Diplomarbeit

Towards an Hierarchical Kalman Filter Approach to Robot Localisation and Mapping

By Jeannette Bohg
Born on 15th March 1981 in Cottbus

TECHNISCHE
UNIVERSITÄT
DRESDEN

Submitted on 31st December 2005
Overseeing Professor: Steffen Hölldobler
Supervisor: Tobias Pietzsch

Contents

Chapter 1

Introduction

The ability of a mobile robot to negotiate its way in an unknown environment largely determines its possible application and performance. The related research area is known as *Simultaneous Localisation and Mapping* (SLAM). Leonard and Durrant-Whyte [28] coined this term for the task of a mobile robot to continuously build a map of an unknown static environment and to use this map at the same time to localise itself. The resulting information is based on the measurements of fixed landmarks or features in this environment obtained by the sensor devices mounted on the robot.

Our long term goal is to develop a SLAM system capable of processing in real time. It should facilitate conventional cameras as senor devices. Its field of application should be indoor space as, for example, an office environment.

Approaching this problem can be divided into the subtasks of developing the following three methods

1. A real-time SLAM technique that determines 3D point features explicitly associated with certain objects in the scene.
2. A segmentation method that subsumes 3D point features to surfaces. These surface can then be used to represent objects.
3. A localisation method that can facilitate features of higher complexity than points.

This work addresses solely the first problem which has to be solved before the two others can be approached.

The most common approach to SLAM is to use the *Extended Kalman Filter* (EKF). It was firstly applied by Smith et al. [42]. Here, the setting of a mobile robot moving within a static environment is seen as a *dynamical system*. The positions of the robot and of the landmarks within a world coordinate system at a particular point in time are summarised in the term of the system's *state*. The task is to optimally estimate the new state for the next point in time in the sense that the possible error with respect to the unknown real state is minimised. The old state, the last motion of the vehicle and the new observations provided by some sensor device contribute to the estimation. We need to model the robot's motion and the process of measuring landmarks to obtain accurate estimates for the new state.

The EKF involves the error concerning the position of the robot and of the landmarks as well as the discrepancy between the models and reality. These errors and uncertainties are modelled stochastically as noise or covariance matrices, respectively. They are assumed to be independent and to have a Gaussian profile.

EKF based SLAM appeared to be a robust and powerful method to approach the area of application in which we are interested. This was shown in a number of research work as, for example, by Davison [8, 9], Knight [26].

Nevertheless, besides its advantages, EKF SLAM suffers from some drawbacks. Due to the linearisation of the mentioned motion and observation models, the robustness and correctness of the algorithm can be affected by linearisation errors. However, the major problem of EKF SLAM is its poor scalability. Its time and space complexity is quadratic in the total number n of features maintained in the map. This is due to the fact that the whole $n \times n$ covariance matrix has to be updated each time after involving a new measurement in the related state estimate. Practically this means that only a few hundred feature points can be maintained by the filter such that it is still capable of processing in real-time.

The problem of enhancing the EKF-based SLAM approach in terms of efficiency is considered by many researchers in the SLAM community. The approaches can be divided in methods where all cross correlations are maintained and in methods where a certain amount of them is neglected. In the former one the quadratic complexity is left unchanged. A higher efficiency is achieved, e.g., by postponing the expensive update of the whole map to a later stage. In the second kind of approaches where cross correlation values are omitted, accuracy is traded with efficiency. The minor accuracy is due to the missing constraints normally imposed on the state estimation by the neglected cross correlations.

A very popular technique in this context is to subdivide the global map into local submaps. Significant computational savings are made by only operating in a small part of the map. The idea followed in this work, is to apply a submapping approach to a hierarchically structured map. This map regards to rather complex objects, like surfaces, on the higher level, and to basic features, like points, on the lower level. In this example, submaps gather point features that are situated on the same surface.

Problems arise in the maintenance of the correlations between each submap and the robot state. Before applying a submapping based algorithm to our specific application, we need to collect experiences regarding the amount of loss in accuracy when traversing between submaps and regarding its robustness and performance. In order to analyse these points, we need to re-implement an existing approach and to evaluate it experimentally. We decided to analyse the *Compressed Kalman Filter*(CKF) by Guivant and Nebot [16, 18] regarding the specific requirements when applied to the application considered in this work.

We have chosen the CKF because of its theoretical soundness. It uses a map representation which enables to neglect certain cross correlations since these values are close to zero anyway. There is a version of the CKF which can be classified as a method maintaining the full correlation. The mentioned low values are not omitted and the algorithm stays optimal. The second version neglects these low values and is therefore a suboptimal method. For the evaluation we used a simulation environment instead of real world data to compare the suboptimal with the optimal version of the CKF.

The work is structured as follows. In the next chapter, we will give an overview about existing solutions for the scalability problem of EKF SLAM. In Chapter 3 the basics of the Kalman Filter algorithm are repeated. This is followed by a detailed presentation of the CKF. In Chapter 4 we point out the basic requirements that have to be fulfilled by a submapping algorithm. The CKF is discussed in terms of them. After that, the experimental evaluation of the CKF is presented. The last part of this chapter describes ideas to approach a hierarchical Kalman Filter. The conclusions are drawn in Chapter 5

Chapter 2

Related Work

2.1 Robot Localisation and Mapping

One of the fundamental tasks in the field of robotics is robot navigation. This is one of the most basic abilities a robot should have to fulfil complex tasks in real-world environments. Robot navigation is usually subdivided into the tasks of *localisation, mapping* and *planning*.

Localisation deals with the problem of determining the robot position in a given map. One of the most popular techniques is *Monte Carlo Localisation* [43]. In this method the belief about the robot location is represented using a particle filter. A set of samples, each representing a potential robot position and weighted with a likelihood, is predicted according to the motion model of the robot and newly weighted after incorporating a measurement.

Mapping is concerned with the acquisition of a map of the environment with given robot poses in that environment. One of the first and most naive mapping techniques is *Occupancy Grid mapping* [43]. Here the map is tessellated into a discrete grid. Each cell carries a value which describes its degree of occupancy by an obstacle. When detecting an obstacle in a certain cell, the according value will be increased while the values of the cells between the vehicle and the occupied cell will be decreased.

Path planning is part of robot navigation and is concerned with finding a route from the current position to a specified known goal position. Path planning is not considered in this work.

Instead, we focus on the integration of localisation and mapping, usually referred to as *Simultaneous Localisation and Mapping* (SLAM) or *Concurrent Localisation and Mapping* (CML). The robot position and the map have to be estimated simultaneously by incorporating measurements of the environment. In contrast to localisation and mapping considered as single tasks both entities are not given in advance. Two of the most popular algorithms are *FastSLAM* initially presented by Montemerlo et al. [32] and *Extended Kalman Filter* (EKF) based SLAM as introduced by Smith et al. [42].

FastSLAM is a technique based on a particle filter. This filter exploits the fact that knowing the robot's path renders the individual landmark measurements independent. Each particle possesses k EKFs that estimate the k landmark positions conditioned on the estimated robot position. By using a sophisticated tree-based data structure, a time complexity of $O(m \log k)$ is achieved where m is the number of particles used at each mobile robot's position estimate. Julier and Uhlmann [23] stated that the FastSLAM algorithm is difficult to analyse theoretically and that empirical examinations have produced mixed results with regard to accuracy and stability.

In this work we will focus on SLAM techniques that are based on the EKF or optimal Kalman Filter. Here, it is assumed that the robot moves in an environment with fixed point landmarks that can be measured by a distance sensor. The position of the landmarks as well as of the robot at a particular point in time are summarised in the *state* of the system. The task is to determine a belief about the state for the next point in time given the last motion of the robot and new feature observations provided by the sensor. The belief is given by a mean, describing the estimated state, and a covariance matrix representing the uncertainty about the state estimate. Usually, we need to model the robot motion and its sensors to be able to provide accurate estimates. EKF SLAM involves the error concerning the position of the robot as well as of the landmarks in the estimation process. These errors are assumed to be independent and to have a Gaussian profile.

There is a variety of techniques facilitating EKF SLAM, e.g., by Davison [8, 9], Dellaert and Stroupe [10], Hähnel et al. [19], Leonard and Durrant-Whyte [28], Smith et al. [42], using either a 2D or 3D world representation and different kinds of sensor devices such as laser range finders or cameras.

Nevertheless, there are two major drawbacks of the EKF based approach to SLAM. Among others Guivant and Nebot [16, p.14], stated that

> [...] any Kalman Filter based system is prone to catastrophic failures under the data association problem.

This can be improved by a significant amount of effort in feature detectors and by using multiple and more accurate sensor devices. In our previous work [2], we analysed the applicability of a stereo camera to SLAM in comparison to a monocular camera. It was experimentally shown that the stereo camera provides much better results due to its ability to obtain depth information.

The second drawback lies in the complexity of the EKF. The covariance matrix carries n^2 elements where n is the number of landmarks. It represents the uncertainty about the components of the system's state at an instance of time and their mutual dependencies. Dissanayake et al. [13, p.230], show that

> The entire structure of the SLAM problem critically depends on maintaining complete knowledge of the cross correlation between landmark estimates.

Thus, after incorporating a new measurement, this matrix has to be fully updated. The time and space complexity is then quadratic in the number of landmarks. A maximum of a few hundred landmarks can be maintained by the full filter to be able to process in real-time. This is obviously not enough to map a natural large-scale environment. In the following the standard EKF will be also referred to as *full filter*.

Recent research in the area of SLAM was focused on enhancing this approach in terms of efficiency. Knight [25] distinguishes between *optimal* and *suboptimal* methods in comparison to the original EKF-based solution.

In the SLAM community, the full EKF is described as *optimal* even if it is not optimal in the statistical sense. Rather, it is a well-understood tool and based on an optimal technique (the classic Kalman Filter) which is guaranteed to find the true solution for a linear problem. The EKF's stochastic suboptimality is due to the linearisation process that is necessary to apply the Kalman Filter algorithm to a non-linear problem. Nevertheless, good modelling and a careful choice of the noise variances will considerably reduce the effects of linearisation [25].

Nevertheless, instead of the term *optimal* we will use the term *full correlation* to indicate methods that do not neglect any of the cross correlation terms between components of the state of the system. In Section 2.2 some examples are presented and we will illustrate that these do not reduce the quadratic time complexity.

Instead of *suboptimal* we will use *reduced correlation* to label algorithms that speed up things in comparison to the full EKF by removing some interrelations and therefore simplifying the problem. A selection of them is presented in Section 2.3.

A more comprehensive review of the various techniques concerned with complexity is given in [25].

2.2 Full Correlation Techniques

Firstly we have techniques which delete map features falling below a certain quality bound from the state. Strictly speaking, these methods do not necessarily have to be mentioned in this section because they do not use all the information available. Nevertheless, they are presented here because they do not neglect any correlation between entities contained in the state. These algorithms emphasise the quality of the feature points within the state rather than their quantity. Dissanayake et al. [11, 12] claim that the quality of a feature is characterised by the provided information content in the sense of Fisher. It is shown that the convergence of the robot localisation is mainly governed by the features with the highest quality.

On the one hand, deleting low quality landmarks leads to a better result in the subtask of localisation but on the other hand we are still just able to derive a sparse map. The upper bound is not changed for approximately a few hundred known landmarks.

To derive constant-time updates is in the main focus of the algorithms belonging to the second category of full information SLAM methods as for example described by Guivant and Nebot [18], Knight et al. [27]. Just the partial state and error covariance matrix corresponding to the vehicle itself and its local environment is updated each filter cycle. The full calculation is postponed to a later stage where more processing time might be available. Again, we do not have a change in the quadratic complexity at all. The advantage is that we do not have to do the full update every filter cycle.

A deeper insight in the technique presented by Guivant and Nebot [18] is given in 3.3.1.

2.3 Reduced Correlation Techniques

In this section, we will present some methods that trade accuracy with complexity by neglecting a certain amount of cross correlations. The term *decoupling* describes the separation of the different entities within the error covariance matrix from each other by setting the according values to zero. The techniques presented in the following differ in the amount of neglected covariance values. Submapping is a special case of decoupling and will be introduced in a separate section.

But before we go into detail concerning decoupling and submapping, the relative map approach is presented. Here, no cross correlations are explicitly neglected but rather the map describing the environment is modelled in a way that its estimation is independent of the robot state. The cross correlations between landmarks and the vehicle are therefore not regarded.

2.3.1 Relative Maps

Csorba [7] and Newman [36] introduced the concept of relative maps. These maps consist of distances between landmarks rather than of their absolute positions. These distances are measured and can be modelled independently of the vehicle position. Usually, the mobile robot is not maintained in the state of the system because it would complicate the state transition model [25]. Instead, the estimation of the map and the robot state is decoupled.

The main advantage of a SLAM algorithm using a relative map consists of its constant-time filter update. This is due to both the structure of the error covariance matrix, whose blocks are centred along the diagonal, and to the fact that just one block per observation is updated.

Knight [25] stated that the removal of the robot from the state leads on the one hand to the advantageous time complexity but on the other hand complicates the unique identification of a feature. The latter is known as the *data association problem* and becomes more complicated in relative maps due to the missing absolute position information. The position information of the robot imposes a strong constraint. Therefore, we focus in this work on absolute maps where the vehicle is maintained in the state of the system.

2.3.2 Decoupling

Decoupling describes methods that omit cross correlations between components of the system state by setting them to zero. These correlations describe to what extent the according variables take values in mutual dependency and, when zeroed, they are lost. In general, we distinguish between full decoupling and partial decoupling techniques.

Full Decoupling

In full decoupling techniques, all cross correlations are set to zero. All components of the state are filtered independently of each other. The correlations between the mobile robot and each landmark as well as between the landmarks themselves are lost. This leads to a linear time and space complexity in the number of known features.

Yu et al. [46, 47] used this technique to reconstruct a three-dimensional object in space from a short sequence of pictures derived by a monocular camera. They stated that the loss in accuracy is little and acceptable in real applications.

The main focus of Yu's work lies in the reconstruction of objects rather than reliable navigation. For a long camera run, Davison [8] and Castellanos et al. [5] analysed the effects of full decoupling. They came to the conclusion that the filter estimates the camera location and orientation and the structure of the map optimistically. This means that the uncertainties about the respective entities derived by the fully decoupled filter are much smaller than by the full filter. Characterised by these uncertainties, the estimates are usually referred to as *non-conservative*. In the limit, this leads to a diverging map and robot trajectory.

Example of Full Decoupling. The reason for the diverging state is illustrated in Figure 2.1 which is adapted from [8]. A top view on a 3D scene is presented. The scene consists of a mobile robot and eleven landmarks in a row. Real positions of these entities are plotted as dark points whereas the estimates are light points. The ellipses represent the uncertainty about the according estimates given in the form of the error covariance matrix from the top view. Because the world is three-dimensional, the uncertainty also has to be considered in three dimensions and is therefore represented as an ellipsoid.

In the left picture, we can see the estimation result when carrying the full covariance matrix. The right picture shows the estimation result derived by a fully decoupled filter.

The vehicle started its run from the grid origin of a global reference frame and drove forward in a straight line. Every second landmark was measured and tracked for a short while. At the end it drove backwards and measured the features it had not previously seen. This order is partially labelled by the black numbers to the left of the feature points. By Step 26 the vehicle returned to the origin.

