural landmarks, called simply *features* [9, 10]. They benefit from the fact that the environment does not have to be enhanced and controlled in advance the robot deployment. This benefit can also be a disadvantage when dynamic changes happen like rain or snow. Detecting natural landmarks is computationally more demanding, and thus it may limit the robot from performing other tasks.

Artificial landmarks are made with the purpose of localisation, and they are intentionally added to the environment, which often has a lack of natural ones. They have a priori known characteristics which make them easy to detect and calculate their position and orientation relative to a sensor. In the case of active artificial landmarks, they are called *beacons*. They can emit light or transmit any radio or sound signal as in [11]. The passive ones vary significantly in their shapes, colours or materials because they are simpler to produce and set up as the lack of energy source. The artificial landmarks are robust and reliable relative to the natural ones. Simple barcodes, retroreflective targets, or infrared beacons are examples of those landmarks.

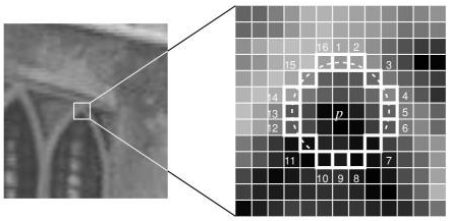


Figure 4: FAST feature for corner detection

Figure 5: Ultrasonic beacons

Landmark localisation is principally based on the comparison of the landmarks at known positions with the current landmarks observed by the sensors [12]. The precision and detectability of landmarks decrease with increasing distance and angle under which the sensor observes them. Therefore, it is often combined with the odometry to overcome situations when landmarks are not visible with enough certainty. There are two basic ways how to localise in space using landmarks [13]. *Trilateration* is measuring the robot position using only the distances from landmarks. *Triangulation* is using the angles relative to the longitudinal axis to estimate the position.

External localisation systems

External localisation systems provide a reliable and precise estimation of the absolute position within a given operational frame. They are widely spread in the mobile robotics, especially for assessing the ground truth of position or closed-loop robot control. For a reliable system, high precision and speed of measurement are necessary; however, expensive and closed-source hardware and software are often unavoidable. The possible issue might also be in low scalability for multi-robot systems or swarm robotics.

Vicon is a high-end optical passive motion capture system used for 3D position estimation [14]. It consists of a set of retroreflective markers which are tracked by high-resolution infrared cameras. Each camera captures synchronised images and computes the 2D position of the reflected targets. Once the overlapping points are detected, the 3D position of them is calculated using triangulation. Although initially developed for gait analysis, it became popular in robotics, biomechanical or augmented reality field because of its precision and flexibility [15]. However, it cannot be easily deployed in the exteriors because of the naturally occurring reflections of infrared light corrupting the infrared tracking.

OptiTrack is a framework for motion capture providing 3D positional data through a passive optical system [16]. As the Vicon system, the OptiTrack system requires multiple high-speed synchronised cameras to cover the operational field. The tracked objects are equipped with retroreflective targets which reflects the infrared light emitted by the cameras. The 3D position of the markers is obtained via triangulation. There are cases similar to the Vicon when even the deployment in indoors is not possible because of the use of infrared sensors which are commonly utilised in swarm robotics. In [17], the authors state that the OptiTrack, despite its low-cost, is comparable to the Vicon system, and it is suitable for applications with a bit lower requirements to precision.

Phoenix represents an optical active motion capture system based on the LED markers providing a 3D position information [18]. The hardware components are similar to the mentioned optical passive systems; however, the critical difference is in the markers which actively emit light. Optical active systems do not require any supporting strobes or infrared light, unlike the passive systems. Thus, there is not a problem with the tracking cameras facing and blinding each other. The disadvantage of the active approach is the need to power supply the markers, which requires charging wires and batteries.

Figure 6: Vicon/OptiTrack markers

Figure 7: Phoenix markers with control unit

Fiducial markers

Fiducial markers are observable artificial objects in an image which can be used for measurement or serve as a point of reference. In robotics, the markers are engaged in localisation, navigation, and tracking using a camera either for ego-localisation or external localisation. The most popular fiducials are typically planar black-and-white tags with known shape and dimensions. Thus, machine vision techniques can estimate their relative position and or orientation to the camera. Thanks to their planar nature, they can be variously placed on walls, floors or other robots to enrich the environment with artificial passive landmarks. Therefore, many applications can be found like to support robot Handle (Figure 8) by Boston Dynamics [19] or help while the docking phase of robot Linda (Figure 9) developed during STRANDS project [20]. Their deployment in distance estimation between autonomous vehicles is thoroughly examined in [21].

AprilTag is a fiducial of a square proportion with identification encoded within. It provides a full 6 degrees of freedom localisation [22]. The detection core at first detects all line segments in the image and searches for those which form a rectangular shape, leading to low false negatives, yet high false-positive rate. Thus, several restrictions on the outer and inner areas of the pattern are applied to reduce the number of objects to process further. The inner area is sampled for white and black pixels, then thresholded to provide a 2D binary code. The coding system produces lexicographic encoding with a highly scalable number of possible tags, also with a flexible false positive rate robustness. The fiducial system is capable of detecting, identifying and localising the marker even when partially covered when using an appropriate coding system parameters. In [23], the improved version of AprilTag is presented introducing non-square markers with the possibility to include non-marker related content inside the inner encoding area.

ArUco is a square-based fiducial marker incorporating binary matrix identification encoded in its inner area [24]. The detection core performs similarly to the mentioned one

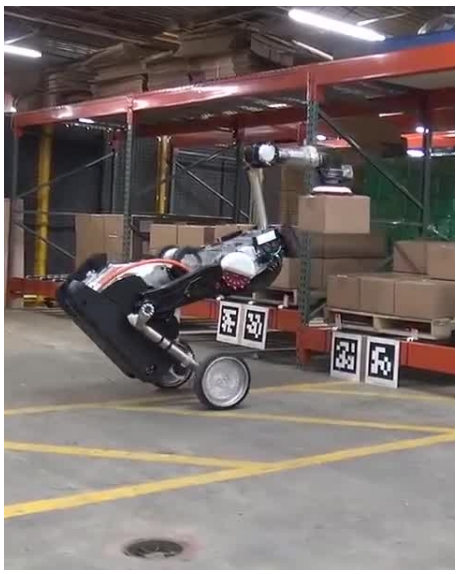
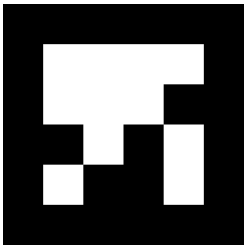


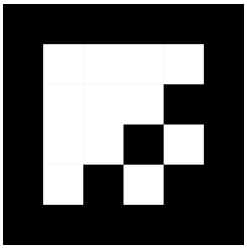
Figure 8: AprilTag markers as package and rack labels for warehouse robot

Figure 9: WhyCon markers as letter o in the "Robot station" sign to support docking into a charging station

in AprilTag. Thus, there is an edge detector on the thresholded grey-scale image whose output is processed to form rectangular objects. However, it does not include such line segments detection, and so contrary to the AprilTag, the ArUco is not capable of dealing with occlusion. Instead, the authors solve this problem by using a marker board, which consists of multiple smaller markers which can be detected to substitute the occluded ones. The authors later present a modification of this fiducial system detection core which provides faster detection by using temporal marker information from a previously observed image sequence [25]. In general, the square-based markers have the drawback of the variable estimation time as in [21], which might cause unexpected delays in processing.



(a) AprilTag



(b) ArUco

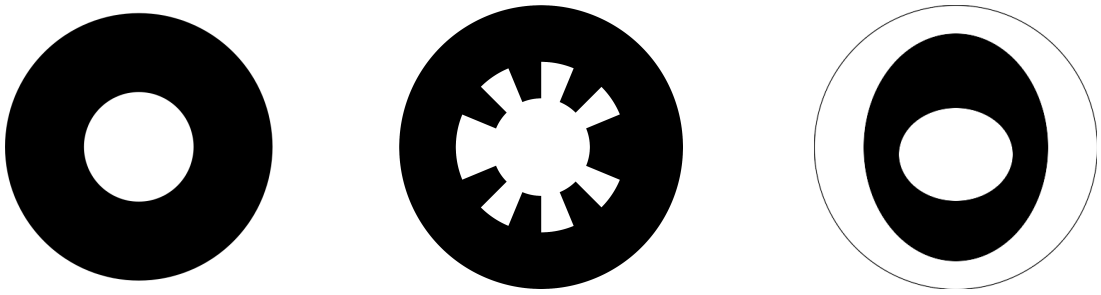
Figure 10: Square-based fiducial markers for vision-based localisation

WhyCon outlines a real-time, low-cost visual localisation system based on black-and-white roundel markers [26]. The marker is in the form of two concentric circles with known diameters where the outer circle is black, and the inner one is white. The system provides

only 5 degrees of freedom, meaning it is not able to estimate the rotation around its surface normal. However, in situations when one does not care about the orientation and needs only the 3D position, WhyCon serves well and provides the precision in the order of millimetres. The performance of this system is significant because it can track several thousands of tags in the time of a few milliseconds. The main advantage of WhyCon is low memory and computational requirements. Due to the WhyCon's popularity, there are systems which build on top of it and address the missing orientation problem - WhyCode and SwarmCon.

WhyCode is a localisation system which builds on top of WhyCon system, and thus it is an enhanced circular marker [27, 28]. Therefore, it features similar robustness, precision, and performance as the WhyCon mentioned before. The WhyCode adds a unique and distinguishable identification and an estimation of the rotation around the marker's surface normal vector. It provides the full 6 degrees of freedom while maintaining its low computational requirements. The identification is achieved through a scalable circular barcode on the inside of the outer black circle. The barcode is the Necklace binary code which is rotationally invariant; thus based on the rotation of the code, the surface rotational angle is recovered. However, the encoding system reduces in the possible detectable and identifiable angles and distances due to the possible lack of visibility of the encoded bits.

SwarmCon marker extends the WhyCon system by introducing the angle estimation around the surface normal and distinguishability of markers [29]. The concentric circles are replaced by ellipses which allow obtaining the heading from the rotation of the ellipse covertices. However, this approach is possible only when the marker plane is perpendicular to the camera axis. The unique identification of individual markers is achievable by combining differently scaled and rotated concentric ellipses. There is also an additional build-in feature allowing automatic calibration and transformation to the swarm arena coordinate system.



(a) WhyCon (b) WhyCode (c) SwarmCon

Figure 11: Roundel-based fiducial markers for vision-based localisation

2.2 Mapping

Mapping is done by robots to capture their surrounding and transform it into maps, an abstract representation of the operational environment. Thus, having the landmarks mentioned above in the environment, the robots can use them to build a map and to

localise in it, which is particularly essential for ego-localisation. The robots use sensory information, which is usually additionally filtered and evaluated to achieve robustness and reliability, to describe the environment. Maps can be created in two ways. On the one hand, we can prepare the maps by ourselves and provide them to robots, or on the other hand, they can be created by robots. When the robots are building the maps, they either autonomously move and explore on their own, or they are teleoperated. In both cases, the environment is at first unknown to them. In general, maps form two basic categories, metric and topological, depending on the way the sensory data are abstracted [30].

2.2.1 Metric maps

In metric representation, the map description is stored as coordinates [31, 32]. Thus, all the objects, points of interest, and obstacles are represented by their positions within the map coordinate space. Therefore, multiple sensors can be incorporated because a robot only needs to know how to transform the data to the map metrically. Another benefit is the human-readability and verifiability of the stored representation. However, the disadvantages of this approach are the sensitivity to the noise or miscalibration of sensors. Also, metric maps can be memory demanding, which depends on information stored - free space, occupied space, objects properties.

Occupancy grids represent the environment in a discrete way, divided uniformly into cells. Each cell contains information describing the area it covers. This method requires many memory resources, yet it is a straightforward representation and easy to implement.

Geometric maps models the environment as a combination of geometric primitives. These maps can be seen as a more abstract occupancy grid by joining cells with the same descriptive information. They are less memory demanding, but they are harder to create because the sensory data need an interpretation to form a geometric shape. Also, they are not so detailed because they represent more substantial and more significant areas.

Landmark maps contain only the descriptions and positions of landmarks. They are very memory efficient and easy to create. The disadvantage of such maps is the need of having a controlled environment without landmark changes. There might also be redundancy in supplemental information when multiple same markers are connected to the same location.

2.2.2 Topological maps

Topological maps are better in the abstraction as they are modelling the world as a graph [33, 34]. The different places are represented as the graph nodes, and the edges are the connections in between them. As the nodes contain descriptive data, the maps are less detailed because it would require the creation of additional nodes and edges, and thus leads to higher memory and computational requirements. The benefit is that there is no need for metric transformations of sensory data into the global representation. Thus, a node can

only contain the relative representation of a place which, however, raises a need for proper detecting places. In the case of erroneous place recognition, this may lead to failure in localisation and navigation, especially in places without robust distinguishable descriptive elements.

