A Bayesian Framework for Multi-Stage Robot, Map and Target Localization

Ioannis Papakis

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Tomonari Furukawa, Chair
Erik Komendera
Anuj Karpatne

July 23, 2019
Blacksburg, Virginia

Keywords: EKF, SLAM, target, tracking, mapping
Copyright 2019, Ioannis Papakis

A Bayesian Framework for Multi-Stage Robot, Map and Target Localization

Ioannis Papakis

(ABSTRACT)

This thesis presents a generalized Bayesian framework for a mobile robot to localize itself and a target, while building a map of the environment. The proposed technique builds upon the Bayesian Simultaneous Robot Localization and Mapping (SLAM) method, to allow the robot to localize itself and the environment using map features or landmarks in close proximity. The target feature is distinguished from the rest of features since the robot has to navigate to its location and thus needs to be observed from a long distance. The contribution of the proposed approach is on enabling the robot to track a target object or region, using a multi-stage technique. In the first stage, the target state is corrected sequentially to the robot correction in the Recursive Bayesian Estimation. In the second stage, with the target being closer, the target state is corrected simultaneously with the robot and the landmarks. The process allows the robot's state uncertainty to be propagated into the estimated target's state, bridging the gap between tracking only methods where the target is estimated assuming known observer state and SLAM methods where only landmarks are considered. When the robot is located far, the sequential stage is efficient in tracking the target position while maintaining an accurate robot state using close only features. Also, target belief is always maintained in comparison to temporary tracking methods such as image-tracking. When the robot is closer to the target and most of its field of view is covered by the target, it is shown that simultaneous correction needs to be used in order to minimize robot, target and map entropies in the absence of other landmarks.

A Bayesian Framework for Multi-Stage Robot, Map and Target Localization

Ioannis Papakis

(GENERAL AUDIENCE ABSTRACT)

This thesis presents a generalized framework with the goal of allowing a robot to localize itself and a static target, while building a map of the environment. This map is used as in the Simultaneous Localization and Mapping (SLAM) framework to enhance robot accuracy and with close features. Target, here, is distinguished from the rest of features since the robot has to navigate to its location and thus needs to be continuously observed from a long distance. The contribution of the proposed approach is on enabling the robot to track a target object or region, using a multi-stage technique. In the first stage, the robot and close landmarks are estimated simultaneously and they are both corrected. Using the robot's uncertainty in its estimate, the target state is then estimated sequentially, considering known robot state. That decouples the target estimation from the rest of the process. In the second stage, with the target being closer, target, robot and landmarks are estimated simultaneously. When the robot is located far, the sequential stage is efficient in tracking the target position while maintaining an accurate robot state using close only features. When the robot is closer to the target and most of its field of view is covered by the target, it is shown that simultaneous correction needs to be used in order to minimize robot, target and map uncertainties in the absence of other landmarks.

Dedication

To my parents and my brother.

Acknowledgments

First and foremost, I would like to thank my advisor Tomonari Furukawa for his support, guidance and continuous trust.

I had the chance to be involved in many activities and projects under his advising, such as the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020, Mahindra & Mahindra Autononous Robotic Harvester and engage in academic research through publications. It was a unique chance to develop technical and theoretical skills that greatly enhanced my ability to work efficiently, innovatively and solve real-world problems. Professor Furukawa entrusted me to be involved in the MBZIRC competition as the team leader for graduate students and main advisor for the senior design team which manufactured an unmanned ground vehicle. I deeply thank him for giving me those opportunities which greatly strengthened my connections, my communication and organizational skills.

I will always remember our moments through our trips and our everyday interaction. It has been wonderful to have a person by my side who is not only my advisor, but a friend and spiritual father and I hope to always maintain our communication and friendship in the future.

I would also like to express my gratitude towards Professor Robin Ott. Through our cooperation as a teaching assistant in her senior design class, I learned a lot about communicating with students and teaching. She was very supportive, friendly and helped me a lot throughout the almost two years of our continuous work.

I would also like to thank one by one all the members of the lab and the undergraduate students which were involved in our lab's activities. For your support, friendship, advises and moments we shared, I want to thank Jeeseop Kim, Abdullah Abdul-Dayem, Phillip Grambo, Josiah Steckenrider, Dean Conte, George Kontoudis, Diya Li, Murat Ambarkutuk, Spencer Leamy, David Jensen, Mickey Cowden, Tamer Attia, Mengyu Song, Yazhe Hu, Cong Chen, Ashrarul Sifat and Jonathan Hodges.

For the people who have always been and will always be by my side in Greece through our friendship, I would like to thank Thanos Koumarelas, Alexandros Kariofillakis and Charalampos Papakwnstantinou. I owe thank you also to my beloved uncle Evangelos Chiotis for his friendship and constant motivation to pursue my dreams.

Finally, I would like to deeply thank my parents, grandparents and my brother for being by my side and supporting me in every possible way.

Contents

List of Figures viii

List of Tables x

1 Introduction 1

1.1 Background . . . 1

1.2 Objectives . . . 2

1.3 Contribution of the proposed framework . . . 2

1.4 Publications . . . 2

1.5 Outline . . . 3

2 Review of Literature 4

2.1 Fundamentals of Simultaneous Robot Localization and Mapping (SLAM) . . . 4

2.2 Target Search, Tracking & Engagement . . . 7

3 Bayesian Estimation, Robot and Target Localization 9

3.1 Probability theory . . . 9

3.2 Recursive Bayesian Estimation . . . 10

3.3 Robot Localization . . . 12

3.4 Target and Robot Localization . . . 12

4 Multi-Stage Bayesian Framework for Robot, Map and Target Localization 17

4.1 Overview . . . 17

4.2 Robot, Map and Target Localization . . . 19

4.2.1 Definitions . . . 19

4.2.2 Sequential Stage . . . 19

4.2.3 Simultaneous Stage . . . 21

4.2.4 Framework Implementation 23
4.2.5 Framework Extension 25

5 Experimental Analysis **28**
5.1 Objectives 28
5.2 Hardware & Software 29
5.3 Environment 32
5.4 System Modelling 33
5.5 Framework Testing & Analysis 34
5.5.1 Engaging with single target 34
5.5.2 Engaging with multiple targets 44
5.5.3 Engaging with a target region 47

6 Summary **52**

Bibliography **53**

List of Figures

2.1 Inference progression in filtering and smoothing [49] 5
2.2 SLAM application 6

3.1 Propagating uncertainties through motion and sensor models 11
3.2 Robot, Target uncertainties in origin frame 13
3.3 Belief space 14
3.4 Multi-stage robot and target localization framework [24] 16

4.1 Multi-stage SLAM and target localization framework 18
4.2 Noise calculation during each stage's states estimation 20
4.3 Propagating uncertainty between stages 22
4.4 Implemented recursive back-end 25
4.5 Implemented front-end 25
4.6 Targets and Landmarks 26
4.7 Target regions defined by multiple levels of precision 27
4.8 Hierarchical labelling and multi-stage localization scheme 27

5.1 Clearpath Robotics' Husky with installed robot and sensor suite 29
5.2 Proprioceptive Sensors 29
5.3 Exteroceptive Sensors 30
5.4 Overall software design 31
5.5 Detection and depth estimation examples 32
5.6 Outdoor facilities 32
5.7 Measuring noise level for various distances 33
5.8 Sensor model noise estimation and selection 33
5.9 Front-end vision based perception. Detection and depth registered on left-camera view 35

5.10 Test A, first-person view 35
5.11 Robot and target positions 36
5.12 Progressive map creation 36
5.13 State error vs time 37
5.14 Entropy vs time 38
5.15 Map created progressively by previous approach 38
5.16 Information gain of previous and proposed method 39
5.17 Different settings tested 40
5.18 Map created at each environment 40
5.19 Robot error 41
5.20 Target error 41
5.21 Robot entropy 41
5.22 Target entropy 42
5.23 Robot error 43
5.24 Target error 43
5.25 Test environment 44
5.26 Camera view 45
5.27 Progressive map creation 45
5.28 Error 46
5.29 Entropies 46
5.30 Tested Environments 47
5.31 Region labelling 48
5.32 Test A, first person view 48
5.33 Test B, first person view 49
5.34 Created map 49
5.35 Test A, error and entropies 50
5.36 Test B, error and entropies 51

List of Tables

5.1 System and process parameters 34

List of Abbreviations

EKF Extended Kalman Filter

FOV field of view

LOC level of confidence

ML Maximum Likelihood

PDF probability density function

PF Particle Filter

POD probability of detection

RBE recursive Bayesian estimation

ROI region of interest

SLAM Simultaneous Localization and Mapping

Chapter 1

Introduction

1.1 Background

Robotics has been a growing field for the past decades, integrating advances and innovations from different areas, which foster its growth. An increasing demand for autonomous tasks currently performed by humans, can be recognized in today's societies.

A commonly known industry to be associated with robots is the manufacturing. Due to its structured environment where many of the repetitive tasks can be deterministically designed, robot practical use and return of investment have made them prevalent. However, due to the continuous increase in their autonomous capabilities, their use has also been extended to the security, intelligent transportation, services and medical sectors among others [3].

Robotic research has been expanding in many areas. Those include the study of their kinematics, dynamics, sensing and estimation, motion planning, control and artificial intelligence to name a few [47]. Robot perception and localization is one of the fundamental areas of research, assisting robot operation in complex scenarios. Its vital role in enabling robot autonomy is commonly found in dynamic, unstructured and unknown environments, such as urban areas. Under those conditions, probabilistic techniques instead of deterministic methods are applied that deal with uncertainty in the robot perception of its environment and self-state.

Although many advances have been made in the field of localization, several challenges need to be addressed, such as long-term capability, operating in varying environment conditions and with many dynamic objects among others [9]. For a robot often needed to engage with a specific target and perform a task, such as drop off a package or manipulate an object, the ability to locate its target but also maintain an accurate self-localization is necessary. The need to develop robust systems through which the robot can perceive and localize itself and its environment, remains one of the pillars of ongoing research.

1.2 Objectives

This thesis' goals are to:

- Develop a global frame target localization method to estimate and approach a target from a long distance using a mobile robot platform.
- Develop a framework for robot localization and mapping while allowing for the use of target tracking.
- Investigate the effectiveness of the approach in experimental outdoor environments.
- Implement the approach using a variety of target objects.
- Compare the technique to similar approaches and previously proposed methods.

1.3 Contribution of the proposed framework

- A Bayesian method based on the Extended Kalman Filter that enables target tracking and Simultaneous Robot Localization and Mapping (SLAM) to be combined under a common formulation, using a multi-stage approach through:
  - The development of an uncertainty propagation method from robot to target in the tracking stage.
  - A target uncertainty transfer procedure from the tracking to the SLAM stage.
  - A sensor modelling process to estimate noise characteristics for observations of each stage.
- The ability to maintain belief for targets outside the robot field of view.
- A framework extension to deal with target regions in a multi-stage process by sequentially tracking each subregion and reach the target for engagement with high-precision.

1.4 Publications

I. **Ioannis Papakis**, Spencer Leamy, Tomonari Furukawa. Autonomous Navigation to a Targeted Distant Position Using Multi-Stage SLAM and Tracking. Proceedings of the 5th International Symposium on Future Active Safety Technology toward Zero Accidents, 2019.

1.5 Outline

Chapter 2 presents fundamental literature on the topic of robot localization using probabilistic methods. Also, it delves into the topic of target localization for a robot which has the task of locating and engaging with a target. Chapter 3 presents fundamental formulations concerning probability theory, robot and target localization. Chapter 4 proposes a framework for robot, map and target localization that can be applied in real-world scenarios. Chapter 5 is focusing on the experimental validation and analysis of the proposed approach, also in comparison with other techniques.

Chapter 2

Review of Literature

This chapter presents an overview of the SLAM history, with its past and current state of the art methods, as well as its connection to the tracking literature through various applications. Both SLAM and tracking have seen large number of uses in different domains and many times have crossed paths to enhance robot autonomy. Therefore, they will also be presented through some of the common applications they have shared in the context of robotics.

2.1 Fundamentals of Simultaneous Robot Localization and Mapping (SLAM)

SLAM refers to the simultaneous state estimation of robot and a model of the environment as perceived by on-board the robot sensors. Robot state is usually defined as pose, referring to position and orientation. Environment state usually refers to the position of environment features, extracted from sensor data or could also be raw data information.

SLAM has applications mainly in cases that robot localization is not feasible with other methods such as GPS or ad hoc localization infrastructure such as radio beacons. Such applications can be found at homes, factories, any kind of indoor facility with no GPS and also natural environments such as caves and underwater. Also in any case that an a-priori map is not available.

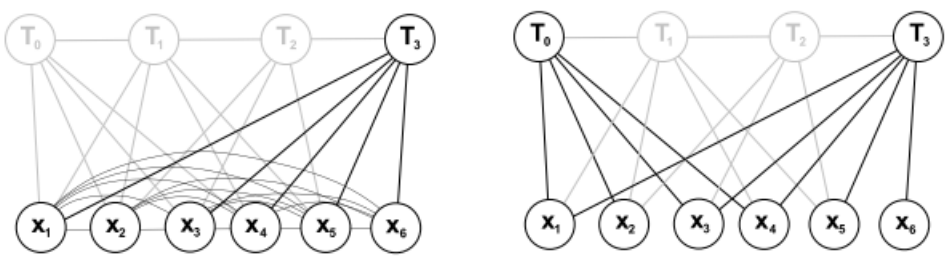
For the robot operating autonomously, it is important to localize itself using its surrounding information. The robot will be able to do mission and motion planning by deciding where is the next position to move and avoid obstacles in its path. Even in the case that the robot is operated by a human operator, SLAM can be used to provide visualization information.

Here it is important to make a distinction between SLAM and odometry only based localization. By odometry, we refer to measuring the robot displacement using sensors that measure sequential small displacement steps such as wheel odometers, inertial measurement units, cameras by tracking feature points, etc. Such sensors and techniques can only estimate the discrete incremental and sequential progression of robot displacement without being able to recognize if an environment has been revisited. The result is that by accumulating error, robot estimated position will eventually become inconsistent with the actual one. Recogniz-

ing previously visited areas, is a process termed as loop-closing and is the fundamental trait of SLAM systems in combination with odometry localization.

To provide a historical overview of SLAM, two main development phases can be recognized [10]. The classical age of SLAM (1986-2004), refers to the problem definition and early solving through probabilistic techniques, such as the Extended Kalman Filter (EKF) [21], Particle Filter (PF)[35] and maximum-likelihood estimation (MLE)[25]. Many versions of those techniques were developed and algorithms were created to deal with data association, memory and efficiency issues. The next period is the algorithmic analysis age (2004-2015), which tackled problems regarding observability, convergence and consistency of SLAM. During this period many modern and now also state of the art techniques were implemented based on the three main techniques of the previous era. The most notable techniques nowadays use the maximum-likelihood estimation, exploiting the sparsity of the SLAM problem and using efficient optimization solvers.

To define the SLAM problem, as mentioned previously, techniques such as the EKF, PF and MLE have been developed [49]. Although, this chapter won't present in detail the implementation for those techniques, the general framework will be presented to differentiate between them. The filtering techniques[27] are considered to be the EKF and PF, whereas the MLE techniques are usually referred to as smoothing techniques. The basic distinction can be seen in figure 2.1, where the filtering techniques tackle the online SLAM problem by estimating the current time step state using the previous only state, whereas the smoothing techniques use a set of previous states.



(a) Filtering (b) Smoothing

Figure 2.1: Inference progression in filtering and smoothing [49]

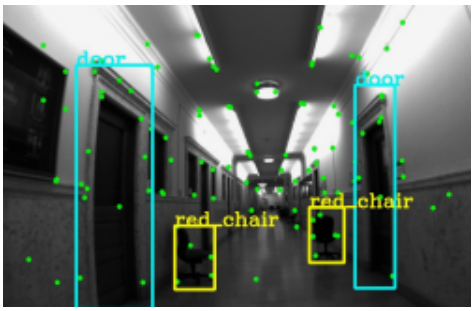
Filtering approaches are commonly expressed through the use of probabilities which model state probabilistically. Seeing figure 2.1a, the robot state is defined is defined by T and inputs such as measurements by x. The goal is to maximize the probability of T, given the set of x using Recursive Bayesian Estimation. Here all poses except the current one, are marginalized, as seen in the figure. This means essentially that only historic measurements and not robot states are used for the estimation.

Smoothing, known to the computer vision community as Bundle-Adjustment [49], is based on computing the state using historic measurements and a subset of historic states. The

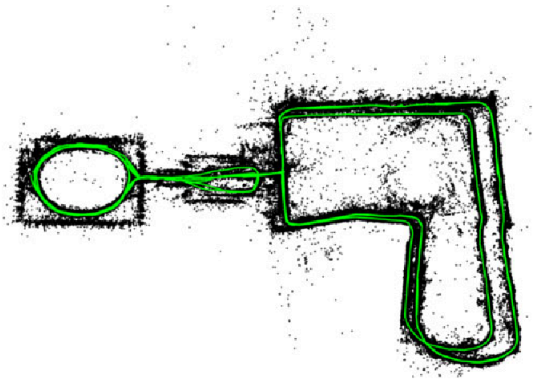
problem is usually dealt with the use of graphs in which factors represent probabilistic constraints over nodes. Marginalizing the set of poses, as seen in figure 2.1b, makes the problem sparse, without having to involve all past states, and computationally efficient to be solved by optimization algorithms.

So far, the focus was on the back-end of SLAM, however the front-end is a vital part of the process. Front-end usually refers to sensor input, data processing and data association. The first process determines the rules to identify landmarks such as points, lines, planes, objects or just process raw data. Many sensors can be used to achieve that, with cameras and lidars being the most dominant. In the age of semantics, semantic information can be used also to derive environment information and labels to identify features. Maps can therefore be distinguished as metric maps and semantic maps [10]. Metric maps usually employ landmarks or occupancy grids. Landmark or feature based maps identify a number of environment features which are extracted from raw data. Occupancy maps are generally more dense representation of the environment and exploit most of the raw data of the sensor to represent the environment. Semantic mapping on the other hand, assigns class labels to places and objects to recognize previously visited places in comparison to metric only map models.

The data association is also critical for the matching of features. By matching, it is meant discovering the connection of one feature to another previously observed. Data association is very important for short-term data association as well as long-term, finding applications in visual odometry and loop-closure respectively. Many data association techniques have been developed to tackle the uncertainty in dealing with sparse and dense data. Popular techniques include RANSAC, Joint Compatibility and Hough Transform [1].



(a) Features extraction and semantic labelling [8]



(b) SLAM with loop closing [37]

Figure 2.2: SLAM application

2.2 Target Search, Tracking & Engagement

Autonomous robots have many applications in which approaching a target and engaging through an autonomous task with it is necessary. The need for developing such robot capabilities can be found in warehouses, manufacturing environments, the package delivering industry, search and rescue operations and many more [29],[5]. The same demands and challenges have been introduced in robotics competitions [2] in order to test existing state of the art techniques and their limitations and develop novel methods, tackling particular sub-problems. The ability to localize a target from a far distance and finally approach and engage has been the main goal.