Regarding the left picture, the estimated robot location is quite offset from its real position. This is also represented by the uncertainty, which is not shown

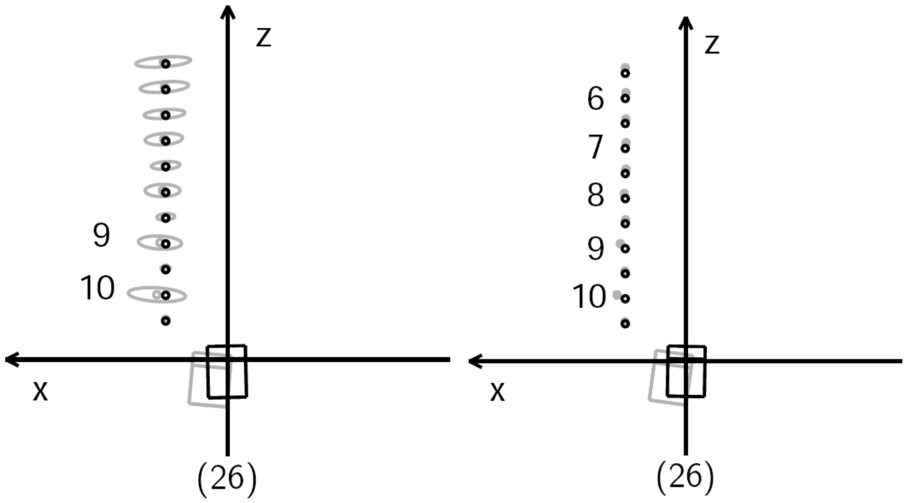


Figure 2.1: *Estimated Location of a Robot and Estimated Map in Step 26 of a Period of Navigation. (Left) Result by Carrying the Full Covariance Matrix. (Right) Result by Using Uncoupled Covariances. Picture adapted from Davison [8, pp.86,88].*

in the figure. The later a feature is measured during the vehicle's motion the more uncertain is its estimate and the greater is the discrepancy between real and estimated position. Feature 0, which is depicted by the bottom circle in each picture, is estimated best. It was initially measured from the origin, where the robot was totally sure about its position. Feature 10 is the most uncertainly estimated landmark but the real position still lies within the uncertainty ellipse.

In the next step, the robot will re-measure Feature 0. Due to the accurately known landmark position, it will reaffirm its true location and the uncertainty about the robot position will drop. The uncertainties about the other features will also decrease because they are coupled with the uncertainty about the vehicle.

Regarding to the right picture, we can see that the accuracy in the state estimate is similar to the left picture. However, the uncertainties about this estimate have not grown correctly after the period of navigation. For example Feature 10 is estimated quite incorrectly but the uncertainty ellipsoid does not reflect that in the same way as for the usage of the full filter.

As a result, the robot will not be able to re-find Feature 0 because it will not be at the expected place. As a result it will not register with the grid origin in the next step as it occurs when using the full filter. The map is doomed to diverge.

What is not taken into account by the fully decoupled filter is the coupling between the uncertainty in the feature and vehicle estimates but rather the uncertainty about their position within the global reference frame. Situations within the global reference frame where the positions of each of these entities are not very well known, but their relative positions are well determined, cannot be represented by fully uncoupled filters. The missing constraint of relative positions, not just between the vehicle and the landmarks but also between the landmarks themselves, leads to overconfident estimates.

Covariance Intersection. Julier and Uhlmann [22] developed a technique referred to as *Covariance Intersection*. It is assumed that no cross correlation information is available and thus the state has to be estimated without it.

Covariance Intersection is based on the observation that when merging a previous estimate for the uncertainty of a map feature with the uncertainty of a new measurement of this feature, the result always lies in the intersection of the two covariance matrices. The new estimate for the uncertainty is *conservative* which means that the uncertainty about the according feature estimate is higher than when using the full filter. Due to this conservatism, Covariance Intersection converges slower. Nevertheless, it is said to yield consistent estimates for any degree of correlation between the two input values [25].

Partial Decoupling

Knight et al. [27] and Guivant and Nebot [18] introduced *Partial Decoupling*. This is based on the according optimal techniques presented in the same articles and already mentioned in this work in Section 2.2. Only part of the full error covariance is kept up to date. In the optimal version, the full update is postponed to a later stage of the navigation process. Here in the suboptimal version, the *passive* part of the state is decoupled from the *active* one. How *active* and *passive* are defined differs for each algorithm, but the fundamental idea behind them is that features in the active part are less correlated with features of the passive part. The algorithm developed by Guivant and Nebot [16] is presented in more detail in Section 3.3.2.

In the Postponement algorithm by Knight et al. [27] active features are landmarks that were recently observed. Passive landmarks were lastly measured a long time ago. Postponement with partial decoupling is evaluated in [27, p.7], as follows:

> The algorithm is not consistent, but it can be used to produce distorted but navigable maps which do not diverge. [...] Partial decoupling simply provides a compromise between the processing expense of the optimal filter and the inaccuracy of a fully decoupled filter.

This statement should be underlined by Figure 2.2. We can see two top views of the same 3D scene. In each picture the vehicle started at the bottom left edge and drove one circuit in this corridor.

In the left view, the full filter was used. The resulting map is quite accurate and carries 264 landmarks. The features nearest to the initial robot position are most certainly estimated whereas the features furthest away are not. The amount of uncertainty does not depend on the sequence of their initial observation but on the distance to the most certainly known robot state.

In the right view, partial decoupling was used to estimate the structure of the map. This led to an inconsistent map, quite inaccurate in comparison to the map depicted in the left picture. In quite a lot of cases, the uncertainty ellipses do not contain the position of the according real landmark position. The filter observed just 248 features.

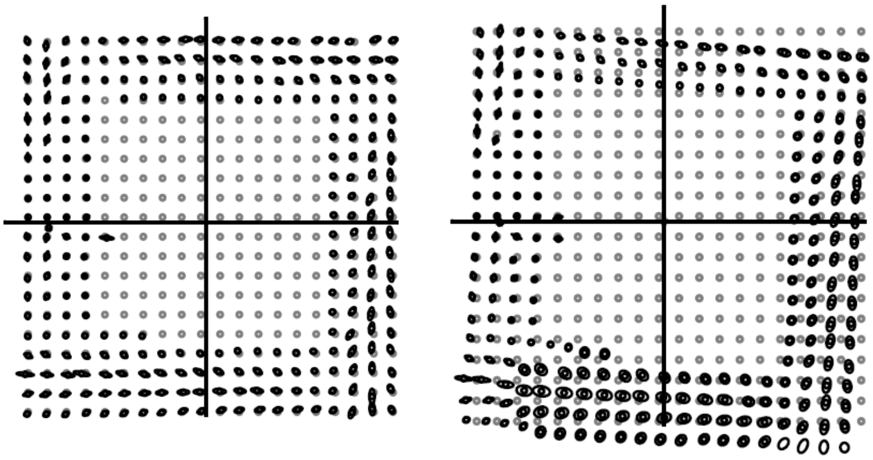


Figure 2.2: *40m × 40m Corridor Map Generated in one Circuit by the Optimal Filter (left) and by Using Partial Decoupling (right). 264 features were observed in the left picture and 248 in the right one. The pictures are adapted from Knight [25, p.27].*

The uncertainty about the known feature points does not depend on the distance to the initial robot position as it could be observed for the use of the full filter.Instead, the amount of uncertainty is influenced by the sequence in which the features are measured. Landmarks measured in the end are much more uncertain than features measured in the beginning. This is due to the missing correlations between the active landmarks and landmarks that were measured a long time ago.

The inconsistent result of partial decoupling is due to the fact that the fundamental assumption does not hold in the limit. The amount of correlation between features that were measured at distant points in time is not low for all times and can therefore not simply be omitted. Dissanayake et al. [13] showed that in the limit as the number of observations increases, the landmark estimates become fully correlated.

2.3.3 Submapping

The submapping approach to SLAM as one possibility to decouple certain components in the state enjoys great popularity. Many researchers analyse the effect of dividing the global map into local submaps, e.g. in Bosse et al. [3], Castellanos et al. [4], Chong and Kleeman [6], Estrada et al. [14], Guivant and Nebot [16], Leonard and Newman [30], Lisien et al. [31], Newman et al. [37]. It is based on the principle that the update of one feature will not affect a feature which is far away. Their correlation will be very small and it is therefore not worth storing their covariance values. Thus, cross correlations between features belonging to different local maps are not maintained. Guivant and Nebot [16] showed that this assumption holds when the map contains features referred relative to local frames and the origins of these frames which are regarded absolutely. Strong correlations persist between the base landmarks defining a frame and the vehicle as well as between features belonging to the same local submap. Cross correlations between relative features belonging to different submaps tend to stay near zero. This is discussed in more detail in Section 3.3.2.

In general, we can distinguish between two different submapping techniques. The first one is perhaps best described by the following statement from Bosse et al. [3, p.1], describing the basic idea of their so-called *Atlas* framework.

> Rather than operating on a single map of ever-increasing complexity, the Atlas framework simply switches its focus to a new or adjacent map-frame.

This means that the EKF works in one active submap during one filter cycle and the vehicle is only represented in this active submap. When it is necessary, the EKF switches to an adjacent submap or creates a new one. Within the Atlas framework, the relative position of two submaps is described by a metric transformation between the local reference frames and this transformation carries an uncertainty.

The second submapping technique works in one active submap as well, but additionally updates the global map after a local correction. The robot is correlated with all submaps.

In the following, we will discuss another example for switching between submaps.

Switching between Submaps

Newman et al. [37] developed a SLAM algorithm based on submapping. The global map is divided into several submaps containing landmarks and the vehicle state is referred to a local coordinate frame. The origin of these frames are landmarks that are addressed absolutely in the world coordinate frame. There is always just one active submap which is updated by taking measurements of related feature points.

Submaps can overlap in several feature points. This enables the algorithm to include information from adjacent maps to improve the map and vehicle estimates.

The big advantage of this approach is that it can operate in constant-time because it only considers one submap per filter cycle. The disadvantage occurs when the vehicle crosses between maps and thus loses information. Not just correlations between feature points belonging to different submaps are lost but also information on the relation between the vehicle and the local maps. Correlations to the old map are lost and correlations to the new map are unknown. The missing information can be partially reconstructed by involving information from the overlapping features also belonging to the old map. Thus, as long as there are a relatively small number of submap traversals, the algorithm will provide conservative estimation results.

Keeping Track of the Global Map

Castellanos et al. [4] provided a SLAM approach based on submapping that works in two levels. In the first, the so-called landmark level, features are addressed relative to one of the several reference frames known as landmarks dividing the global map. The origin of these landmarks are absolutely positioned in the global level. Correlations are maintained between features addressed in the same local reference frame and between the globally referenced origins of the landmarks. The vehicle is represented on both levels and therefore correlations between the robot and the according feature in the landmark or global level are constantly maintained.

Every time a particular feature is measured both the local submap containing the feature as well as the global level are updated. This leads to a time complexity of $O(N_L^2 + N_{FL}^2)$ where N_L denotes the number of submaps and $N_F L$ the maximum number of features per landmark. Unlike the previous example of switching between submaps, this is not a constant-time solution. However the correlations between features addressed in different landmarks are omitted. This is realised by handling the estimates for the local and global robot state independently which they are of course not. Discrepancies could arise in two places: either between the estimates themselves; or more likely between the amount of uncertainty about these estimates in the global and landmark levels.

Despite the fact that the technique is not time-constant, Knight [25] stated that Castellanos solution is useful to solve the problem of structuring large-scale environments.

Submap Management

In any technique facilitating submaps, the question arises of how to organise features into submaps. Different techniques are applied. The decision when to found a new submap is made based on the following criterias:

- The maximum amount of features belonging to one submap is reached, e.g., applied by Estrada et al. [14], Leonard and Newman [30], Newman et al. [37].
- The maximum amount of uncertainty about the vehicle is reached, e.g., applied by Chong and Kleeman [6], Estrada et al. [14].
- The maximum distance from the currently active submap origin is reached, e.g., applied by Leonard and Newman [30], Newman et al. [37].
- No matchings are found for the current sensor measurement, e.g., applied by Estrada et al. [14].

Mainly, the principle of locality holds which means that feature points are gathered that are positioned in an immediate vicinity.

2.4 Extraction of Higher Level Features from Images

In all the works mentioned in this chapter, maps consisted of features based on point landmarks. As already mentioned, to be able to process a full EKF in real-time only a few hundred features can be maintained. An idea to reduce complexity, besides neglecting certain cross correlations, would be to use higher level features like lines, surfaces or even whole objects seen as cubes. This would reduce the number of known landmarks within the state.

Point features could still be used as a basis to abstract to these more complex features. It has to be decided which landmark belongs to which higher level feature. This decision could be made based on a segmentation of the environment into, e.g., areas or volumes defining these complex landmarks. Features that are detected to be within or part of a specific segmented region are gathered to one submap. As mentioned in the previous section, the majority of research work in the field of submapping exploits the principle of locality to subdivide the global map. Our idea additionally includes semantics, e.g., the submap just consists of surface features that belong to the same cube.

Supposing that we have a number of point features detected by a stereo camera sensor and our goal is to group them if they belong to one surface, the patchlet method introduced by Murray and Little [35] could be applied. Patchlets are locally planar surface elements which have a position, surface normal, size and confidence on the position and normal direction associated with it. These additional information in comparison to ordinary point features make a stereo matching algorithm more robust against the data association problem.

Murray and Little [33, 34] applied their patchlet method to surface segmentation from stereo images. The segmentation process is divided into two steps. Firstly, Random Sample Consensus (RANSAC) is applied to estimate the number of surfaces in the scene and to obtain reasonable initial estimates. Secondly, the Expectation Maximisation (EM) algorithm is used to refine the output of the first step.

Pietzsch and Großmann [38] improved this method by obtaining higher quality confidence measures for the position and surface orientation. This was achieved by involving the intensity patterns in the determination of position and orientation of the patchlets. More reliable confidence values allow a sophisticated choice of which patchlets belong to a surface and thus produce a more reliable surface segmentation of a stereo image.

Chapter 3

Foundations

In this chapter, we provide a basis for the understanding of the contributions presented in the following chapters. First of all, we introduce the notational conventions used throughout this work. This is followed by a short review of the *Kalman Filter* (KF) and its application to the problem of robot localisation and mapping. In the last section the general idea and mathematical formulation of the *Compressed Extended Kalman Filter* (CEKF) presented by Guivant and Nebot [18] is given.

3.1 Notation

Throughout this work, the following notation is used. It has been largely adopted from the textbook by Thrun et al. [43].

- Scalars are denoted by italic symbols, e.g., a, b, c.
- Vectors, regardless of dimension, are denoted in sans-serif lower case symbols, e.g., $\mathsf{x} = (x, y)^\top$. Usually, the dimension of a vector is clear from the context. Nevertheless, this is sometimes declared by $\mathsf{x} \in \mathbb{R}^2$.
- Matrices, regardless of dimension, are denoted in sans-serif capital letters, e.g., $\mathsf{A}, \mathsf{B}, \mathsf{C}$. Denoting their dimension is done equivalently to vectors, e.g., $\mathsf{A} \in \mathbb{R}^{3\times 3}$
- In this work, we will extensively use algorithms based on the Kalman Filter. It is used to derive a belief about the current state of the dynamical system considered. This belief is represented by a mean μ and a covariance Σ.
- Generally speaking, the Kalman Filter loops through two steps: a prediction and an update of the belief in the current state of the system. A predicted belief will be indicated by a bar superscript, e.g., $\overline{\mu}$, $\overline{\Sigma}$. Updated beliefs are not provided with an additional attribute and thus denoted by μ and Σ.