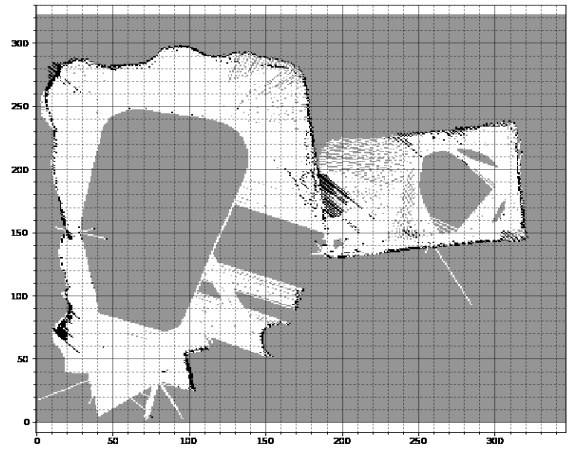


Figure 12: Occupancy grid map

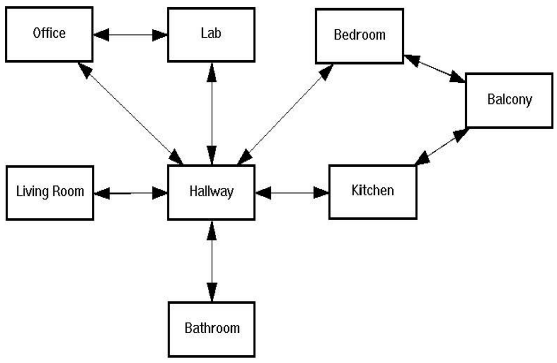


Figure 13: Topological map

2.2.3 Hybrid maps

The combination of multiple types of maps, mainly metric and topological, forms hybrid maps [35]. They are popular because they reduce the drawbacks of individual map categories. Topological maps can form the high-level representation, thus benefiting from the abstraction, while the lower-level can use the metric maps to maintain the precision. These maps are useful when there is a high chance of mistaking the places, and thus falsely localisation within the map.

2.3 Simultaneous localisation and mapping

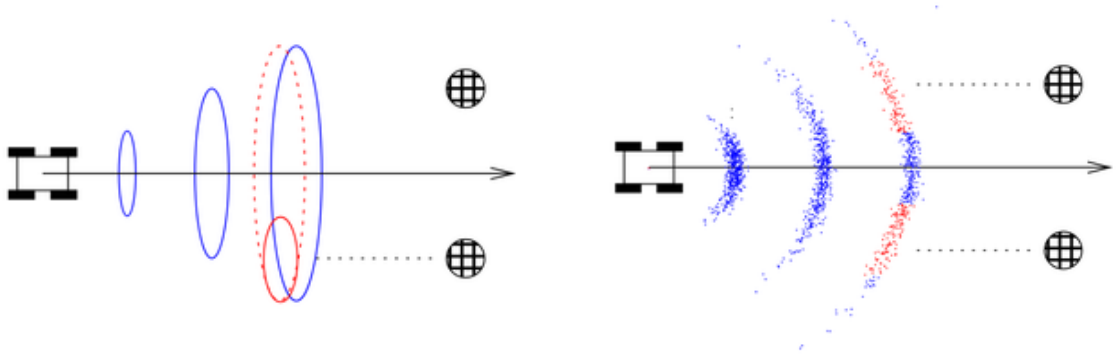
Simultaneous localisation and mapping (SLAM) is a well-known challenging problem in robotics. It describes a situation when a robot has to create a representation of an unknown environment while estimating its location in it. The main issue, apart from proper simultaneous processing, is the correct treating of the uncertainty of the robot's position and creating a map based on sensory information which is often noisy. Therefore, to represent the sensory observations and to model the robot motion, probabilistic methods or filters are applied to reduce the noise by creating hypothesises of the actual state. The most popular methods for estimating the SLAM state are extended Kalman filters and particle filters [36].

Extended Kalman filters

Extended Kalman filter (EKF) creates a model of the uncertainty of the robot position and representation of the map by the normal distribution. Assuming the position estimation is uncertain; therefore, even the estimation of the measured data within the map is not absolute. Thus, the EKF-based SLAMs are favoured in applications where the map can be build from precise measurements. The problem of the EKF algorithms is their quadratic complexity concerning the landmarks on the map, which restricts large-scale maps. The map size can be resolved by breaking the map into sub-maps which are processed individually. EKF SLAM is typically incorporated in feature-based applications [37, 38].

Particle filters

The uncertainty of the robot and feature positions are expressed as individual particles forming a probability density. As the density function models the position, there can be multiple hypotheses of the robot position at the same time. Depending on the observed data, the positions of particles are reassessed and reweighted based on their likelihood of appearance. The amount of particles highly grows as there have to be multiple particles per landmark to has high likelihood. Therefore, particle filters are applicable mainly for localisation and not for building a map. The solution to the particle filters and their modification to lower memory complexity presents FastSLAM [39].



(a) extended Kalman filter

(b) Particle filter

Figure 14: Robot position estimation by SLAM filters [1]

2.3.1 Visual SLAM

The solution to the SLAM problem through cameras only has been thoroughly addressed by many authors, mainly because the camera is often affordable and straightforward sensor to handle. However, the sensory information is challenging to process correctly in order to obtain enough data for SLAM underlying algorithms. Since the camera produces visual data only, those approaches are called visual SLAM (vSLAM) techniques. Based on how

the image is processed, there are two main categories of vSLAM - feature-based and direct methods [40, 41].

MonoSLAM

Monocular SLAM (MonoSLAM) is a feature-based SLAM using the SIFT descriptors, and it represents the first visual SLAM technique presented in [42]. The mapping initialises by observing a known object with a defined global coordinate system. The extended Kalman filter is the underlying 3D position estimation algorithm. The disadvantage of this method is a massive growth of the computational requirements concerning the size of the operational area. The problem of estimating the depth of the features solves a feature tracking in multiple consecutive frames. However, the tracking leads to another issue where the camera movement has to be slow enough to ensure successful tracking. It can be partially overcome by increasing the camera capture speed; however, such modification results in an explosion of map size and cannot be processed in real-time.

LSD-SLAM

Large-Scale Direct Monocular SLAM (LSD-SLAM) is a real-time direct monocular visual SLAM which does not work with extracted image features [43]. This methods directly incorporates the image intensities for all the aspects - mapping, tracking, localisation. Compared to feature-based SLAMs, this method uses the image as a whole; thus, it benefits from all information and not only a few high-contrast patches. As the authors present a modified direct image alignment, the LSD-SLAM reduces the problem of scale-drift in large environments. The depth in the map is estimated from many per-pixel stereo comparisons. It results in more precise and stable localisation and 3D reconstruction in sparse and low-textured environments. The LSD-SLAM was extended to stereo [44] and omnidirectional [45] cameras.

ORB-SLAM

ORB-SLAM is a versatile and scalable monocular SLAM system based on visual ORB features [46]. The ORB features are fast binary descriptors based on the BRIEF features, but they are noise-resistant and also rotation invariant. They also allow real-time performance even without GPUs. The co-visibility graph is used for map representation which allows operating in large environments because the localisation is done only in a local co-visible area. The loop closure is done through a spanning tree of a graph, which reduces the graph size to process, and thus the error distribution along the loop is faster. In [47], the ORB-SLAM2 is presented, which is an improvement of this method. It enables the application of not just monocular camera, but also stereo and depth cameras. So it solves the problem like the scale drift of the map or trajectory tracking. Another improvement is

a lightweight operation mode which provides only localisation without mapping. Thus an already created map can be reused without creating a new one every time.

UcoSLAM

UcoSLAM combines the natural and artificial landmark localisation to overcome the volatility and repetitions of natural landmarks which are used by most of the vSLAMs [48]. The authors use the square-based ArUco fiducial as an artificial landmark to support the localisation and map building in repetitive environments. The system can use only the marker, keypoint features, or a combination of them; thus, it maintains the versatility of the traditional approaches. The markers support the determination of the map scale as their dimensions are a priory known. In terms of the features, UcoSLAM is designed to incorporate any type of image descriptors.

3 Fiducial marker modifications

This chapter describes the selected fiducial markers, their improvements and extensions, which are the goal of this thesis. There are many popular markers to choose from as we introduced before. The markers examined further are WhyCon and WhyCode systems as they offer tracking, localisation and maintain high-speed performance even with poor quality cameras. The selected methods applications vary from accurate and real-time localisation to post-processing a robotic experiment where is a need for precise and low-cost ground-truth.

WhyCon and the further evolved WhyCode fiducials are open-source vision-based localisation systems with millimetre accuracy which needs only a simple web camera to detect and localise up to thousands of markers per image frame. Even though nowadays robot hardware is more and more powerful, those markers maintain the original purpose of being lightweight and thus are noticeably less computationally demanding than the square-based markers, ArUco and AprilTag. They share a common core which is thoroughly explained in [26], while in [49], there is described a modification for embedded systems. The fast detection of the black-and-white pattern is performed using flood-fill segmentation, reusing the previously calculated characteristics and on-demand thresholding, which allows processing only the found segment instead of the whole image. As the system performs the detection phase, the pattern segment undergoes a sequence of tests whether it complies with the updated pattern characteristics which speed up the process by immediately dropping the unsuitable pattern candidates. Another benefit of remembering the characteristics is tracking the marker to speed up the detection because it can be assumed that the change in position of the fiducial is not significant between individual frames. Therefore, the system performance is increased by looking for the roundel at the previous location, and so processing less image area.

In [27], the authors present a flexible encoding system that introduces unique identification and an ability to estimate the rotation around the marker surface normal. The area between the concentric circles is enhanced by a circular binary code which translates to the marker ID. The code represents the Manchester-encoded Necklace binary pattern [50] which we call *teeth* because of the resemblance. The rotational invariant characteristics of the Necklace code allows obtaining the ID by reading the binary code and rotating it around until the smallest value is reached. The resulting value then corresponds with the marker ID. The angle is recovered from the required rotation of the binary code.

However, to achieve robust and reliable position and orientation of the markers, they require additional modifications, mainly as the 3D orientation estimation is not addressed at all. There is also unresolved ambiguity in the result calculation in the system core, which causes significant and unpredictable changes affecting both the identification and pose. Those issues have not been tackled since both of the markers were mainly used for position estimation and identification under small-angle only. The missing features have been thoroughly addressed by [21].

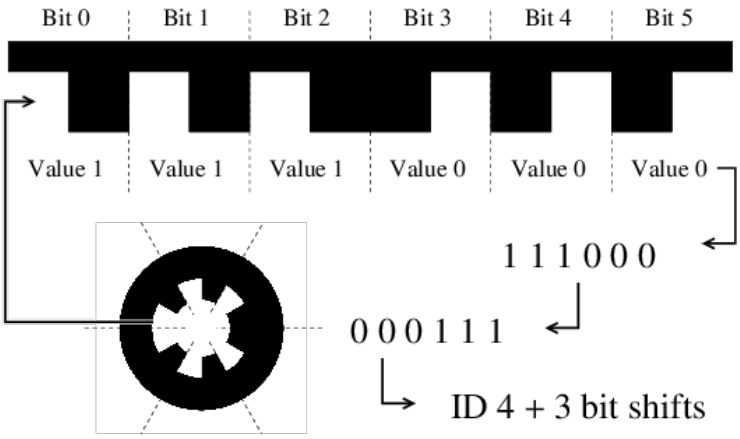


Figure 15: Manchester-encoded Necklace ID overview. The binary string is unwrapped and shifted to the lowest value matching a Necklace code. Based on the number of bit-shifts, the rotation is recovered.

To make the methods truly competitive with others or even prioritised, they have to be comfortable to users and developers in various situations. Therefore, both fiducials were provided with an interface to the *Robot Operating System* (ROS) [2] framework in version Melodic to unify the means of communication with other robotic software. Users also need to visualise or debug their system output, so a new user interface (UI) is built using the rqt plugin based on the Qt graphics framework [51].

3.1 Resolving ambiguity

Position estimation is based on the same core for both markers, and it begins once the system detects an image segment which fulfils all the validity tests. Since the segment area is even, the pattern centre is calculated as the mean of the segment pixel positions. The area is also circular, or elliptical when under perspective, so we can calculate the covariance matrix of it and apply the eigenvalue decomposition. Then, using the pinhole camera model, the canonical camera coordinates of axes and centre are obtained from the ellipse characteristics which allow us to express the characteristic ellipse equation of the marker as $\mathbf{X}^T\mathbf{Q}\mathbf{X} = 0$, where $\mathbf{Q}$ is the conic section of the segment border. Further described in [52, 27], the marker surface unit normal vector $\mathbf{n}$ and position vector $\mathbf{x}$ are calculated as below:

$$\mathbf{x} = s_3 \frac{r}{\sqrt{-\lambda_0\lambda_2}} \left(s_1\lambda_2\mathbf{q_0}\sqrt{\frac{\lambda_0-\lambda_1}{\lambda_0-\lambda_2}} + s_2\lambda_0\mathbf{q_2}\sqrt{\frac{\lambda_1-\lambda_2}{\lambda_0-\lambda_2}} \right) \tag{1}$$

$$\mathbf{n} = s_1\lambda_2\mathbf{q_0}\sqrt{\frac{\lambda_0-\lambda_1}{\lambda_0-\lambda_2}} + s_2\lambda_0\mathbf{q_2}\sqrt{\frac{\lambda_1-\lambda_2}{\lambda_0-\lambda_2}} \tag{2}$$

where r is the outer circle radius, eigenvalue analysis of $\mathbf{Q}$ yeilds eigenvalues $\lambda_0 \geq \lambda_1 > 0 > \lambda_2$ and their respective eigenvectors $\mathbf{q_0}, \mathbf{q_1}, \mathbf{q_2}$, and s_1, s_2, s_3 are unknown signs.