Localizing a target requires the use of many different robotic perception and information processing methods. Various sensors with different data types exist, creating the need to process the data for each different sensor. Lidar sensors have been used to track moving [17],[31] and static objects [55] using point clouds. Information from cameras, monocular [18] and stereo cameras [56] has also been extensively applied to detect objects using images. Those sensors as well as others such as radars are commonly fused together to detect and estimate the target's state more reliably and accurately [14],[41],[4].

Those sensors provide the output and several methods are then applied to extract useful information. Detection techniques are very broad and utilize color, edges, optical flow, texture and point cloud geometry. They can be categorized to point detectors [33],[34], segmentation methods [20],[30], background modellers [15] and supervised classifiers with the use of deep of deep neural networks [11], support vector machines [40] and others.

The first step in the process of engaging with a target after defining a sensor suite and detection scheme, is to first search for the target. Several techniques for searching have been proposed, depending on the application. A search agent has to localize itself and also search for the target [39]. Searching for a target in unknown environments is also commonly dealt with a team of robots such as UAVs, for scenarios in which a target has been lost or not yet detected [45],[23], such as in search and rescue scenarios. In those scenarios, multiple robots and their cooperation generalizes the concept of searching by enabling the usage of all available search agents. Searching for targets in occluded environments has moreover been addressed [54], as well as searching for multiple targets [22]. Optimizing search operations to minimize time and space coverage is another task commonly dealt with a team of robots [38] and by handling non-Gaussian beliefs [7].

After the target has been found, its state involving its position, pose, velocity and other characteristics have to be estimated and continuously updated in order to improve accuracy and also maintain the information. Several techniques exist for tracking applications. Point tracking techniques are commonly dealt with deterministic and probabilistic methods such as Kalman [52], Particle [26] and PHD filters [19]. Kernel and Silhouette trackers make use of appearence based models, contours and shapes. Common techniques involve Mean-Shift [50], KLT [46] and Hough transform [44] trackers. A more thorough analysis of various

detection and tracking methods can be found also in [53], [43].

Once the robot is located close to the target, control schemes can be used to engage with it using various manipulators. Control schemes usually include visual servoing [12], [13], in which features in the image frame provide guidance for a robot to approach a location using minimization of desired to current features position. Control laws such as position and force based, can be applied to a specially designed gripper and robotic arm to reach a specific position and apply a desired force.

The majority of the techniques presented for target localization, assuming that the agent is searching, estimating and engaging, do not formulate the problem to include the global frame for target localization. SLAM, which provides the robot with the capability to estimate the robot's position in a global frame, has been introduced with tracking applications for moving objects [16]. Those kinds of techniques can help an autonomous agent deal with complex environments where moving objects have to be avoided [51] and also distinguished from fixed landmarks [32]. The ability to combine target tracking and robot localization when a target is static has however only recently been introduced in a common formulation [28]. By this definition, the robot's position is unknown and has to be estimated, while the target is being sequentially tracked or simultaneously estimated with the robot depending on their distance using a multi-stage method.

This thesis proposes a method to perform robot, target and map localization using observations from sensors on-board the robot. The scenario considered, in this case, is a target being located far from the robot when detected. Enabling the capability for the robot to localize itself using surrounding close map features enhances its capability for accurate localization in unknown environments. At the same time, the target can be estimated by propagating the robot's uncertainty into the target estimate, tracking its position and maintaining belief even when the target is outside the sensor's field of view. Moreover, the target can be simultaneously corrected with the robot when the target is close and not many other environment features can be observed. Its main contributions are that the current framework propagates robot uncertainty to target estimation continuously for both stages, reducing monotonously the target uncertainty and allowing the robot to estimate features by tracking as well.

Chapter 3

Bayesian Estimation, Robot and Target Localization

This chapter presents an overview of Bayesian Estimation and its use for robot and target localization. First, it describes fundamental rules of probability theory. It then formulates Recursive Bayesian Estimation as it handles a generalized estimation problem for a dynamic system. Next, it presents how it can be applied to the problem of robot localization through the recursive procedure and appropriate notations. Finally, it describes the problem of robot and target localization through a generalized formulation as well as through the Extended Kalman Filter using a multi-stage process.

3.1 Probability theory

In probability theory, variables are considered random, meaning that their value depends on probabilistic inference laws. The outcome of an event is thus subject to a randomness which is described through a function p. If X represents all possible discrete values, then the aforementioned function satisfies the following properties:

$$\begin{cases} p(x) \in [0,1] & \forall x \in X \\ \sum_i p(x_i) = 1 \end{cases} \tag{3.1}$$

The continuous space is used for a random variable taking a range of values, instead of only one specific. For the case of a continuous random variable, the probability density function (pdf) can be defined. Here, the function represents the probability within an interval $x_1 \leq x \leq x_2$ of the continuous X space. The probability is therefore defined within intervals such that:

$$\begin{cases} p(x_1 \rightarrow x_2) = & \int_{x_1}^{x_2} p(x)dx \\ \int_x p(x)dx = & 1 \end{cases} \tag{3.2}$$

Many probability distribution functions exist for continuous variables. The most common probability distribution function is the Gaussian, due to its ability to represent many phe-

nomena. The Gaussian probability density is given by:

$$p(x) = \frac{e^{(-0.5*(\frac{(x-\mu)}{\sigma})^2)}}{(2\pi\sigma^2)^{-0.5}}$$

in the uni-variate case and

$$p(x) = det(2\pi\Sigma)^{0.5} e^{-0.5(x-\mu)^T \Sigma^{-1}(x-\mu)})$$

in the multi-variate case.

Moreover, when an event is associated with another event, this relationship can be expressed through the conditional PDF. It describes the probability of x_1 occurring when x_2 is known to hold. That is written through:

$$p(x_1 \mid x_2) = \frac{p(x_1, x_2)}{p(x_2)}$$

and this means that if a and b are independent, meaning $p(x_1 \mid x_2) = p(x_1)$, then $p(x_1, x_2) = p(x_1)p(x_2)$, defining that way the conditional independence.

An important addition is the Bayes rule. Bayes rule is another probabilistic law which relates a conditional probability $p(x_1 \mid x_2)$ to $p(x_2 \mid x_1)$, which is its inverse. The rule takes the form:

$$p(x_1 \mid x_2) = \frac{p(x_2 \mid x_1)p(x_1)}{p(x_2)}$$

Those rules and associations are the building blocks for probabilistic inference and are used to derive the relationships among system states.

3.2 Recursive Bayesian Estimation

Recursive Bayesian Estimation refers to constantly updating belief about the system's state. This operation is necessary and very commonly found in most systems which rely on noisy sensor input to perform actions. In fact, when there is some initial belief about the state, this can be constantly updated, using the Bayes rule and probability theory to derive the new stochastic belief.

Stochastic state estimate is a necessary means of representing belief about system's state. In Bayesian estimation theory, that belief is broken down into a motion and a sensor model. The motion model is used to describe a dynamic system and the connection between a control action u_k and the system state x_k at time step k. That connection is commonly expressed as $x_k = f(x_{1:k-1}, u_k, v_k)$ or otherwise $p(x_k \mid x_{1:k-1}, u_k, v_k)$, with the model dynamics expressed through function f. Moreover, the connection between the state and the input measurements

is described with the sensor model $z_k = h(x_k, w_k)$ or $p(x_k \mid z_k, w_k)$ where w_k is the modelled noise. The modelled noises v_k, w_k will be omitted in the following derivations.

Due to the fact that the process of evaluating system state through f is based on the modelled only dynamics and prior to the time step's k measurements, the first step utilizing it is called prediction. Prediction is written as:

$$\begin{aligned} p(x_k \mid z_{1:k-1}, u_k) = \quad & f_{pred}(p(x_{k-1} \mid z_{1:k-1})p(x_k \mid x_{k-1}, u_k)) \rightarrow \\ & \int_X (p(x_{k-1} \mid z_{1:k-1})p(x_k \mid x_{k-1}, u_k)dx \end{aligned} \tag{3.3}$$

where $x_0, ..., x_{k-2}$ has been omitted for a Markovian motion model where $p(x_k \mid x_0, ..., x_{k-1}) = p(x_k \mid x_{k-1})$

The next step is to update this belief with the measurements or else called observations at time step k. Due to the fact that these measurements are able to update the predicted belief, they constitute the correction step. The correction step can be probabilistically derived as:

$$\begin{aligned} p(x_k \mid z_{1:k}) = \quad & f_{cor}(p(x_k \mid z_{1:k-1}), p(z_k \mid x_k)) \rightarrow \\ & \frac{l(x_k \mid z_k)p(x_k \mid z_{1:k-1})}{\int_X l(x_k \mid z_k)p(x_k \mid z_{1:k-1})dx} \end{aligned} \tag{3.4}$$

with $l(x_k \mid z_k) = p(z_k \mid x_k)$, defined as likelihood. Prediction and correction are the two necessary components and represent the modelled system state input fused with the actual system input from the measurements. The weight of the importance to either of those factors determines the extent to which the system relies on the dynamics and the measurements. Due to the probabilistic nature of the states, uncertainties are maintained to express the historic state progression and provide relative weights for future beliefs.

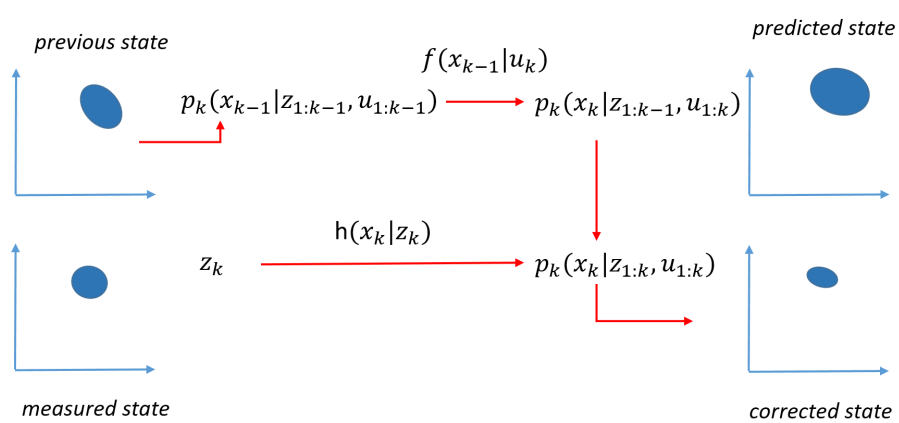


Figure 3.1: Propagating uncertainties through motion and sensor models

3.3 Robot Localization

A whole field in robotics is dedicated to the Bayesian estimation of the robot's state. Since uncertainties are dominant in many robotics platforms such as unmanned vehicles and walking robots, those need to be dealt with probabilistic inference laws to continue maintaining a robust to noise belief.

The most common of all states that could be estimated, the pose (position and orientation) is dealt with. Having a good estimate of it, provides the robot with the capability to choose motion actions while being in a coordinate frame defined as its original position or some other global frame such as geodesic coordinates. Its position and orientation is subject to uncertainty coordinates. Its position and orientation is subject to uncertainty always due to noisy sensors.

For the robot to achieve robust estimation, the RBE can be employed such that the system state x_k^r is recursively updated using the motion model f^r and sensor model h^r. This operation will be similarly to before expressed as:

$$p(x_k^r \mid z_{1:k-1}^r, u_k^r) = \int p(x_k^r \mid x_{k-1}^r, u_k^r) p(x_{k-1}^r \mid z_{1:k-1}^r, u_{1:k-1}^r) dx_k$$

using the Markovian motion model which means that $p(x_k^r \mid x_{k-1}^r, ..., x_0^r) = p(x_k^r \mid x_{k-1}^r)$. Next, when new external measurements arrive, the updated belief takes the form of:

$$p(x_k^r \mid z_{1:k}^r, u_k^r) = \frac{l(x_k^r \mid z_k^r) p(x_k^r \mid z_{1:k-1}^r)}{\int_X l(x_k^r \mid z_k^r) p(x_k^r \mid z_{1:k-1}^r) dx}$$

In robot platforms, prediction is usually done with odometry information with techniques such as dead reckoning. It relies on measuring wheel revolutions and steering angle in the case of ground vehicles. Usually methods such as this, are prone to error accumulation, which is termed as drift. The correction step is providing external input to the system such as GPS and beacons, which help to globally localize the robot with a less smooth however input. Combining the smooth but drifting input and the more noisy but not drifting input through the RBE helps provide a robust state estimate.

3.4 Target and Robot Localization

A robot usually has to perform a task reaching a targeted position close to an object of interest. Defining that region of interest as target, the problem of robot localization becomes clearly connected to the target localization. Maintaining an accurate belief about the target

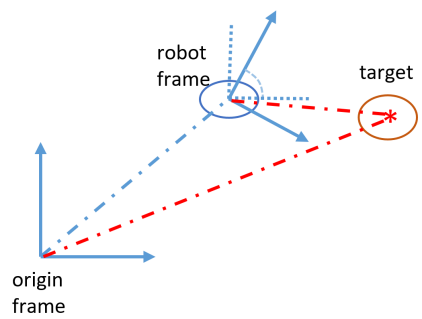


Figure 3.2: Robot, Target uncertainties in origin frame

state is essential for performing autonomous navigation to that targeted position in order to do the task.

Target state is also subject to noisy estimation since it has to be estimated with the use of exteroceptive sensors on-board the robot such as cameras and lidars. Target belief can now be expressed as $p(x_k^t \mid z_k^t, x_k^r)$ since it is dependent on the robot sensor which is located in the robot frame. For the problem of localizing a static target, only the sensor model is defined such that:

$$z_k^t = \begin{cases} h^t(x_{k-1}^t, x_k^r) & \text{if } x^t \in X_o^t, \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

defining as X_o^t the probability of detecting the target in the observable region $X_o^t = \{x^t \mid 0 \leq P_d \leq 1\}$. Moreover, since the target might be detected or not, the observation likelihood can be defined such that:

$$l(x_k^t \mid z_k^t, x_k^r) = \begin{cases} p(z_k^t \mid x_k^t, x_k^r) & z_k \in X_d \\ 1 - P_d & \text{otherwise} \end{cases} \tag{3.6}$$

where X_d can be defined as the detectable region $X_d^t = \{x^t \mid \epsilon \leq P_d \leq 1\}$ and P_d the probability of detection.

After defining these sensors characteristics for target estimation, the RBE can now be implemented. Using a multi-stage approach, previously proposed [28], to engage with a target in a scenario of a large field operation has recently been proposed. The target has been considered to be static, narrowing the scope of the problem. A generalized framework is presented to deal with the problems by means of a simultaneous and sequential approach. In multistage localization, robot and target are located sequentially when the target is faraway from the robot. When their locations are closer though, they are estimated simultaneously. Using this approach, the traditional target tracking and robot localization are handled in a unified framework. The approach is able to first track the target when it is far thus avoiding increasing uncertainties in the robot system and providing a low-precision tracking technique for target localization, since the robot position is well known. The second stage minimizes

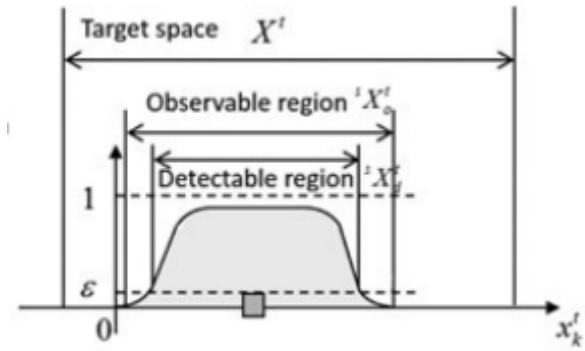


Figure 3.3: Belief space

the entropy of robot and target and increases accuracy which is needed when the robot is close to the target for engagement.

In the following formulations, the robot and target belief will be referred by $p(x_k^r), p(x_k^t)$. The time steps $1:k$ are defined along with the sequence of states, $x_{1:k} = \{x_i \mid \forall i \in \{1, ..., k\}\}$ and observations $z_{1:k} = \{z_i \mid \forall i \in \{1, ..., k\}\}$.

The recursion involves the prediction about the next time step's state and correction of the state for that step. This process is constituted of the following:

Prediction:
Prediction takes place only for the robot. The PDF of the current state $p(x_k^r \mid z_{1:k}^r, u_{1:k}^r)$ is first predicted from the previous belief state $p(x_{k-1}^r)$. Using a Markovian motion model, which means $p(x_k^r \mid x_{k-1}^r, \ldots, x_0^r) = p(x_k^r \mid x_{k-1}^r)$, the Chapman-Kolmogorov equation takes the form:

$$p(x_k^r \mid z_{1:k-1}^r, u_{1:k}^r) = \int_X p(x_k^r \mid x_{k-1}^r, u_k^r) * p(x_{k-1}^r \mid z_{1:k-1}^r u_{1:k-1}^r) dx_k \tag{3.7}$$

The RBE is implemented through the use of the Extended Kalman Filter (EKF), which makes the assumption of Gaussian noise models $w_k \sim N(0, \Sigma^w)$, $v_k \sim N(0, \Sigma^v)$. Prediction for robot motion is given by:

$$x_{k|k-1}^r = f^r(x_{k-1|k-1}^r, u_k^r) \tag{3.8a}$$

$$\Sigma_{k|k-1}^r = \nabla_r f^r \Sigma_{k-1|k-1}^r \nabla_r f^{rT} + \Sigma_{k-1}^w \tag{3.8b}$$

with $\nabla_r f^r$ the motion model Jacobian at $x_{k-1|k-1}$.

Correction:
Sequential stage: The correction step of the process is incorporating sensor measurements from the robot z_k^r and the target z_k^t. The PDF takes the corrected form as previously

mentioned:

$$p(x_k^r \mid z_{1:k}^r, x_{k|k-1}^r) = \frac{l(x_k^r \mid z_k^r)p(x_k^r \mid z_{1:k-1}^r)}{\int_X l(x_k^r \mid z_k^r)p(x_k^r \mid z_{1:k-1}^r)dx} \tag{3.9}$$

Using the EKF this can be written:

$$x_{k|k}^r = x_{k|k-1}^r + W_k^r(z_k^r - h^r(x_{k|k-1}^r)) \tag{3.10a}$$
$$\Sigma_{k|k}^r = (I - W_k^r \nabla_r h^r)\Sigma_{k|k-1}^r \tag{3.10b}$$

When the target is observed, the following belief is updated only for the target:

$$p(x_k^t \mid z_{1:k}^t, x_k^r, x_{k-1}^t) = \frac{l(x_k^t \mid z_k^t, x_k^r)p(x_k^t \mid z_{1:k-1}^t, x_k^r)}{\int_X l(x_k^t \mid z_k^t, x_k^r)p(x_k^t \mid z_{1:k-1}^t, x_k^r)dx} \tag{3.11}$$

Using the EKF the equations take a similar form again as:

$$x_{k|k}^t = x_{k|k-1}^t + W_k^t(z_k^t - h^t(x_{k|k-1}^t, x_{k|k}^r)) \tag{3.12a}$$
$$\Sigma_{k|k}^{t'} = (I - W_k^t \nabla_t h^t)\Sigma_{k|k-1}^t \tag{3.12b}$$
$$\Sigma_{k|k}^t = \Sigma_{k|k}^{t'} + \Sigma_{k|k}^r \tag{3.12c}$$

with the Kalman gain in general defined as $W_k = \nabla h^T(\nabla h \Sigma_{k|k-1} \nabla h^T + \Sigma_k^v)^{-1}$.