- In the following, the specific vectors and matrices used in the Kalman Filter equations are assigned a specific symbol. Their use is briefly described in the next section.
  - A . . . Process model
  - R . . . Process noise covariance
  - ϵ . . . Process noise
  - C . . . Measurement model
  - Q . . . Measurement noise covariance
  - δ . . . Measurement noise
  - z . . . Measurement
  - S . . . Innovation covariance
  - ν . . . Innovation
  - K . . . Kalman Gain
- In this work, we focus on robot localisation and mapping. Therefore, the Kalman Filter is used to estimate the current position of a mobile robot and a map of the environment the robot is moving in. Thus, the state of our dynamical system to be estimated is a vector containing the coordinates of the robot position represented by x and its velocity denoted by v. The robot state is summarised by $\mathsf{r} = \begin{pmatrix}\mathsf{x}^\top & \mathsf{v}^\top\end{pmatrix}^\top$. Providing that maps are constituted by point features, the state of the system considered will additionally contain their coordinates indicated by an y.
- Maps are assumed to be a set of point landmarks. The index of a landmark within the state vector will be denoted by a superscript i, e.g., y^i. Other vectors or matrices, e.g., the measurement vector or model, corresponding to one of the point features will also be labelled by index i.
- The dynamic system we are interested in is estimated at discrete points in time. The current point in time is represented by a subscript t, e.g., μ_t, Σ_t.
- We will frequently deal with entities which are addressed in a local or global reference frame. To distinguish them, we will use an l or g subscript to label features belonging to a local or global reference frame, respectively, e.g., y_l, y_g.
- For the purpose of evaluation we need to transfer entities to the vehicle reference frame. This will be indicated by an x subscript, e.g., y_x.

3.2 The Kalman Filter

In this section, we briefly present the main idea and equations of the Kalman Filter algorithm. It was firstly introduced by Kalman [24]. We simplify the general equations by neglecting the so-called *control input*, commonly used in robotic applications to involve, e.g., odometry data. In this work, we do not regard any control input since we assume a purely distance measurement based SLAM. For more detailed explanations and examples please refer to our previous work [2, Ch.4] or Welch and Bishop [44].

The Kalman Filter is commonly used to derive a belief about the continuous state of a dynamical system when it is not *observable*. Not observable means that the state is not directly measurable. The belief is represented by a Gaussian probability distribution given by a mean μ and a covariance Σ over the state of the system.

Information about the internal state is only accessible via noisy sensor measurements. In order to derive any knowledge from this output by applying the Kalman Filter equations, we need to provide a model for the state transition and for the sensor measurements. The state transition function is also known as the *process model* and has to be linear. The *measurement model* will describe the connection between the output and the internal state of the system. It has to be linear as well.

In practise, it is not possible to represent the system considered with absolute precision. Instead, the computational model will stop at some level of detail. The gap between this model and reality is filled with some probabilistic assumption referred to as *noise*.

Imagine an autonomous mobile robot navigating in a certain environment. It will be possible to model its motion to a certain extent because there is, for example, an upper bound on the robot velocity. Supposing the robot is situated at a given spot in a room, it cannot move to an arbitrary point in the next few seconds. Its next possible position will be in a limited area centred around its old position. However, its true position cannot be inferred directly if no externally given information is available.

The same holds for the measurement model. Sensors always provide measurements which contain a certain amount of noise. Nevertheless, we are usually able to provide a probabilistic model of the sensor properties. This enables us to determine the possible deviation of the measurement from the real value.

The Kalman Filter is able to use these noise quantities to derive a belief about the current state of the system.

3.2.1 Kalman Filter Equations

The Kalman Filter algorithm is summarised in Algorithm 1. One cycle can be divided into two parts. In the *predict step* (lines 2 and 3) the process model A, the process noise covariance R, the current state, and error covariance matrix estimates μ_i and Σ_i are used to derive an *a priori* state and matrix estimates $\overline{\mu}_{i+1}$ and $\overline{\Sigma}_{i+1}$ for the next point in time. R models the standard deviation of the noise introduced by the deviation of the process model from reality. The so-called *process noise* is assumed to have a Gaussian profile with zero mean.

In the *correct step* (lines 4 to 6) a noisy measurement z_{i+1} is obtained to enhance these *a priori* values and derive an improved *a posteriori* estimate. In order to do so, a measurement is predicted by applying the measurement model C to the predicted state. The difference between that and the real measurement

Algorithm 1: Kalman Filter Algorithm

```
   Input  : Belief at time 0: $\mu_0$ and $\Sigma_0$
   Output: Belief at time $t$: $\mu_t$ and $\Sigma_t$
1  for ( $i = 0;\ i < t;\ i++$ ) do
       // Predict Step
2      $\overline{\mu}_{i+1} = \mathsf{A}\mu_i$
3      $\overline{\Sigma}_{i+1} = \mathsf{A}\Sigma_i\mathsf{A} + \mathsf{R}$
       // Update Step
4      $\mathsf{K}_{i+1} = \frac{\overline{\Sigma}_{i+1}\mathsf{C}^\top}{\mathsf{C}\overline{\Sigma}_{i+1}\mathsf{C}^\top + \mathsf{Q}}$
5      $\mu_{i+1} = \overline{\mu}_{i+1} + \mathsf{K}_{i+1}(\mathsf{z}_{i+1} - \mathsf{C}\overline{\mu}_{i+1})$
6      $\Sigma_{i+1} = (\mathsf{I} - \mathsf{K}_{i+1}\mathsf{C})\overline{\Sigma}_{i+1}$
7  end
```

is referred to as residual or innovation $\nu_{i+1} = \mathsf{z}_{i+1} - \mathsf{C}\overline{\mu}_{i+1}$. It is used in line 5 to correct the state estimate after weighting it with the so-called Kalman Gain K_{i+1}. This gain is computed by the equation given in line 4. The denominator is equal to the innovation covariance $\mathsf{S}_{i+1} = \mathsf{C}\overline{\Sigma}_{i+1}\mathsf{C}^\top + \mathsf{Q}$. S_{i+1} represents the uncertainty in the predicted measurement. It is composed of the projection of the predicted covariance $\overline{\Sigma}_{i+1}$ into the measurement space by multiplying it from the left and right with the measurement model C and the measurement noise covariance Q. The latter models the standard deviation of the assumed white Gaussian noise introduced by the sensor device.

The ratio between the measurement noise and the uncertainty about the predicted state determines how large the Kalman Gain is and thus how heavy the residual is weighted. It is weighted more heavily the smaller the measurement noise is. If the uncertainty about the prediction approaches zero, the residual is weighted less.

Using the Kalman Filter to compute a non-observable state of a dynamic system constitutes an optimal solution. In most of the interesting applications , e.g., SLAM, the condition of linearity is not fulfilled. To be able to apply the Kalman Filter approach to these non-linear tasks, we have to linearise the models. The modified algorithm is referred to as *Extended Kalman Filter* (EKF). The disadvantage is that linearisation errors will occur additionally to the noise terms.

In this work, we will focus on simplified, mostly one-dimensional problems where the models are linear. This allows us to evaluate different methods independent of linearisation errors. For a more detailed description of the EKF please refer to Bohg [2].

3.2.2 Complexity of the Kalman Filter Algorithm

If the equations as given in Algorithm 1 are implemented naively, the time and space complexity of the Kalman Filter is of $O(n^3)$, where n is the number of features contained in the map. Considering a sparse process and measurement model, it is possible to reduce the complexity to be quadratic in the number of features. The update and maintenance of the full covariance at the end of one Kalman Filter cycle is responsible for the quadratic time and space complexity.

More detailed information on the complexity is given in our previous work [2].

3.3 The Compressed Kalman Filter

In this section we are presenting the concept of the *Compressed Kalman Filter* (CKF) and its mathematical formulation introduced by Guivant and Nebot [16, 18] and Guivant [15]. Its discussion and experimental evaluation is given in Section 4.2.

As introduced in [16, 18] and also in [15] the presented algorithm was applied to a mobile vehicle moving in two dimensions. Instead of the Kalman Filter, the Extended Kalman Filter (EKF) was used to handle non-linear vehicle rotations. Their algorithm was thus denoted as CEKF. In this work we will restrict the discussion to the linear case and refer to the CKF to avoid linearisation errors.

Guivant and Nebot [16, 18] introduced two versions of the CEKF. Firstly, we have the optimal method as in detail described in [18]. In this case optimal means, that it is absolutely equivalent to the full Extended Kalman Filter. The difference lies solely in the partitioning of the state and covariance into two parts. For a certain period of time, the first or *active* part is updated. The update of the full state and covariance is shifted to a later stage. Thus the complexity of the CEKF is still quadratic in the number of map features, but the full covariance update causing this complexity does not have to be performed every filter cycle.

This method is very similar to the *Postponement* technique presented by Knight et al. [27]. They differ in the way they determine the maintained features: Guivant and Nebot [18] use static geographical boundaries while Knight et al. [27] expand the active subset of map features dynamically.

The second version of the CEKF introduced by Guivant and Nebot [16] is a suboptimal method. Here, suboptimal means that it is not equivalent to the full filter anymore due to omitting certain cross correlations. The partitioning of the map is used to decide which of them can be neglected. Its update then involves significantly less than n^2 multiplications.

In the next section we introduce the optimal version of the CKF. This is followed by a presentation of the suboptimal CKF where its reduced complexity is analysed in more detail.

3.3.1 The Optimal CKF

Guivant and Nebot [18] showed that due to the internal structure of the SLAM problem, it is not necessary to perform a full covariance update when working in a local area. During this operation, the computational requirements of the

SLAM algorithm are then reduced to the order of the number of features in the local vicinity of the vehicle.

This is especially advantageous when the vehicle is operating in the same local area for a long time or when the measurement device is providing information at a high rate. Imagine, for example, the bumblebee™ stereo camera providing stereo images at frequencies of 15 to 30 Hz [39]. Incorporating these measurements requires the full filter to update the whole map while the CEKF just needs to update a usually significantly smaller subset of the map.

The information obtained from a local area is propagated to all the other landmarks, when the vehicle departs from this area.

In this section, we will first show how to build and apply the so called *compressed* matrices. They *gather* the information for the *passive* part of the map during the updates of the *active* part. Next we describe the map management and the technique used to select the local area. Finally the algorithm is summarised.

Analysing a Kalman Filter Cycle

Consider the following scenario. A robot is moving in one dimension equipped with a sensor which provides measurements of the distance between the vehicle and one-dimensional static point features.

The state μ is divided into two groups. The first group is referred to as active and contains a set of landmarks y_i located in an area of a certain size surrounding the vehicle and the vehicle state itself. This vehicle state is described by a one-dimensional position x and translational velocity v. The robot state is summarised by r. Entities referring to this group are labelled with a superscript A.

The second group contains the map features not contained in the area around the robot. Entities referring to this group are labelled with a superscript B. The full covariance is divided equivalently.We have

$$\mu = \begin{pmatrix} \mu^A \\ \mu^B \end{pmatrix} \qquad \begin{matrix} \mu^A \in \mathbb{R}^{n^A+2} \\ \mu^B \in \mathbb{R}^{n^B} \end{matrix} \qquad \begin{matrix} \mu \in \mathbb{R}^{n+2} \\ n = n^A + n^B \end{matrix}$$

$$\Sigma = \begin{bmatrix} \Sigma^{AA} & \Sigma^{AB} \\ \Sigma^{BA} & \Sigma^{BB} \end{bmatrix} \qquad \begin{matrix} \Sigma^{AA} \in \mathbb{R}^{n^A+2 \times n^A+2} \\ \Sigma^{BA} \in \mathbb{R}^{n^B \times n^A+2} \end{matrix} \qquad \begin{matrix} \Sigma^{AB} \in \mathbb{R}^{n^A+2 \times n^B} \\ \Sigma^{BB} \in \mathbb{R}^{n^B \times n^B} \end{matrix}$$

In the following we look at the corresponding prediction and update steps as they would be performed with the standard Kalman Filter. Thereby we analyse them in terms of possible simplification.

Prediction The prediction of the state and full covariance is performed as follows:

$$\begin{aligned} \overline{\Sigma}_{t+1} &= \mathsf{A}\Sigma_t\mathsf{A}^\top + \mathsf{R} \\ \overline{\mu}_{t+1} &= \mathsf{A}\mu_t \end{aligned}$$

We assumed that the environment is static. Thus, the process model A will only affect the vehicle part r of the state.

$$\begin{aligned} \mathsf{A} &= \begin{bmatrix} \mathsf{A}^{AA} & \mathsf{A}^{AB} \\ \mathsf{A}^{BA} & \mathsf{A}^{BB} \end{bmatrix} = \begin{bmatrix} \mathsf{A}^{AA} & 0 \\ 0 & \mathsf{I} \end{bmatrix} & \mathsf{R} &= \begin{bmatrix} \mathsf{R}^{AA} & \mathsf{R}^{AB} \\ \mathsf{R}^{BA} & \mathsf{R}^{BB} \end{bmatrix} = \begin{bmatrix} \mathsf{R}^{AA} & 0 \\ 0 & 0 \end{bmatrix} \\ \mathsf{A}^{AA} &= \begin{bmatrix} \mathsf{A}^{rr} & 0 \\ 0 & \mathsf{I} \end{bmatrix} & \mathsf{R}^{AA} &= \begin{bmatrix} \mathsf{R}^{rr} & 0 \\ 0 & 0 \end{bmatrix} \end{aligned}$$

Performing the matrix multiplication $\mathsf{A}\Sigma_t\mathsf{A}^\top$ explicitly yields

$$\begin{aligned} \overline{\Sigma}_{t+1}^{AA} &= \mathsf{A}^{AA}\Sigma_t^{AA}(\mathsf{A}^{AA})^\top + \mathsf{R}^{AA} \\ \overline{\Sigma}_{t+1}^{AB} &= \mathsf{A}^{AA}\Sigma_t^{AB} \\ \overline{\Sigma}_{t+1}^{BA} &= (\overline{\Sigma}_{t+1}^{AB})^\top \\ \overline{\Sigma}_{t+1}^{BB} &= \Sigma_t^{BB} \end{aligned}$$

Update We assume that for a period of time landmarks contained in the vicinity of the vehicle, thus contained in part A, are measured. The measurement model is then structured as follows:

$$\mathsf{C} = \begin{bmatrix} \mathsf{C}^A & \mathsf{C}^B \end{bmatrix} = \begin{bmatrix} \mathsf{C}^A & 0 \end{bmatrix}$$

Please note, C is then completely independent of the part B of the state.

Performing the equations to yield the Kalman Gain K explicitly, we have:

$$\begin{aligned} \overline{\Sigma}_{t+1}\mathsf{C}^\top &= \begin{bmatrix} \overline{\Sigma}_{t+1}^{AA}(\mathsf{C}^A)^\top \\ \overline{\Sigma}_{t+1}^{BA}(\mathsf{C}^A)^\top \end{bmatrix} \\ \mathsf{C}\overline{\Sigma}_{t+1}\mathsf{C}^\top &= \mathsf{C}^A\overline{\Sigma}_{t+1}^{AA}(\mathsf{C}^A)^\top \\ \mathsf{S}_{t+1} &= \mathsf{C}^A\overline{\Sigma}_{t+1}^{AA}(\mathsf{C}^A)^\top + \mathsf{Q} \\ \mathsf{K}_{t+1} &= \overline{\Sigma}_{t+1}\mathsf{C}\mathsf{S}_{t+1}^{-1} = \begin{bmatrix} \overline{\Sigma}_{t+1}^{AA}(\mathsf{C}^A)^\top\mathsf{S}_{t+1}^{-1} \\ \overline{\Sigma}_{t+1}^{BA}(\mathsf{C}^A)^\top\mathsf{S}_{t+1}^{-1} \end{bmatrix} = \begin{bmatrix} \mathsf{K}_{t+1}^A \\ \mathsf{K}_{t+1}^B \end{bmatrix} \end{aligned}$$

Note that the innovation covariance S_{t+1} and the Kalman Gain regarding part A are functions of $\overline{\Sigma}_{t+1}^{AA}$ and C^A. They are independent of $\overline{\Sigma}_{t+1}^{BB}$, $\overline{\Sigma}_{t+1}^{AB}$, $\overline{\Sigma}_{t+1}^{BA}$ and $\overline{\mu}_{t+1}^B$.