As the authors correctly assume, the marker is only detected when it is in the camera's field of view, thus the normal $\mathbf{n}$ faces to the camera centre and $\mathbf{x}$ lays in front of it. Therefore, the following constrains on the vectors gives us two inequalities as:

$$\mathbf{n}^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} < 0 \tag{3}$$

$$\mathbf{x}^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} > 0 \tag{4}$$

Note, that only two out of the three undetermined signes can be solved from the equations. This leads to two possible solutions $\mathbf{x_1}, \mathbf{x_2}$ and $\mathbf{n_1}, \mathbf{n_2}$, apart from the case of one solution when $\lambda_0 = \lambda_1$. However, this ambiguity of solutions has been treated only partially in the way of only applying the inequality constrain on the position vector. This approach yields four different solutions from which one is arbitrarily chosen and together with the sign uncertainty of the eigenvectors; it is unpredictable whether the correct solution was chosen. The authors in [52] solve the ambiguity by incorporating additional information about the observing camera position and orientation from an auxiliary sensor. However, there is already enough information in the fiducials that we can use, but unfortunately, we cannot use the same approach for both WhyCon and WhyCode as they are structurally different. The differences in solutions are discussed separately for each marker.

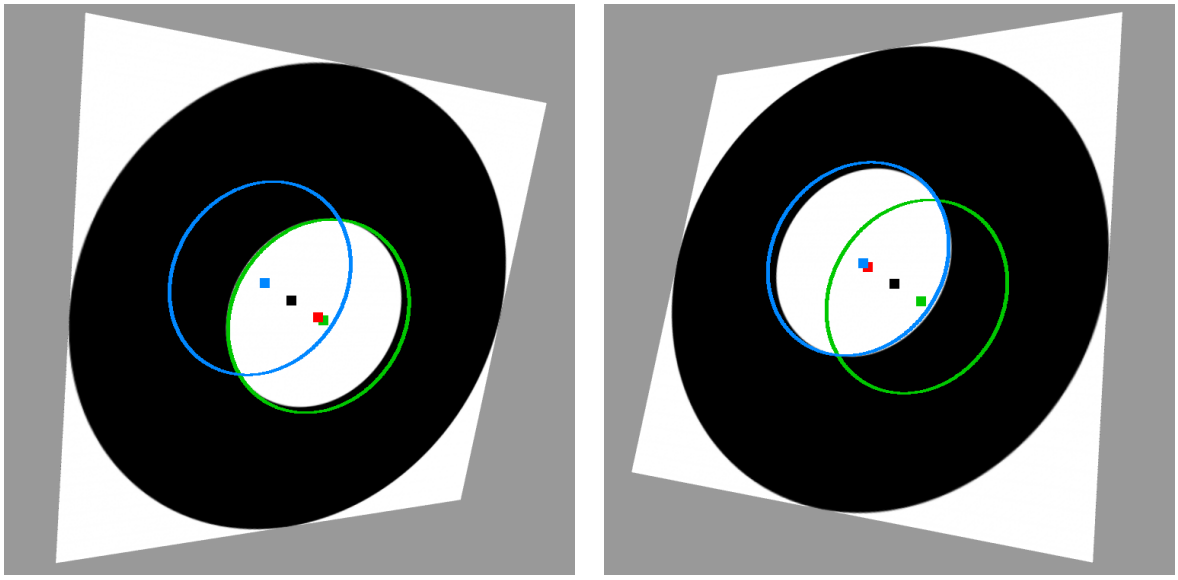
3.1.1 WhyCon's solution

The search for WhyCon's correct estimation starts in the detection phase when the pattern is found at first as one outer black roundel and then the inner white ellipse. The outer one is used for the position estimation as described above. The white ellipse is only meant for the validation together with the black circle in the testing of marker characteristics like the area ratio of the segments, or their radii ratio is sufficient. Although we calculate the centre and semiaxis of both parts, we proceed with just the outer one.

The inner white ellipse can provide us with additional information about the undergoing perspective. Smaller objects are less affected by the camera perspective; therefore, the inner centre pixel position does not shift as much. Thus, it can serve as the reference point because it is closer to the real-world marker centre, which is the same for both due to the co-centricity. We can then compare the ambiguous position vectors with the perspective centre of the inner white circle. So, we calculate all possible vectors $\mathbf{n}$ and $\mathbf{x}$ solving the inequality constrains. The outer centres are in the 3D canonical camera frame coordinates; thus, we project them back to the 2D image coordinates using camera intrinsic matrix and distortion parameters. Then, the results are two equally distanced points, from the original

mean outer centre, in subpixel resolution. The centre closer to the inner one in the means of Euclidean distance is our sought solution together with its corresponding unit surface normal vector.

In Figure 16, there is a visualisation of the decision process described above. There are both candidates, blue and green, for the marker centre, and their distance to the original black segment centre is the same. The perspective effect on the inner and outer centres is noticeable, and one can easily claim that the outer centre drifted more than the inner one. When the Figures 16a 16b are compared, they represent the ambiguous pair as the centres lay on the same line, and the markers are in the opposite orientation. It is clear that the false solution for one is the correct for the second, and the only thing changed in the constellation is the inner centre, displayed in red. Thus, the correct solution for the position vector is genuinely closer to the inner centre. However, this approach is less precise in situations when the marker plane is perpendicular, and the marker is located close to the camera's optical axis because the differences in the distance are less clear.



(a) $\theta = [0, -0.3, 0.5]$rad (b) $\theta = [0, 0.3, -0.5]$rad

Figure 16: Resolving ambiguity of WhyCon with 20 cm in diameter at canonical camera coordinates $[0.3, 0, 0]$m and orientation in Euler angles. There are inner white center in red, outer segment center in black, and both solution candidates in green and blue.

3.1.2 WhyCode's solution

In this case, the ambiguity solution is not as straightforward as the WhyCon's one because not only two solutions have to be resolved but also the occasional misidentification based on various illumination conditions. Even though it is tempting to apply the same approach above and examine the inner area parameters provided by the detection core,

we can use it only for the validation that the pattern has been detected correctly. The reason is the presence of the Necklace encoding in between the black and white circles, which causes the inner area is not round and centred correctly. Thus, we cannot rely on the calculated white centre to be our reference. Although there are cases when we would not notice any difference, it only happens when we set the number of bits to encode high enough and observe under small angles. When using fewer bits for identification, the teeth are more prominent, which causes the inner centre to shifts towards them as it is only the mean of white pixel positions. We could start using more ID bits and consider it as a solution; however, it would lower the detection range and restrict the reliability under high observation angles.

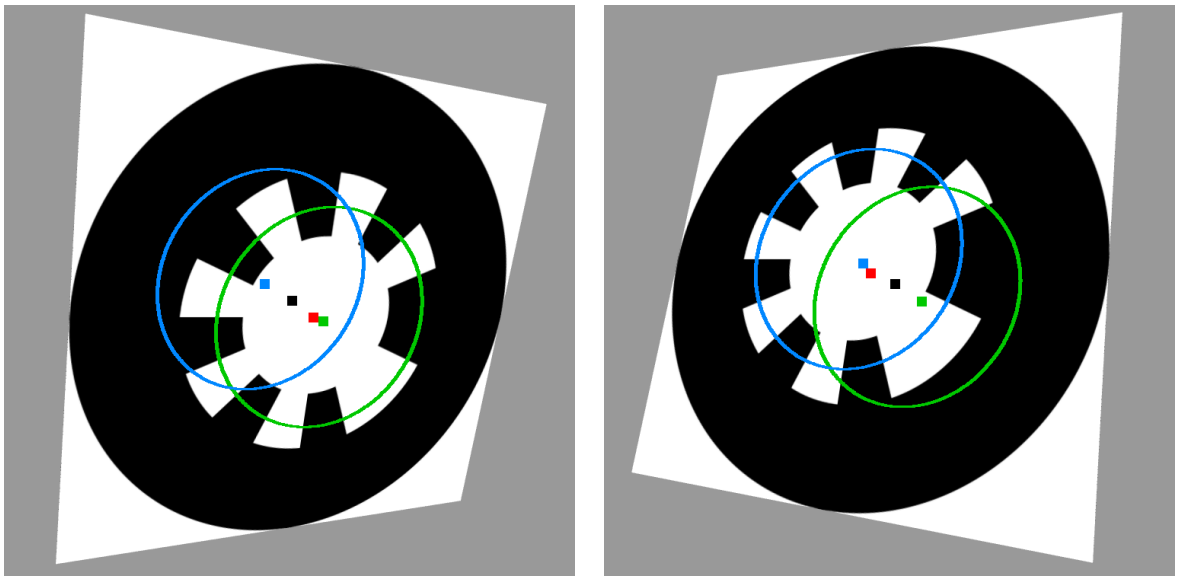
Firstly, we have to deal with problematic binary code extraction under a different light. The image brightness of the raw circular code is obtained with a simple sampling around the marker centre. The original implementation processes the raw code by searching for brightness gradient and smoothing it out. Then, the authors look for the strongest smoothed signal position, which is assumed to be in the middle of the starting tooth from where the code is evaluated by tooth-wide steps. Here comes the first ID detection problem because the brightest point does not have to be in the tooth centre, and thus the whole processing is shifted and may produce incorrect binary code.

This issue can be overcome by different processing of the samples. Instead of trying to maintain the real nature of brightness raw signal, we can threshold it and get the binary signal. The threshold is chosen as the average value. Then, we can look for teeth edges directly and convert the signal into the Manchester-encoded Necklace code.

Now, the ambiguity can be examined as the signal evaluation is robust enough. We follow the same steps as it would be regular WhyCon; thus, we calculate the two candidates for position vector and transform them back in image coordinates. Instead of measuring the distance to the white centre, we proceed as described above and sample around both assumed centres to extract the binary signal of the circular code. The idea is that sampling around the sought solution produces a signal with more equally spaced black-white edges on the circle around it. Therefore, we project the edge positions onto a unit circle and calculate their variation. We compare the variations, and the outer centre, which leads to a lower value, is then considered as the correct one. The drawback is higher computational requirements because the sampling, binarisation and Necklace decoding are calculated twice. However, considering the original performance, the difference is not significant.

We can examine the ambiguity affecting both the position estimation and sampling circle, as described in Figure 17. The influence of the binary encoding between the black and white circles on the inner centre is visible in Figure 17b as it shifts towards the double white tooth. It is even stronger when the teeth are more disproportioned. Therefore, the decision is chosen based on the sampling circle, which under higher observation angle can lay even outside the inner circle area. The markers in Figures 17a 17b are in the opposite orientation, and therefore there represent the ambiguous pair. Based on the visualisation, one can easily judge that the false solution for one is correct for the second. The decision

criterion is shown in the Figure 18, where one solution produces inconsistent binary code with unevenly sized bits compared to the other. However, there applies the same issue as in the WhyCon's case; thus in a perpendicular position close to the optical axis, the decision criterion is less informative. A partial solution might be increasing the number of samples which, on the other hand, slows overall pose estimation.



(a) $\theta = [0, -0.3, 0.5]$rad (b) $\theta = [0, 0.3, -0.5]$rad

Figure 17: Resolving ambiguity of WhyCode with 20 cm in diameter at canonical camera coordinates $[0.3, 0, 0]$m and orientation in Euler angles. There are inner center in red, outer segment center in black, and both solution candidates in green and blue together with the encoding sampling circle for WhyCode teeth.

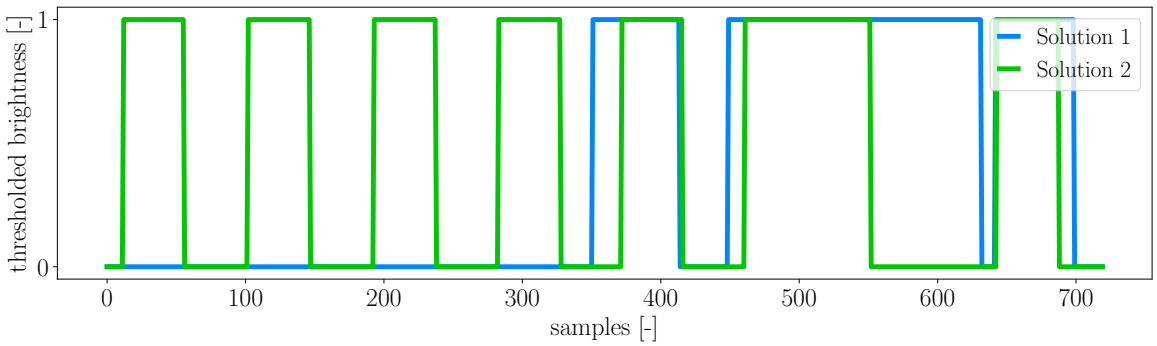


Figure 18: Binarized brightness signal of the solution candidates in Figure 17a

3.2 ROS interface

Integration with the ROS framework is done using the version *Melodic* and programming language C++. The schematic interface implementation overview is in Figure 19. The system is formed from two ROS packages communicating with each other - *rqt_whycon* as user interface and *whycon* as WhyCon/WhyCode backend. The backend is designed to be just a wrapper around the library. The input callback functions take a standard frame buffer and camera intrinsic and distortion parameters and output a newly defined message *MarkerArray* with an array of descriptions of all detected markers. Apart from the mentioned inputs, the ROS wrapper introduces optional control inputs for the action servers within the system which are used for specifying the graphical output, user-defined coordinate system or path to the configuration file. We chose the service approach because of the one-to-one communication nature and the ability to send feedback back to the users in case of errors. Having that, we can fully control the system with either the ROS command-line tool *rosservice* or through buttons in the UI which is based on the *rqt_image_view* [53] in version 0.4.13, see Figure 20. The core codebase had to be rewritten, in order to be easily separated from the UI and ROS, and so the library could run independently with different wrappers. Also, we had to implement a new way of calculating the orientation, which we decided to represent in quaternions as these are natural angle representation in ROS.