Simultaneous stage: This time the prediction is the same as in the sequential stage. The difference is that the positions of the target and robot are estimated simultaneously as:

$$p(x_k^{rt} \mid z_{1:k}^r, z_{1:k}^t, u_{1:k}^r) = \frac{l(x_k^{rt} \mid z_k^r, z_k^t) * p(x_k^{rt} \mid z_{1:k-1}^r, z_{1:k-1}^t, u_{1:k}^r)}{\int_X l(x_k^{rt} \mid z_k^r, z_k^t) * p(x_k^{rt} \mid z_{1:k-1}^r, z_{1:k-1}^t, u_{1:k}^r)dx_{k-1}} \tag{3.13}$$

In EKF form this is written as:

$$\begin{bmatrix} x_{k|k}^r \\ x_{k|k}^t \end{bmatrix} = \begin{bmatrix} x_{k|k-1}^r \\ x_{k|k-1}^t \end{bmatrix} + \begin{bmatrix} W_k^{rr} & W_k^{rt} \\ W_k^{tr} & W_k^{tt} \end{bmatrix} \begin{bmatrix} z^r - h^r(x_{k|k-1}^r) \\ z^t - h^t(x_{k-1|k-1}^t, x_{k|k-1}^r) \end{bmatrix} \tag{3.14a}$$

$$\begin{bmatrix} \Sigma_{k|k}^r & \Sigma_{k|k}^{rt} \\ \Sigma_{k|k}^{tr} & \Sigma_{k|k}^t \end{bmatrix} = \left(\begin{bmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{bmatrix} - \begin{bmatrix} W_k^r & W_k^{rt} \\ W_k^{tr} & W_k^t \end{bmatrix} \begin{bmatrix} \nabla h_k^r & \nabla h_k^r \\ \nabla h_k^t & \nabla h_k^t \end{bmatrix} \right) \begin{bmatrix} \Sigma_{k|k-1}^r & \Sigma_{k|k-1}^{rt} \\ \Sigma_{k|k-1}^{tr} & \Sigma_{k|k-1}^t \end{bmatrix} \tag{3.14b}$$

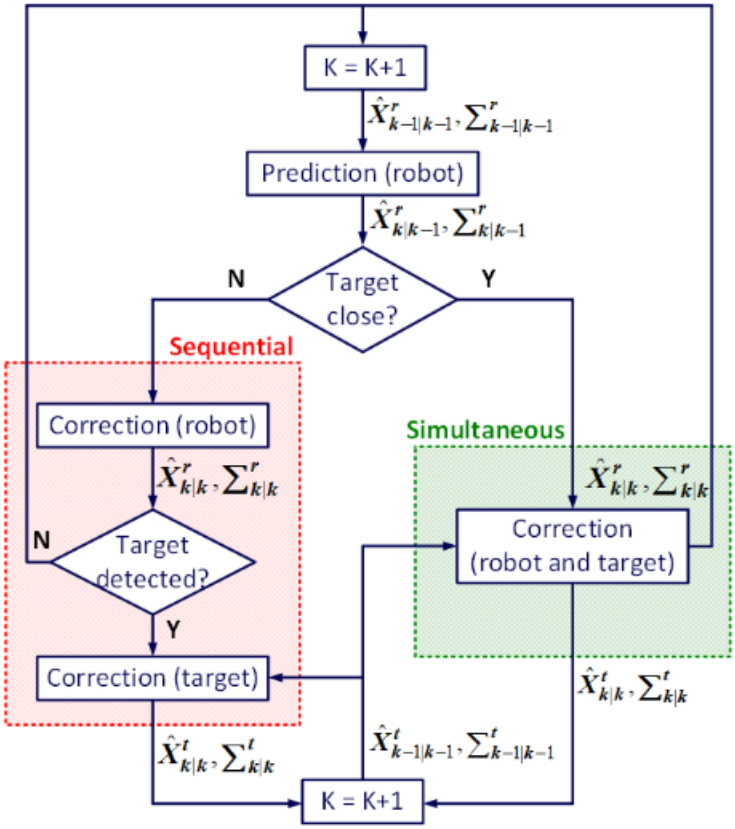


Figure 3.4: Multi-stage robot and target localization framework [24]

Chapter 4

Multi-Stage Bayesian Framework for Robot, Map and Target Localization

This chapter presents a multi-stage framework for a robot to localize itself and a target from a long distance, using surrounding map features. The proposed framework consists of two multi-stage Bayesian techniques, which localize the robot and the environment in a global frame. Features are extracted from raw sensor data and the target is detected. The robot maps the environment and localizes itself by simultaneously correcting its position accurately, using close features as in SLAM. The target estimation which is commonly dealt within the target tracking framework is corrected sequentially by propagating robot uncertainty, in the case that the robot needs to rely only on close features. Target uncertainty is gradually reduced reaching a lower limit. When the robot is closer to the target, it is shown that simultaneous correction needs to be used in order to minimize robot, target and map entropies. The experimental validation of the approach on a single target, on multiple targets and on a target region, using an unmanned ground vehicle shows the applicability and effectiveness of the approach in real-world autonomous operation.

4.1 Overview

The block diagram in figure 4.1 illustrates the recursive process of the proposed framework. Robot position is first predicted through the use of a motion model. Sensor observations are extracted and labelled in a way that separates the target from the rest of the features. Those extracted features are then filtered to determine whether they should be considered as landmarks through techniques such as data association. In the case of target detection, it is first estimated sequentially to the robot, similarly to many target tracking problems. This will be the basic assumption in this study, in the sense that the robot can only gain small information in comparison to more robust close observations. During this stage, target uncertainty is being gradually reduced reaching a lower limit and a stable estimate. Robot uncertainty also remains low by observing other features. When the robot approaches the target, its field of view can be easily covered by mostly the target or only the target without many other features. Thus, a distance criterion can be defined to make simultaneous correction between target and robot take place. Using this as stage selection, although sequential localization is efficient in the first stage, in order to further minimize the target entropy, si-

multaneous correction needs to take place. Also, when target uncertainty has been reduced in the first stage, robot and map uncertainty, being corrected with every target observation, can be reduced more significantly in the second stage.

The proposed technique builds upon the previous framework of Chapter 3. It incorporates map features in the process to provide loop closing capabilities for robot estimation, in addition to the previous technique, achieving lower uncertainties and lower error. This is done by a more accurate method for the propagation of robot uncertainty to target when corrected sequentially. The sensor noise is also modelled based on the estimated states rather than the measurements.

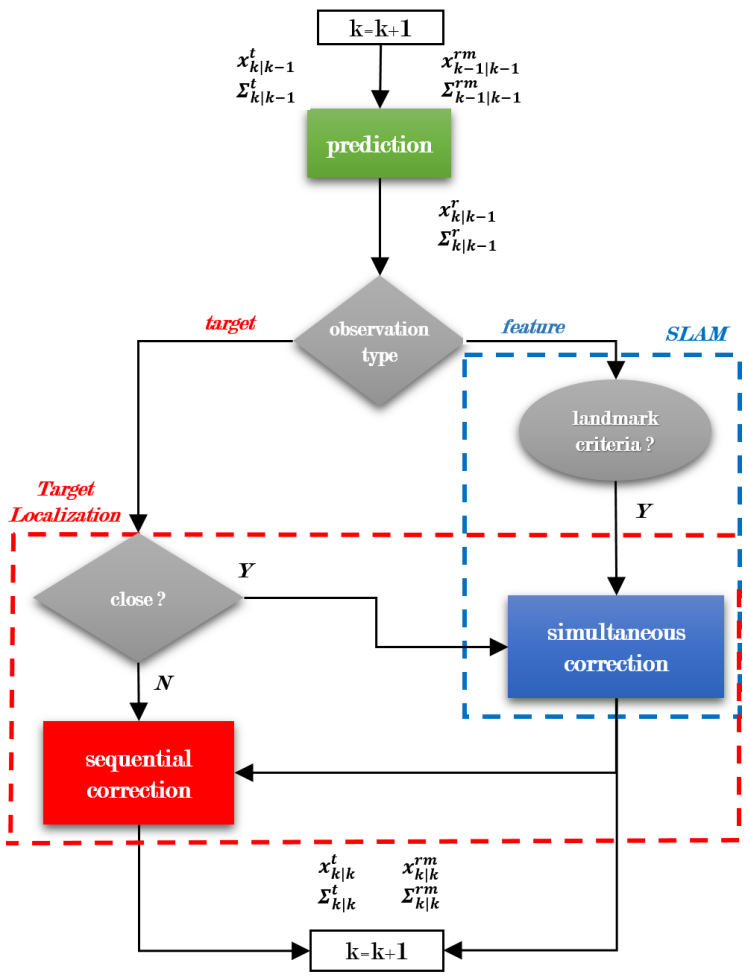


Figure 4.1: Multi-stage SLAM and target localization framework

4.2 Robot, Map and Target Localization

4.2.1 Definitions

Using features in the environment, the robot is able to correct its position at all k time steps. The robot state will be expressed by x_k^r and map state as $x_k^m = [x_k^{m_1}, ..., x_k^{m_n}]$ where n the number of landmarks. Moreover, target state will be again referred by x_k^t. Similarly to the previous approach, a sensor model will be used, defined as heteroscedastic noise model. That means that the expected noise variance is changing depending on the values of the states observed. That is expressed by $\Sigma^w = \Sigma^w(v_1, v_2)$, where u the vectors of variables observed. Since features are considered fixed in the environment, no motion model is defined for them. To narrow down the scope of the problem, target will also be considered static. The motion model thus for the robot only will be referred as f^r and $G = \nabla f^r$ is its derivative, whereas h^m the sensor model for the features and h^t the sensor model for the target. Then:

$$H^m = \nabla h^m = \begin{bmatrix} \frac{\partial v_1}{\partial x^r} & \frac{\partial v_2}{\partial x^m} \\ \frac{\partial v_1}{\partial x^r} & \frac{\partial v_2}{\partial x^m} \end{bmatrix} \tag{4.1a}$$

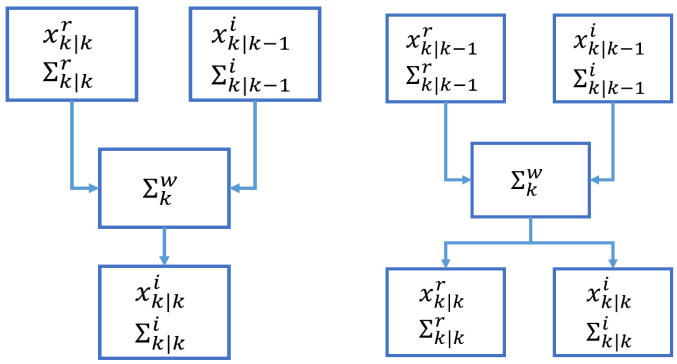
$$H^t = \nabla h^t = \begin{bmatrix} \frac{\partial v_1}{\partial x^r} & \frac{\partial v_2}{\partial x^t} \\ \frac{\partial v_1}{\partial x^r} & \frac{\partial v_2}{\partial x^t} \end{bmatrix} \tag{4.2a}$$

Moreover, in this framework a different technique is used to calculate the noise characteristic when estimating the target. Sensor noise Σ^w may vary based on $v_1, v_2, ..$ values as previously mentioned. Instead of using those observations to calculate the noise characteristics, here is proposed to use the estimated states and derive a relationship among noise and those values such as estimated range, bearing, etc. This can be expressed as $\Sigma^w = \Sigma^w(x^r, x^i)$ where r robot state and i the other feature or target state observed. This is shown graphically for both sequential and simultaneous stages in figure 4.2.

4.2.2 Sequential Stage

The proposed approach builds upon the following recursive process:

Prediction:



(a) Sequential stage (b) Simultaneous stage

Figure 4.2: Noise calculation during each stage's states estimation

For the robot and map:

$$\begin{bmatrix} x^r_{k|k-1} \\ x^m_{k|k-1} \end{bmatrix} = \begin{bmatrix} f^r(x^r_{k-1|k-1}, u^r_k) \\ x^m_{k-1|k-1} \end{bmatrix} \tag{4.3a}$$

$$\begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} \end{bmatrix} = \begin{bmatrix} G_{k-1} & 0 \\ 0 & \mathrm{I} \end{bmatrix} * \begin{bmatrix} \Sigma^r_{k-1|k-1} & (\Sigma^{rm}_{k-1|k-1})^T \\ \Sigma^{rm}_{k-1|k-1} & \Sigma^m_{k-1|k-1} \end{bmatrix} * \begin{bmatrix} (G_{k-1})^T & 0 \\ 0 & \mathrm{I} \end{bmatrix} + \begin{bmatrix} \Sigma^v & 0 \\ 0 & 0 \end{bmatrix} \tag{4.3b}$$

For the target:

$$x^t_{k|k-1} = x^t_{k-1|k-1} \tag{4.4a}$$

$$\Sigma^t_{k|k-1} = \Sigma^t_{k-1|k-1} \tag{4.4b}$$

$$\Sigma^{rt}_{k|k-1} = (G * \Sigma^{tr}_{k-1|k-1})^T \tag{4.4c}$$

Correction: During this step, measurements are incorporated to update the predicted states. For the robot and the features the sequential correction process is:

$$\begin{bmatrix} x^r_{k|k} \\ x^m_{k|k} \end{bmatrix} = \begin{bmatrix} x^r_{k|k-1} \\ x^m_{k|k-1} \end{bmatrix} + \sum_{i=1}^{k} (K^{m^i}_k * (z^{m^i}_k - h^{m^i}_k(x^r_{k|k-1}, x^{m^i}_{k|k-1}))) \tag{4.5a}$$

$$\begin{bmatrix} \Sigma^r_{k|k} & (\Sigma^{rm}_{k|k})^T \\ \Sigma^{rm}_{k|k} & \Sigma^m_{k|k} \end{bmatrix} = (I - \sum_{i=1}^{k} K^{m^i}_k \nabla h^{m^i}_k) * \begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm^i}_{k|k-1})^T \\ \Sigma^{rm^i}_{k|k-1} & \Sigma^{m^i}_{k|k-1} \end{bmatrix} \tag{4.5b}$$

where k the number observed features and:

$$K_k^{m_i} = \begin{bmatrix} \Sigma_{k|k-1}^{r} & (\Sigma_{k|k-1}^{rm^i})^T \\ \Sigma_{k|k-1}^{rm^i} & \Sigma_{k|k-1}^{m^i} \end{bmatrix} * H^{m'} * (H^{m'} * \begin{bmatrix} \Sigma_{k|k-1}^{r} & (\Sigma_{k|k-1}^{rm^i})^T \\ \Sigma_{k|k-1}^{rm^i} & \Sigma_{k|k-1}^{m^i} \end{bmatrix} * H^{m'T} + \Sigma_k^w)^{-1} \tag{4.6a}$$

where $H^{m'} = H^m * F$ with F mapping the low-dimensional H^m such that $H^m \times F_0 = (n_1 + n * n_2)^2$, where n_1 robot states, n_2 landmark states and n landmarks number.

For the target estimate correction:

$$x_{k|k}^t = x_{k|k-1}^t + F_1 * (K_k^t * (z_k^t - h_k^t(x_{k|k}^r, x_{k|k-1}^t))) \tag{4.7a}$$

$$\begin{bmatrix} \Sigma_{k|k}^{tr} \\ \Sigma_{k|k}^{t} \end{bmatrix} = (I - K_k^t \nabla h_k^t) * \begin{bmatrix} \Sigma_{k|k}^{r} & (\Sigma_{k|k-1}^{rt})^T \\ \Sigma_{k|k-1}^{rt} & \Sigma_{k|k-1}^{t} \end{bmatrix} * F_2 \tag{4.7b}$$

with $F_1 = [0, I]$, $F_2 = [0; I]$ and:

$$K_k^t = \begin{bmatrix} \Sigma_{k|k}^{r} & (\Sigma_{k|k-1}^{rt})^T \\ \Sigma_{k|k-1}^{rt} & \Sigma_{k|k-1}^{t} \end{bmatrix} * H^t * (H^t * \begin{bmatrix} \Sigma_{k|k}^{r} & (\Sigma_{k|k-1}^{rt})^T \\ \Sigma_{k|k-1}^{rt} & \Sigma_{k|k-1}^{t} \end{bmatrix} * H^{tT} + \Sigma_k^w)^{-1} \tag{4.8a}$$

Robot uncertainty estimated through map corrections is thus propagated to the target. Notice that maintaining the correlations of Σ^{rt} is as important as maintaining Σ^{rm}, since otherwise estimates become overconfident and uncertainty can be quickly reduced to almost zero.

To initialize target uncertainty, a similar approach can be followed to landmark initialization. That means that:

$$\Sigma_0^t = \nabla h_x^{-1} \Sigma_k^r (\nabla h_x^{-1})^T + \nabla h_v^{-1} \Sigma_k^w (\nabla h_v^{-1})^T$$
$$\Sigma_0^{rt} = \nabla h_x^{-1} \Sigma_k^r$$

4.2.3 Simultaneous Stage

After the stage selection to simultaneous, it is crucial to maintain the information gained by sequentially correcting target position. Target can then be included as in figure 4.3, where $x_{k|k}^t$ is the last updated state of target together with $\Sigma_{k|k}^t$, $\Sigma_{k|k}^{rt}$ while $\Sigma_{k|k}^{mt}$ can be initialized as $\Sigma_{k|k}^{mt} = \nabla g_x^{-1} \Sigma_{k|k}^{rm}$. This process allows transition from sequential to simultaneous with minimum information loss. Notice that target is written separately from the rest of landmarks, since its tracking states might be different such as estimating its 3D position instead of only 2D landmarks.