While no entity of part B of the state is measured, we loose no information when updating part A separately. Updating the full state yields:

$$\mu_{t+1} = \begin{pmatrix} \mu_{t+1}^A \\ \mu_{t+1}^B \end{pmatrix} = \overline{\mu}_{t+1} + \mathsf{K}_{t+1}\nu_{t+1} = \begin{pmatrix} \overline{\mu}_{t+1}^A + \overline{\Sigma}_{t+1}^{AA}(\mathsf{C}^A)^\top\mathsf{S}_{t+1}^{-1}\nu_{t+1} \\ \overline{\mu}_{t+1}^B + \overline{\Sigma}_{t+1}^{BA}(\mathsf{C}^A)^\top\mathsf{S}_{t+1}^{-1}\nu_{t+1} \end{pmatrix} \quad (3.1)$$

From Equation (3.1) we can see that we would loose information about the state B if we would ignore it completely during the update. This also holds for the covariance matrix. The update term of the covariance matrix is of the following structure:

$$\begin{aligned}
\mathsf{K}_{t+1}\mathsf{C}\overline{\Sigma}_{t+1} &= \begin{bmatrix} \overline{\Sigma}^{AA}_{t+1}\ \kappa_{t+1}\ \overline{\Sigma}^{AA}_{t+1} & \xi_{t+1}\ \overline{\Sigma}^{AB}_{t+1} \\ (\xi_{t+1}\ \overline{\Sigma}^{AB}_{t+1})^{\top} & \overline{\Sigma}^{BA}_{t+1}\ \kappa_{t+1}\ \overline{\Sigma}^{AB}_{t+1} \end{bmatrix} \\
\text{with} & \\
\kappa_{t+1} &= (\mathsf{C}^{A})^{\top}\mathsf{S}^{-1}_{t+1}\mathsf{C}^{A} \\
\xi_{t+1} &= \overline{\Sigma}^{AA}_{t+1}\ \kappa_{t+1}
\end{aligned}$$

The updated covariance matrix is then:

$$\Sigma^{AA}_{t+1} = \overline{\Sigma}^{AA}_{t+1} - \overline{\Sigma}^{AA}_{t+1}\ \kappa_{t+1}\ \overline{\Sigma}^{AA}_{t+1} \tag{3.2}$$

$$(\Sigma^{AB}_{t+1})^{\top} = \Sigma^{AB}_{t+1} = \overline{\Sigma}^{AB}_{t+1} - \xi_{t+1}\ \overline{\Sigma}^{AB}_{t+1} = (\mathsf{I} - \xi_{t+1})\overline{\Sigma}^{AB}_{t+1} \tag{3.3}$$

$$\Sigma^{BB}_{t+1} = \overline{\Sigma}^{BB}_{t+1} - \overline{\Sigma}^{BA}_{t+1}\ \kappa_{t+1}\ \overline{\Sigma}^{AB}_{t+1} \tag{3.4}$$

As you can see in Equation (3.2), the active part A of the covariance can be updated independently of any entity related to the passive part B. This does not hold true for the inverse relation: The cross correlation parts $\Sigma_{t+1}t^{AB}$ and Σ^{BA}_{t+1} as well as Σ^{BB}_{t+1} cannot be updated independently of matrices related to part A.

Thus, this information needs to be saved and propagated to the passive part of the state vector and the covariance matrix.

Compressing Information

Guivant and Nebot [18] introduced two auxiliary matrices ϕ and ψ and an auxiliary vector θ to gather and compress the information obtained by updating the active part.

Considering the calculation of Σ^{AB}_{t+1} given in Equation (3.3) after k consecutive full filter cycles, we have:

$$\begin{aligned}
\Sigma^{AB}_{t+k} &= (\mathsf{I} - \xi_{t+k})\ \mathsf{A}^{AA}\ \underbrace{(\mathsf{I} - \xi_{t+k-1})\ \mathsf{A}^{AA} \dots \underbrace{(\mathsf{I} - \xi_{t+2})\ \mathsf{A}^{AA}\ \underbrace{(\mathsf{I} - \xi_{t+1})\ \mathsf{A}^{AA}\ \Sigma^{AB}_{t}}_{\Sigma^{AB}_{t+1}}}_{\Sigma^{AB}_{t+2}}}_{\Sigma^{AB}_{t+k-1}} \\
&= \phi_{t+k}\ \Sigma^{AB}_{t}
\end{aligned}$$

with

$$\phi_{t+k} = (\mathsf{I} - \xi_{t+k})\ \mathsf{A}^{AA}(\mathsf{I} - \xi_{t+k-1})\ \mathsf{A}^{AA} \dots (\mathsf{I} - \xi_{t})\ \mathsf{A}^{AA}$$

It can be easily seen that ϕ can be evaluated recursively. In the following we will distinguish between the predicted $\bar{\phi}$ and the updated ϕ with

$$\begin{aligned}
\bar{\phi}_{t+1} &= \mathsf{A}^{AA}\phi_{t} \\
\phi_{t+1} &= (\mathsf{I} - \xi_{t+1})\bar{\phi}_{t+1}
\end{aligned}$$

Equation (3.4) can be rewritten as

$$
\begin{aligned}
\Sigma^{BB}_{t+1} &= \overline{\Sigma}^{BB}_{t+1} - d\Sigma^{BB}_{t+1} \\
d\Sigma^{BB}_{t+1} &= \overline{\Sigma}^{BA}_{t+1} \; \kappa_{t+1} \; \overline{\Sigma}^{AB}_{t+1}
\end{aligned}
$$

Performing Equation (3.4) k filter cycles to derive Σ^{BB}_{t+k} yields:

$$
\begin{aligned}
\Sigma^{BB}_{t+k} &= \Sigma^{BB}_{t} - (d\Sigma^{BB}_{t+1} + d\Sigma^{BB}_{t+2} + \ldots) \\
d\Sigma^{BB}_{t+1} &= \underbrace{\Sigma^{BA}_{t} \; (\mathsf{A}^{AA})^\top}_{\overline{\Sigma}^{BA}_{t+1}} \; \kappa_{t+1} \; \underbrace{\mathsf{A}^{AA} \; \Sigma^{AB}_{t}}_{\overline{\Sigma}^{AB}_{t+1}} \\
d\Sigma^{BB}_{t+2} &= \underbrace{\underbrace{\Sigma^{BA}_{t} \; (\mathsf{A}^{AA})^\top \; (\mathsf{I} - \xi_{t+1})^\top}_{\Sigma^{BA}_{t+1}} \; (\mathsf{A}^{AA})^\top}_{\overline{\Sigma}^{BA}_{t+2}} \; \kappa_{t+2} \; \underbrace{\mathsf{A}^{AA} \; \underbrace{(\mathsf{I} - \xi_{t+1}) \; \mathsf{A}^{AA} \; \Sigma^{AB}_{t}}_{\Sigma^{AB}_{t+1}}}_{\overline{\Sigma}^{AB}_{t+2}}
\end{aligned}
$$

This can be simplified to

$$\Sigma^{BB}_{t+k} = \Sigma^{BB}_{t} - \Sigma^{BA}_{t} \; \psi_{k+t} \; \Sigma^{AB}_{t}$$

with

$$\psi_{t+k} = \sum_{i=t+1}^{t+k} \bar{\phi}_i^\top \; \kappa_i \; \bar{\phi}_i$$

Also ψ can be evaluated recursively:

$$\psi_{t+1} = \psi_t + \bar{\phi}^\top_{t+1} \; \kappa_{t+1} \; \bar{\phi}_{t+1}$$

Considering the part B of the state after k consecutive full filter cycles as described by Equation 3.1, we have

$$
\begin{aligned}
\mu^B_{t+k} &= \overline{\mu}^B_{t+k} + \sum_{t=i}^{t+k} \mathsf{K}^B_i \; \nu_i \\
&= \overline{\mu}^B_{t+k} + \underbrace{\underbrace{\Sigma^{BA}_t \; \bar{\phi}_{t+1}}_{\overline{\Sigma}^{BA}_{t+1}} \; (\mathsf{C}^A)^\top \; \mathsf{S}^{-1}_{t+1} \; \nu_{t+1}}_{d\overline{\mu}^B_{t+1}} \\
&\quad + \underbrace{\underbrace{\Sigma^{BA}_t \; \bar{\phi}_{t+2}}_{\overline{\Sigma}^{BA}_{t+2}} \; (\mathsf{C}^A)^\top \; \mathsf{S}^{-1}_{t+2} \; \nu_{t+2}}_{d\overline{\mu}^B_{t+2}} \\
&\qquad \vdots \\
&\quad + \underbrace{\underbrace{\Sigma^{BA}_t \; \bar{\phi}_{t+k}}_{\overline{\Sigma}^{BA}_{t+k}} \; (\mathsf{C}^A)^\top \; \mathsf{S}^{-1}_{t+k} \; \nu_{t+k}}_{d\overline{\mu}^B_{t+k}}
\end{aligned}
$$

This can be simplified to

$$
\begin{aligned}
\mu^B_{t+k} &= \overline{\mu}^B_{t+k} + \Sigma^{BA}_t \sum_{t=i}^{t+k} \bar{\phi}_i \; (\mathsf{C}^A)^\top \; \mathsf{S}^{-1}_i \; \nu_i \\
&= \overline{\mu}^B_{t+k} + \Sigma^{BA}_t \; \theta_{t+k}
\end{aligned}
$$

with

$$
\theta_{t+k} = \sum_{t=i}^{t+k} \bar{\phi}_i \; (\mathsf{C}^A)^\top \; \mathsf{S}^{-1}_i \; \nu_i
$$

θ can also be evaluated recursively by

$$
\theta_{t+1} = \theta_t + \bar{\phi}_{t+1} \; (\mathsf{C}^A)^\top \; \mathsf{S}^{-1}_{t+1} \; \nu_{t+1}
$$

Our goal was to find a method to maintain the information about the changes in the passive parts of the state and covariance while updating the active part. In the previous equations we can see that we can compress this information in two auxiliary matrices ϕ and ψ and an auxiliary vector θ. Constructing them is done recursively every filter cycle in the part A of the scenario.

Before running the Kalman Filter equations on a subset of the whole map, the auxiliary structures need to be initialised.

Initialisation:

$$\begin{aligned} \phi_{t_0} &= \mathsf{I} \\ \psi_{t_0} &= \mathsf{0} \\ \theta_{t_0} &= \mathsf{0} \end{aligned}$$

In every predict and update step we have to evaluate the following additional equations:

Predict Step:

$$\bar{\phi}_{t+1} = \mathsf{A}^{AA}\ \phi_t$$

Update Step:

$$\begin{aligned} \phi_{t+1} &= (\mathsf{I} - \xi_{k+1})\ \bar{\phi}_{t+1} \\ \psi_{t+1} &= \psi_t + \bar{\phi}_{t+1}^{\top}\ \kappa_{t+1}\ \bar{\phi}_{t+1} \\ \theta_{t+1} &= \theta_t + \bar{\phi}_{t+1}\ (\mathsf{C}^A)^{\top}\ \mathsf{S}_{t+1}^{-1}\ \nu_{t+1} \end{aligned}$$

Global Update

If a global update has to be performed, the information obtained in the local area is propagated to the part of the map not contained in this local area. The matrices Σ^{AB}, Σ^{BA} and Σ^{BB} are updated by

$$\begin{aligned} \Sigma_{t+k}^{AB} &= \phi_{t+k}\ \Sigma_t^{AB} = (\Sigma_{t+k}^{AB})^{\top} \\ \Sigma_{t+k}^{BB} &= \Sigma_t^{BB} - \Sigma_t^{BA}\ \psi_{t+k}\ \Sigma_t^{AB} \end{aligned}$$

The state μ^B is globally updated by

$$\mu_{t+k}^B = \overline{\mu}_{t+k}^B + \Sigma_t^{BA}\ \theta_{t+k}$$

Map Representation

Until now we stated that a global update is necessary when the robot leaves a local area. But how are the borders of such a local area defined? Guivant and Nebot [16] propose a regular subdivision of the whole environment based on the known range of the sensor device.

Supposing the scenario of a vehicle moving in one dimension, the local area would span the region l in which the vehicle is currently situated and its two neighbours to the left and right. This situation is depicted in Figure 3.1. Thus, part A of the map will include all the landmarks located within these three regions, each the size of the sensor range. As long as the vehicle moves within region l it only needs to update the active part of the state and covariance. When the robot leaves this region, a global update is required at full SLAM algorithm cost.

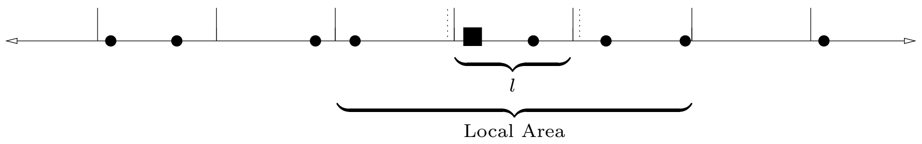


Figure 3.1: *One-dimensional map subdivided into regions. The robot, depicted as a square, is situated in region l. The local region consists of region l and its two adjacent regions to the left and right. The landmarks, depicted as circles, contained in this local area constitute the active part of the map. The dotted lines mark the hysteresis region around the central region l (please refer to text). A global update is just performed if the vehicle leaves l and the hysteresis belt surrounding it.*

When transitioning to a new region it becomes the new central area of the filter. It will consist of the point features situated in the central region and its two adjacent areas.

To avoid multiple global updates when the robot is moving close to a border, Guivant and Nebot [18] proposed a *hysteresis* region expanding the border of the current central region[1]. If the vehicle is not situated anymore within region l but within the hysteresis belt, no global update will be performed. This is also depicted in Figure 3.1 with dotted lines. The algorithm is quite sensitive to the sizes of the regions and of the hysteresis and these must be chosen carefully. If the regions are smaller than the range of the sensor or if the hysteresis is made too large, there is a chance that the vehicle will measure a landmark not contained in the active part of the map. As a result, the CKF would not be optimal as already mentioned in Section 3.3.1. This measurement could not be associated with a landmark in the local region and thus could not be incorporated into the state and covariance estimate. It would be lost.

Summary

Exploiting the internal structure of the SLAM problem, the CKF algorithm imposes an optimal solution where the expensive update of the full covariance matrix is postponed to a later stage. Between two global updates, just a subset of all the landmarks constituting the full map is updated. The rest is not considered at all. Information important for the global update and gained at these local updates is compressed in auxiliary data structures.

The whole method is summarised in Algorithm 2.

3.3.2 The Suboptimal CKF

The time and space complexity of the CKF stays the same as for the full Kalman Filter. It is quadratic in the number of features belonging to the map. The

[1] *Hysteresis* describes the continuity of an effect even if its cause disappears.