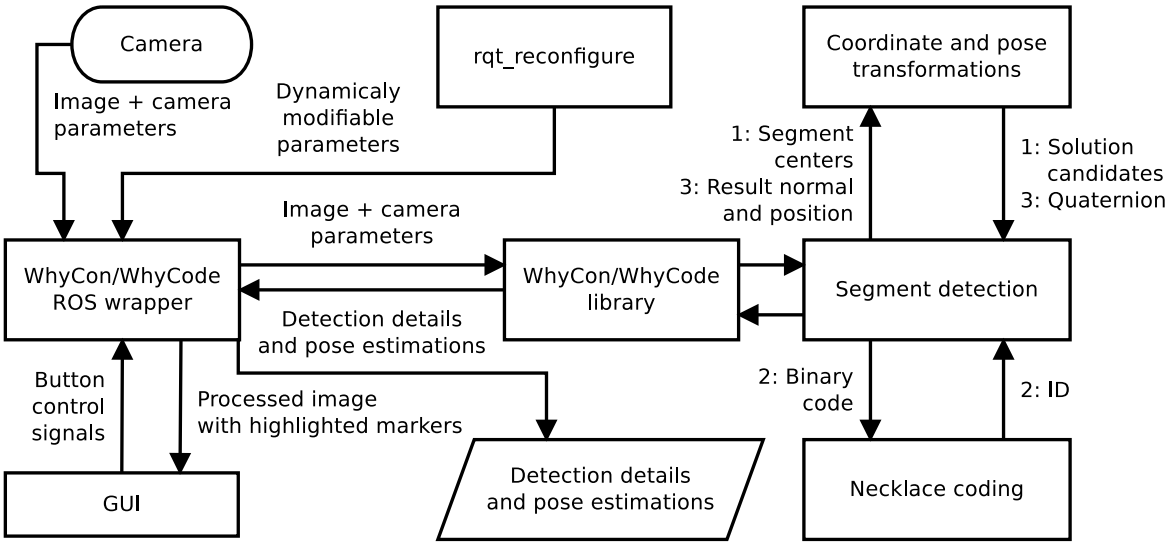


Figure 19: WhyCon/WhyCode simplified system diagram in ROS

The whole system is meant to be easily used by newcomers and at the same time, offer high user-configuration for advanced users. Therefore, the threshold parameters for segment validity tests are possible to change from the default values. There are two options to achieve it; one can adjust the values on-the-fly through the dynamic reconfigure server or modify the default values directly in the configuration file. It is possible to dynamically scale the diameter, set how many markers to look for, and toggle whether to start

identification of the ID. The user can also comfortably input a path to custom coordinate system calibration files or save a screenshot. Other settings which are unchangeable during the runtime are available at the program start-up either temporally as a command-line argument or persistently in the launch file.

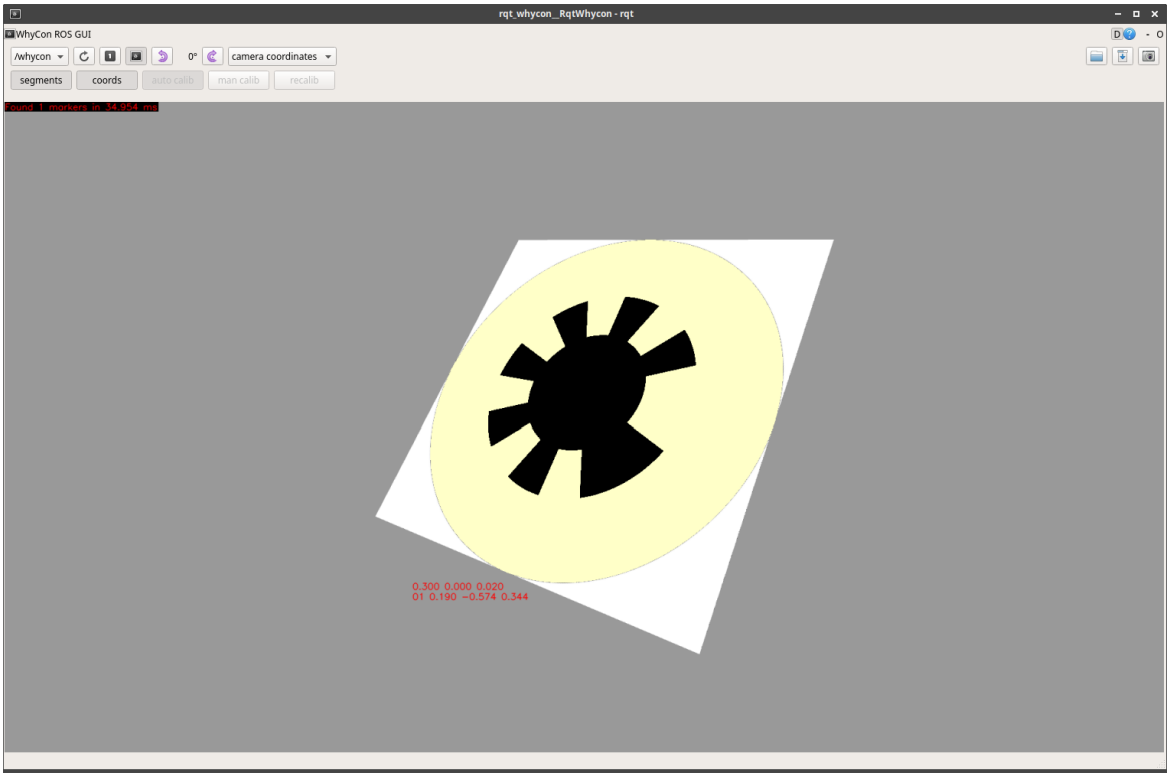


Figure 20: WhyCode's rqt based user interface

3.2.1 Quaternion from the surface normal vector

Once the means of data exchange within the framework were established, we had to incorporate the conventions for units and data representations. Thus, to express the orientation of the marker, we decided to use the quaternions to avoid unnecessary conversions. Quaternions are widespread across many fields as in robotics where they can represent the rotation and orientation in the 3D space [54]. They are also implementation-wise beneficial compared to the rotational matrices because they are more numerically stable and are formed by only four elements as they extend the complex numbers. However, for better user interpretation of the resulting marker's pose, the quaternion is also converted to the Euler angles.

The output quaternion is obtained from the surface normal vector $\mathbf{n}$ and the rotational angle calculated from the encoded ID in two steps. In the case of WhyCon, we do not have

the ID, so we perform only the first step, which is results only in partial orientation. The quaternion is calculated through the axis-angle representation. Firstly, we need to know the axis of revolution which is obtained as the cross-product between the optical axis - null rotation - and the surface normal unit vector $\mathbf{n}$.

$$\mathbf{z}' = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \mathbf{n} \tag{5}$$

The angle of revolution is then the angle between $\mathbf{n}$ and $\mathbf{z}'$ and as they are unit vectors, we can omit the division by their norms

$$\theta = -\arccos(\begin{pmatrix} 0, & 0, & 1 \end{pmatrix} \cdot \mathbf{n}) \tag{6}$$

Thus, the first quaternion can be now formed

$$\mathbf{q_1} = \begin{pmatrix} z'_1 \cdot \sin(\theta/2)\ \mathbf{i} \\ z'_2 \cdot \sin(\theta/2)\ \mathbf{j} \\ z'_3 \cdot \sin(\theta/2)\ \mathbf{k} \\ \cos(\theta/2) \end{pmatrix} \tag{7}$$

The second one $\mathbf{q_2}$ is obtained similarly, where θ is the rotational angle from encoded ID, and the axis of revolution is the marker surface normal $\mathbf{n}$. Then, the final orientation is composed as:

$$\mathbf{q} = \mathbf{q_2} \circ \mathbf{q_1} = \begin{pmatrix} q_{2w}q_{1x} + q_{2x}q_{1w} + q_{2y}q_{1z} - q_{2z}q_{1y}\ \mathbf{i} \\ q_{2w}q_{1y} - q_{2x}q_{1z} + q_{2y}q_{1w} + q_{2z}q_{1x}\ \mathbf{j} \\ q_{2w}q_{1z} + q_{2x}q_{1y} - q_{2y}q_{1x} + q_{2z}q_{1w}\ \mathbf{k} \\ q_{2w}q_{1w} - q_{2x}q_{1x} - q_{2y}q_{1y} - q_{2z}q_{1z} \end{pmatrix} \tag{8}$$

where $\circ$ is the Hamilton product which is based on the distributive law and the product of the basis elements as above, q_w denotes to the real part, and q_x, q_y, q_z are the imaginary parts.

4 Datasets

To compare the modified methods with the other fiducials, they were evaluated on both real-world and artificial simulation-based datasets. All the markers achieve a high and up to millimetre precision, and some of them were primarily designed to provide a ground-truth for mobile robot experiments. Therefore, the problem araised as many other ground-truth systems provides similar accuracy. Thus, the simulated artificial environment was built to verify the functional and reliable aspects of the localisation in a scalable and noise-free environment. However, the methods were also evaluated on the real-world dataset to demonstrate their actual performance under a naturally occurring noise and imperfect image frames.

4.1 Real-world dataset

The real-world dataset represents the external localisation application of the markers in swarm robotics for providing the low-cost ground-truth system. This application of the fiducials is popular [55, 56, 57] as the swarms can then operate in various conditions without the need of setting up complex localisation systems of multiple cameras and synchronisation units. Also, their sensors do not interfere with the infrared light emitted by the localisation systems. The swarm arena in the dataset is captured by an off-the-shelf USB camera recording approximately 2.5 m above the arena. MONA platform [58] is used to simulate a swarm as it is a small open-source, open-hardware, and affordable platform for research and education. It is based on the microcontroller ATmega 328, which is the same as in Arduino Mini/Pro; thus, there is enormous compatibility of libraries and sensors.

The robots are equipped with a board on top with four fiducials, AprilTag, ArUco, WhyCode, and SwarmCon in the corners as in Figure 21. All the markers have the same size of 4 cm and have been generated with their reasonably small set of IDs. Thus, SwarmCon generated 3×3 unique ellipses, ArUco binary matrix IDs are from the 4×4 dictionary, WhyCode markers have 6-bit encoding, and AprilTag identifies markers with $16h5$ ID set. The video stream has the resolution of 1920×1080 and a frame rate of approximately 30 frames per second due to occasional frame dropping while saving the stream. The Vicon localisation system obtains the ground-truth measurements as it can provide a submillimeter precision.

Two different datasets were created based on the mentioned set up, MONA_1 and MONA_3. Dataset MONA_1 contains only one robot moving around the arena centre while it holds markers with ID 1. The video length is one minute and seven seconds, consisting of 2016 frames. The second one added other two robots with IDs 2 in the left and 9 in the right part of the arena. The dataset is a bit longer with 2062 frames and a minute and eight seconds.

(a) MONA_1 dataset (b) MONA_3 dataset

Figure 21: Sample frames from the real-world datasets of swarm arena

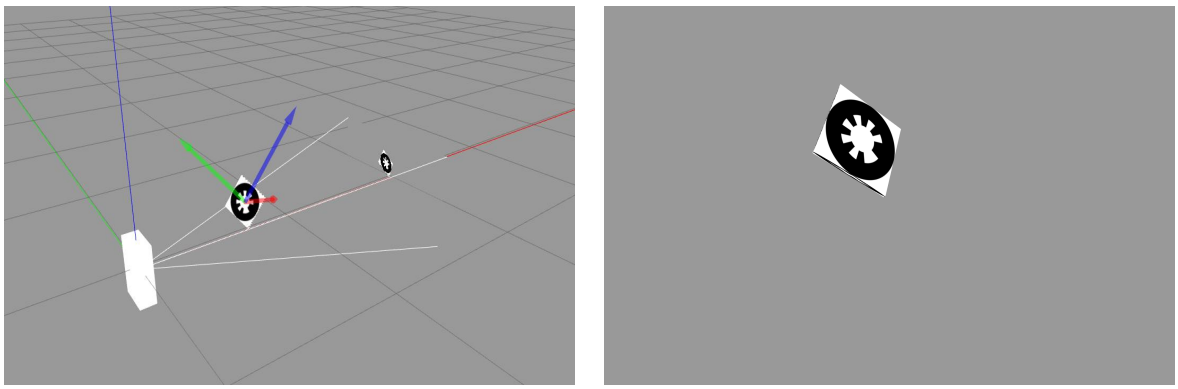
4.2 Artificial dataset

The artificial dataset was made in the Gazebo [59] simulator framework because it provides a precise and reliable environment simulation with high ROS integration. The framework offers a sophisticated physics engine, user-friendly interface, high modularity and easy-to-use premade packages. We created three testing worlds to evaluate localisation scenarios - ego-localisation, 2D external localisation and 3D external localisation. The simulation environment is controlled by central coordinator ROS nodes, which can repeatedly run individual scenarios of marker position and orientation, and they provide the synthetic camera image with stable parameters. This node initiates the whole testing framework and then sequentially send commands to the Gazebo action server to change either the camera or marker position. The camera is set up with default parameters and without any added noise and distortion for stable measurements. The fiducials are sized to have the same width varying depending on the scenario.