With the new state vector, simultaneous localization is written as:

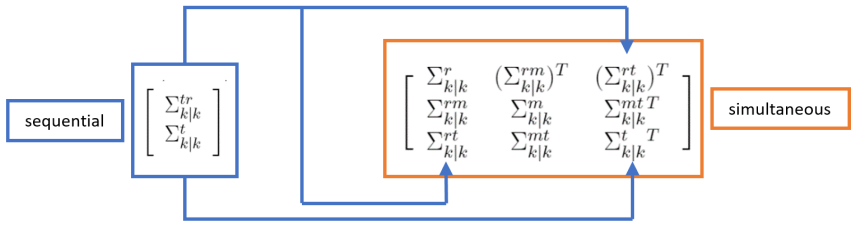


Figure 4.3: Propagating uncertainty between stages

Prediction:

$$\begin{bmatrix} x^r_{k|k-1} \\ x^m_{k|k-1} \\ x^t_{k|k-1} \end{bmatrix} = \begin{bmatrix} f^r(x^r_{k-1|k-1}, u^r_k) \\ x^m_{k-1|k-1} \\ x^t_{k-1|k-1} \end{bmatrix} \tag{4.9a}$$

$$\begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T & (\Sigma^{rt}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} & \Sigma^{mt}_{k|k-1}{}^T \\ \Sigma^{rt}_{k|k-1} & \Sigma^{mt}_{k|k-1} & \Sigma^t_{k|k-1}{}^T \end{bmatrix} = \begin{bmatrix} G_{k-1} & 0 & 0 \\ 0 & \mathrm{I} & 0 \\ 0 & 0 & \mathrm{I} \end{bmatrix} * \begin{bmatrix} \Sigma^r_{k-1|k-1} & (\Sigma^{rm}_{k-1|k-1})^T & (\Sigma^{rt}_{k-1|k-1})^T \\ \Sigma^{rm}_{k-1|k-1} & \Sigma^m_{k-1|k-1} & \Sigma^{mt}_{k-1|k-1}{}^T \\ \Sigma^{rt}_{k-1|k-1} & \Sigma^{mt}_{k-1|k-1} & \Sigma^t_{k-1|k-1}{}^T \end{bmatrix} * \begin{bmatrix} (G_{k-1})^T & 0 & 0 \\ 0 & \mathrm{I} & 0 \\ 0 & 0 & \mathrm{I} \end{bmatrix} + \begin{bmatrix} \Sigma^v & 0 & 0 \\ 0 & \mathrm{I} & 0 \\ 0 & 0 & \mathrm{I} \end{bmatrix} \tag{4.9b}$$

Correction:

Next, the correction step is:

$$\begin{bmatrix} x^r_{k|k} \\ x^m_{k|k} \\ x^t_{k|k} \end{bmatrix} = \begin{bmatrix} x^r_{k|k-1} \\ x^m_{k|k-1} \\ x^t_{k|k-1} \end{bmatrix} + \sum_{i=1}^{k} (K^{m^i}_k * (z^{m^i}_k - h^{m^i}_k(x^r_{k|k-1}, x^{m^i}_{k|k-1}))) + K^t_k * (z^t_k - h^t_k(x^r_{k|k-1}, x^t_{k|k-1})) \tag{4.10a}$$

$$\begin{bmatrix} \Sigma^r_{k|k} & (\Sigma^{rm}_{k|k})^T & (\Sigma^{rt}_{k|k})^T \\ \Sigma^{rm}_{k|k} & \Sigma^m_{k|k} & \Sigma^{mt}_{k|k}{}^T \\ \Sigma^{rt}_{k|k} & \Sigma^{mt}_{k|k} & \Sigma^t_{k|k}{}^T \end{bmatrix} = (I - \sum_{i=1}^{k} K^{m^i}_k \nabla h^{m^i}_k - K^t_k \nabla h^t_k) * \begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T & (\Sigma^{rt}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} & \Sigma^{mt}_{k|k-1}{}^T \\ \Sigma^{rt}_{k|k-1} & \Sigma^{mt}_{k|k-1} & \Sigma^t_{k|k-1}{}^T \end{bmatrix} \tag{4.10b}$$

where:

$$\begin{aligned} K^{m_i}_k = & \begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T & (\Sigma^{rt}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} & \Sigma^{mt}_{k|k-1}{}^T \\ \Sigma^{rt}_{k|k-1} & \Sigma^{mt}_{k|k-1} & \Sigma^t_{k|k-1}{}^T \end{bmatrix} * H^{m'} \\ & * (H^{m'} * \begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T & (\Sigma^{rt}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} & \Sigma^{mt}_{k|k-1}{}^T \\ \Sigma^{rt}_{k|k-1} & \Sigma^{mt}_{k|k-1} & \Sigma^t_{k|k-1}{}^T \end{bmatrix} * H^{m'T} + \Sigma^w_k)^{-1} \end{aligned} \tag{4.11a}$$

where $H^{m'} = H^m * F_0$ with F mapping the low-dimensional H^m such that $H^m \times F_0 = (n_1 + n * n_2 + n_3)^2$, where n_1 robot states, n_2 landmark states, n landmarks number and n_3 the target states.

and

$$K_k^t = \begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T & (\Sigma^{rt}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} & \Sigma^{mt}_{k|k-1}{}^T \\ \Sigma^{rt}_{k|k-1} & \Sigma^{mt}_{k|k-1} & \Sigma^t_{k|k-1}{}^T \end{bmatrix} * H^{t'} \\ * (H^{t'} * \begin{bmatrix} \Sigma^r_{k|k-1} & (\Sigma^{rm}_{k|k-1})^T & (\Sigma^{rt}_{k|k-1})^T \\ \Sigma^{rm}_{k|k-1} & \Sigma^m_{k|k-1} & \Sigma^{mt}_{k|k-1}{}^T \\ \Sigma^{rt}_{k|k-1} & \Sigma^{mt}_{k|k-1} & \Sigma^t_{k|k-1}{}^T \end{bmatrix} * H^{t'T} + \Sigma^w_k)^{-1} \tag{4.12a}$$

where $H^{t'} = H^t * F_1$ with F_1 mapping it similarly to the high dimensional space.

Further things to notice is that this formulation is able to differentiate target state and landmark states in the simultaneous and sequential stage. It propagates robot uncertainty to the target differently and in a way that target uncertainty is bounded and monotonously decreasing, in comparison to the previous approach. Last, it is able to maintain belief for target even when changing stage without loss of information.

4.2.4 Framework Implementation

The presented and proposed approach for long to close range target localization using surrounding map features is implemented on an unmanned ground vehicle (UGV). Basic assumptions will be made here to formulate a complete solution to the framework. The robot is localizing 2D landmarks positions using range-bearing observations. 2D position will be assumed for the vehicle as well and bearing. To simplify the problem formulation, target will also be assumed to have a 2D position to be tracked although any number of states different from other landmarks could be estimated.

These states will be expressed as $X^r_k = [x^r_k, y^r_k, \theta^r_k]$, $X^t_k = [x^t_k, y^t_k]$ and $X^m_k = [x^m_k, y^m_k]$. Moreover, the covariances are:

$$\Sigma^r_{k|k} = \begin{bmatrix} \Sigma^{xx}_{k|k} & \Sigma^{yx}_{k|k} & \Sigma^{\theta x}_{k|k} \\ \Sigma^{xy}_{k|k} & \Sigma^{yy}_{k|k} & \Sigma^{\theta y}_{k|k} \\ \Sigma^{x\theta}_{k|k} & \Sigma^{y\theta}_{k|k} & \Sigma^{\theta\theta}_{k|k} \end{bmatrix}, \qquad \Sigma^t_{k|k} = \begin{bmatrix} \Sigma^{xx}_{k|k} & \Sigma^{yx}_{k|k} \\ \Sigma^{xy}_{k|k} & \Sigma^{yy}_{k|k} \end{bmatrix}, \qquad \Sigma^m_{k|k} = \begin{bmatrix} \Sigma^{xx}_{k|k} & \Sigma^{yx}_{k|k} \\ \Sigma^{xy}_{k|k} & \Sigma^{yy}_{k|k} \end{bmatrix} \tag{4.13a}$$

Further definitions involve motion and sensor models. The motion model is $f^r = x^r_{k-1|k-1} + (x^{odom}_k - x^{odom}_{k-1})$, where $(x^{odom}_k - x^{odom}_{k-1})$ the odometry interval from last correction. In addition, landmarks are first corrected and provide absolute localization for the robot instead of only

relative such as odometry. The sensor model for those is range bearing and formulation is seen below. Notice the difference in the formulations for each stage:

$$h = h^m = h^t = \begin{bmatrix} \sqrt{q} \\ atan(\frac{dy}{dx}) - \theta^r \end{bmatrix} \tag{4.14a}$$

and

$$\nabla h^m = \nabla h^t = \frac{1}{q} \begin{bmatrix} -\sqrt{q}dx & \text{-}\sqrt{q}dy & 0 & \sqrt{q}dx & \sqrt{q}dy \\ dy & \text{-dx} & \text{-q} & \text{-dy} & \text{dx} \end{bmatrix} \tag{4.15a}$$

where $q = d^T d$ with:

$$d^{m_i} = \begin{bmatrix} x^m_{k|k-1} - x^r_{k|k-1};\ y^m_{k|k-1} - y^r_{k|k-1} \end{bmatrix} \tag{4.16a}$$

for landmarks and:

$$d^t = \begin{bmatrix} x^t_{k|k-1} - x^r_{k|k};\ y^t_{k|k-1} - y^r_{k|k} \end{bmatrix} \tag{4.17a}$$

in the sequential stage while:

$$d^t = \begin{bmatrix} x^t_{k|k-1} - x^r_{k|k-1};\ y^t_{k|k-1} - y^r_{k|k-1} \end{bmatrix} \tag{4.18a}$$

in the simultaneous stage.

As previously mentioned, F_0 is used to map ∇h to the dimensions of Σ for correction, thus:

$$F_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

where the first omitted size is $2j - 2$ and the second $2N - 2j$ with N number of landmarks plus target if included and j the ID of current observation. If target had more states then the rows would have to be increased. Overall, the back-end design will have the structure of figure 4.4.

For the data association, 4.5, among observations, map features are the only ones considered to be similar while target is different and no association is needed. Thus for the target to landmarks identification a semantic rule is established which identifies both. For the data association which is needed since landmarks are similar, any data association rule can be established as mentioned in Chapter 2, such as nearest neighbor.

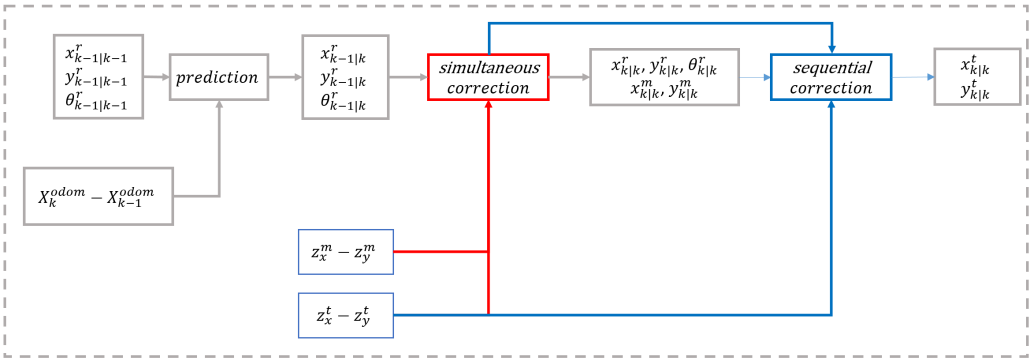


Figure 4.4: Implemented recursive back-end

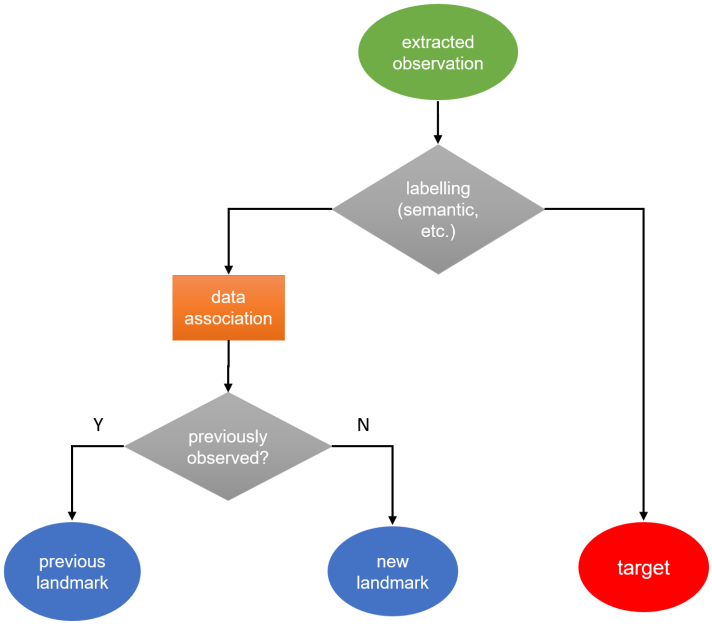


Figure 4.5: Implemented front-end

4.2.5 Framework Extension

This section is adding to the previous distinction of targets and features based on distance of observation. Target engagement is a crucial part for many robotic tasks. Many applications require that the target is detected from long range, localized in the environment and finally manipulated by the robot. In those kinds of scenarios, there are two areas that need to be addressed concerning the target's special characteristics and the mission for target engagement. The proposed framework will be extended here and generalized as a technique to engage sequentially with multiple regions of interest, expressed as target region.

First, concerning the target's characteristics which can be seen as potential landmarks to be used in SLAM, there are several differences. These differences which can be identified to represent a generalized scenario for robot-target engagement are summarized in figure 4.6. Those differences are broken down in usage and spatio-temporal characteristics.

	usage	spatial characteristics	temporal characteristics
features	improve localization, close to mid range	points/lines/planes/objects, fixed or expected size	permanent/fixed
target	provide navigation goal, close to long range	points/objects/regions, varying size and shape	subject to transport/manipulation

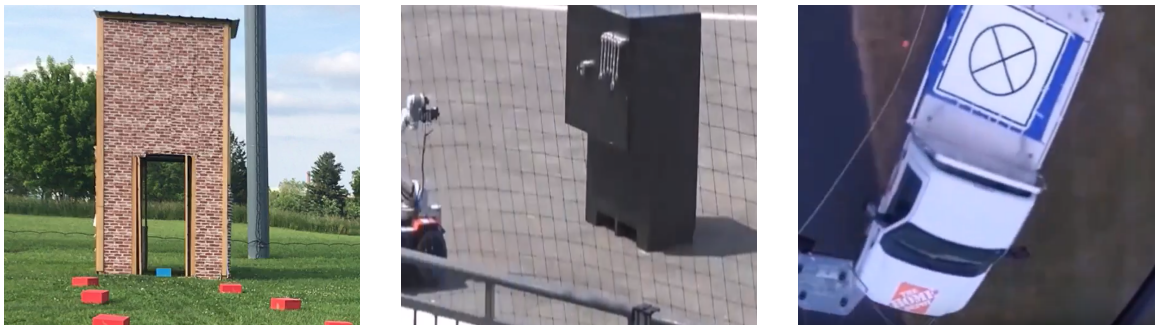
Figure 4.6: Targets and Landmarks

In fact, a target position needs to be estimated maximally meaning from the first observation. Although map features observations which are far from the robot could be ignored and the robot can rely on closest features, a target needs to be estimated from far for the robot to approach it by selecting motion actions. Those characteristics differentiate the target from the features and therefore change the localization technique to be implemented.

Also, a target being an object or a target region, might be completely different in its spatio-temporal characteristics. Its size might be much larger since it can be an object. The detection method might place only a bounded box region around the target space, defining a region rather than a specific feature. Moreover, it might only be a temporary feature and it can be moved after the robot approaches it to pick it up and place it in a different position.

The robot task is the next topic to discuss on the design of a target localization framework. As discussed previously, targets have spatial characteristics and those play a role for the engagement task. Moreover, they can be detected differently depending on the distance by which they are observed. As seen in figures 4.7, there are cases such as approaching a building structure with some target of interest inside it, grabbing a tool on larger board or landing on helipad marked by segments of different size. All these operations require different levels of precision based on how the target is detected. A large feature seen from far can be estimated with a good tracking method. This feature may no longer be observed when closer however. Moreover, it might not be an appropriate feature to use for robot localization when closer or might not be interesting since the final target location might be detected.

Defining as target region of interest (ROI) in those cases, the same region which involves all target of interest features, a multi-stage localization framework can be implemented to start from lower precision and move towards higher. If a higher level segmentation algorithm is implemented to first detect a target feature which is larger than the smaller area of final targeted position, the robot needs to localize first this element and then progress in smaller



(a) (b) (c)

Figure 4.7: Target regions defined by multiple levels of precision

regions. It can be more adequately expressed as an a-priori known detection and localization scheme by which the target can be estimated in multiple levels.

The process is shown in figure 4.8, depicting a multiple level of precision approach in which a target region is segmented by a detection algorithm and then applied to define what an ROI stands for. If those labels are recognized as higher or lower hierarchy based on our a-priori knowledge, the structure is created based on label architecture:

$$S_1 \subseteq S_2 \subseteq ...Sn$$

defining any labelled regions of interest.

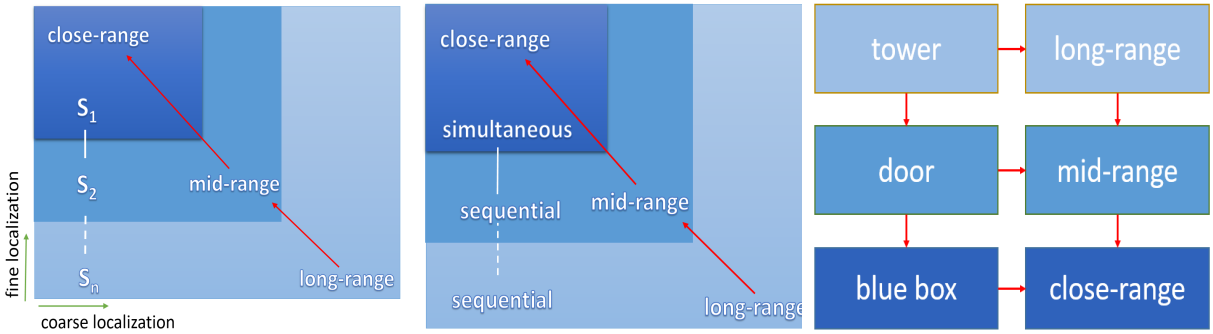


Figure 4.8: Hierarchical labelling and multi-stage localization scheme

A multi-stage approach as presented could be applied to those scenarios as a method to track any static object of temporary interest based on the distance of observation and the required level of precision. The sequential stage can be used to track features of the target region which seen from far can be good features for tracking whereas the simultaneous stage could be used to provide high precision with the area of the target region that the robot wants to engage with when closer.

Chapter 5

Experimental Analysis

5.1 Objectives

The experimental analysis will focus on three main parts.