Algorithm 2: Compressed Kalman Filter Algorithm

```
Input  : Belief at time 0: $\mu_0$ and $\Sigma_0$
         Sensor range
Output: Belief at time $t$: $\mu_t$ and $\Sigma_t$

Initialisation of auxiliary structures $\phi_0 = \mathsf{I}$, $\psi_0 = \mathsf{0}$ and $\theta_0 = \mathsf{0}$
Dividing the map into regions based on sensor range;
Determine region $l$ in which the vehicle is situated
Division of $\mu_0$ and $\Sigma_0$ into $\mu_0^A$, $\mu_0^B$, $\Sigma_0^{AA}$, $\Sigma_0^{BB}$, $\Sigma_0^{AB}$ and $\Sigma_0^{BA}$
$k = 0$
for ( $i = 0;\ i < t;\ i++$ ) do
    // Predict Step
    $\overline{\mu}_{i+1}^A = \mathsf{A}^{AA}\mu_i^A$
    $\overline{\Sigma}_{i+1}^{AA} = \mathsf{A}^{AA}\Sigma_i^{AA}\mathsf{A}^{AA} + \mathsf{R}$
    $\bar{\phi}_{i+1} = \mathsf{A}^{AA}\ \phi_i$
    // Update Step
    $\mathsf{K}_{i+1}^A = \frac{\overline{\Sigma}_{i+1}^{AA}\mathsf{C}^\top}{\mathsf{C}\overline{\Sigma}_{i+1}^{AA}\mathsf{C}^\top + \mathsf{Q}}$
    $\mu_{i+1}^A = \overline{\mu}_{i+1}^A + \mathsf{K}_{i+1}^A(\mathsf{z}_{i+1}^A - \mathsf{C}\overline{\mu}_{i+1}^A)$
    $\Sigma_{i+1}^{AA} = (\mathsf{I} - \mathsf{K}_{i+1}^A\mathsf{C})\overline{\Sigma}_{i+1}^{AA}$
    $\phi_{i+1} = (\mathsf{I} - \xi_{i+1})\ \bar{\phi}_{i+1}$
    $\psi_{i+1} = \psi_i + \bar{\phi}_{i+1}^\top\ \kappa_{i+1}\ \bar{\phi}_{i+1}$
    $\theta_{i+1} = \theta_i + \bar{\phi}_{i+1}\ \mathsf{C}^A\ \mathsf{S}_{i+1}^{-1}\ \nu_{i+1}$
    if $\mathsf{x}_{i+1}$ not within $l$ + hysteresis then
        // Global Update
        $\Sigma_{i+1}^{AB} = \phi_{i+1}\ \Sigma_k^{AB} = (\Sigma_{i+1}^{AB})^\top$
        $\Sigma_{i+1}^{BB} = \Sigma_k^{BB} - \Sigma_k^{BA}\ \psi_{i+1}\ \Sigma_k^{AB}$
        $\mu_{i+1}^B = \mu_k^B + \Sigma_k^{BA}\ \theta_{i+1}$
        // Determine new active subset of map
        Determine region $l$ in which the vehicle is situated
        Division of $\mu_{i+1}$ and $\Sigma_{i+1}$ into $\mu_{i+1}^A$, $\mu_{i+1}^B$, $\Sigma_{i+1}^{AA}$, $\Sigma_{i+1}^{BB}$, $\Sigma_{i+1}^{AB}$ and
        $\Sigma_{i+1}^{BA}$
        Initialisation of auxiliary structures $\phi_{i+1} = \mathsf{I}$, $\psi_{i+1} = \mathsf{0}$ and
        $\theta_{i+1} = \mathsf{0}$
        $k = i + 1$
    end
end
```

expensive update of the whole covariance matrix is shifted to a later stage and not performed every filter cycle.

Davison [8] and Castellanos et al. [5] showed that it is essential to maintain the full covariance matrix in order to avoid a diverging map and robot trajectory. This holds especially for covariance matrices where the cross correlation between state entities is strong.

Nevertheless, Guivant and Nebot [16] claim that a decorrelation, which involves neglecting certain cross correlations, is possible with a close to optimal estimation result. A precondition for this is that the decorrelated state entities are weakly correlated. In order to ensure these weak correlations, they used a *Relative Landmark Representation* (RLR). The following quotation from [16, p.5], sums up the characteristic of RLR:

> This representation divides the map into sub-regions where the landmarks are defined respect to local coordinates frames [15]. For the 2-D case each local frame is defined based on two local landmarks represented in global coordinates. The high correlation characteristics persists but only between the frame base landmarks and the vehicle states. These landmarks represent a small subset of the total landmark representation.

For the one-dimensional case as we consider it in this work, each local reference frame is defined by the global position of one landmark since no orientation is needed. In the following we will also refer to these globally addressed landmarks as *anchors*. An anchor constitutes the local origin for all the relative features belonging to the according submap or subregion. Since no further information is given on that issue in [16, 18], we assume that the assignment of relative features to certain subregions is made arbitrarily. We suppose that there is a maximum amount of features a submap can carry. Relative features are added to a map as long as the maximum is not reached. This implies that one submap carries relative features which are quite close to each other.

In Figure 3.2 which is adapted from Guivant and Nebot [16, p.6] we can see that the correlations between relative landmarks belonging to different constellations tend to be extremely low. This holds especially for submaps which are separated by a large distance and thus not adjacent. Furthermore, the uncertainty concerning the relative landmarks will be much smaller than the uncertainty about the globally referred anchors. This is due to the fact the relative positions between landmarks and even between the robot and landmarks can be determined much more exactly than their global position. Estimates of global coordinates are dependent on the uncertainty about the global robot position and the measurement noise covariance when the according landmark was measured. Regarding relative positions, only the measurement error covariance becomes important.

In order to save some time and space when performing global updates, decorrelation can be applied to neglect these low cross correlation values between features belonging to different submaps. According to Guivant and Nebot [16] this will just cause a small loss of the estimation quality.

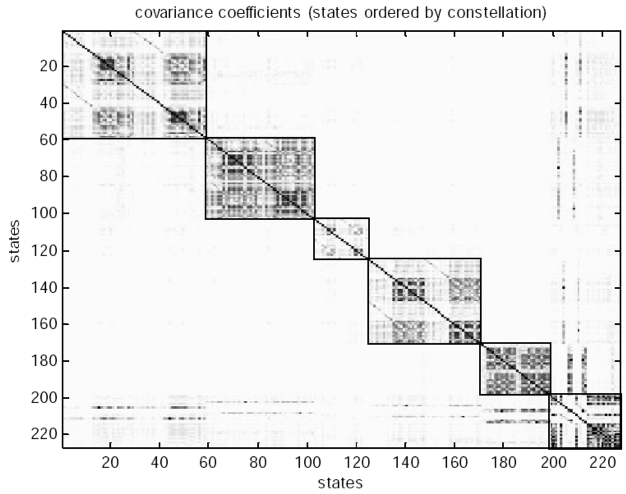


Figure 3.2: *Correlation coefficients using a relative landmark representation. The bottom right block has the vehicle state and all the absolute landmarks. All the other blocks contain the relative states in each local submap. The darker the label, the higher is the coefficient. The picture is adapted from Guivant and Nebot [16, p.6].*

The suboptimal version of the CKF introduced in [16] is basically a combination of the optimal Compressed Filter using RLR and a decorrelation technique. In this section we begin by summarising the latter and continue by describing how to apply this decorrelation to the CKF.

A Decorrelation Technique

As summarised by Knight [25, p.8], many researchers, e.g., Leonard and Feder [29] or Guivant et al. [17], state that

> [...] any estimated covariance which is provably more uncertain than the optimal matrix is likely to leave the validity of the Kalman Filter unaffected.

To put it in another way, the estimated covariance has to be conservative. In order to avoid a non-conservative decorrelated covariance matrix, it is not possible to simply neglect the cross correlation terms between different submaps. Some changes to the diagonal part need to be done.

Guivant [15] showed that a symmetric positive definite matrix $\mathsf{P} \in \mathbb{R}^{2\times 2}$ can be decorrelated such that the result is conservative compared to P. Suppose P to be defined as follows:

$$\mathsf{P} = \begin{bmatrix} p^{11} & p^{12} \\ p^{21} & p^{22} \end{bmatrix}$$

P can be rewritten as

$$\mathsf{P} = \mathsf{G} - \tau = \begin{bmatrix} p^{11} + \rho \cdot |p^{12}| & 0 \\ 0 & p^{22} + \frac{|p^{21}|}{\rho} \end{bmatrix} - \begin{bmatrix} \rho \cdot |p^{12}| & -p^{12} \\ -p^{21} & \frac{|p^{21}|}{\rho} \end{bmatrix} \tag{3.5}$$

If $\rho > 0$ then τ is a nonnegative definite matrix. This is denoted as

$$\begin{bmatrix} \rho \cdot |p^{12}| & -p^{12} \\ -p^{21} & \frac{|p^{21}|}{\rho} \end{bmatrix} \geq \mathsf{0}$$

From this it follows that G is the conservative result of the decorrelation of P. This is denoted by

$$\mathsf{G} - \tau \leq \mathsf{G}$$

A similar technique can be applied to covariance matrices we like to decorrelate. In the following we will assume a block matrix Σ denoted as

$$\Sigma = \begin{bmatrix} \alpha & \mathsf{D} & \mathsf{E} \\ \mathsf{D}^\top & \beta & \mathsf{F} \\ \mathsf{E}^\top & \mathsf{F}^\top & \gamma \end{bmatrix} \quad \text{with} \quad \mathsf{D} = \begin{bmatrix} d_{11} & \dots & d_{1m} \\ \vdots & \ddots & \vdots \\ d_{n1} & \dots & d_{nm} \end{bmatrix}$$

Suppose that we want to partially decorrelate Σ by setting the cross correlation D between α and β to zero. The rest of the matrix namely E, F and γ are left untouched. The to be decorrelated part of Σ can be rewritten as

$$\begin{aligned} \begin{bmatrix} \alpha & \mathsf{D} \\ \mathsf{D}^\top & \beta \end{bmatrix} &= \begin{bmatrix} \alpha & \mathsf{0} \\ \mathsf{0}^\top & \beta \end{bmatrix} - \begin{bmatrix} \tilde{\alpha} & -\mathsf{D} \\ -\mathsf{D}^\top & \tilde{\beta} \end{bmatrix} + \begin{bmatrix} \tilde{\alpha} & \mathsf{0} \\ \mathsf{0}^\top & \tilde{\beta} \end{bmatrix} \\ &= \begin{bmatrix} \alpha + \tilde{\alpha} & \mathsf{0} \\ \mathsf{0}^\top & \beta + \tilde{\beta} \end{bmatrix} - \begin{bmatrix} \tilde{\alpha} & -\mathsf{D} \\ -\mathsf{D}^\top & \tilde{\beta} \end{bmatrix} \end{aligned}$$

The condition is that the decorrelated version of Σ needs to be conservative regarding Σ. This denoted as:

$$\begin{bmatrix} \alpha & \mathsf{D} \\ \mathsf{D}^\top & \beta \end{bmatrix} \leq \begin{bmatrix} \alpha + \tilde{\alpha} & \mathsf{0} \\ \mathsf{0}^\top & \beta + \tilde{\beta} \end{bmatrix}$$

This is just fulfilled if

$$\begin{bmatrix} \tilde{\alpha} & -\mathsf{D} \\ -\mathsf{D}^\top & \tilde{\beta} \end{bmatrix} \geq \mathsf{0}$$

In order to fulfil this condition of non-negativity, according to Equation (3.5), $\tilde{\alpha}$ and $\tilde{\beta}$ have to be chosen as follows

$$\begin{aligned} \tilde{\alpha} &= \begin{cases} \sum\limits_{k=1}^{m} \rho_{ik} \cdot |d_{ik}|, i = j \\ 0, i \neq j \end{cases} \\ \tilde{\beta} &= \begin{cases} \sum\limits_{k=1}^{n} \frac{|d_{ki}|}{\rho_{ki}}, i = j \\ 0, i \neq j \end{cases} \end{aligned}$$

with $\rho_{ik} > 0$ for all i and k. In this work we will choose the simplest and most conservative technique and set $\rho_{ik} = 1$ for all i and k.

$$
\begin{aligned}
\tilde{\alpha}_{ii} &= \sum_{k=1}^{m} |d_{ik}| \\
\tilde{\beta}_{jj} &= \sum_{k=1}^{n} |d_{kj}|
\end{aligned}
$$

Due to the fact that the cross correlations between different submaps are close to zero when RLR is used, the increment in the diagonal terms of the decoupled covariance matrix will be small. According to Guivant and Nebot [16] this will make the suboptimal CKF approach close to optimal.

Applying Decorrelation to the CKF

To be able to reduce the quadratic complexity of the global update of the CKF, it is integrated with the decoupling technique described in the previous section. This technique is used to decouple the relative features belonging to different submaps from each other.

Information obtained during a local update will then affect the vehicle state, all the anchors, the active relative point features, and relative point features which are part of a partially active submap since their cross correlations are maintained. Improvements regarding relative landmarks which are not part of an active constellation are completely ignored due to the omitted cross correlations between different submaps. The quality of these landmarks will thus stay the same after a global update. To compensate this loss of information, the diagonal of the passive covariance part is incremented when applying the decoupling technique. This is similar to adding some uncertainty to the passive states. Also the active relative landmarks are *inflated* because they are decoupled from the passive relative landmarks.

The method summarised in Algorithm 2 stays essentially the same. Changes are made in the subdivision of the state into an active and passive part and by applying the decorrelation after a global update.

The difference in the partitioning of the state is due to the occurrence of anchor features and relative features. In the conventional optimal CKF all features were given with respect to the global coordinate frame.

We have five different categories of features. In Figure 3.3 their characteristics are illustrated.

- The first group denoted as y_l^A contains all relative features which are located within the local area around the vehicle. In Figure 3.3 they are depicted as white circles.
- The group denoted by y_g^A covers all anchors which are located within the local area around the vehicle. Additionally, this group contains anchors which are not located in this area but which define a local reference frame for some active relative features from group y_l^A. In Figure 3.3 they are depicted by white triangles.

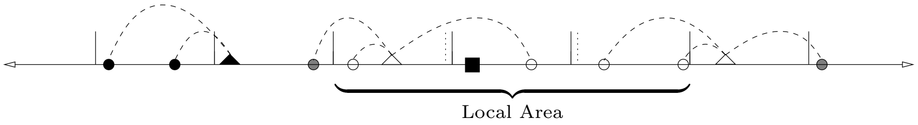


Figure 3.3: *One-dimensional robot navigation situation. The vehicle is depicted as a black square. Essentially the same situation as in Figure 3.1 is depicted but with a relative landmark representation, thus with additional anchor features. These anchor features are depicted as triangles. They define a local reference frame for submap. The dashed lines label which relative features belong to the according submap. The different colours of the symbols are explained in the text.*

- Group three contains all anchors which are not within the local area and does not form a base for features from y_l^A. They are illustrated in the mentioned picture by black triangles. This group is named y_g^B
- There might be relative features which belong to a partially active submap but are not located within the local area. They are gathered under the name $\mathsf{y}_{l_1}^B$ and are depicted by grey circles.
- $\mathsf{y}_{l_2}^B$ describes relative features addressed within a local reference frame defined by an anchor of group y_g^B. They are not contained in the local area and labelled by black circles.

The active part of the state will then consist of all the anchors contained in group y_g^A and the relative features covered by group y_l^A. The passive part contains the rest of the features whether anchor or relative, which are out of range from the sensor: y_g^B, $\mathsf{y}_{l_1}^B$ and $\mathsf{y}_{l_2}^B$.

Please note that the relative landmark representation and hence the according division of the state can also be applied for the optimal CKF. It is nevertheless a condition for applying decorrelation without loosing significant information.

Supposing that the decorrelation is applied before the full covariance matrix was restructured, the resulting covariance used by the suboptimal CKF is depicted in Figure 3.4. What is not visible from the picture are the proportions between part A and part B. Usually, the passive subset of the map will be significantly larger than the active part.

Complexity of the Suboptimal Compressed Kalman Filter

The complexity of the optimal CKF is dramatically reduced. Supposing that we have n features altogether and that there is a maximum amount n^A of features within the local area, the time and space complexity is approximately $O(n \cdot n^A)$. Usually, n^A will be significantly lower than n especially in large scale environments.