The Gazebo words settings are based on the default empty world with a few modifications. The RGBA ambient light value was increased to the maximum value in order to allow uniform illumination and to evaluate markers without the aspect of undergoing different light conditions. Also, to make the simulated environment stable, the gravity force parameters were zeroed, so the camera or the markers do not slide away after displacement. The camera is represented as a model with added standard sensor plugin *gazebo_ros_camera* and resolution of 1920×1080. Individual markers are added to the world through the visual parameter of a simple box-sized model. The markers were generated to provide enough unique variants to cover all the simulator scenarios; thus, we need a total of 28 markers. So for AprilTag we chosen the 16h5 ID family of 30 markers, ArUco IDs are from the 4×4 dictionary of 50 markers, and WhyCode markers were generated with 8-bit encoding resulting in 30 unique markers.

4.2.1 3D external localisation scenario

The environment layout is simple as there are only a camera and one marker present, see Figure 22. The marker size is $20\,\text{cm} \times 20\,\text{cm}$ as it is approximately the size one can print on an A4 paper; thus, the simulation and the results are more comfortable to imagine and interpret. The camera is arbitrarily positioned at the origin of the coordinate system. The marker is then randomly placed in the field of view as needed. The random distribution of displacement is uniform and ranges in the distance from $3\,\text{m}$ to $5\,\text{m}$, the vertical axis from $-1.5\,\text{m}$ to $1.5\,\text{m}$, and in the horizontal axis from $-1.5\,\text{m}$ to $1.5\,\text{m}$. The orientation of the marker varies similarly; thus, the rotation around the marker surface normal goes from $0\,\text{rad}$ to $2\pi\,\text{rad}$. In terms of the rotation around the two other axes, we can choose between small and large angles; therefore the both angles are either from the interval $[-\pi/6, \pi/6]$ rad or from $[-\pi/3, -\pi/6] \cup [\pi/6, \pi/3]$ rad respectively. This scenario allows us to test the fiducial markers at various positions and even extreme angles which would be difficult in the real world because it would require expensive hardware to provide the ground-truth.



(a) (b)

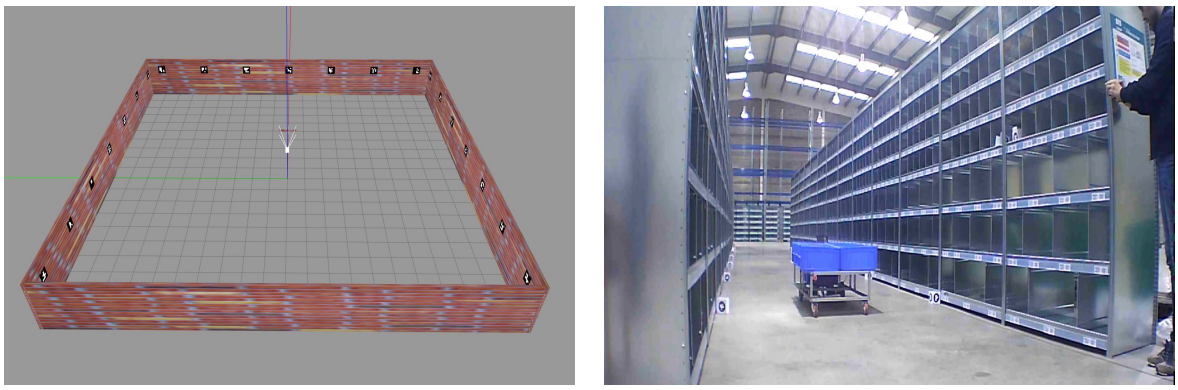
Figure 22: 3D external localisation simulator of WhyCode where (a) is the simulator models layout and (b) is the image produced by the camera sensor.

4.2.2 Ego-localisation scenario

The purpose of this scenario is to test the fiducial marker usefulness for the 2D onboard localisation of a mobile robot. It has been inspired by a project with Synergy Logistics Ltd., which is a UK-based warehouse and logistic company. They requested an absolute onboard localisation system based on fiducial markers placed around the whole warehouse. The reason why one could consider this application of the markers is their uniqueness in the environment compared to the racks or concrete walls. The traditional SLAM methods based on the extracted image features or lidar point clouds are powerless in an environment with such repetitive and plain characteristics causing unexpected loops and feature mismatching. The project outcome was a ROS-based system that can perform absolute localisation by

fitting the observed markers to a global map and then fusing the visual odometry with the wheel odometry by the extended Kalman filter.

To perform the ego-localisation tests, we created a testing square arena with a camera and markers on the border walls as in Figure 23. The dimension of the arena is 19.6 m × 19.6 m. There are 28 markers, seven on each wall, equally spaced by 3 m, and their size is 50 cm in diameter. This scenario is for testing the kidnapped-robot problem by randomly changing the robot planar position and orientation within the arena. The robot then observes the markers at different position and orientation, and it has to localise itself in a given map using standard methods. The markers are matched by their IDs to create the correspondence pairs. The robot is reduced only into the camera sensor because there is no reason to simulate the whole robot in this scenario. The random position is uniformly distributed from −8 m to 8 m along both planar axes. The rotation around the vertical axis goes also uniformly from $-\pi$ rad to π rad.



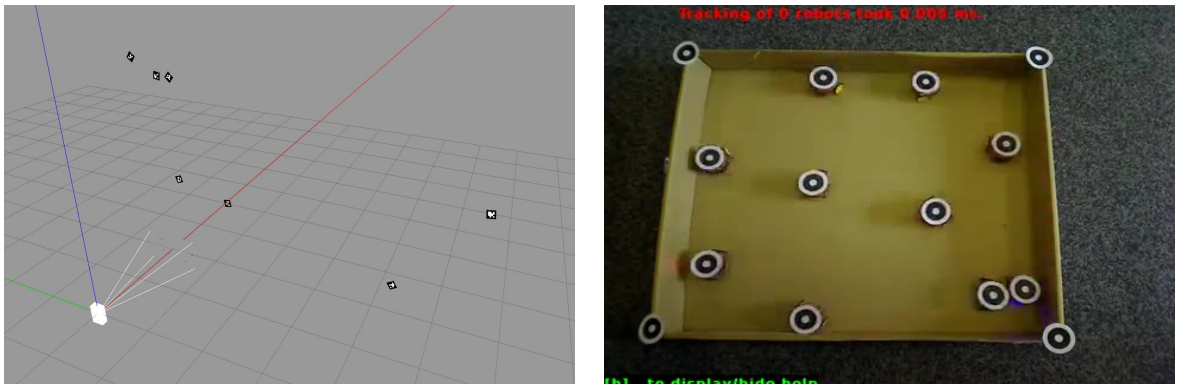
(a) (b)

Figure 23: Ego-localisation overview where (a) is the simulator of ArUco models layout and (b) is the application of WhyCode in a warehouse at Synergy Logistics Ltd. where the markers are placed on the ground along the racks.

4.2.3 2D external localisation scenario

The 2D external localisation scenario is simulated as a planar multi-robot swarm, see Figure 24, where the markers are meant to provide the pose of the robots as ground truth. This scenario is similar to the real-world one; thus, we can evaluate the method performances on image frames without any noise or distortion, and the markers can achieve the best theoretical precision. As it is often in mobile swarm robotics, we estimate the two-dimensional position and heading of the robots only. In the simulator, there are seven fiducials representing the actual robots and their size is the same as in the 3D external localisation scenario, 20 cm. The markers are positioned randomly with a uniform distribution in a non-overlapping way over the arena. The actual movement of the robots is not simulated continuously. They are instead placed in a grid with 50 cm cells in order to avoid

any possible interaction between the markers. The dimensions are 9 m × 5 m and all the markers in the same perpendicular plane at the distance of 5 m. They are also randomly rotated around their surface normal axes by the angle in the range from $-\pi$ rad to π rad. The camera is centred above $[0, 0]$ position of the robot plane, so it can evenly observe the operational space.



(a) (b)

Figure 24: 2D external localisation overview where (a) is the simulator of AprilTag models layout and (b) is the application of WhyCon to track AMiR microrobot platforms [60] in robotic swarm.

5 Experiments

In this chapter, we evaluate the introduced fiducial marker methods on both real-world and artificial datasets for their quality of robustness and localisation. While localising by a real camera, there is always noise present, and we have to rely on that the camera has been calibrated properly and the environment does not change in a way to affect the lens characteristics. There is also a marker deformation by the motion blur or non-uniform illumination conditions. All of those aspects add their effect on the produced image frame, which is then processed by the fiducial marker methods. Therefore, we assume the results will differ between the simulator and real-world.

In order to determine the quality of localisation methods and compare them with each other, we established the following criteria. We start with the real-world dataset, on which we firstly examine the detection *reliability* of the methods mentioned above. Other two more tests are done on the swarm dataset, *2D and 3D position* errors. The artificial dataset scenarios are all tested in two ways, position and orientation errors. The 3D external localisation scenario is evaluated in the full *6 degrees of freedom* as it provides a synthetic way to test many random poses. The 2D external localisation scenario tests only the *planar position and orientation* error. Then, we test the ability to maintain the planarity of data, so we compare how is the *distance* estimation precise. The last two criteria are related to the ego-localisation of a mobile robot; thus, we observe the 2D position and yaw error of the camera. Those criteria are mainly influenced by real robotic and experiment-evaluation applications. Since there are many measurements obtained from each dataset, it might not be clear which method is better; and therefore, we compare the results also by statistical testing.

5.1 Statistical evaluation

Student's t-test is applied for testing whether the resulting mean errors of one marker differ significantly from the results of the others. To make it possible for the t-test to be used, we have to assume that the tested samples have a normal distribution; however, we cannot be sure. In that case, we apply the central limit theorem as an unknown distribution approximates the normal distribution with an increasing number of samples. We use sample sets large enough with varying size from over 2000 to 3000 samples depending on the tested criteria. We test the paired samples t-test where the random variable represents the localisation errors associated with the same image frame.

The paired samples t-test can be transformed into the one-sample t-test. We need to lable samples from one group as u_i and from the second group as v_i, then we can construct a new random variable $x_i = u_i - v_i$ with the mean value of $\mu = 0$. Testing statistics is then calculated as

$$T = \frac{\bar{X} - \mu_0}{S}\sqrt{n}, \tag{9}$$

where $\bar{X}$ is the sample mean value of the random variable X, S is its sample standard deviation, μ_0 is the mean value of the normal distribution and n is the size of the sample set. The null hypothesis is that the mean value of the normal distribution of this random variable is equal to zero when testing on two-sided confidence interval. It is rejected if the sample mean differs from zero a lot, resulting in considering the paired distributions does not have equal means. We may also test on the one-sided confidence interval; thus, the null hypothesis is $H_0 \colon \mu_u \leq \mu_v$, which, when rejected, can provide an ordering of the measured mean values. All the tests were done with the confidence level set to $\alpha = 5\%$.

In the experiments, we also statistically test the mean errors of orientation estimations. However, the mean and standard deviation cannot be calculated regularly as in terms of position because the angles are circular quantities. Thus, the directional statistics approach has to be applied [61]. The sample mean angle $\bar{\theta}$ is calculated as

$$\bar{\theta} = \arctan2\left(\frac{1}{n}\sum_{i=1}^{n}\cos\theta_i, \frac{1}{n}\sum_{i=1}^{n}\sin\theta_i\right), \tag{10}$$

where n is the number of samples, and θ_i is the small-angle difference between the measured angle samples. The sample standard deviation is obtained as

$$\begin{aligned} S &= \sqrt{-2\ln R} \\ R &= \left\| \frac{1}{n}\sum_{j=1}^{n}(\cos\theta_j + i\sin\theta_j) \right\|, \end{aligned} \tag{11}$$

where the the small angle θ_j is represented as a unit complex number, and R is the length of the resulting vector.

5.2 Data generation

All experiment measurements are generated on Linux operating system Ubuntu 18.04.4 LTS with Hardware Enablement stack enabled, thus running with kernel 5.4. The ROS framework is used in distribution *Melodic* which applies on all dependent packages which were upgraded before conducting the experiments. The artificial dataset simulator Gazebo is in the version 9.0.0. All the codes required for experiments are written in C++11 and compiled with standardly used optimisation flag *-O2*. AprilTag and ArUco are used in their open-source implementations in computer vision library OpenCV 3.4.10 as it is the most up to date version supported in the ROS. They run with default parameters apart from the *minMarkerPerimeterRate*, which is set to 0.005 to lower the minimum number of marker perimeter pixels to be considered for detection. It then allows detecting square markers even when they are placed far away from the camera. The same applies to WhyCode; it is linked to the same version of OpenCV, and the parameter *minSize* is set to 10.

Hardware set up is simple laptop Dell Latitude E7250 with processor Intel Core i5-5300U, 16 GB RAM, and 256 GB mSATA SSD. It turned out this set up was a little

limiting in terms of the number of available cores because each ROS node is at least one process, and thus it would be more efficient to run on more cores. Since the simulator was run with graphical interface turned off, there was no significant limitation from the GPU point.