First part:

- Evaluate the performance of the proposed multi-stage method for a distant located target from the start position of the robot, sections 4.1 and 4.2.1-4.2.4.
- Compare the approach with the previously proposed multi-stage method of section 3.4 in terms of accuracy and uncertainty estimation, while adding the landmarks to both stages in order to compare with the same setup of sensors. Comparison will be made for the sequential only stage formulations, where target is estimated using 3.12, since simultaneous will be identical.
- Compare the approach with the sequential only method, similarly to a tracking only method, using the proposed framework of chapter 4, without the simultaneous part, to show the effectiveness of simultaneous stage.

Second part:

- Demonstrate the ability to handle multiple targets under the same process.
- Demonstrate the ability to maintain belief during target field of view tracking loss.

Third part:

- Demonstrate the ability to localize a target region which is comprised of multiple levels of precision.
- Demonstrate the applicability of the technique, section 4.2.5, in cases of targets whose characteristics differ significantly from other landmarks.

5.2 Hardware & Software

The robotic vehicle used for the experiments is the Clearpath Robotics' Husky, as seen in figure 5.1, equipped with localization sensors. The Unmanned Ground Vehicle (UGV) used, is capable of traversing both indoor and outdoor environments and is mainly designed for outdoor rugged terrain. It features differential steering wheels and encoders that provide odometry information. For its ability to perform various robotics tasks, a robotic manipulator from Rethink Robotics, Sawyer, is also installed on the UGV.

(a)

Figure 5.1: Clearpath Robotics' Husky with installed robot and sensor suite

Proprioceptive sensors provide measurements internally to the system, such as battery level, joint angles, etc. The proprioceptive sensors used in this setup include the wheel encoders and an inertial measurement unit (IMU) of type UM7 from CH Robotics. Exteroceptive

(a) UM7 IMU

Figure 5.2: Proprioceptive Sensors

sensors refer to sensors that deal with environment observations such as cameras, lidars, radars, ultrasonic devices, sonar and others. The exteroceptive sensor suite installed on

the UGV includes the ZED stereo camera from Stereolabs, figure 5.3a which can provide resolution from VGA to 2K and a frame rate ranging from 15 to 100 fps. Its two cameras have a baseline of 120 mm and by specs with the onboard chipset, it can provide up to 30 m depth range. Its main advantage against other depth sensing cameras such as RGB-D cameras that project infrared light (IR), is that it can work very well outdoors, being a passive sensor. The setup further includes a 3D lidar from Quanergy Systems and a Garmin 18x OEM Global Positioning Sensor (GPS).

(a) ZED stereo camera (b) Quanergy 3D Lidar

Figure 5.3: Exteroceptive Sensors

The software used is based upon the Robot Operating System (ROS). ROS was first introduced in 2007 by Willow Garage. Rather than being an operating system, ROS provides services designed for a heterogeneous computer cluster. It can be used across different hardware platforms through its hardware abstraction, providing several packages and frameworks for various tasks such as control, sensor management, planning, actuation and other.

The version of ROS used, is the Kinetic, compatible with the operating system of Ubuntu v.16. For the low-level fusion of wheel odometry and IMU to provide what will be commonly referred to as odometry, the Robot-Localization package open-sourced by Charles River Analytics [36], will be used in the experiments. It can fuse arbitrary sensor information such as GPS, any odometry measurement and accelerometer or gyroscopic measurements. Therefore, it provides absolute robot position estimation and available fusion schemes include the Extended and Unscented Kalman Filters (EKF and UKF).

The package however does not include the ability to perform Simultaneous Localization and Mapping (SLAM), since it does not deal with observations relative to the robot. For that purpose a custom implementation of EKF-SLAM has been developed based on C++. The package is combined together with the custom EKF-SLAM package, based on the architecture of figure 5.4.

In the experiments, the stereo camera was used to provide range-bearing measurements to objects in the environment. In comparison to the lidar it provides detection in further distances, it is capable of recognizing objects faster and more reliably than the lidar in daytime conditions and since it is in a stereo setup, it can provide depth information.

Object detection is based on the Deep-Learning framework of "You Only Look Once: Unified,

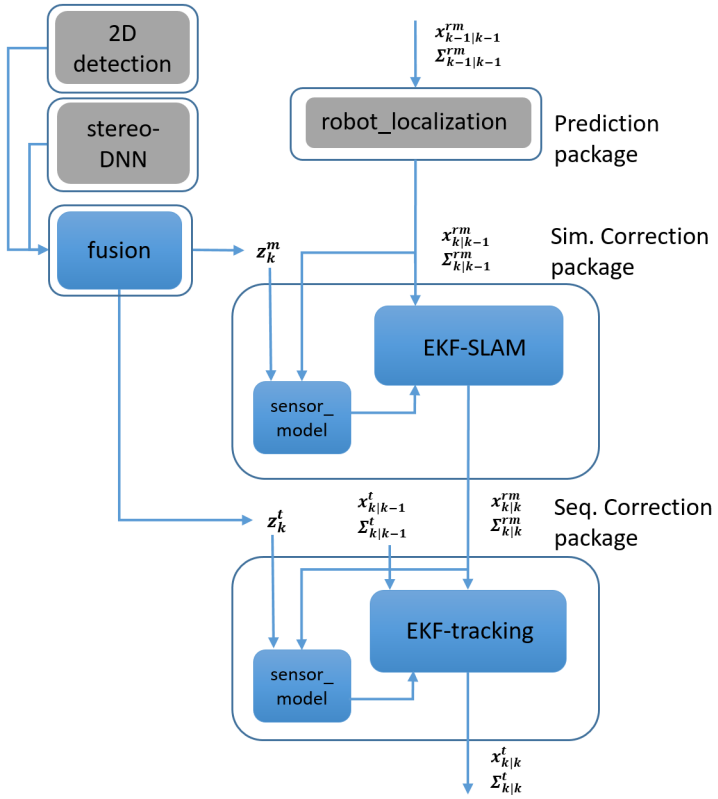
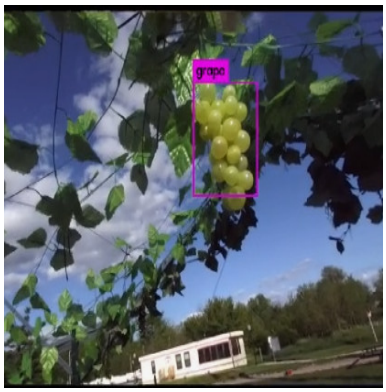


Figure 5.4: Overall software design

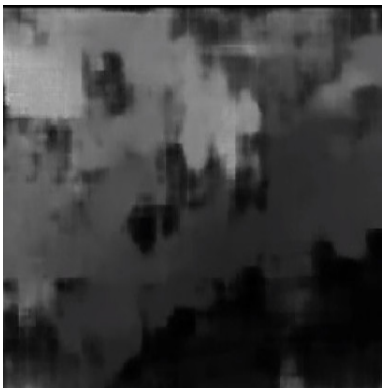
Real-Time Object Detection" [6], [42]. A set of images have been used for the objects to detected in the environment and which serve as landmarks. It is based on a single convolutional neural network that simultaneously predicts multiple bounding boxes and class probabilities for those boxes. It can detect objects at 45 frames per second. Training the network requires a set of images to be used as training set and another set as test set and both need to be manually labelled with the object of interest.

For the depth estimation, a visual technique called Stereo-DNN [48] is used. Feature extractors are applied to each one of the two image frames. Those features are used to extract disparity maps from both image pairs which are used as input images to a network that has been trained on lidar ground truth. The machine-learned output function is finally used to provide depth estimation. In comparison to other techniques such as OpenCV which rely only on feature extraction, the semi-supervised DNN approach provides depth estimation at much greater distances and with more accuracy.

Furthermore, a final equipment used for ground truth distance estimation, is a Garmin Lidar-Lite 3 laser rangefinder which can provide an accuracy of 2.5 cm and has a range of 0-40 m.



(a) Detection with Yolo v3



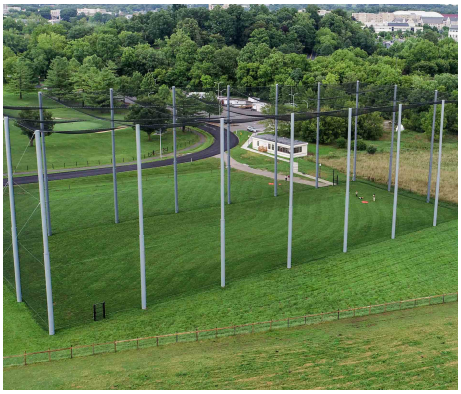
(b) Depth estimation with Stereo-DNN

Figure 5.5: Detection and depth estimation examples

5.3 Environment

The facility in which the experimental tests were performed is outdoors.

The purpose of testing outdoors is to design a robust localization system that can deal with noise and vibrations from rough terrain and under varying environment conditions. The outdoor environment is the Drone Park Facility in Virginia Tech, a netted structure on campus, which has a grass almost flat terrain. The facility also includes a tower structure, designed and built by the MBZIRC 2020 team Victor as part of the environment for the competition challenges, and is used in the third part of experiments as an extra localization feature.



(a) Drone park

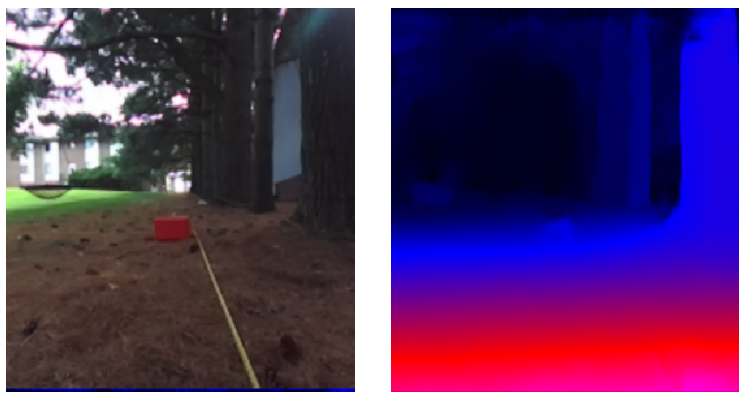
(b) Tower

Figure 5.6: Outdoor facilities

5.4 System Modelling

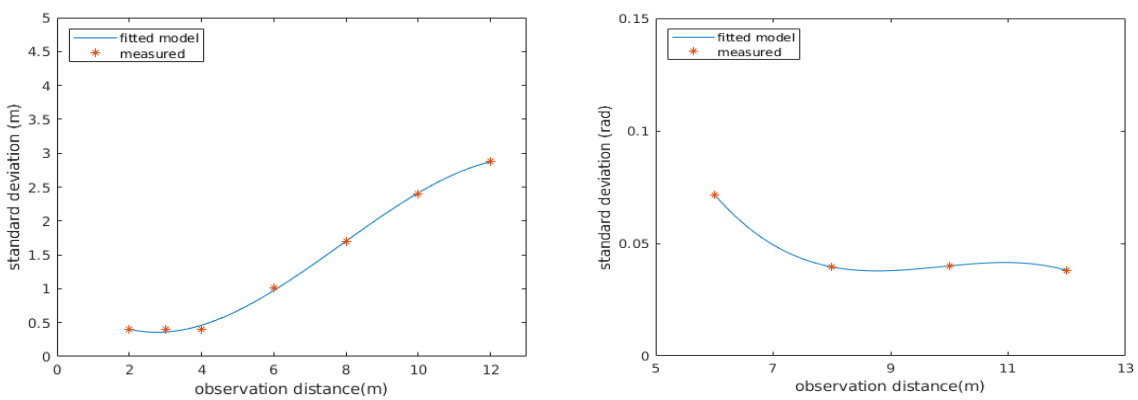
The choice of systems parameters is the core point of this section and will be used in the following ones as well. The EKF-SLAM framework requires the use of a sensor and motion model.

The value of the sensor noise was experimentally measured to match with the squared standard deviation of the measurements at different distances. In figures 5.8, the noise characteristics of range and bearing measurements. The standard deviation for each one of those distances has been estimated by looking at the global position measurements.



(a) Image (b) Depth

Figure 5.7: Measuring noise level for various distances



(a) Range (b) Bearing

Figure 5.8: Sensor model noise estimation and selection

It can be seen that varying the distance of measurement, varies the noise level as well. That is expected given that the sensor does not have the same quality in pixel information for longer distances.

Parameters	Values
Sensor Noise (m^2)	$[(\text{-}0.0049{*}\text{d}_{rt}^3 + 0.1151 * d_{rt}^2 - 0.5241 d_{rt} + 1.038)^2, 0; 0, 0.08^2]$;
System Noise (m^2)	[0.005,0.005,0.001] additive with every step
Detection	Bounding box
Depth	Centroid depth value of box
Stage selection (m)	distance robot-target= 5 m
Association radius for landmarks	1.5 m

Table 5.1: System and process parameters

The motion model noise is calibrated to counterbalance the effect of sensor measurements and odometry measurements. It is selected so that an accurate map with the close landmarks can be created after the sensor model selection. Its values are incrementally added to the robot covariance which is the only moving element, at the rate of odometry calculation. The value of prediction's squared standard deviation is increased by a rate such that after a run of 40 seconds the standard deviation is around 2 meters. The same motion model will be used in all tests. Also, the UGV moves at a constant velocity of around 0.5 m/s.

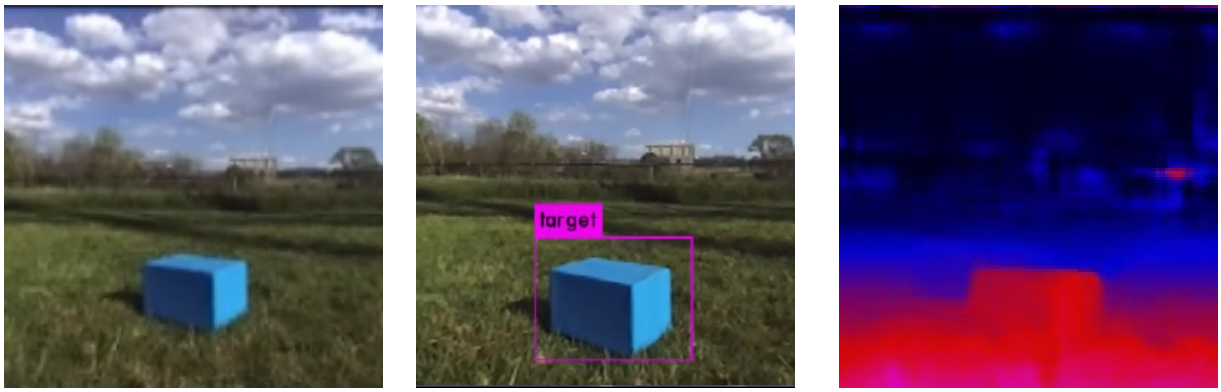
5.5 Framework Testing & Analysis

5.5.1 Engaging with single target

The first part of experiments is focused on the position estimation of a distant target. To be able to detect the target from a far distance, but also from the start position of the UGV, it has been placed at a distance of 15 to 25 m in all experiments. In this setup a single target has been placed whereas several landmarks are placed along the path that the UGV follows to approach the target.

For the robot localization, landmarks are being used to correct its position. For the target, a distinct and of different color object is used as seen in figure 5.10. The target is placed far from the rest of the landmarks which are equally apart along the path to the target. The target here is the blue box whereas the rest of landmarks are the red boxes. The front-end of SLAM is performed through the detection and depth estimation of the stereo camera.

A complete list of all system parameters must also define how data association is performed in this case. In this first set of experiments, the target is one single object to be localized from far. Thus, no data association is performed for it. For the landmarks, which are placed in distinct positions, a position-based association rule is applied so that observations belonging to a specific radius are used to identify one landmark. Otherwise, observations will be used to identify new landmarks. Table 5.1, shows the values used. Also, multi-stage



(a) Rectified camera-view (b) 2-D Detection (c) Depth estimation

Figure 5.9: Front-end vision based perception. Detection and depth registered on left-camera view

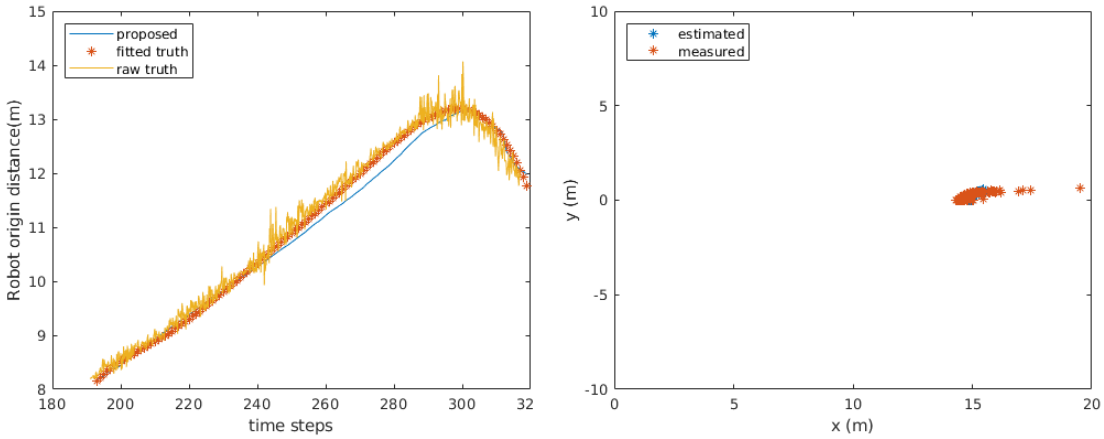
localization for the target can be applied when the robot is closer to it. This in this case has been adapted based on its size and total distance covered. The robot distance travelled as well as the target position with their estimated values are shown in figure 5.11a and figure 5.11b respectively.

(a) (b) (c)

Figure 5.10: Test A, first-person view

For the qualitative performance, a sequential map creation, figure 5.12, is shown to demonstrate the estimation process of the robot, map and target using their position and uncertainties. Those features help localize the UGV more accurately and improve odometry based localization. The target is seen from a long distance and as the vehicle gets closer, the uncertainty is reduced and target position is more accurately estimated. With red color are the landmarks or target if corrected simultaneously and with green color, only the target when corrected sequentially.