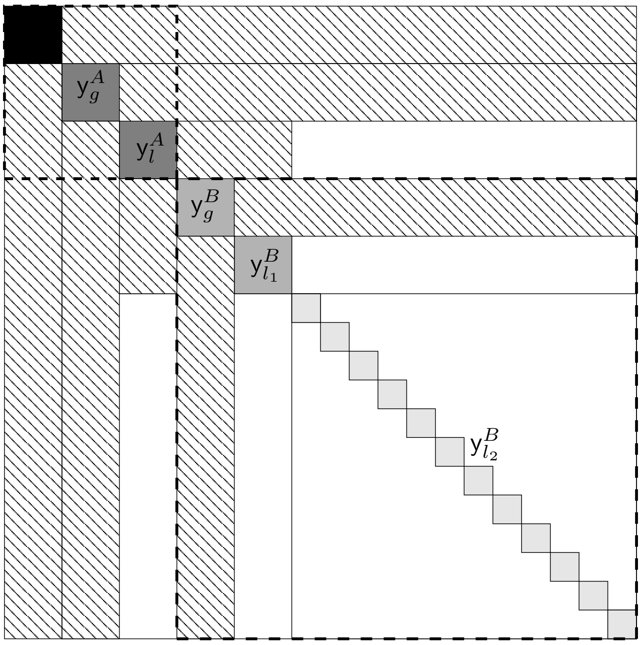


Figure 3.4: *The full covariance matrix restructured. The black box is depicting the covariance part regarding the vehicle. Maintained cross correlations are indicated by hatched parts. As you can see, they are situated between the vehicle and all states, between all anchor features, and between relative features belonging to the same submap. Omitted cross correlations are left white. They are situated between relative features belonging to different constellations.The upper left dashed box marks part A, the lower right dashed box, part B.*

Chapter 4

Towards a New Submapping Approach

The goal of this work is to develop the idea of a submapping algorithm that can handle a hierarchical state structure. We will first of all point out the main requirements for a shortcut algorithm. The Compressed Kalman Filter facilitating the concept of submaps is discussed regarding these requirements.

The CKF is subdividing the environment based on the sensor range into a number of regions. Under the assumption that region transitions are rare, the algorithm will provide close to optimal results. In the long term we want to use a stereo camera as a measurement device. Since its range is not that easy to determine, such a subdivision is not applicable when applying a camera sensor.

In order to overcome this limitation we propose a new approach for a submapping algorithm which is related to the CKF.

The chapter is structured as follows. First of all the basic requirements of shortcut methods are presented. After that CKF is evaluated with simulated data. In the last section, we propose our own idea.

4.1 Requirements for a Submapping Algorithm

In this section, we are presenting the fundamental requirements for methods that are trying to improve the Kalman Filter equations in terms of efficiency. Because of its theoretical soundness, the Compressed Kalman Filter is a promising attempt to deal with large maps by facilitating the concept of submapping. Submapping means to divide the global map into multiple smaller maps. Instead of using the whole data, a smaller amount of it will be facilitated. In that point, this field seems to be related to human navigation. A human being is not maintaining the whole map of the known world while navigating. Instead a smaller part of it referring to the current local environment will be used and refined. The CKF in terms of submapping will be discussed later in this section.

4.1.1 General Requirements for Shortcut Methods

In 2001, Knight [25] presented a review of so-called *shortcut methods*. With this term he refers to all the methods to improve the standard Kalman Filter equations in terms of efficiency. Particularly, he is interested if these different algorithms could be proven to be valid. He concluded [25, p.1]

> [...] that such proof is actually impossible in all but the simplest cases, and only sufficient empirical testing can be seen as true evidence.

Nevertheless, he suggested some basic requirements for algorithms concerned with robot localisation and mapping and how they could be theoretically validated besides using empirical methods.

As fundamental requirements, he emphasises *stability* and *correctness*. Stability in the case of a state estimator like the Kalman Filter means that the state estimate converges. Correctness means that the state converges to the true value. Knight [25] mentioned two indicators for the fulfilment of the requirements: *consistency* and *conservatism*.

His definition of *basic consistency* is adopted from Bar-Shalom [1] and says that a random estimator of a non-random variable is consistent if it converges in the limit to the true value. More precisely, the expected value of the mean square error between the estimated and true value should be zero in the limit.

$$\lim_{k \to \infty} E\{[x(k) - x_0]^2\} = 0$$

Please recall, an estimate is *conservative* if its estimated uncertainty is larger or equal to the true uncertainty. This is provided by the EKF when all assumptions, e.g., suitable process and measurement model, Gaussian noise, hold. Supposing that $\mathcal{P}$ denotes the covariance matrix yielded by the EKF, a conservative covariance matrix P regarding $\mathcal{P}$ is denoted by

$$\mathsf{P} \geq \mathcal{P}$$

As already mentioned in Section 3.3.2 and Section 2.3.2, conservative estimates of a state are unlikely to corrupt the basic consistency requirement.

This assumptions is based on the consideration that a smaller uncertainty compared to the real uncertainty is likely to miss out the true value of the estimated variable. As shown in Figure 2.1, where an example run of a vehicle is shown. A fully decoupled filter was used. The map was estimated optimistically such that the true positions of some of the landmarks where outside of the ellipse bounding the uncertainty region. This ellipse is used to bound the area where each landmark is searched to achieve a measurement. To associate the measurements with landmarks will be hard or even impossible since their true position is not within the area expected.

These two criteria of consistency and conservatism are nevertheless quite ambivalent. First of all there are no theoretical proofs of consistency in the general case. Secondly, there is no proof that non-conservative estimates invalidate consistency.

Another requirement imposed by Knight [25] is that of constant time complexity. He claims that unless the number of map features is bound, in the long term, the computation of the map could exceed the capabilities of the system. Since our idea involves the possibility to abstract from lower to higher level features, the number of landmarks within the map may also be reduced over time. Thus in this work, constant time is not considered to be of the same importance as stability and correctness.

4.1.2 Advantages and Disadvantages of CKF

As mentioned before, we regard the Compressed Kalman Filter as a principled and suitable representative of the shortcut methods. As mentioned in Section 3.3.2, its time and space complexity is approximately $O(n \cdot n^A)$ where n is the number of all landmarks known and n^A the maximum amount of features in the local vicinity around the vehicle. Its efficiency is obtained by omitting cross correlations between entities contained in different submaps.

In contrast to methods where just a number of submaps is maintained, the CKF additionally keeps track of the global map. This is particularly advantageous regarding the correlation between the vehicle and landmarks. While the *switching* algorithms ignore these correlations in large part, the CKF maintains it completely. Information gained on the local level especially about the vehicle, is completely propagated to the whole map.

The decision to omit cross correlations between relative features belonging to different submaps is made by purpose. The relative landmark representation (RLR) is chosen to ensure that these values are close to zero.

In order to ensure that this decorrelated covariance matrix stays conservative, the cross correlations are not simply set to zero. Depending on their values, the diagonal part of the covariance is inflated. This is equivalent to adding uncertainty to the estimates.

For Guivant et al. [17] the test if a method provides conservative uncertainty estimates is a consistency test at the same time. But as Knight [25] points out, there is no proof that conservative methods will be automatically consistent. He refers to an implementation of the CEKF as described in Guivant and Nebot [18] in the Oxford system which always resulted in a distorted map although it is proven to be conservative. Knight [25] was not ruling out the possibility of an error in the implementation since the method was working for Guivant and Nebot [18]. But in his opinion the instability could be also due to an exacerbation of linearisation errors.

A further investigation of that problem is not given by Knight [25]. By using the CKF instead of the CEKF we rule out the occurrence of linearisation errors. Any instability or diverging map derived in the next section where experimental results are presented cannot be due to linearisation errors.

Besides this potential problem, there are other disadvantages in the suboptimal compressed filter. First of all, the constant time criteria imposed by Knight [25] is not satisfied either. But as already mentioned, this is not regarded that important compared to some other drawbacks.

For example, it is assumed that the vehicle navigates for a considerable amount of time in the same region and thus, the global update is just performed rarely. The computational savings will then be considerable since during a long phase in one region just the active part of the features is updated every filter cycle. Instead in the order of $n \cdot n^A$ multiplications just $n^A \cdot n^A$ multiplications have to be performed. n^A denoting the amount of active feature is usually much smaller than the total number n of landmarks in the map. By avoiding frequent global updates the frequent loss in correlation information by decorrelation is reduced, too. To support this, a hysteresis belt is introduced by Guivant and Nebot [18] which expands the current region the robot is situated in. Multiple global updates are then avoided when the vehicle is moving close to the boundary of a region.

But equally, we can assume that the more global updates are made the more correlation information is lost and the more poorly the suboptimal CKF will perform in terms of stability and correctness.

This assumption is also supported by Julier [21] where the stability of covariance inflation methods are analysed considering the Compressed Kalman Filter as an example. Julier [21] proves that there is a *lower bound* on the covariance matrix. The uncertainty of a system can never be reduced, e.g., by a sensor measurement, below this value. It is shown that by adding values to the diagonal of the covariance matrix when decorrelating, this lower bound is increased. This can degrade steady-state performance, thus convergence, of the filter.

When frequently performing global updates, correlation values are added more often to the diagonal of the covariance matrix. The stability of the suboptimal CKF could then be even more endangered.

Furthermore, it is questionable whether there is a maximum amount n^A of landmarks in the local area around the vehicle. Even if the submap size is bounded this is no guarantee for a bounded number of features within an area covering three times the sensor range. Nevertheless, it is reasonable to assume that in a large scale map the local area carries significantly less features than contained in the whole map.

Guivant and Nebot [16, 18] used a laser range finder (LRF) as a sensor to detect natural beacons in an outdoor environment. The most common beacons were trees. The range of this LRF is quite clearly defined. It amounts to 40 metres. Thus, regions of 40 square metres in size were used to divide the whole area. Now, assume that we use a camera sensor instead of an LRF. Depending on the resolution of the provided picture, the range of a camera sensor is quite large. Additionally, we can detect landmarks within three dimensions instead of just two. Trees that are more than 40 metres distant can be detected. One could suggest that we simply enlarge the size of a region to adopt to this extended scope. However, this procedure would demand a determination of a maximum range. For a camera this is far not as clearly defined as for an LRF. The division of the whole environment based on the range of sight is thus not that convenient for the usage of such a sophisticated sensor.

In general, we can state that the mentioned drawbacks seem to lead back to the rigid division of the whole map.

In the next section we attempt an evaluation of the CKF in terms of stability and correctness when facing frequent region transition.

4.2 Evaluating the Compressed Kalman Filter

In this section we present an empirical evaluation of the CKF algorithm with simulated data. We will especially compare the behaviour of the optimal and suboptimal algorithm facing more or less region transitions. In particular, we want to confirm our hypothesis that the more region transitions and thus global updates are performed, the less accurate the suboptimal CKF algorithm estimates the map and vehicle position.

In order to avoid linearisation error we have selected a setup in which a vehicle is moving in one dimension only.

We also discuss which quantities are measured and evaluated. Finally the experimental results are given.

4.2.1 Experimental Setup

Supposing a robot moving randomly in one dimension. It is equipped with a sensor capable of measuring the relative distance to randomly distributed one-dimensional point features. In the following we will describe the Compressed Kalman Filter to track the position and velocity of the robot as well as to learn the map of the environment. The map will be in the RLR.

State Representation

The state of the system consists of the position and velocity of the vehicle and the known beacons constituting the map. The map is divided into submaps. We distinguish between two kinds of beacons: anchor feature and relative features. They differ in the reference frame they are regarded to. The anchors are addressed in the global coordinate frame. At the same time they define the origin of the local reference frame for the relative features belonging to the according submap. The whole state μ_t at time t is then denoted as

$$\mu_t = \begin{pmatrix} x \\ v \\ a_1 \\ y_{1;1} \\ \vdots \\ y_{1;m_1} \\ \vdots \\ a_n \\ y_{n;1} \\ \vdots \\ y_{n;m_n} \end{pmatrix} = \begin{pmatrix} \text{Robot position} \\ \text{Robot velocity} \\ \text{First anchor} \\ \text{First relative feature in first submap} \\ \vdots \\ \text{Last relative feature in first submap} \\ \vdots \\ \text{Last anchor} \\ \text{First relative feature in last submap} \\ \vdots \\ \text{Last relative feature in last submap} \end{pmatrix}$$

Please note that the time index t is omitted for the state entities in order to simplify the notation. The variable x denotes the current position of the robot at time t and v its velocity between the two consecutive points in time t and $t-1$. We summarise the entities regarding the robot in the vector $\mathsf{r} = (x\ v)^\top$. The variable a_i denotes the anchors in the map with $i = 1, \ldots, n$. The variable $y_{i,j}$ refers to a relative feature in submap i for all $j = 1, \ldots, m_i$. It is thus addressed within the local reference frame defined by anchor a_i. The total amount N of landmarks in the map is then computed by

$$N = n + \sum_{i=1}^{n} m_i$$

Then we have $\mu \in \mathbb{R}^{2+N}$. The according covariance matrix is $\Sigma_t \in \mathbb{R}^{2+N\times 2+N}$.

Linear Process Model

Assuming that the active state of the robot is given by μ_t^A, the motion of the vehicle can be modelled by

$$\mu_{t+1}^A = \mathsf{A}_t \mu_t^A + \epsilon_t$$

A_t is the state transition matrix that projects the current active part of the state into the next point in time. The variable ϵ_t is a vector of temporally uncorrelated process errors with zero mean and covariance R_t compensating the inaccuracy of the process model.

We assume that the robot is moving at a constant velocity. Acceleration and deceleration are modelled as a. The landmarks are assumed to be static. The process model therefore just affects the state entities regarding the robot.

$$\begin{aligned} \begin{pmatrix} x_{t+1} \\ v_{t+1} \end{pmatrix} &= \begin{pmatrix} x_t & +v_t\Delta t & +\frac{1}{2}a\Delta t^2 \\ & v_t & +a\Delta t^2 \end{pmatrix} \\ &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} r_t + \begin{pmatrix} \frac{1}{2}a\Delta t^2 \\ a\Delta t^2 \end{pmatrix} \end{aligned}$$

Because the landmarks are not changing over time, the transition matrix and process noise regarding the active part of the state are as follows

$$\mathsf{A}_t = \begin{bmatrix} 1 & \Delta t & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \qquad \epsilon_t = \begin{pmatrix} \frac{1}{2}a\Delta t^2 \\ a\Delta t^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Let a be the variable of the acceleration of the robot, i.e., and Σ_a be the covariance of a, i.e.,

$$\Sigma_a = (\sigma_a^2)$$

then the covariance matrix R_t of the noise in the process model is obtained by the following projection.

$$\mathsf{R}_t = \frac{\partial \epsilon_t}{\partial a} \; \Sigma_a \; \frac{\partial \epsilon_t}{\partial a}^T$$

Given

$$\frac{\partial \epsilon_t}{\partial a} = \begin{pmatrix} \frac{\partial \epsilon_{t,1}}{\partial a} \\ \frac{\partial \epsilon_{t,2}}{\partial a} \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix}$$

$$\frac{\partial \epsilon_t}{\partial a} \; \Sigma_a = \begin{pmatrix} \frac{1}{2}\Delta t^2 \sigma_a^2 \\ \Delta t \sigma_a^2 \end{pmatrix}$$

we obtain

$$\mathsf{R}_t = \begin{bmatrix} \frac{1}{4}\Delta t^4 \sigma_a^2 & \frac{1}{2}\Delta t^3 \sigma_a^2 \\ \frac{1}{2}\Delta t^3 \sigma_a^2 & \Delta t^2 \sigma_a^2 \end{bmatrix}$$

Linear Observation Model

We assume that the robot is equipped with a sensor device that can measure the distances of point features relative to the robot. Supposing that the observation model, describing how to derive the measurement vector z_t, is linear, it can be written as follows

$$\mathsf{z}_t = \mathsf{C}\mu_t^A + \delta_t$$

where δ_t is a vector of temporally uncorrelated observation errors with zero mean and covariance Q_t. C is the observation matrix relating the measurement vector z_t to the active part of the state vector μ_t.