5.3 Real-world tests

In those tests, we let the methods to perform external localisation of moving robots as they can substitute for high-end localisation devices. All the methods were provided with the same 2016 frames (MONA_1) or 2062 frames (MONA_3), camera parameters and the size of the marker. However, the Vicon system, which measures the ground truth, provides measurements at a higher frequency, approximately three times more than the camera frame rate. This issue occurs to be shared in high precision systems, and as of that, we at first had to match the ground truth with the frames. It was done by aligning the beginning and the end of the data with the video stream, then we iterated over the video frames and assigned them the closest Vicon measurements. In order to make the data comparable, we also have to reduce the number of samples, so there will be corresponding pairs for each video frame. Having data correspondence, we could evaluate the results using the criteria below.

Firstly, we examined if the methods are capable of providing a measurement for every frame and if not, then what is the number of dropped frames. This primary criterion was chosen because, for the applications of providing the ground truth, one needs a stable and robust method which can measure continuously throughout the experiment. Together with the continuity, the number of outliers is another vital parameter to look at, because in case of high outlier rate, we might rather consider dropping the uncertain frames instead of providing a misleading measurement. To determine the outliers, we evaluated the data manually, see Figure 25.

Based on the results in Table 1, we can say about dataset MONA_1 that SwarmCon provides the most stable results. In terms of square fiducials, ArUco dropped more than four times fewer frames and produced fewer outliers than AprilTag. The roundel markers, compared to the rectangular ones, can measure across more frames continuously, and the difference is by order of magnitude. WhyCode benefits from the roundel shape in the dropped-frame criterion; however, due to the nature of encoding the ID, it cannot outperform SwarmCon because the teeth are not visible enough. Also, we have to conclude that the dataset MONA_3 is unfortunately not suitable for thorough testing as the square markers are not able to produce stable and continuous measurements. AprilTag performs especially poorly because it cannot detect all three moving robots in more than 16 frames. Almost half of the data had to be dropped in case of ArUco from the same reason. After examination of the video stream, we assume that the different illumination in MONA_3 dataset caused more challenging conditions for edge detection and following steps in the square marker methods. Thus, we proceed to the next criteria only with the MONA_1;

otherwise, we would not have enough measurements for statistical tests.

	MONA_1		MONA_3
	Dropped frames	Outliers	Dropped frames
AprilTag	621	1	2046
Aruco	149	2	976
WhyCode	14	1	265
SwarmCon	0	0	4

Table 1: Evaluation of detection reliability on the real-world dataset

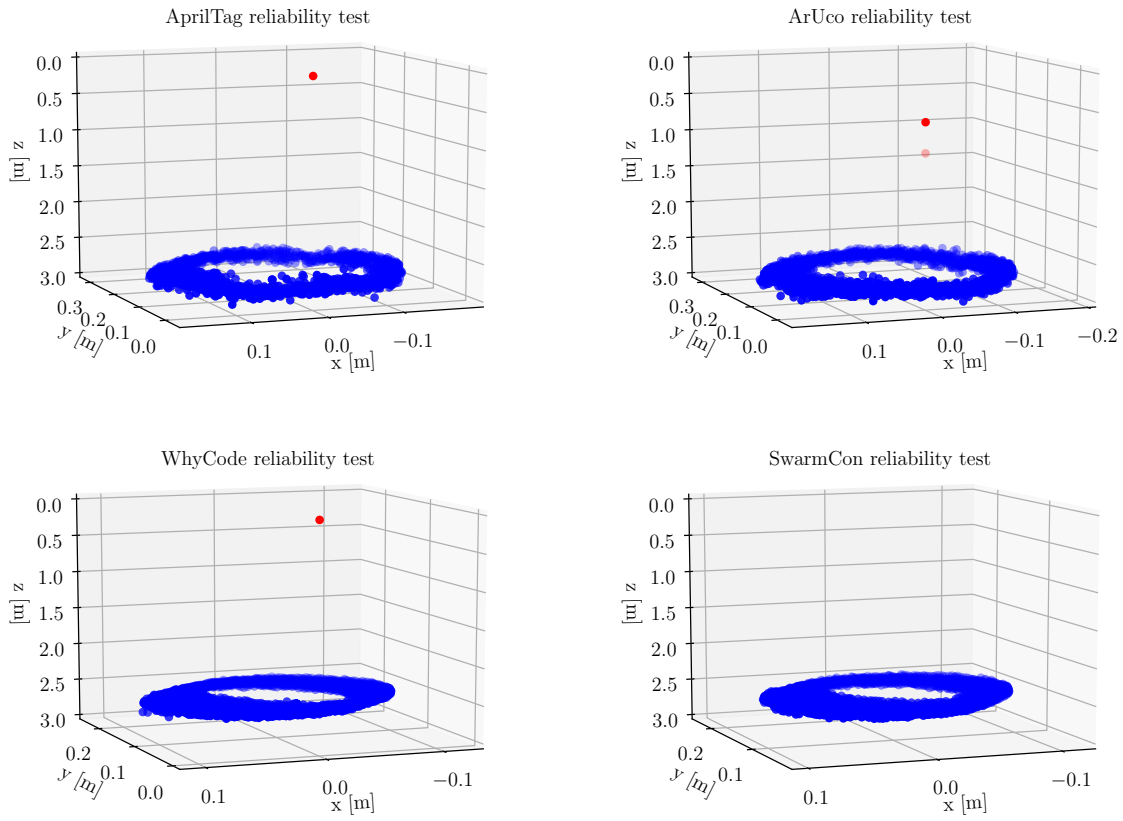


Figure 25: Outlier evaluation on the real-world dataset. Points are displayed in canonical camera coordinates where x is the right-left axis, y is the down-up axis, and z is the distance of the image plane. The correct data are displayed in blue and outliers in red.

Further, we focus on the ability to estimate the position in both 2D and 3D precisely. Those two criteria are fundamental for the fiducial markers deployment as ground truth in the experimental evaluation. We start with the 2D position localisation assuming SwarmCon will score the best as it did not drop a single measurement. Based on the p-value

result in Table 2, we reject all the null hypotheses for 2D apart from testing WhyCode with SwarmCode in both one-sided and two-sided tests. Therefore, we can make only a partial assumption of how to sort the markers because $\mu_A rUco > \mu_A prilTag$ and they are higher than in the case of roundel markers. However, we cannot claim anything about the relation between the mean errors of WhyCode and SwarmCon.

In terms of the 3D position evaluation, we can make assumptions of the results based on the 2D tests. We expect the roundel markers will outperform the square ones. However, this was verified only for AprilTag and SwarmCon, but we cannot distinguish between WhyCode and ArUco. Thus, SwarmCon marker outperformed the others in both the 2D and 3D tests.

H_0	p-value	
	2D	3D
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00	0.93
$\mu_{WhyCode} = \mu_{SwarmCon}$	0.46	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.00	0.00
$\mu_{AprilTag} = \mu_{SwarmCon}$	0.00	0.00
$\mu_{ArUco} = \mu_{SwarmCon}$	0.00	0.00

(a)

H_0	p-value	
	2D	3D
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00	0.00
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00	0.46
$\mu_{SwarmCon} \leq \mu_{WhyCode}$	0.77	0.99
$\mu_{ArUco} \leq \mu_{AprilTag}$	0.00	0.99
$\mu_{AprilTag} \leq \mu_{SwarmCon}$	0.00	0.00
$\mu_{ArUco} \leq \mu_{SwarmCon}$	0.00	0.00

(b)

Table 2: Real-world dataset results of mean 2D and 3D position errors. Based on the rejected hypotheses, the means can be in ascending order as (2D) {SwarmCon, WhyCode} < AprilTag < ArUco and (3D) SwarmCon < {WhyCode, ArUco} < AprilTag.

5.4 3D external localisation

The 3D external localisation scenario can be considered as the synthetic testing of estimating the position and orientation. We would not usually encounter such extreme situations as were generated by the simulator, but we cannot neglect them in testing because they can stress well the methods. The motivation to perform this experiment is simple as it is the necessary examination of the overall marker system. The marker has to at first detect the shape, then calculate the segment position, and also correctly decode the identification within. Even though we consider only the position and orientation error, all mention is reflected in how well the system can serve its general purpose. During the experiment, we encountered an issue of the enormous simulated state space. The random draws of orientation angles were set to cover the maximum range in the closer distance; however, when the markers were further away, the detection was almost impossible. In order to provide better inside of the orientation estimation, we split the experiment into two parts which differ by the range of random angles. The small-angle test has boundary values $\pm\pi/6\,\mathrm{rad}$, while in the large-angle test the values are $\pm\pi/3\,\mathrm{rad}$.

To accurately compare quaternions, we have to choose a proper and easily explainable criterion on them. In [62], the author presents several metrics for orientations in 3D.

$$\begin{aligned} &\Phi\colon S^3 \times S^3 \to \mathbb{R}^+ \\ &\Phi(\mathbf{q_1}, \mathbf{q_2}) = \arccos |\mathbf{q_1} \cdot \mathbf{q_2}|, \end{aligned} \tag{12}$$

where $S^3 = \{\mathbf{q} \in \mathbb{R}^4 | \|\mathbf{q}\|^2 = 1\}$, $\cdot$ is a dot product and since Φ has to be non-negative, we restrict the range of values to $[0, \pi/2]$ radians. Applying this metric on quaternions outputs the smallest rotation angle required to get from one orientation to another.

Both the scenarios were tested on 3000 samples which were then filtered from the misidentified or undetectable cases. Despite the consideration of all potential issues previously mentioned, all methods managed to provide estimation for each small-angle test. In the large-angle test, we had to drop 348 records due to both reasons. Wrong ID decoding happened to WhyCode in 3 tests. ArUco did not detect the marker in 66, AprilTag 72, and WhyCode in 334 cases.

One could assume that WhyCode would produce worse results than the square markers as it could not detect the marker in more cases. In terms of the position, it was naive to claim that as based on Table 3. The position precision is in favour of WhyCode, see Figure 26. We are even able to reject all the interval tests; thus, we can order the mean error values ascendingly as WhyCode $<$ ArUco $<$ AprilTag. Those relations maintains for bot large and small angle testing; thus, we can make further deductions. WhyCode's ability to reliably estimate the position better than the others is not tied to the observation angles.

In order to compare the angles between orientations, we have to apply the directional statistics as described before. Once the mean angle is compared as in Table 4, we can see the original assumption was not entirely wrong. During the small-angle test, WhyCode achieved the smallest mean error allowing us to claim it is the most precise marker in terms of orientation within the angle of observation between $\pm\pi/6$ rad. The square markers scored almost similarly, but now it is AprilTag which provides a better result than ArUco. However, when the maximum angle was increased to $\pm\pi/3$ rad, the results changed drastically as in Figure 27. The orientation errors for the square markers are undecisive, and they are also reaching very high errors, but their mean values maintain small. Even though WhyCode has a lower maximum error, it cannot overweight the mean to be smaller as one would not guess from the histogram. Thus, we can only partially order the fiducials by precision as WhyCode lost its position to the AprilTag and ArUco.

5.5 2D external localisation

This test is similar to the real-world 2D position test because it is also focused on the planar localisation ability. The difference from the real-world test is that we evaluate on noise-free image frames, perfect intrinsic parameters and no distortions. Also, the markers are not moving but being placed randomly, which affects the computational requirements

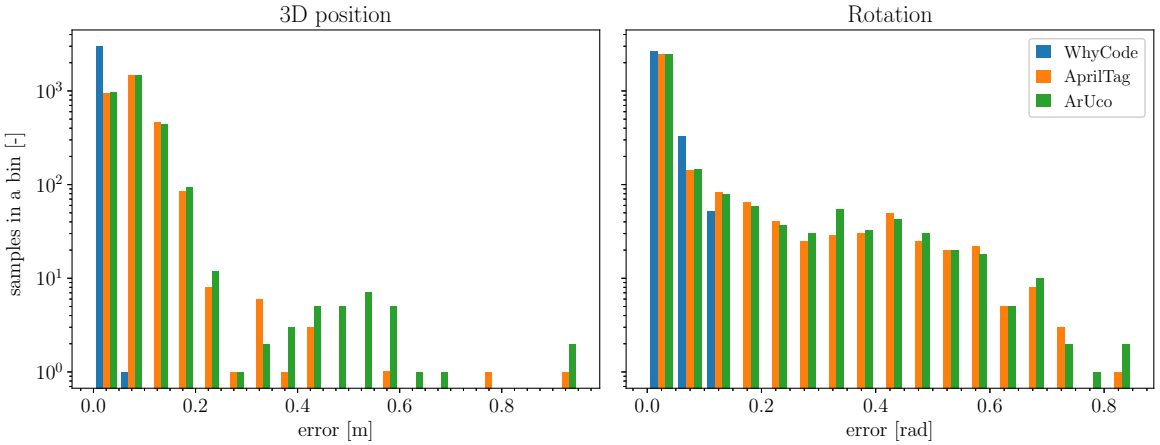


Figure 26: Histograms of 3D external small-angle localisation errors with 5 cm bins for position and 0.05 rad bins for angles. Vertical axis is in logarithmic scale of decadic base

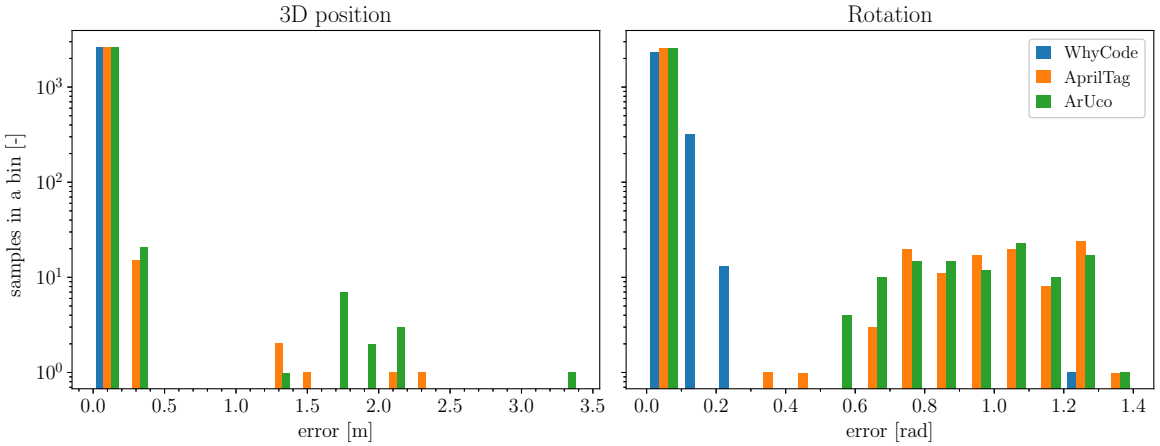


Figure 27: Histograms of 3D external large-angle localisation errors with 20 cm bins for position and 0.1 rad bins for angles. Vertical axis is in logarithmic scale of decadic base

as some of the markers always process the whole image, and the others tend to track the markers through frames to improve the detection time. The results will give us more understanding if the fiducials are suitable for the applications of swarms or multi-robot systems. Since one needs only a camera and printed paper marker, the fiducial markers are especially useful for easy and portable ground-truth localisation system for evaluation of experiments.