Since the target is located far initially, its position is only approximately known meaning



(a) Robot distance from origin (b) Target position

Figure 5.11: Robot and target positions

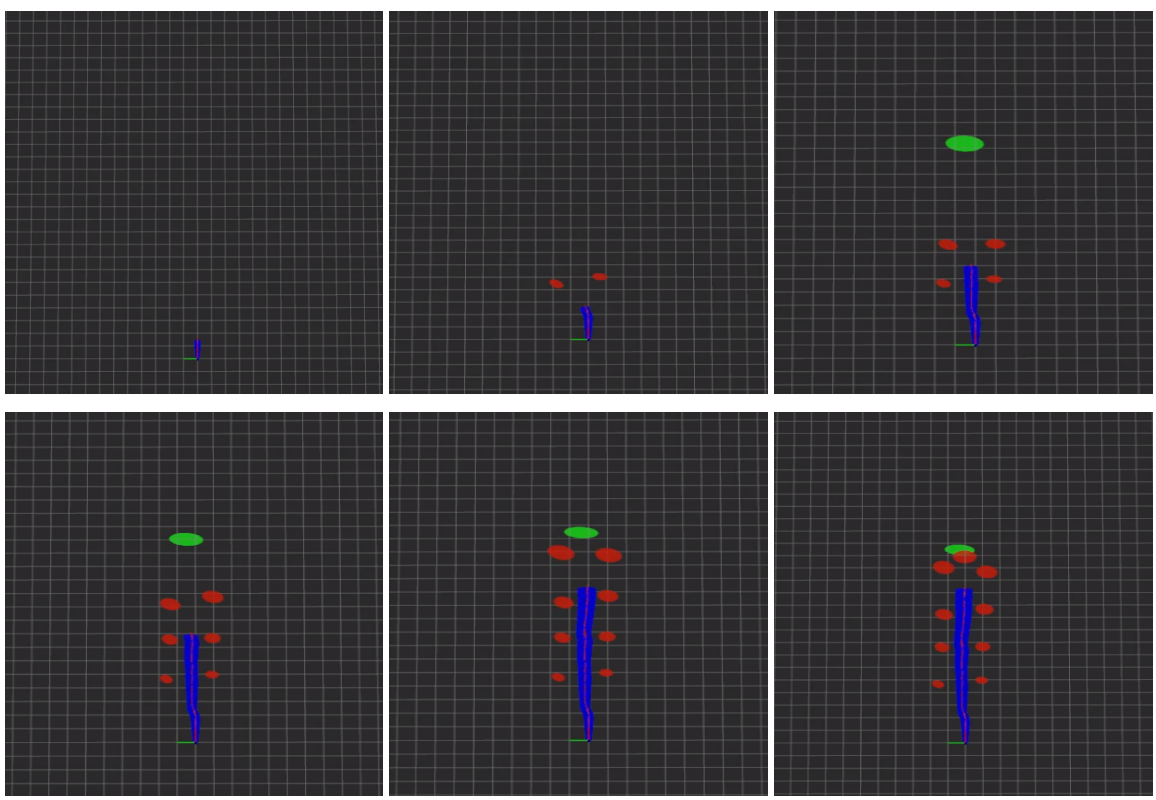
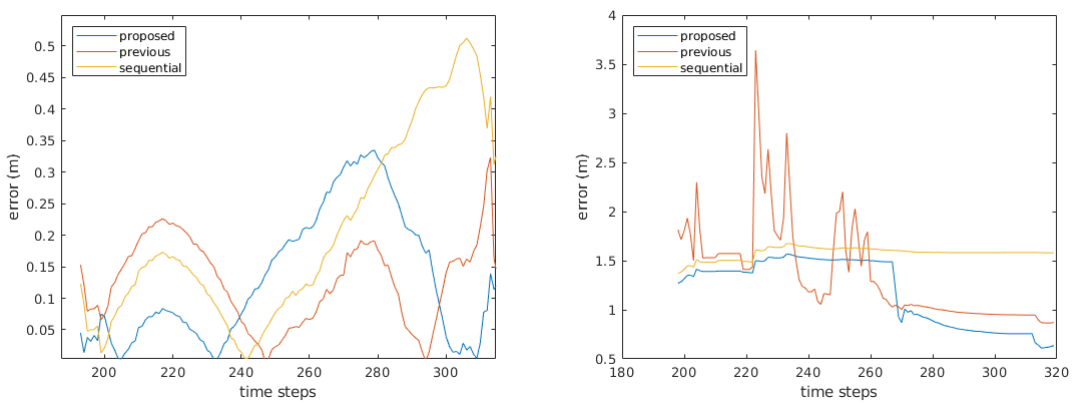


Figure 5.12: Progressive map creation

with less accuracy in comparison to close landmarks. However, as the robot progresses and finally gets closer, it can more accurately estimate it and use it as a landmark since not many other landmarks can be seen, due to close proximity to the target only.

Next, a comparison is made both qualitatively and quantitatively to compare all the approaches that relate to the framework and can be considered alternative solutions. One comparison is with the previously proposed framework 3.4, where although map features were not introduced, here they are added to ensure a common ground for comparison. Its main difference is the propagation method for uncertainties of the robot to the target. Also, the noise estimation is based on measured values in that previous method rather than the estimated ones in the proposed. Here, however, for comparison the estimated ones will be used, after determining that it is a more accurate approach. The second comparison is with the tracking principle applied while the vehicle position is corrected with landmarks. Although not definitely large differences are expected in terms of robot error, certainly uncertainty reduction will be the key. The proposed technique and previous as well as the tracking only method with globally corrected robot position are compared for the first scenario in terms of error 5.13 and entropy 5.14. , defined as $0.5 * ln((2\pi e)^n|\Sigma_{k|k}|)$.



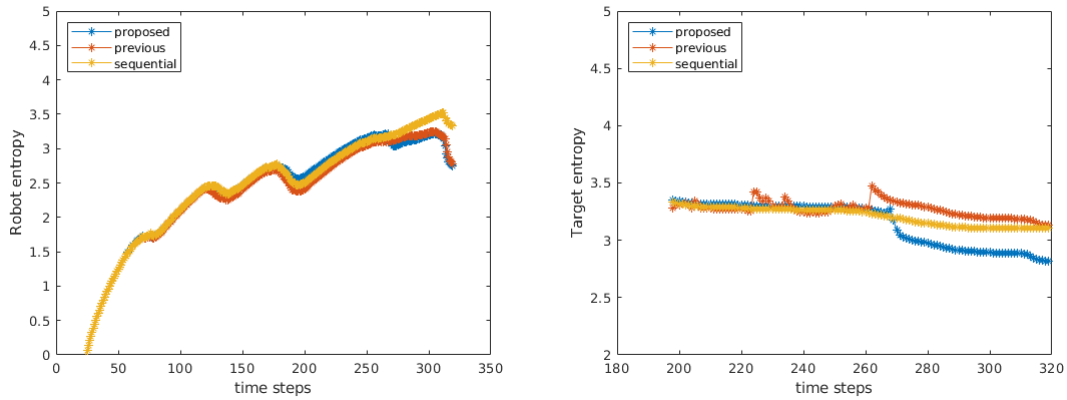
(a) Vehicle error

(b) Target error

Figure 5.13: State error vs time

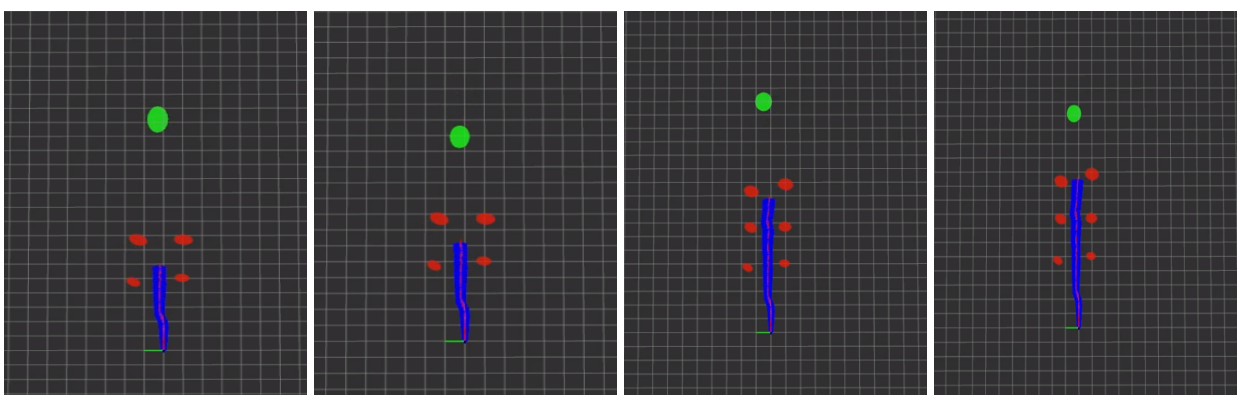
The results are shown in figures 5.14, 5.13. Starting the evaluation from the entropies, those results indicate lower uncertainty in the proposed technique. This is explained in comparison to the sequential technique, due to the reduction in robot uncertainty 5.14a, when doing simultaneous correction with one more element. The reduced robot uncertainty corrected with the target can lower its uncertainty as well as a result 5.14b. Also, the previous method is constantly adding the robot uncertainty and target uncertainty is not monotonically reduced.

When stage change happens in the multi-stage approach, the proposed with the previous framework come closer due to their common way of simultaneously correcting robot and target positions. However, the proposed technique ensures lower uncertainties throughout the whole process and produces less average and final errors 5.13b, mainly in the target estimation since the robot isn't much affected by one extra landmark. Also the error is lower than the sequential also.



(a) Vehicle entropy (b) Target entropy

Figure 5.14: Entropy vs time



(a) (b) (c) (d)

Figure 5.15: Map created progressively by previous approach

Estimating the target from a long distance sequentially ensures robust robot localization using close features and then accurate target estimation. The error remained lower in the case of the proposed technique by ensuring monotonic decrease in target estimation uncertainty as well as maximization of information usage when close to the target.

To further explain the difference between the proposed and previous technique, the information acquired both from robot correction and target correction is defined. The information gain is an information theoretic measure used to define the amount of novel information the system is able to get from measurements. It can be defined as:

$$im_{k|k-1} = -0.5 * ln((2\pi e)^n |\Sigma_{k|k-1}|)$$

$$im_{k|k} = -0.5 * ln((2\pi e)^n |\Sigma_{k|k}|)$$

$$IG_{k|k} = im_{k|k} - im_{k|k-1}$$

with $\Sigma_{k|k}$ the covariance of the state after a correction step, $\Sigma_{k|k-1}$ at the prediction step and information gain IG showing the reduction in uncertainty using any feature for correction using n states. Then:

$$I - K_k H_k = \begin{bmatrix} I_r & 0 \\ 0 & \mathrm{I}_t \end{bmatrix} - \begin{bmatrix} K_r \\ K_t \end{bmatrix} \begin{bmatrix} H_r \\ H_t \end{bmatrix}^T = \begin{bmatrix} I_r - K_r H_r & \text{-K}_r H_t \\ -K_t H_r & \mathrm{I}_t - K_t H_t \end{bmatrix}$$

where K^t and K^r. Thus the information gain following every target correction and showing the amount of new information with every correction is:

$$IG^t_{k|k} = 0.5 * ln\left(\frac{|\Sigma^t_{k|k-1}|}{|\Sigma^t_{k|k}|}\right) = 0.5 * ln\left(\frac{|\Sigma^t_{k|k-1}|}{|-K_t H_r \Sigma^{rt}_{k|k-1}{}^T + (I_t - K_t H_t)\Sigma^t_{k|k-1}|}\right)$$

In figure 5.16, the steady reducing information gain in comparison to the previous approach shows a more robust behavior in state estimation. Also, when the stage changes to simultaneous, information gain of the target is increasing due to the extra information by correcting robot position at the same time.

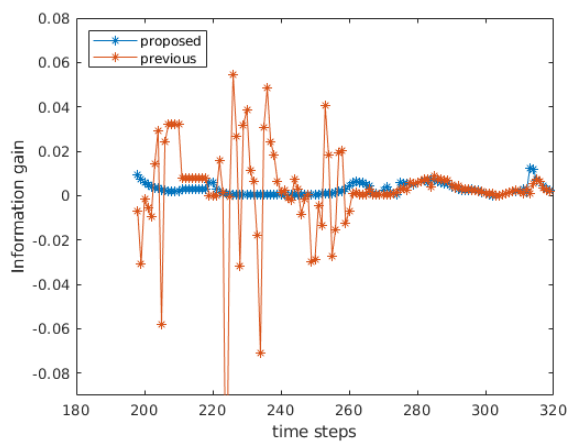
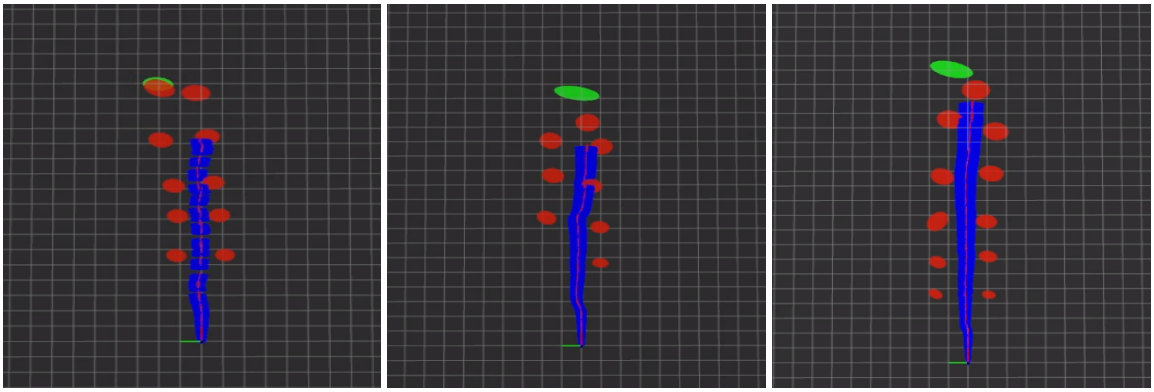


Figure 5.16: Information gain of previous and proposed method

To receive a more representative sample of data collection and evaluate the repeatability of the process, three more environments were tested again in the same field by randomizing the configuration as seen in figures 5.17. A single target is located at a further distance and stage selection happens as in the previous experiment. Now aggregated results can be derived and evaluate the validity of the framework. To do that, first a qualitative map creation is shown in figures 5.18 for each experiment, for the proposed technique.

(a) Test B (b) Test C (c) Test D

Figure 5.17: Different settings tested



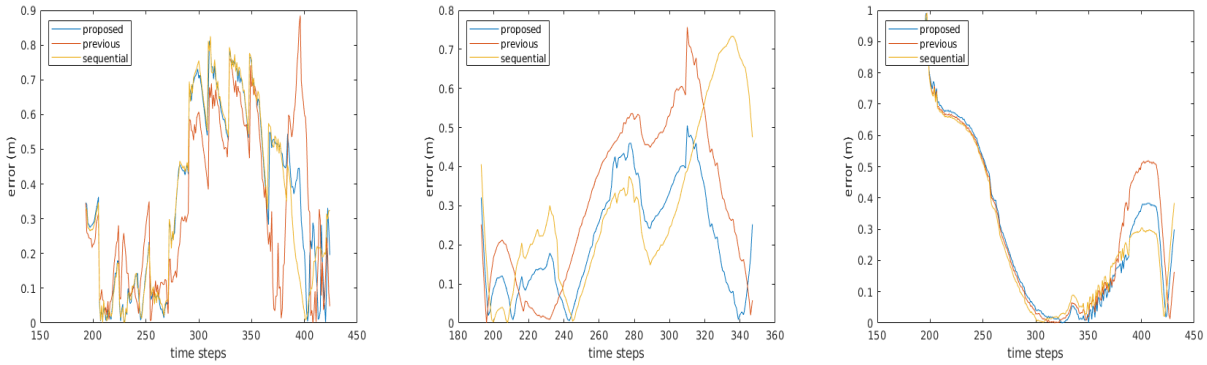
(a) Test B (b) Test C (c) Test D

Figure 5.18: Map created at each environment

Next for each case, metrics are shown similar to previously, regarding the error 5.19 of robot and target 5.20 as well as their entropy 5.21, 5.22.

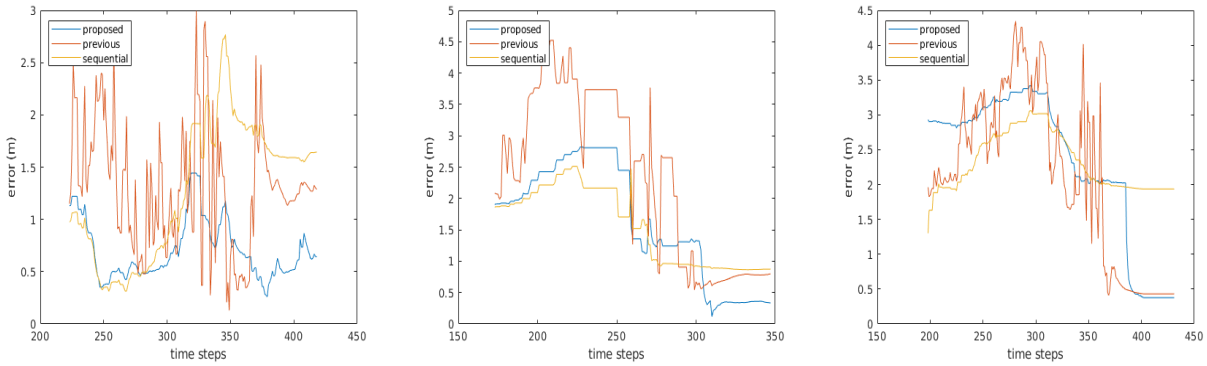
As predicted through the formulations, uncertainties remain lower in the proposed technique in comparison to the other ones. The error in all cases can be quantitatively evaluated by taking the average and final target errors. Thus, the average and final error for robot and target are shown in figures 5.23 and 5.24.