Due to the occurrence of two different kinds of landmarks, namely anchors and relative features, we have two different measurement models. Assuming that we obtain a measurement $z_{i;t}$ of the distance between the vehicle and the anchor a_i, we can predict that measurement by

$$\begin{aligned} z_{i;t} &= a_i - x_t \\ &= \begin{bmatrix} -1 & 0 & 0 & \dots & 1 & \dots & 0 \end{bmatrix} \mu_t^A \end{aligned}$$

In the case we measure a relative feature $y_{i;j}$, the according measurement model is given as

$$\begin{aligned} z_{(i;j);t} &= (a_i + y_{i;j}) - x_t \\ &= \begin{bmatrix} -1 & 0 & 0 & \dots & 1 & \dots & 1 & \dots & 0 \end{bmatrix} \mu_t^A \end{aligned}$$

For the sake of simplification, we assume that the correspondences between measurements and landmarks in the state are known. The complete observation matrix C can then be composed of these different models depending on the observed entities in the measurement vector z_t.

As an example let us suppose the following instance of an observation step. The active part of the state μ_t^A is structured as follows

$$\mu_t^A = \begin{pmatrix} x \\ v \\ a_i \\ y_{i;1} \\ y_{i;2} \end{pmatrix}$$

It just contains one submap with anchor a_i and two relative features $y_{i;1}$ and $y_{i;2}$. Suppose now that after predicting the state, a measurement of the anchor and the second relative feature is obtained. The measurement vector z_{t+1} is then predicted as follows.

$$\begin{pmatrix} z_{1;t+1} \\ z_{(1;2);t+1} \end{pmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 \end{bmatrix} \overline{\mu}_{t+1}$$

In this work, we assume that the covariance matrix Q_t of the noise in the observation model is given by

$$\mathsf{Q}_t = \begin{bmatrix} \sigma_s^2 & 0 & \dots & 0 \\ 0 & \sigma_s^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & \sigma_s^2 \end{bmatrix}$$

σ_s is a fixed measure determined by the accuracy of the sensor.

4.2.2 Scenarios and Criteria

Our aim is to compare the optimal and suboptimal version of the CKF in terms of stability and correctness. Before we describe the quantities we evaluate in the experiments, we present how the simulation of each algorithm is realised and the particular scenarios look like.

Realisation of the Simulation

The position of the robot in the one-dimensional world is to be limited in range. In our case, the robot is controlled not to move outside the range of $[-1, \dots, 1]$. Landmarks are also randomly distributed within the same area.

In general, each method investigated is an instance of Algorithm 2. The determination of the active and passive subset of the state underlies the rules presented in Section 3.3.2.

Please recall, we have five groups of landmarks. Generally speaking, active landmarks are in the local vicinity around the vehicle and passive landmarks are not.

For the suboptimal version, we additionally apply the decorrelation technique presented in Section 3.3.2. The cross correlations of relative features of submap i with relative features of different submaps $j \neq i$ are added to the diagonal of submap i.

Initialisation of the State and Covariance. We assume that the robot state is initially known with some uncertainty. This is reasonable since the global coordinate system is usually chosen arbitrarily. Davison [8], for example, choose the global coordinate frame to be equal to the initial robot position. This starting position was then even known without any uncertainty.

For the sake of simplification we also assume that the position of a number of landmarks is already known with some uncertainty. This known position is perturbed with noise according to the chosen initial uncertainty. Because the optimal and suboptimal version of the algorithm start with exactly the same values in the state and covariance this simplification will not corrupt the result of the comparison.

The initial state and covariance are then

$$\mu_0 = \begin{pmatrix} x \\ v \\ a_1 \\ \vdots \\ y_{n,m_n} \end{pmatrix} \qquad \Sigma_0 = \begin{bmatrix} \sigma_x^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_v^2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_s^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_s^2 \end{bmatrix}$$

No features will be added to the system during one filter run. The technique how to realise the initialisation of new features is given by Guivant [15].

The division of the environment from -1 to 1 into submaps is also fixed in advance. This division will not be regular. Instead, 4 points are set randomly in the mentioned range and impose the end or the beginning, respectively, of each submap. Then, a certain amount of landmarks is distributed within the submap. The first of them is declared as the anchor point of this submap. The other ones are the corresponding relative features.

Simulating the Robot Motion. Within the simulation the position and motion of the robot is described using

- the *position* $x \in \mathbb{R}$ of the robot
- the *velocity* $v \in \mathbb{R}$ of the robot
- the *acceleration* $a \in \mathbb{R}$ of the robot

Initially, the robot is stationary at position x_0. At every time step, we select a random acceleration value a_t according to the normal distribution $\mathcal{N}(0, \sigma_a^2)$. The velocity at time $t + 1$ is given by $v_{t+1} = v_t + a_t$. In addition, we would like to limit the range of the robot pose to the actual map range of -1 till 1. Therefore, we introduce a maximum absolute value of the velocity v_{max} of the robot. In particular, we require $a > 0$ if $x < -0.5$ and $a < 0$ if $x < 0.5$.

Practically, this means that the motion does not have a Gaussian profile. But when considering a robot moving in the real world and probably equipped with some special task, random motion with Gaussian profile is always just an approximation. The research work regarding the SLAM problem presented in Chapter 2 usually use this approximation and have proven it as reasonable.

Simulation of the Observation Process. The range of sight of the robot is limited. With a certain constant probability p_m the robot is able to perceive the landmarks within a fixed range r_s. The distance measurements of the robot are subject to Gaussian error according to the normal distribution $\mathcal{N}(0, \sigma_s^2)$.

The Simulation Environment. In Figure 4.1, a screen shot of the simulation is shown. The picture shows the results of the state estimation after a number of filter cycles performed by the optimal CKF or the full filter, respectively. An exact description of the simulation environment is given in the caption. Each algorithm got exactly the same input. The uncertainty about either the landmarks or the vehicle is given as Gaussian error distribution curves. The illustration is scaled such that the highest peak is still visible. The three landmarks located on the left side which were not measured yet by none of the filters are still known with the initial uncertainty. Due to the scaling, the peaks of the according curves have a different height in the CKF and full filter window although the are of the same value.

The different scale is caused by the fact that the relative landmarks in the CKF are more certainly known than the anchors or the feature estimated by the full filter. In the full filter all the feature points just differ slightly in the corresponding uncertainty values.

The landmarks in the full filter and the anchors in the CKF are estimated with respect to the global coordinate frame and relative to the vehicle. Thus, the uncertainty about the global robot position contributes to the uncertainty about the landmarks or anchors, respectively. The relative features in the CKF are estimated with respect to the anchor features. Thus, their estimation is independent of the global uncertainty about the according anchor feature. Instead, it depends on the relative position of the anchor to the vehicle which is quite well known in comparison to their global location.

Two Scenarios to Evaluate

Our hypothesis is that the suboptimal CKF performs worse, when the assumption of long term runs in a single region does not hold anymore. We will therefore compare each version of the CKF in two scenarios. In the first scenario, the mentioned assumption will hold. This means that a global update will be just performed occasionally.

In the second, we will manually force a frequent global update independent of a real region transition. This has no influence on the optimal CKF regarding the estimated state and covariance. Of course, the computational advantages are lost since the expensive global update is performed more frequent.

When the decorrelated CKF is globally updated independent of a region transition, more cross correlations are neglected. In the first scenario, just the values between features belonging to different submaps which are currently passive are disregarded. In the second scenario the active submaps are decorrelated even if there is no real region transition.

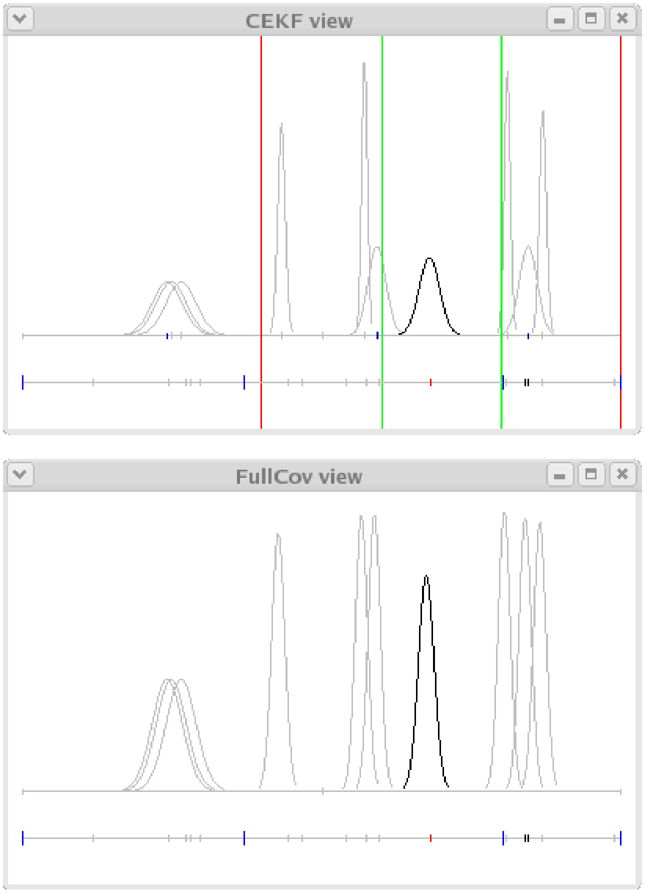


Figure 4.1: *A screen shot of the simulation program. The topmost window shows the optimal CKF using RLR. The bottommost window shows the Kalman Filter using the absolute map representation. In this example we have three submaps containing 3 landmarks including the anchor.*
The bottommost horizontal lines in each window shows the true one-dimensional environment. Small grey dashes mark the true positions of the landmarks. The red dash labels the real position of the vehicle. Black dashes mark the landmarks currently measured. Blue dashes define the borders for the submaps.
In the Full Filter window the topmost horizontal line just marks the origin of the global coordinate system and the range $[-1, \ldots, 1]$ *by grey dashes in the middle or on the left and right. In the CKF window, the topmost horizontal line additionally marks the estimated positions of the relative landmarks by grey dashes. Blue dashes label the anchors of each submap.*
The uncertainty about the estimates of either the landmarks or the vehicle is illustrated by Gaussian error distribution curves. The black curve depicts the uncertainty about the vehicle position. The higher their peak the more narrow is the curve and thus the more certain is the corresponding estimate.
In the CKF window you can see four vertical lines. The green ones bound the region the vehicle is currently situated in. The red lines limit the local area defining which landmarks belong to the active part of the state.

In each case the map will initially contain 25 features. They are distributed over five submaps, each containing one anchor and 4 relative features. The submaps are not overlapping.

In the following table, the values used in the experiments for all the parameters like initial uncertainty, range of sight, etc. are given.

Initial Uncertainty about Robot Position	$\sigma_x = 0.05$
Initial Uncertainty about Velocity	$\sigma_v = 0.1$
Initial Uncertainty about Landmark	$\sigma_s = 0.05$
Measurement Noise Variance	$\sigma_s = 0.05$
Uncertainty about Acceleration	$\sigma_a = 0.001$
Range of Sight	$r_s = 0.4$
Probability of Detection	$p_m = 0.5$

We will evaluate 10 × 200 filter runs for each scenario. We will compare the optimal and suboptimal algorithm over time regarding some criteria introduced in the next section.

Evaluation Criteria

As already introduced in Section 4.1.1 there are two main requirements for state estimators: stability and correctness. This means that the estimated state should converge to a true value. Criteria that are commonly used to ensure stability and correctness are consistency and conservatism.

Consistency. The former criteria is reached if the Euclidean distance between the estimated and real state of a systems converges. The camera in our system behaves erratic. Therefore, the error between its estimated and real position will never converge to a certain value or even to zero. Instead we say that the filter behaves correctly if the error stays within the uncertainty bounds imposed by the according covariance value.

As mentioned earlier, Julier [21] showed that there is a lower error bound to the covariance matrix which cannot be exceeded by, e.g., measurements. Thus, the error between the estimated positions of the landmarks and their real position should converge to this lower bound.

In order to compare the positions of the real and estimated map, we will transfer it to the vehicle reference frame. This will eliminate uncertainties in the map estimates introduced by the estimated global vehicle. Suppose that we have an anchor point a_i that is referred in the global reference frame. To project it into the vehicle reference frame to yield $a_{i,x}$ we have to calculate

$$a_{i,x} = a_i - x \tag{4.1}$$

Rewriting Equation 4.1 as a matrix T_a, we yield

$$\mathsf{T}_a = \begin{bmatrix} -1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{4.2}$$

Multiplying T_a with the state vector x yields the anchor $a_{i,x}$ referenced in the vehicle reference frame. Furthermore, we will transfer all the relative landmarks into the same reference frame as the anchor points, thus into the vehicle reference frame. Suppose that we have y_j as a relative feature referenced to the anchor a_i. Its position $y_{j,x}$ within the vehicle frame is calculated by

$$y_{j,x} = x - (a_i + y_j) \tag{4.3}$$

Rewriting Equation 4.3 as a matrix T_y we have

$$\mathsf{T}_y = \begin{bmatrix} -1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{4.4}$$

Multiplying this matrix with the state vector yields the relative feature $y_{j,x}$. After this map transformation, we will evaluate the *mapping error* as the Euclidean distance between this estimated transformed map and the real map as provided by the simulator.

Sim and Roy [41] analysed exploration strategies for collecting map data when using the Kalman Filter. They evaluated their experiments in two dimensions and stated that arbitrary error could be introduced before the map comparison by rotating it around the robot start pose. Thus they applied a correction procedure to rotate the map back such that the error between the real and estimated map is minimised.

We will apply a similar technique for our one-dimensional scenario. *Brent's method* as presented by Press et al. [40] is used to shift the landmarks until the minimum deviation between the estimated and real map is reached. The output after each filter cycle will be a minimum mapping error ω_m at that particular point in time and an offset ω_p denoting how much the map had to be shifted until this minimum error had been reached. Suppose that $\bar{\mathsf{y}}$ denotes the estimated transformed map and y the transformed map as provided by the simulator each containing N landmarks. The mapping error is then computed as

$$\omega_m = \sqrt{\sum_{k=1}^{N} ((\bar{y}_i + \omega_p) - y_i)^2}$$

Since we are searching for the minimum error ω_m we have to minimise $\frac{\partial \omega_m}{\partial \omega_p}$.

The offset ω_p used to shift the whole map to the best match is applied to correct the vehicle position. It will be subtracted from the estimated global vehicle position x to compare the resulting value with the true robot position x_{true}. The difference is referred to as *position error* ω_x.

$$\omega_x = (x + \omega_p) - x_{true}$$

Conservatism. The optimal and suboptimal version of the Compressed Kalman Filter are proven to be conservative (Section 3.3). Thus, the criteria of conservatism is already reached. But as mentioned in Section 4.1.1, this is neither a proof for correctness nor for stability. It is just likely that conservative estimates do not corrupt the validity of the Kalman Filter. But it is also possible that it causes instability as discussed in Section 4.1.2. Thus, we like to analyse how the uncertainty about the state estimates develops when using the suboptimal CKF with frequent global updates.

Sim and Roy [41] argued to use the trace of the covariance matrix as a measure for uncertainty. In our previous work, we applied the technique as presented by Davison [8, 9], Whaite and Ferrie [45]. We evaluated the volume of the uncertainty ellipse or ellipsoid in three dimensions, respectively, as a measure of uncertainty. This volume V is the product of the eigenvalues λ_i of a covariance matrix Σ.

$$V(\Sigma) = \prod_{i=1}^{n} \lambda_i$$

Sim and Roy [41] claim that this does not account for the case that we obtain information for one dimension but not for the others. The matrix trace T instead is equal to the sum of all eigenvalues of the matrix.

$$T(\Sigma) = \sum_{i=1}^{n} \lambda_i \tag{4.5}$$

In [41] a better map accuracy was yielded when using this sum as a measure of uncertainty. The sum of the eigenvalues of a matrix is at the same time the sum of the variances σ_i^2 contained on the diagonal of a covariance matrix.