The simulated experiment was done over 3000 different planar positions and orientations of seven markers. All the marker systems were able to perform successful detection and provide pose estimation. However, as the testing framework is asynchronous, and we were

H_0	p-value	
	small	large
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.00	0.03

(a)

H_0	p-value	
	small	large
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00	0.00
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00	0.00
$\mu_{ArUco} \leq \mu_{AprilTag}$	0.00	0.02

(b)

Table 3: 3D external localisation results of mean 3D position errors. Based on the rejected hypotheses, the means can be in ascending order as WhyCode < ArUco < AprilTag for both small and large angle of observation.

H_0	p-value	
	small	large
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.10	0.70

(a)

H_0	p-value	
	small	large
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00	0.99
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00	0.99
$\mu_{ArUco} \leq \mu_{AprilTag}$	0.05	0.65

(b)

Table 4: 3D external localisation results of mean orientation errors. Based on the rejected hypotheses, the means can be in ascending order as (small angle) WhyCode < AprilTag < ArUco and (large angle) {ArUco, AprilTag} < WhyCode.

able to change the pose of only one marker at the time, it happened that a few data samples were incorrectly recorded with previous reference positions. So, it was necessary to verify the data manually for such errors.

In this test, we simply omitted the third distance coordinate and only focused on the 2D position error. Based on the previous results in the experiment on the real-world dataset, we expect the differences between individual markers will be more visible, but still similar. Thus, as summarised in Table 5, we can notice one change. This time, the mean error of AprilTag is higher than the ArUco's, which may be an effect of the noise-free generate images and perfectly known camera parameters. WhyCode maintains the previous results and has the smallest mean error.

In the angle measurements, we did not calculate the angle between orientations as in the 3D external localisation experiment, but we only considered the angle of revolution around the surface normal. In the light of the 3D external localisation results, we may make an assumption, that the WhyCode will not score well because the markers can be observed under a higher angle than $\pm\pi/6$ rad. We can see that the hypotheses about AprilTag and ArUco are undecisive; however, they are about WhyCode, see Table 6. Contrary to what we assumed, the WhyCode's planar angle estimation turned out to be the most precise than the others. As the square markers are based on a similar approach, it is not surprising they achieve results so close to each other.

Distance error is another essential criterion determining how well the methods can maintain planarity of the measurements. The inspiration comes from an application in a swarm and multi-robot systems, where the robots have to keep a certain distance from others or keep a formation. To calculate the error, we took the absolute difference between the estimated distance and the marker plane, which was 5 m away from the camera. In Table 5, the null hypotheses of the mean equality were rejected together with the one-sided interval test. Having such straight-forward results, we can order the fiducials from the smallest mean distance error as WhyCode < ArUco < AprilTag.

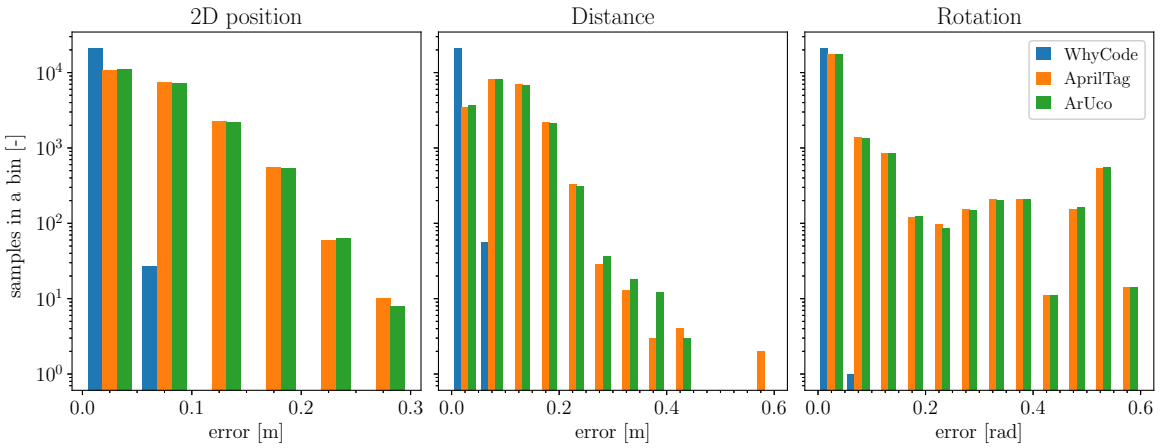


Figure 28: Histograms of 2D external localisation errors with 5 cm bins for position and 0.05 rad bins for angles. Vertical axis is in logarithmic scale of decadic base

H_0	p-value	
	2D	distance
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.00	0.00

(a)

H_0	p-value	
	2D	distance
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00	0.00
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00	0.00
$\mu_{AprilTag} \leq \mu_{ArUco}$	0.00	0.00

(b)

Table 5: 2D external localisation results of mean position errors. Based on the rejected hypotheses, the means can be in ascending order as WhyCode < ArUco < AprilTag for both 2D and distance.

5.6 Ego-localisation

In this experiment, we created a simplified testing environment of the randomly positioned camera and used the standard localisation methods to estimate the camera's 2D

H_0	p-value
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.98

(a)

H_0	p-value
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00
$\mu_{ArUco} \leq \mu_{AprilTag}$	0.49

(b)

Table 6: 2D external localisation results of mean angle errors. Based on the rejected hypotheses, the means can be in ascending order as WhyCode < {ArUco, AprilTag}.

position and orientation. We performed 3000 random tests where 2767 were successful. During the experiment, there were such positions of the camera where the methods could not detect enough markers correctly, or they identified more markers with the same ID resulting in wrong correspondence pairs and erroneous results. We decided to filter the data in a way that would be possible in the localisation pipeline, thus tests with the same multiple IDs or a higher reprojection error than 0.5 m were dropped. Most of the omitted samples were produced by WhyCode, which was the only one producing duplicities or wrong IDs in 83 cases. There was even a higher number of excluded tests in case of the reprojection error which for WhyCode was 90 and ArUco 5 tests. The third case of dropping the measurements was when the methods could not detect the markers, thus not providing enough information for the localisation. This case happened to all of them, AprilTag and ArUco behaved the same, and they failed in 55 samples each, while WhyCode scored a bit better with 47 dropped samples.

In Table 7a, we can see all the null hypotheses testing if the mean errors are equal were rejected. This observation is also possible to make based on the histograms of the position errors in Figure 29. Therefore, the one-sided null hypotheses were tested as summarised in Table 7b and based on the p-value results, we rejected them also. Thus, the mean error values can be ascendingly sorted as WhyCode < AprilTag < ArUco.

To test the angle precision, we considered the error as the small angle between the estimated and reference ones. We had to again apply the directional statistics as before. The angle precision test outcomes are in Table 8, and the results turned out to be similar to the position testing. Thus, it allows us to order the mean angle errors in ascending order as WhyCode < AprilTag < ArUco.

Evaluating together the position and angle precision, we can conclude that WhyCode performs better than the square fiducials and should be considered in ego-localisation applications. However, as we noted before, it also produced the most unusable measurements, which included all three mentioned issues. So, even though it is more precise than the others, the users have to be aware of such drawbacks, which can be addressed as we did. In terms of the square markers, one would benefit from AprilTag because it produces less false positives and has smaller errors than ArUco.

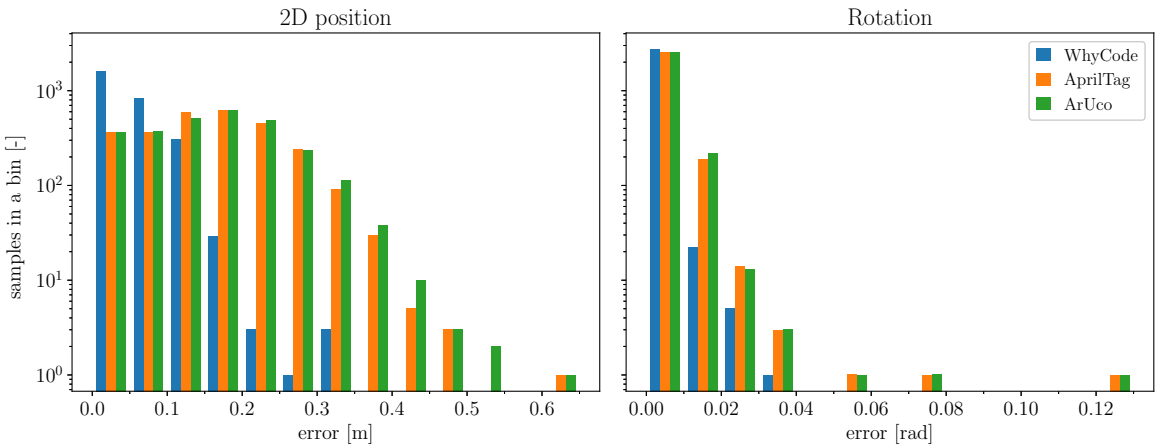


Figure 29: Histograms of ego-localisation errors with 5 cm bins for position and 0.01 rad bins for angles. Vertical axis is in logarithmic scale of decadic base

H_0	p-value
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.00

(a)

H_0	p-value
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00
$\mu_{ArUco} \leq \mu_{AprilTag}$	0.00

(b)

Table 7: Ego-localisation results of mean 2D position errors. Based on the rejected hypotheses, the means can be in ascending order as WhyCode < AprilTag < ArUco.

H_0	p-value
$\mu_{WhyCode} = \mu_{AprilTag}$	0.00
$\mu_{WhyCode} = \mu_{ArUco}$	0.00
$\mu_{AprilTag} = \mu_{ArUco}$	0.00

(a)

H_0	p-value
$\mu_{AprilTag} \leq \mu_{WhyCode}$	0.00
$\mu_{ArUco} \leq \mu_{WhyCode}$	0.00
$\mu_{ArUco} \leq \mu_{AprilTag}$	0.00

(b)

Table 8: Ego-localisation results of mean angle errors. Based on the rejected hypotheses, the means can be in ascending order as WhyCode < AprilTag < ArUco.

5.7 Experimental results summary and discussion

To provide the final decision on the best marker for both external and ego-localisation, all measures tested above have to be considered 9. Relying only on one criterion can lead to significant failures as in real-world applications, one is never truly sure about the undergoing observation conditions. Thus, if a marker localisation system scores well in one of the experiments, it may lose in another.

	Ascending order of the mean errors
Real-world reliability	SwarmCon < WhyCode < ArUco < AprilTag
Real-world 2D position	{SwarmCon, WhyCode} < AprilTag < ArUco
Real-world 3D position	SwarmCon < {WhyCode, ArUco} < AprilTag
3D external position	WhyCode < ArUco < AprilTag
3D external small-angle orientation	WhyCode < AprilTag < ArUco
3D external large-angle orientation	{ArUco < AprilTag} < WhyCode
2D external position	WhyCode < ArUco < AprilTag
2D external orientation	WhyCode < {ArUco, AprilTag}
Ego-localisation position	WhyCode < AprilTag < ArUco
Ego-localisation orientation	WhyCode < AprilTag < ArUco

Table 9: Results summary of all tested criteria. The differences of markers in curly brackets are statisticaly insignificant on the confidence level $\alpha = 5\%$.

AprilTag outperformed ArUco in position and orientation when tested for 3D external localisation under the small angle of observation. However, when the marker was at a higher angle, the orientation estimation became more erroneous, and it only maintained its performance in the position. This test was also the only one when it managed to provide better results than WhyCode. AprilTag showed poor results in 2D external localisation which might imply that in case of such restricted environment where we care only about the 2D position and planar angle, so only partial information from the provided, it loses its overall precision. Its position estimation strength was shown in the ego-localisation experiment where the markers were not in the same distance, and thus, it could localise the closer ones better than ArUco. The suitable application for it would be situations where it is expected to observe the fiducials from various positions and higher angles.