It can be noticed that the robot error as final and average values does not show great difference for each method due to its robust estimation with the other landmarks. The proposed method shows nevertheless the smallest values in error. Next, the target which is more affected is shown to have the best estimation performance in the proposed technique. Both average and final errors remained lower than the other two approaches. Average target error in the proposed method was up to 2 times less for errors less than 3 m. Final target error was similarly smaller, at 0.5 to 2 m for less than 2 m error. From a robot-centric perception of errors, namely the entropy of the target and robot also remain the lowest in all tests for the proposed method. Next, in tests C and D that the target errors in terms of



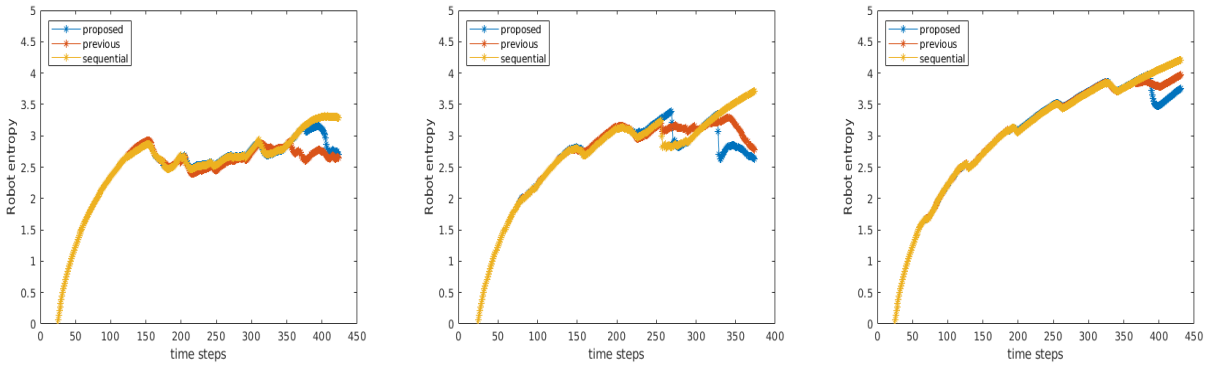
(a) Test B (b) Test C (c) Test D

Figure 5.19: Robot error



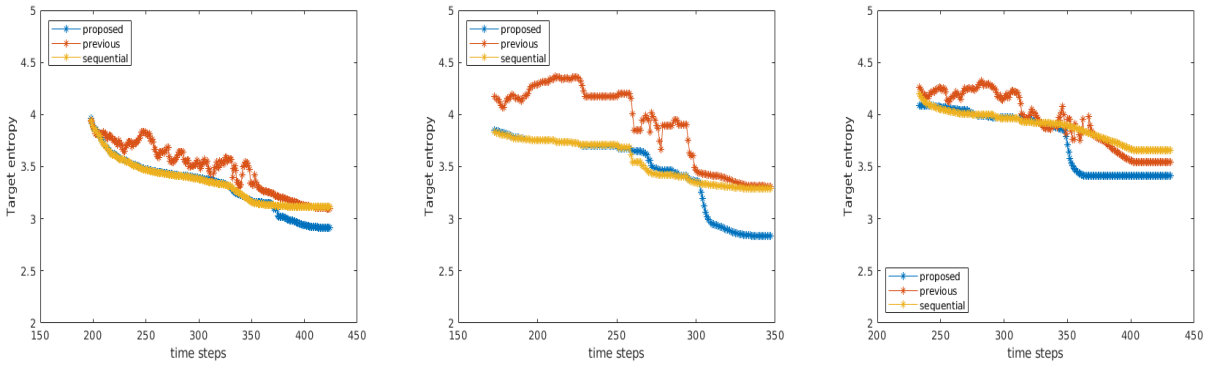
(a) Test B (b) Test C (c) Test D

Figure 5.20: Target error



(a) Test B (b) Test C (c) Test D

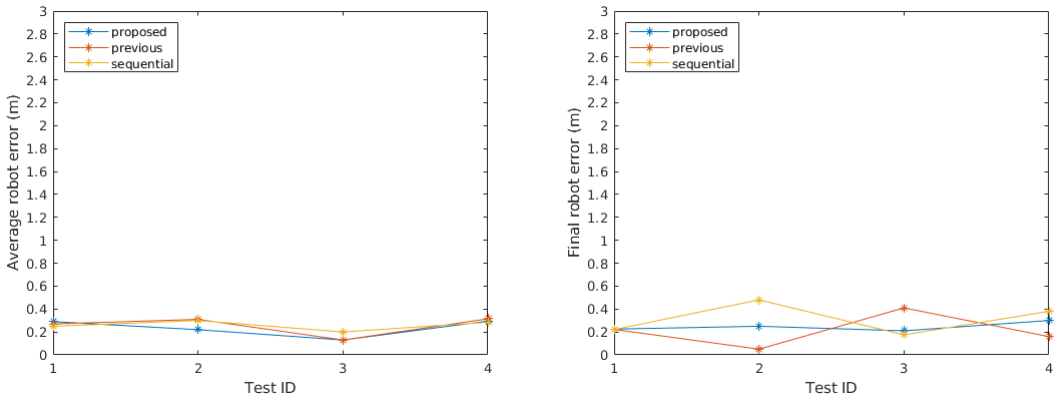
Figure 5.21: Robot entropy



(a) Test B (b) Test C (c) Test D

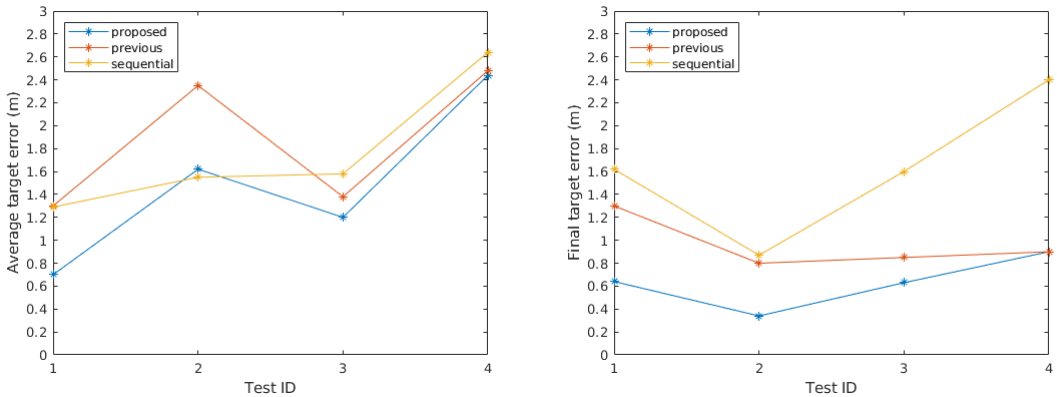
Figure 5.22: Target entropy

average and final are very close for the proposed and previous approach, the error shows much more stable reduction than the constantly increasing and decreasing error in the previous approach. The proposed method results therefore in error reduction, uncertainty reduction as well as more stable and robust target state estimation.



(a) Average (b) Final

Figure 5.23: Robot error



(a) Average (b) Final

Figure 5.24: Target error

5.5.2 Engaging with multiple targets

Next, the implemented framework is tested for the localization of several targets which need to be located from far. That can be seen as the case of having to engage with targets in different locations, while the need to maintain tracking state is necessary since the field of view of the robot can change and a target will not be able to be tracked in the image frame. The setup involves three distinct targets located in separate locations.

Figure 5.25: Test environment

An important difference from the previous case is that they also have to be associated like the rest of landmarks. A similar association rule is formed based on the distance, while to simplify the problem, they are separated by a longer distance than the landmarks. The tested environment can be seen in figure 5.25.

The figures below illustrate how within the tracking and SLAM framework, multiple target locations can be estimated. Far-located targets can be tracked while the vehicle is estimating its position using the closest landmarks. As it can be seen in figure 5.26, targets are seen but since the robot is moving they are very soon lost from the field of view.

The robot is however able to estimate their location using the proposed tracking technique instead of some other visual tracking method which only temporarily tracks their location. Uncertainty is maintained this way as seen in figure 5.27.

The progressive map creation as well as the uncertainties reduction shows how belief is maintained. Localization of the targets with the proposed technique remains accurate and targets have the same properties as landmarks with their belief maintained throughout the whole localization process. That means their uncertainty is reduced always, they can be associated when re-observed from different viewpoints and finally when the robot is closer to each of those, it corrects its position.

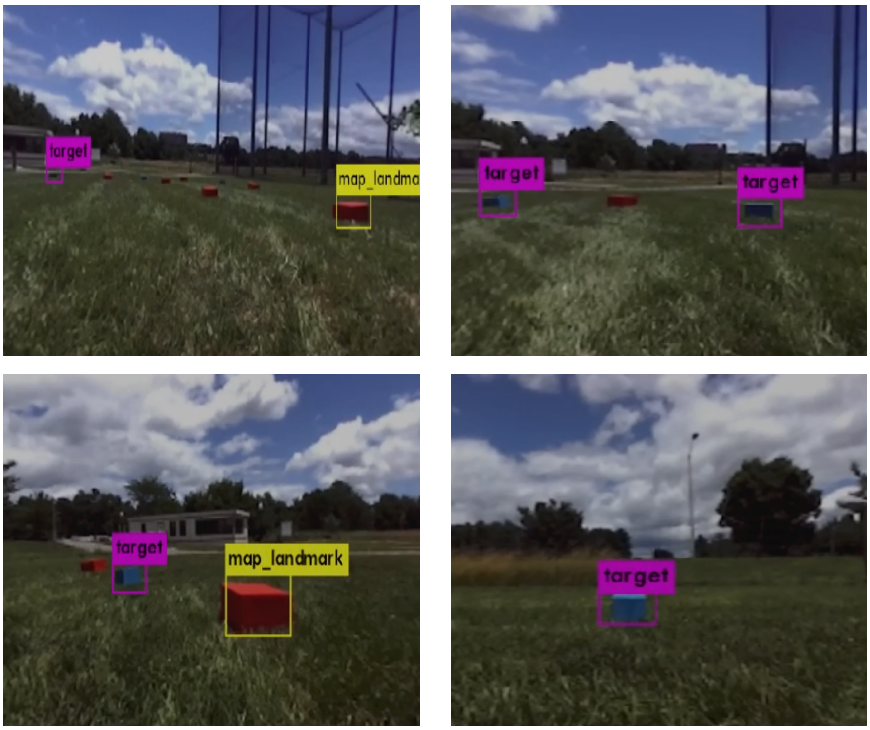


Figure 5.26: Camera view

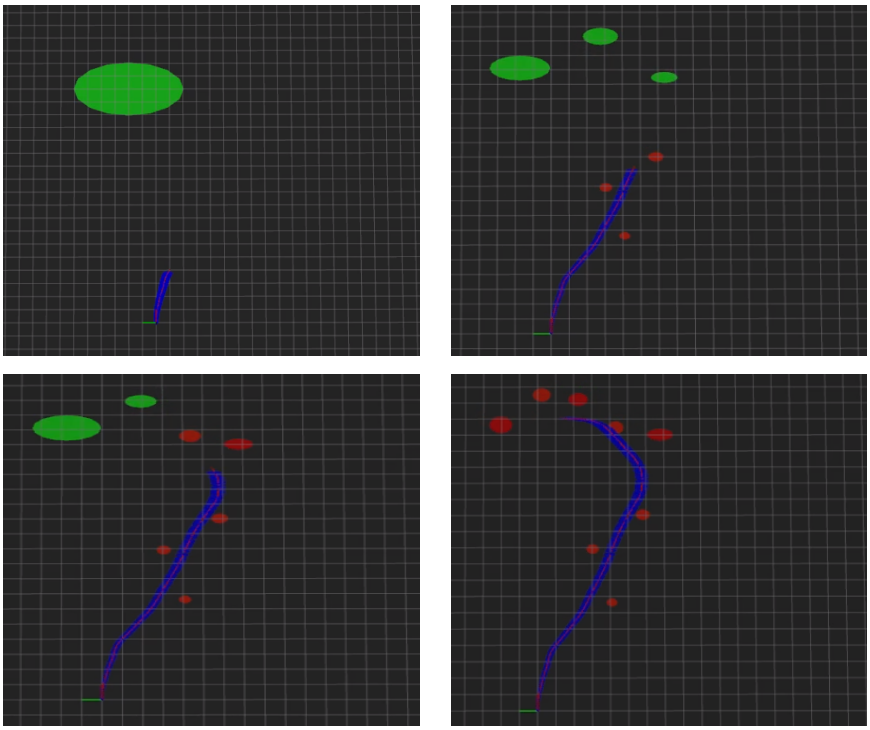
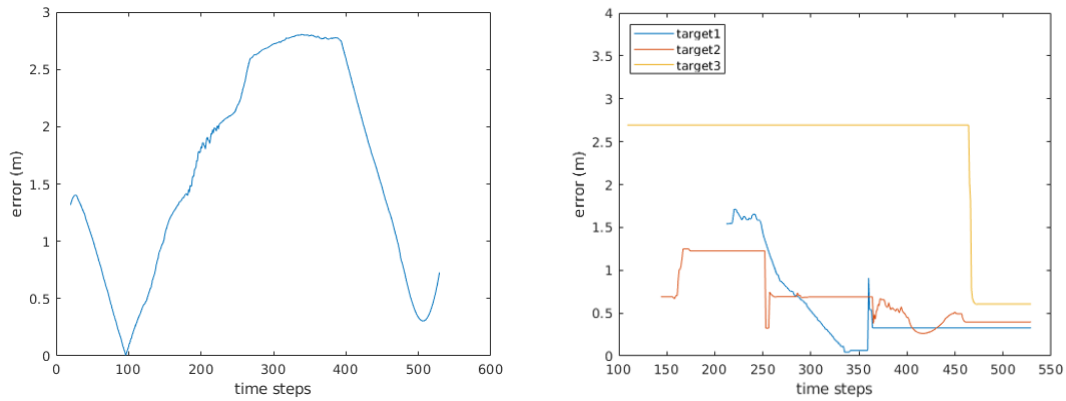


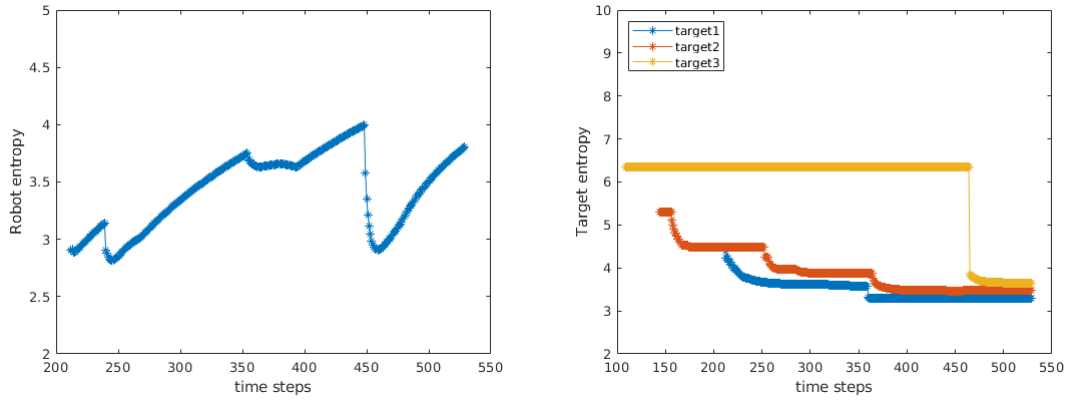
Figure 5.27: Progressive map creation

The robot and targets error and entropies are shown in figures 5.28 and 5.29. It can be observed that robot uncertainty is reduced a lot when observing the target which was seen in longer distance and then corrected simultaneously when being close, since its position is more certain. Target uncertainties are also decreased continuously together with their error.



(a) Robot error (b) Targets error

Figure 5.28: Error



(a) Robot entropy (b) Targets entropy

Figure 5.29: Entropies

5.5.3 Engaging with a target region

The proposed extended framework was also implemented on a target region. The target in this study is considered to be consisting of different regions which can be localized sequentially using label information to derive the sequence for engaging.

This sequence is defined based on semantic labelling and the setup is tested under two different scenarios. First a large region is defined to represent the object with larger size, seen from far. This can be easily detected from a long range. Next, this feature covers a large area of the field of view and loses its potential to be localized efficiently either because of no detection, close distance detection noise due to object size, our interest to keep observing it or because of its properties change such as see through openings.

Two different settings were tested and seen in figures 5.30.

(a) Test A (b) Test B

Figure 5.30: Tested Environments

In the first one, the semantic labels to localize sequentially are seen in figure 5.31a and for the second one in 5.31b. The first setting is for a building rescue scenario simulation while the second one simulates a robot approach a board to pick up a tool.

The framework applies sequential localization to the larger region of the target as detected from far first. The target's entropy is reduced following each correction. Since the targeted region for engagement can be situated in a displaced position of the area, uncertainty is reinitialized allowing for robust estimation. The simultaneous correction takes place when the smaller region is detected and by that point, the robot is also located closer. Since the larger object area can be more noisy by being located far, the error is expected to be larger in the first step but indicates the coarser precision localization. The smaller region should give more precision and less error.

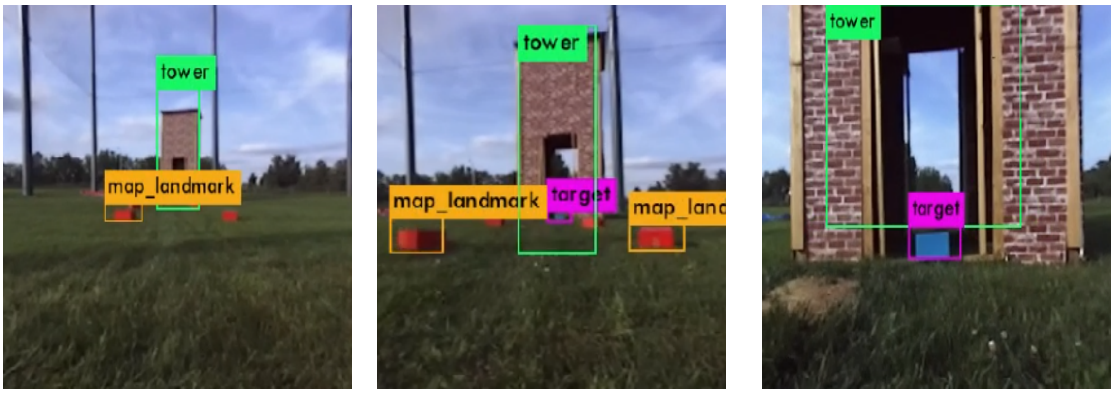
The different views from the camera with the detection of the several features are shown in

(a) Test A (b) Test B

Figure 5.31: Region labelling

figures 5.32 and 5.33. The created map in each case is shown in figure 5.34. The error and entropies are shown in figures 5.35 and 5.36. As predicted, the target error is greatly reduced when the smaller region is detected due to its smaller size and characteristics providing better detection and more accuracy. The larger region detection and localization is however enabling longer estimation and state estimation for a long to close distance target approach.

The technique is able to achieve maximum information gain for the target location going beyond the traditional SLAM framework which considers all static objects to be landmarks. The advantage to that is that the target can be localized based on its properties from different distances while the robot relies on robust and stable environment features to achieve precision localization. The framework with sequential tracking method can be applied to localize targets which in contrast to features have a geometry which can be localized in different levels according to each target object and the distance from which it is observed.