ArUco followed mostly the results of AprilTag as they are both based on similar principals. There were many situations when the difference between those two was not clear like in the distance maintaining test or orientation both under the small and large angle. It was also verified that ArUco produces a lower error when localising markers at higher distances than AprilTag as in 2D external localisation. ArUco achieved more precise estimations than WhyCode in large-angle 3D external localisation or real-world 3D position. It seems it does not improve performance so much when the markers are closer to the camera as in ego-localisation and 3D external localisation experiments it had the worst results. Nevertheless, one should consider ArUco for extreme orientation estimation and possibly deployments with lower camera quality similar in the real-world experiment.

WhyCode was able to determine the 2D and 3D position the best from all the tested markers in the simulated scenarios. In the real-world test, it was beaten by SwarmCon and scored similarly to ArUco, which might be because the way the ID is encoded into the marker affects the segmentation. Despite the fact, WhyCode performed significantly better in the real-world dataset by detecting the markers in frames when the square-based systems did not. In terms of orientation, it cannot handle markers under higher angles of

observations compared to the winning precision in small-angle test. The most important issue one has to treat was the amount of the situations when misidentification happened, or the marker was not detected at all, which might lead to unreliable localisation when not addressed correctly.

Based on the evaluated and summarised criteria above for each marker, we managed to enhance WhyCode so it can compete with the other fiducial localisation systems. Even though its provided data might be erroneous or not continuous under extreme conditions, in most cases, it outperformed the state-of-the-art methods and assumed from the graphs, it even produced smaller maximum errors. Those aspects make WhyCode the best candidate to take into account for external or ego-localisation.

6 Conclusion

The purpose of this thesis was to introduce a new vision-based localisation system for the full 6 degrees of freedom estimation, real-time image processing and stable identification of multiple markers suitable for the external and ego-localisation. The system is based on the widely used method WhyCode, which is a black-and-white roundel marker with a binary-encoded ID within. The current state of the system does not perform well because of the untreated ambiguity in solutions of position and orientation estimations, which becomes more and more significant under higher observation angles. The mentioned uncertainty in solutions also affects the ability to identify individual markers reliably. Also, the implementation relies on obsolete libraries and do not allow dynamically or automatically updatable parameters like image resolution.

In order to solve the issues, the method was thoroughly examined to find sources of errors and potential parts for extensions. The improved method is based on considering both solution candidates and evaluation of their possible fitness to the pattern. We could not apply the same ambiguity decision to WhyCode as well as its underlying WhyCon algorithm. Thus, we had to develop two new criteria to determine which solution of the estimated pose to chose as we wanted to maintain the compatibility with WhyCon. Those introduced criteria improved the performance of the marker system in position estimation, which was even before considerably precise, and especially in orientation, meaning the marker is now able of the full 6 DOF pose estimation.

The system was not only enhanced as of precision, but also provided the extension to the ROS framework to broaden the range of possible applications. During the process, we eliminated the depreciated dependencies and transformed the codebase into a library with a thin wrapper around it. Having the software prepared, we then created a simulated environment able to run repetitively and automatically series of tests of three different scenarios to verify the improved marker is comparable with the state-of-the-art fiducials. To be able to assess the performance accurately, we also collected a real-world dataset of robotic swarm carrying markers to verify the ground truth possibilities.

We evaluated the precision of the markers in the scenarios as mentioned earlier, focusing on the individual aspects of each such as 2D external localisation or 3D orientation. Based on the experimental results, it was showed that WhyCode could successfully outperform the currently most popular markers in the majority of the proposed measures achieving the goal of raising it to at least comparable state.

In future works, it would be beneficial for the ego-motion application to change the segmentation method from the flood-fill search for each marker to one which can process the whole image as in [49]. It would allow introducing a new criterion of computational time as WhyCode is currently tracking the markers, so it is not comparable yet. Another issue is the misidentification, which could be approached either from not providing any information if not sure about the ID or, as it usually happens in extreme situations, one could set it to unknown and still produce partially correct position without orientation, which would

allow continuous tracking. From the user perspective, it is necessary to prepare the code base for the upcoming ROS2 or consider it for release as OpenCV contributor module.

7 References

[1] Tomáš Krajnık. *Large-scale mobile robot navigation and map building.* PhD thesis, Czech Technical University in Prague, 2011.

[2] Ros.org — powering the world's robots. `https://www.ros.org/`. Accessed: 2020-04-29.

[3] Johann Borenstein, Hobart R Everett, Liqiang Feng, and David Wehe. Mobile robot positioning: Sensors and techniques. *Journal of robotic systems*, 14(4):231–249, 1997.

[4] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme. Robust localization using relative and absolute position estimates. In *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1134–1140 vol.2, 1999.

[5] Frédéric Chenavier and James L Crowley. Position estimation for a mobile robot using vision and odometry. In *Proceedings 1992 IEEE international conference on robotics and automation*, pages 2588–2589. IEEE Computer Society, 1992.

[6] Billur Barshan and Hugh F Durrant-Whyte. Inertial navigation systems for mobile robots. *IEEE transactions on robotics and automation*, 11(3):328–342, 1995.

[7] Surachai Suksakulchai, Siripun Thongchai, D Mitchell Wilkes, and Kazuhiko Kawamura. Mobile robot localization using an electronic compass for corridor environment. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.*, volume 5, pages 3354–3359. IEEE, 2000.

[8] Tomáš Rouček, Martin Pecka, Petr Čížek, Tomáš Petříček, Jan Bayer, Vojtěch Šalanskỳ, Daniel Heřt, Matěj Petrlík, Tomáš Báča, Voječh Spurnỳ, et al. Darpa subterranean challenge: Multi-robotic exploration of underground environments. In *International Conference on Modelling and Simulation for Autonomous Systesm*, pages 274–290. Springer, 2019.

[9] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[10] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[11] AR Jimenez, F Seco, R Ceres, and L Calderon. Absolute localization using active beacons: A survey and iai-csic contributions. *Institute for Industrial Automation, CSIC Madrid*, 2004.

[12] Margrit Betke and Leonid Gurvits. Mobile robot localization using landmarks. *IEEE transactions on robotics and automation*, 13(2):251–263, 1997.

[13] Adam Chrzanowski and Gottfried Konecny. Theoretical comparison of triangulation, trilateration and traversing. *The Canadian Surveyor*, 19(4):353–366, 1965.

[14] Vicon — award winning motion capture systems. `https://www.vicon.com/`. Accessed: 2020-04-29.

[15] Pierre Merriaux, Yohan Dupuis, Rémi Boutteau, Pascal Vasseur, and Xavier Savatier. A study of vicon system positioning performance. *Sensors*, 17(7):1591, 2017.

[16] Optitrack - motion capture systems. `https://optitrack.com/`. Accessed: 2020-04-29.

[17] Bruce Carse, Barry Meadows, Roy Bowers, and Philip Rowe. Affordable clinical gait analysis: An assessment of the marker tracking accuracy of a new low-cost optical 3d motion analysis system. *Physiotherapy*, 99(4):347–351, 2013.

[18] Pti phoenix technologies 3d motion capture systems. `http://www.ptiphoenix.com`. Accessed: 2020-04-29.

[19] Handle™ — boston dynamics. `https://www.bostondynamics.com/handle`. Accessed: 2020-04-29.

[20] Nick Hawes, Christopher Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrova, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.

[21] Patrick Irmisch. Camera-based distance estimation for autonomous vehicles. Master's thesis, Technische Universität Berlin, 2017.

[22] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE, 2011.

[23] Maximilian Krogius, Acshi Haggenmiller, and Edwin Olson. Flexible layouts for fiducial tags. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[24] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[25] Francisco J Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and vision Computing*, 76:38–47, 2018.

[26] Tomáš Krajník, Matías Nitsche, Jan Faigl, Petr Vaněk, Martin Saska, Libor Přeučil, Tom Duckett, and Marta Mejail. A practical multirobot localization system. *Journal of Intelligent & Robotic Systems*, 76(3-4):539–562, 2014.

[27] Peter Lightbody, Tomáš Krajník, and Marc Hanheide. A versatile high-performance visual fiducial marker detection system with scalable identity encoding. In *Proceedings of the Symposium on Applied Computing*, pages 276–282, 2017.

[28] Jiří Ulrich, Peter Lightbody, Aaron Weinstein, Filip Majer, and Tomáš Krajník. Whycode: Efficient and versatile fiducial localisation system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) - Late-breaking result*. IEEE, 2018.

[29] Farshad Arvin, Tomáš Krajník, Ali Emre Turgut, and Shigang Yue. Cosϕ: artificial pheromone system for robotic swarms research. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 407–412. IEEE, 2015.

[30] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[31] Hans P. Moravec. Sensor fusion in certainty grids for mobile robots. In *Sensor devices and systems for robotics*, pages 253–276. Springer, 1989.

[32] Alberto Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(3):249–265, 1987.

[33] David Kortenkamp and Terry Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *AAAI*, volume 94, pages 979–984, 1994.

[34] Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 8(1-2):47–63, 1991.

[35] Sebastian Thrun and Arno Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 944–951, 1996.

[36] Josep Aulinas, Yvan R Petillot, Joaquim Salvi, and Xavier Lladó. The slam problem: a survey. *CCIA*, 184(1):363–371, 2008.

[37] Andrew J Davison and David W Murray. Simultaneous localization and map-building using active vision. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):865–880, 2002.

[38] John Leonard and Paul Newman. Consistent, convergent, and constant-time slam. In *IJCAI*, pages 1143–1150, 2003.

[39] Michael Montemerlo, Sebastian Thrun, Daphne Koller, Ben Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 593598, 2002.

[40] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015.

[41] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, 2017.

[42] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

[43] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014.

[44] Jakob Engel, Jörg Stückler, and Daniel Cremers. Large-scale direct slam with stereo cameras. In *Proceedings 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1935–1942. IEEE, 2015.

[45] David Caruso, Jakob Engel, and Daniel Cremers. Large-scale direct slam for omnidirectional cameras. In *Proceedings 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148. IEEE, 2015.

[46] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[47] 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*, 33(5):1255–1262, 2017.

[48] Rafael Muñoz-Salinas and Rafael Medina-Carnicer. Ucoslam: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. *Pattern Recognition*, 101:107193, 2020.

[49] Jan Faigl, Tomáš Krajník, Jan Chudoba, Libor Přeučil, and Martin Saska. Low-cost embedded system for relative localization in robotic swarms. In *2013 IEEE International Conference on Robotics and Automation*, pages 993–998. IEEE, 2013.

[50] William YC Chen and James D Louck. Necklaces, mss sequences, and dna sequences. *Advances in applied mathematics*, 18(1):18–32, 1997.

[51] Qt — cross-platform software development for embedded & desktop. `https://www.qt.io/`. Accessed: 2020-04-29.

[52] Shaowu Yang, Sebastian A Scherer, and Andreas Zell. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1-4):499–515, 2013.

[53] rqt_image_view - ros wiki. `http://wiki.ros.org/rqt_image_view`. Accessed: 2020-04-29.

[54] Leandra Vicci. Quaternions and rotations in 3-space: The algebra and its geometric interpretation. *Chapel Hill, NC, USA, Tech. Rep*, 2001.

[55] Farshad Arvin, Ali Emre Turgut, Tomáš Krajník, Salar Rahimi, Ilkin Ege Okay, Shigang Yue, Simon Watson, and Barry Lennox. ϕ clust: Pheromone-based aggregation for robotic swarms. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4288–4294. IEEE, 2018.

[56] Seongin Na, Yiping Qiu, Ali E Turgut, Jiří Ulrich, Tomáš Krajník, Shigang Yue, Barry Lennox, and Farshad Arvin. Bio-inspired artificial pheromone system for swarm robotics applications. *Adaptive Behavior*, page 1059712320918936, 2020.

[57] Seongin Na, Mohsen Raoufi, Ali Emre Turgut, Tomáš Krajník, and Farshad Arvin. Extended artificial pheromone system for swarm robotic applications. In *Artificial life conference proceedings*, pages 608–615. MIT Press, 2019.

[58] Farshad Arvin, Jose Espinosa, Benjamin Bird, Andrew West, Simon Watson, and Barry Lennox. Mona: an affordable open-source mobile robot for education and research. *Journal of Intelligent & Robotic Systems*, 94(3-4):761–775, 2019.

[59] Gazebo. `http://gazebosim.org/`. Accessed: 2020-04-29.

[60] Farshad Arvin, Khairulmizam Samsudin, Abdul Rahman Ramli, et al. Development of a miniature robot for swarm robotic application. *International Journal of Computer and Electrical Engineering*, 1(4):436–442, 2009.

[61] N. I. Fisher. *Statistical Analysis of Circular Data*. Cambridge University Press, 1993.

[62] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

Appendix

CD Content

In Table 10, there are listed names of all root directories with content description.

Directory name	**Description**
CosPhi	SwarmCon marker system
marker_msg	unified ROS message for all the markers
rqt-whycon-ros	rqt WhyCon/WhyCode interface
simulation_description	Gazebo simulator worlds and experiments coordinators
squares_opencv	OpenCV-based ROS ArUco and AprilTag marker systems
video_publisher	ROS interface for real-world dataset
whycon-ros	ROS WhyCon/WhyCode marker system

Table 10: CD Content