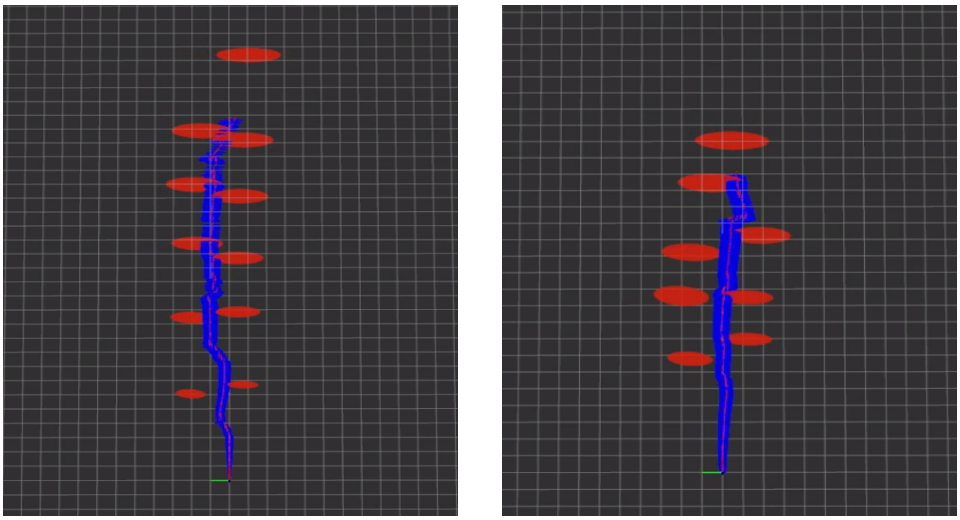


(a) (b) (c)

Figure 5.32: Test A, first person view

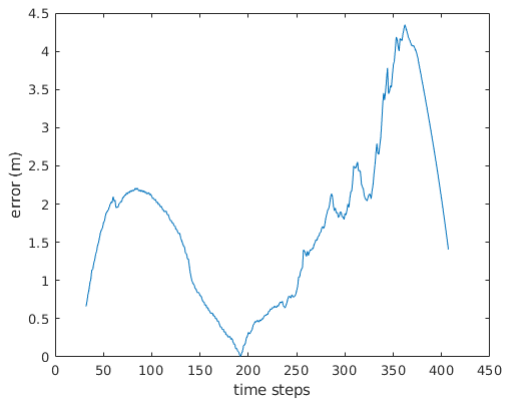
(a) (b) (c)

Figure 5.33: Test B, first person view

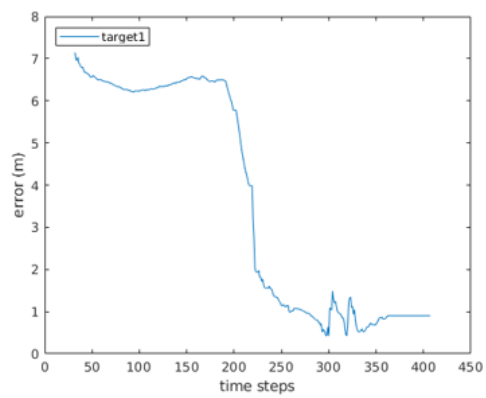


(a) Test A (b) Test B

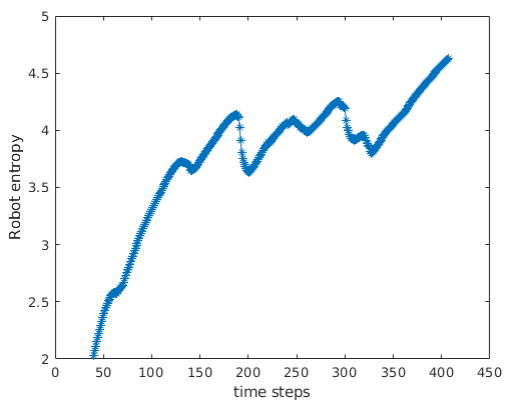
Figure 5.34: Created map



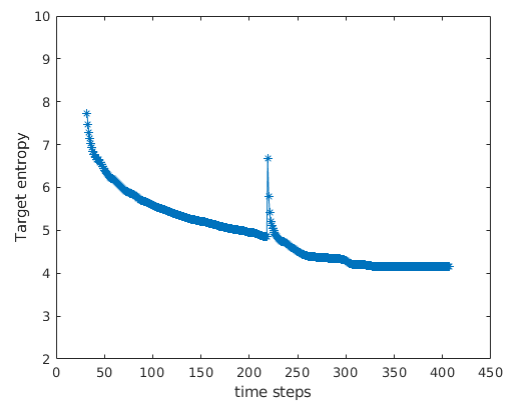
(a) Robot error



(b) Target error

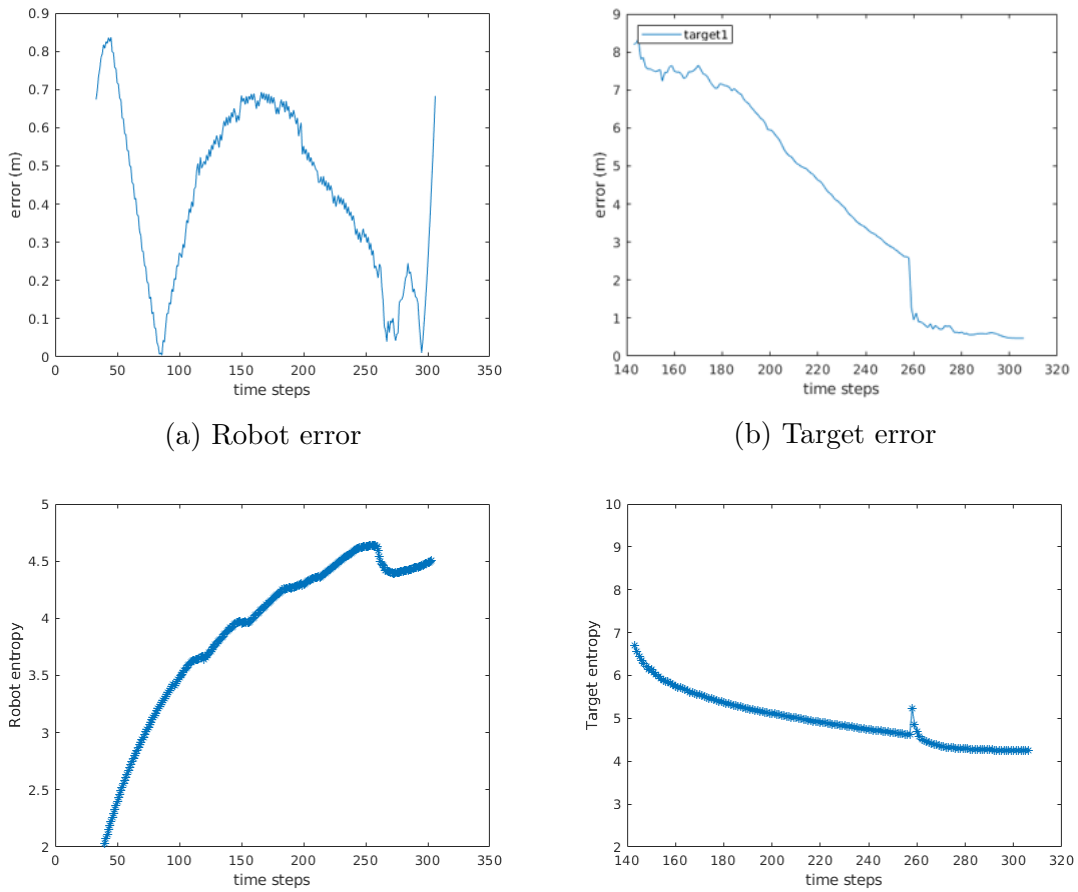


(c) Robot entropy



(d) Target entropy

Figure 5.35: Test A, error and entropies



(a) Robot error

(b) Target error

(c) Robot entropy

(d) Target entropy

Figure 5.36: Test B, error and entropies

Chapter 6

Summary

The goal of this thesis was to propose and evaluate a framework by which a robot can achieve self, map and target localization. The common formulation of mapping and tracking with the usage of the EKF enables the practical application of a multi-stage technique so that localizing a targeted feature can be done either using the sequential method, tracking, or the simultaneous method, SLAM. The technique has several advantages such as allowing the robot to select a stage based on the target characteristics and its mission for engagement. In comparison to temporary tracking methods, such as image tracking, belief for target location can be achieved and maintained maximally.

The proposed framework was implemented on settings in which the target of interest is located far from the point of initial detection by the robot. That was applied in the case of a single target, multiple targets as well as a target structure which involves labelled target regions detected at different distances. The experimental results showed the effectiveness in achieving low error and maintaining low uncertainty for the robot and its mapped surroundings. The final errors remained small and stable for the robot as well as the target. A comparison was made also with the multi-stage framework proposed in a previous work, where the necessity to propagate robot uncertainty in the state estimation of the target based on the new formulation was discovered. The experimental analysis proved also that using the simultaneous method, when the target is located far increases almost certainly the accuracy.

Further analysis can however be done to ensure that the operation can accommodate more complex scenarios. Enhancing that capability with the fusion of more sensors such as lidar and radar would provide more accurate localization in more environments. Moreover, the development of improved depth estimation algorithms with cameras, would further enhance the perception capability. Additional capabilities such as 3D localization of the features or the target can be embedded into the system, while applying information-based control for the robot. Applying the technique in real-world settings could make the proof of concept achieve the level of practicality to allow robots to operate in complex environments using perception and localization as the backbone of their autonomous capabilities.

Bibliography

[1] Data association in slam. https://web.mit.edu/2.166/www/handouts/2166_data_association_jose_neira.pdf. 2019.

[2] Mbzirc. https://www.mbzirc.com/challenge/2020.

[3] NSF. https://www.nsf.gov/pubs/2014/nsf14500/nsf14500.htm, 2019.

[4] A. Asvadi, P. Girão, P. Peixoto, and U. Nunes. 3d object tracking using rgb and lidar data. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1255–1260, Nov 2016. doi: 10.1109/ITSC.2016.7795718.

[5] Mordechai Ben-Ari and Francesco Mondada. *Robots and Their Applications*, pages 1–20. Springer International Publishing, Cham, 2018. ISBN 978-3-319-62533-1. doi: 10.1007/978-3-319-62533-1_1. URL https://doi.org/10.1007/978-3-319-62533-1_1.

[6] Marko Bjelonic. YOLO ROS: Real-time object detection for ROS. https://github.com/leggedrobotics/darknet_ros, 2016–2018.

[7] Frédéric Bourgault, Tomonari Furukawa, and Hugh F. Durrant-Whyte. *Optimal Search for a Lost Target in a Bayesian World*, pages 209–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-32854-4. doi: 10.1007/10991459_21. URL https://doi.org/10.1007/10991459_21.

[8] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic slam. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, May 2017. doi: 10.1109/ICRA.2017.7989203.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, Dec 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2624754.

[10] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[11] Zhaowei Cai, Quanfu Fan, Rogerio S. Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 354–370, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

[12] F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics Automation Magazine*, 13(4):82–90, Dec 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.250573.

[13] François Chaumette, Seth Hutchinson, and Peter Corke. *Visual Servoing*, pages 841–866. Springer International Publishing, Cham, 2016. ISBN 978-3-319-32552-1. doi: 10.1007/978-3-319-32552-1_34. URL https://doi.org/10.1007/978-3-319-32552-1_34.

[14] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. *CoRR*, abs/1611.07759, 2016. URL http://arxiv.org/abs/1611.07759.

[15] Dmitry Chetverikov, Sándor Fazekas, and Michal Haindl. Dynamic texture as foreground and background. *Machine Vision and Applications*, 22(5):741–750, Sep 2011. ISSN 1432-1769. doi: 10.1007/s00138-010-0251-6. URL https://doi.org/10.1007/s00138-010-0251-6.

[16] Chieh-Chih Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 3, pages 2918–2924 vol.3, May 2002. doi: 10.1109/ROBOT.2002.1013675.

[17] Jaebum Choi, Simon Ulbrich, Bernd Lichte, and Markus Maurer. Multi-target tracking using a 3d-lidar sensor for autonomous vehicles. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 881–886, 2013.

[18] D. Chwa, A. P. Dani, and W. E. Dixon. Range and motion estimation of a monocular camera using static and moving objects. *IEEE Transactions on Control Systems Technology*, 24(4):1174–1183, July 2016. ISSN 1063-6536. doi: 10.1109/TCST.2015.2508001.

[19] D. E. Clark, J. Bell, Y. de Saint-Pern, and Y. Petillot. Phd filter multi-target tracking in 3d sonar. In *Europe Oceans 2005*, volume 1, pages 265–270, June 2005. doi: 10.1109/OCEANSE.2005.1511723.

[20] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1197–1203 vol.2, Sep. 1999. doi: 10.1109/ICCV.1999.790416.

[21] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001. ISSN 1042-296X. doi: 10.1109/70.938381.

[22] El-Mane Wong, F. Bourgault, and T. Furukawa. Multi-vehicle bayesian search for multiple lost targets. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3169–3174, April 2005. doi: 10.1109/ROBOT.2005.1570598.

[23] T. Furukawa, F. Bourgault, B. Lavis, and H. F. Durrant-Whyte. Recursive bayesian search-and-tracking using coordinated uavs for lost targets. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2521–2526, May 2006. doi: 10.1109/ROBOT.2006.1642081.

[24] Tomonari Furukawa, Gamini Dissanayake, Tamer Attia, and Jonathan Hodges. A bayesian framework for simultaneous robot localization and target detection and engagement. 10 2018. doi: 10.1109/IROS.2018.8593747.

[25] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, winter 2010. ISSN 1939-1390. doi: 10.1109/MITS.2010.939925.

[26] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. . Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437, Feb 2002. ISSN 1053-587X. doi: 10.1109/78.978396.

[27] T. S. Ho, Y. C. Fai, and E. S. L. Ming. Simultaneous localization and mapping survey based on filtering techniques. In *2015 10th Asian Control Conference (ASCC)*, pages 1–6, May 2015. doi: 10.1109/ASCC.2015.7244836.

[28] Jonathan Hodges, Tamer Attia, Janindu Arukgoda, Changkoo Kang, Mickey Cowden, Luan Doan, Ravindra Ranasinghe, Karim Abdelatty, Gamini Dissanayake, and Tomonari Furukawa. Multistage bayesian autonomy for high-precision operation in a large field. *Journal of Field Robotics*, 36(1):183–203. doi: 10.1002/rob.21829. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21829.

[29] F. Iida. Autonomous robots: From biological inspiration to implementation and control. george a. bekey. (2005, mit press.) hardcover, 577 pages. isbn 0262025787. *Artificial Life*, 13(4):419–421, Oct 2007. ISSN 1064-5462. doi: 10.1162/artl.2007.13.4.419.

[30] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000. ISSN 0162-8828. doi: 10.1109/34.868688.

[31] M. Kusenbach, M. Himmelsbach, and H. Wuensche. A new geometric 3d lidar feature for model creation and classification of moving objects. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 272–278, June 2016. doi: 10.1109/IVS.2016.7535397.

[32] K. Lin and C. Wang. Stereo-based simultaneous localization, mapping and moving object tracking. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3975–3980, Oct 2010. doi: 10.1109/IROS.2010.5649653.

[33] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004. ISSN 1573-1405. doi: 10.1023/B:VISI.0000029664.99615.94. URL https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[34] Krystian Mikolajczyk and Cordelia Schmid. An affine invariant interest point detector. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision — ECCV 2002*, pages 128–142, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-47969-7.

[35] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 2, pages 1985–1991 vol.2, Sep. 2003. doi: 10.1109/ROBOT.2003.1241885.

[36] T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[37] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31:1147–1163, 2015.

[38] Bahareh Nakisa, Mohammad Naim Rastgoo, and Mohd Zakree Ahmad Nazri. Target searching in unknown environment of multi-robot system using a hybrid particle swarm optimization. *Journal of Theoretical and Applied Information Technology*, 96(13):4055–4065, 2018.

[39] M. Nomatsu, Y. Suganuma, Y. Yui, and Y. Uchimura. Development of an autonomous mobile robot with self-localization and searching target in a real environment. *Journal of Robotics and Mechatronics*, 27(4): 356–364, 2015. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84940736904&partnerID=40&md5=ebbad804433426f85b0ca81094bebf25. cited By 3.

[40] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, Jun 2000. ISSN 1573-1405. doi: 10.1023/A:1008162616689. URL https://doi.org/10.1023/A:1008162616689.

[41] C. Premebida, J. Carreira, J. Batista, and U. Nunes. Pedestrian detection combining rgb and dense lidar data. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4112–4117, Sep. 2014. doi: 10.1109/IROS.2014.6943141.

[42] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[43] Cyril Robin and Simon Lacroix. Multi-robot target detection and tracking: Taxonomy and survey. *Autonomous Robots*, 40, 08 2015. doi: 10.1007/s10514-015-9491-7.

[44] Koichi Sato and J.K. Aggarwal. Aggarwal, j.: Temporal spatio-velocity transform and its application to tracking and interaction. comput. vis. image underst. 96(2), 100-128. *Computer Vision and Image Understanding*, 96:100–128, 11 2004. doi: 10.1016/j.cviu.2004.02.003.

[45] C. Shen, Y. Zhang, Z. Li, F. Gao, and S. Shen. Collaborative air-ground target searching in complex environments. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 230–237, Oct 2017. doi: 10.1109/SSRR.2017.8088168.

[46] Jianbo Shi and Carlo Tomasi. Good features to track. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 600, 03 2000. doi: 10.1109/CVPR.1994.323794.

[47] Bruno Siciliano and Oussama Khatib. *Springer Handbook of Robotics*. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 354023957X.

[48] Nikolai Smolyanskiy, Alexey Kamenev, and Stan Birchfield. On the importance of stereo for accurate depth estimation: An efficient semi-supervised deep neural network approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1007–1015, 2018.

[49] Hauke Strasdat, J.M.M. Montiel, and Andrew J. Davison. Visual slam: Why filter? *Image and Vision Computing*, 30(2):65 – 77, 2012. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2012.02.009. URL http://www.sciencedirect.com/science/article/pii/S0262885612000248.

[50] T. Vik, F. Heitz, and P. Charbonnier. Mean shift-based bayesian image reconstruction into visual subspace. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 1, pages I–697, Sep. 2003. doi: 10.1109/ICIP.2003.1247057.

[51] S. Wangsiripitak and D. W. Murray. Avoiding moving outliers in visual slam by tracking moving objects. In *2009 IEEE International Conference on Robotics and Automation*, pages 375–380, May 2009. doi: 10.1109/ROBOT.2009.5152290.

[52] Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. Video object tracking using adaptive kalman filter. *Journal of Visual Communication and Image Representation*, 17

(6):1190 – 1208, 2006. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2006.03.004. URL http://www.sciencedirect.com/science/article/pii/S1047320306000113.

[53] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38:13, 2006.

[54] Yu-Chi Lin, Shao-Ting Wei, Shih-An Yang, and Li-Chen Fu. Planning on searching occluded target object with a mobile robot manipulator. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3110–3115, May 2015. doi: 10.1109/ICRA.2015.7139626.

[55] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *CoRR*, abs/1711.06396, 2017. URL http://arxiv.org/abs/1711.06396.

[56] O. Zoidi, N. Nikolaidis, and I. Pitas. Appearance based object tracking in stereo sequences. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2434–2438, May 2013. doi: 10.1109/ICASSP.2013.6638092.