For evaluating the behaviour of the uncertainty about the map, we will use the covariance matrix when transformed into the vehicle reference frame. Firstly, we need to project the uncertainty values Σ_{a_i} concerning the anchors a_i into the vehicle frame. Secondly, the covariances Σ_{y_j} related to the relative features y_j have to be transformed into the vehicle frame. In order to do that we multiply the whole covariance matrix Σ from the left and right with the matrix X. This matrix X will contain a transformation matrix T_a as given in Equation (4.2) for every anchor in the submap and a transformation matrix T_y as given in Equation (4.4). X will then be as follows

$$\mathsf{X} = [\ \mathsf{T}_a \ \underbrace{\mathsf{T}_y \ \dots \ \mathsf{T}_y}_{\text{First Submap}} \ \dots \ \mathsf{T}_a \ \underbrace{\mathsf{T}_y \ \dots \ \mathsf{T}_y}_{\text{Last Submap}}]^\top$$

The covariance matrix Σ_x is derived as follows

$$\Sigma_x = \mathsf{X}\Sigma\mathsf{X}^\top \tag{4.6}$$

The additional uncertainty introduced by the global vehicle position is then eliminated. To derive the *mapping uncertainty* Ω_m we calculate the following:

$$\Omega_m = 3 \times \sqrt{\sum_{i=1}^{n} \sigma_i^2}$$

This means that we sum up the variances σ_i^2 of the transformed matrix Σ_x as in Equation (4.5). Then, we extract the square root of this sum to derive the sum of the standard deviations σ_i. The probability that the possible realisation of a standard deviated random variable lies within the $3 \times \sigma_i$ region around the mean of the distribution is approximately 99% ([20, p.1119]). Thus, the sum of deviations is multiplied by three to derive error bounds which cover as many potential true feature locations as possible.

The *position uncertainty* Ω_x will be calculated equivalently to the mapping uncertainty based on the standard deviation σ_x about the global vehicle position.

$$\Omega_x = 3 \times \sigma_x$$

Ω_x describes the maximum and minimum possible deviation of the estimated robot position from the real vehicle pose.

4.2.3 Results

As already mentioned we will analyse two scenarios. In the first scenario, we will compare the optimal CKF with the decorrelated CKF when the assumption holds that the vehicle stays for a long time in one region. In the second scenario we will force a global update more often even if there is no region transition.

The following measures are evaluated

Position Error. The Euclidean distance between the estimated and real vehicle position within the global coordinate system at a certain point in time. The estimated position is shifted about an offset to achieve the minimum mapping error.

Position Uncertainty. The uncertainty value of the vehicle within the global reference frame at a certain point in time.

Mapping Error. The Euclidean distance between all the estimated and true map features with respect to the vehicle reference frame at a certain point in time. The estimated map is shifted about an offset to achieve the minimum mapping error.

Mapping Uncertainty. The sum of the diagonal entries of the covariance matrix describing the uncertainty of the estimated map with respect to the vehicle reference frame.

In the following we will compare each scenario regarding the mentioned evaluation measures. Please recall, that we evaluate 10 passes of the algorithm at 200 filter cycles each. The according diagrams show average values over these 10 passes. Additionally, the maximum and minimum values are indicated by error bars.

Position Error

The position error produced by the optimal and suboptimal version of the CKF is compared for the first scenario in Figure 4.2. In general we can state that the position error is not converging to a fixed value. Instead the average position error is oscillating between 0.02 and 0.06 units.

Furthermore, there are just slight differences between the optimal and suboptimal CKF. They are just visible in the maximum or minimum error. The average position error is essentially the same.

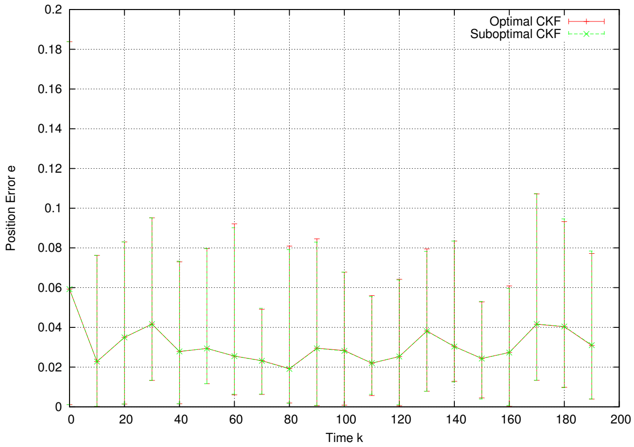


Figure 4.2: *First Scenario. A global update is performed after a region transition. Position Error.*

This supports the already mentioned statement of Guivant and Nebot [16] that the decorrelated CKF behaves close to optimal when the assumption of rare region transition holds.

The position error for the second scenario is illustrated in Figure 4.3. Again, the average position error is not converging against a certain value. Here there are not just differences visible in the error range, but also in the average error. In the very beginning the average error is almost the same. In the middle part from filter cycle 50 till 120, the suboptimal CKF behaves even better than the optimal filter.In the end this relation is inverted such that the optimal filter yields better results.

If we compare both figures there is no clear tendency recognisable for a worse behaviour of the suboptimal filter in the second scenario.

Position Uncertainty

Figure 4.4 shows the position uncertainty for the first scenario. Generally spoken, we can see a decrease of position uncertainty over time. Again the average values are practically the same for either of the two versions of the algorithm. Even the uncertainty ranges are almost the same. Just from filter cycle 130 till the end it is recognisable that the maximum and minimum uncertainty of the optimal filter are lower than that of the suboptimal filter.

The result for the second scenario regarding position uncertainty are given in Figure 4.5. The decrease of uncertainty over time is visible. Considering the average position uncertainty we can realise a slightly better behaviour of the optimal filter. Also the error bars show slightly better values.

If we compare both figures, we can state that the uncertainty about the vehicle

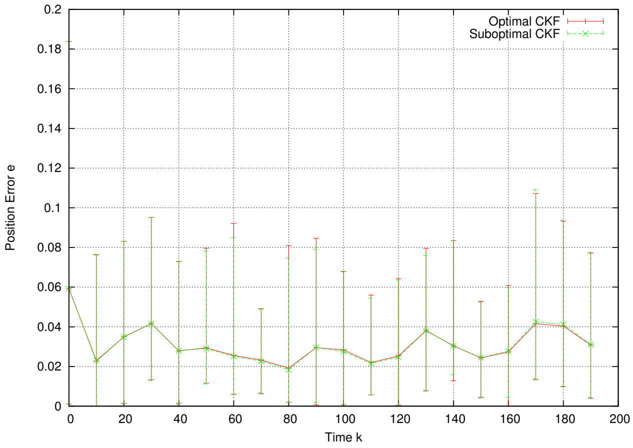


Figure 4.3: *Second Scenario. A global update is performed every second filter cycle. Position Error.*

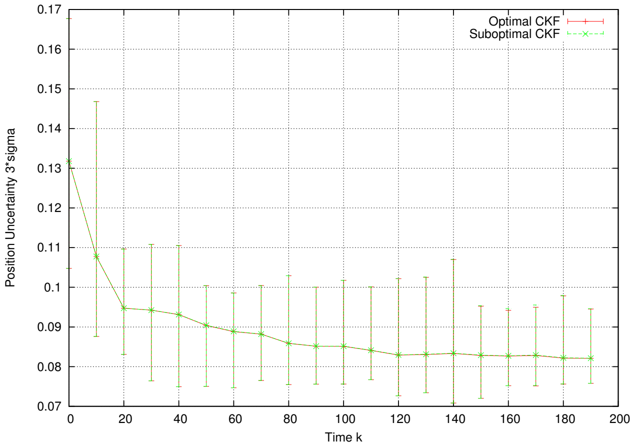


Figure 4.4: *First Scenario. A global update is performed after a region transition. Position Uncertainty.*

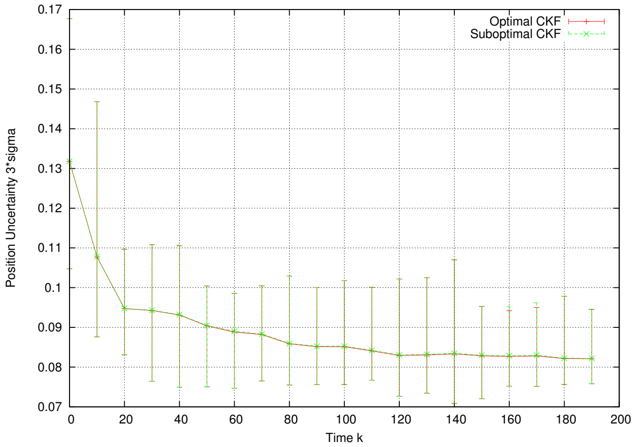


Figure 4.5: *Second Scenario. A global update is performed every second filter cycle. Position Uncertainty.*

position rises with an increasing number of global updates. On the one had this supports the conservatism of the covariance of the suboptimal CKF. On the other hand nothing is explicitly added to the uncertainty about the vehicle when the covariance is decorrelated. Thus, the uncertainty about the robot is indirectly increased by increasing the uncertainty values of the map.

Comparing the position error and uncertainty for each scenario shows that the maximum position error exceeds the maximum position uncertainty just once at filter cycle 170. Of course this does not tell us anything about what happens at single filter cycles. When running the experiments, a test was performed to determine whether the position error exceeds the position uncertainty. This problem appeared just occasionally, on average at 2 filter cycles per run, and did not lead to a diverging state. Please recall, that we multiplied the deviation by three to ensure a 99% coverage of all possible robot positions. Thus, we have one percent of potentially true vehicle positions not covered. These cases are not distinguished from cases where the position error really exceeds the predicted amount.

Mapping Error

In Figure 4.6 the mapping error regarding the first scenario for the optimal and suboptimal filter is given. Due to more and more measurements taken of the map features over time, the filter can estimate them with fewer mapping error. This is visible by a decreasing mapping error over time.

As you can see, for this case where the assumption of rare region transition holds, both versions produce almost the same results. Towards the end, the maximum and minimum mapping error values produced by the optimal filter are slightly lower than for the suboptimal filter.

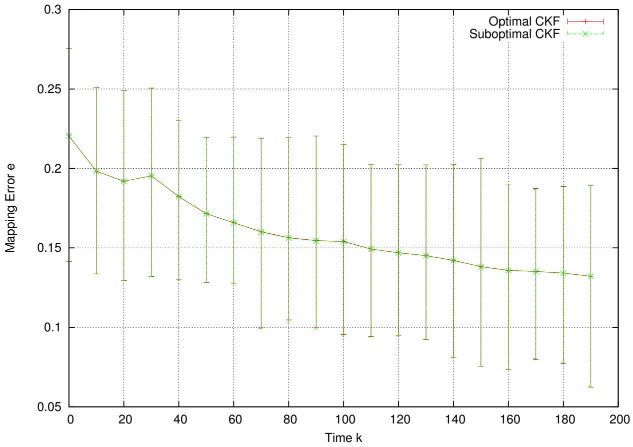


Figure 4.6: *First Scenario. A global update is performed after a region transition. Mapping Error.*

In Figure 4.7 the mapping error for the second scenario regarding the optimal and suboptimal version of the CKF is depicted. In general, it is decreasing over time. From filter cycle 70 we can notice a slightly lower average mapping error produced by the suboptimal filter.

Mapping Uncertainty

In Figure 4.8 the mapping uncertainty in the first scenario for the optimal and suboptimal filter is shown. It can be seen that the average uncertainty as well as the maximum and minimum uncertainty are always slightly lower for the optimal filter.

This is due to the decoupling where the cross correlation values between features belonging to different submaps are added to the trace of the covariance matrix.

Figure 4.9 illustrates the mapping uncertainty in the second scenario for the optimal and suboptimal filter. The conservatism of the covariance built by the suboptimal filter in the first scenario can also be observed for the second scenario. Here, this is even more clearly recognisable. This is due to the more frequent global updates which cause a more frequent addition of values to the covariance matrix trace.

4.2.4 Discussion

The experiments from the previous section were supposed to support our hypothesis that the more region transitions and thus global updates are performed,

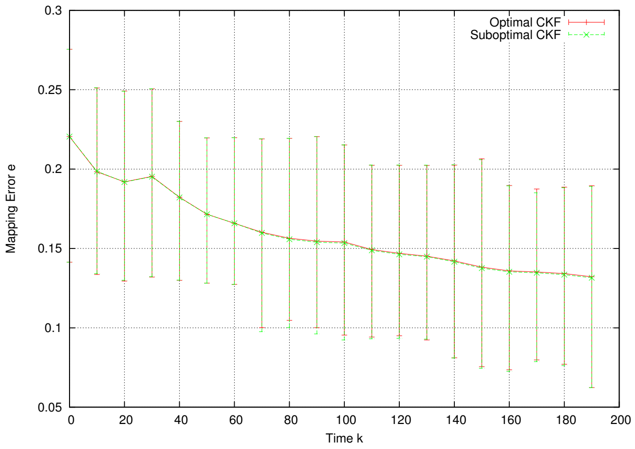


Figure 4.7: *Second Scenario. A global update is performed every second filter cycle. Mapping Error.*

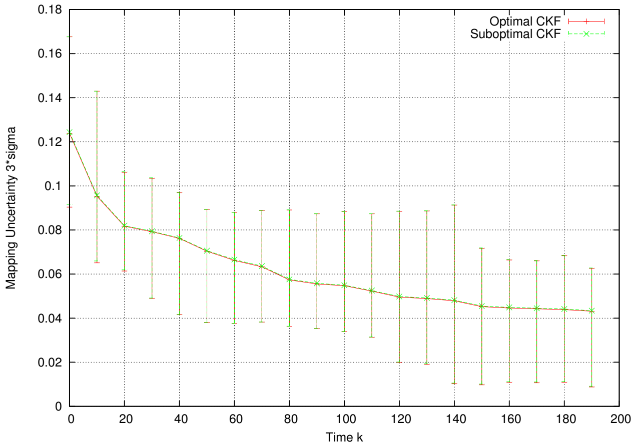


Figure 4.8: *First Scenario. A global update is performed after a region transition. Mapping Uncertainty.*

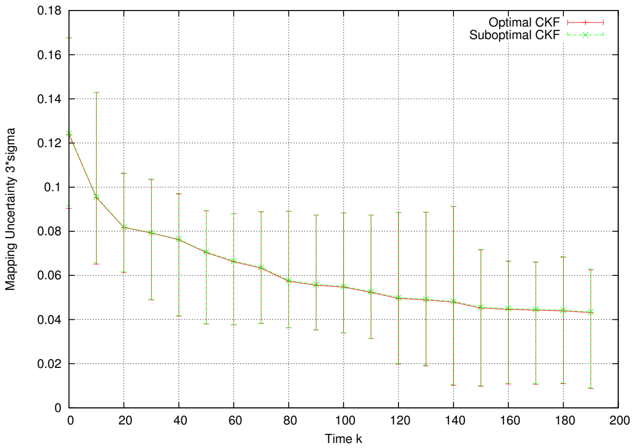


Figure 4.9: *Second Scenario. A global update is performed every second filter cycle. Mapping Uncertainty.*

the less accurate the suboptimal CKF algorithm estimates the map and vehicle position.

The experiments analysing the position and mapping uncertainty support the hypothesis that the respective uncertainty grows with increasing global updates. However, the difference between the scenarios is very small.

Considering the position and mapping error, the experiments do not give a clear answer whether the suboptimal filter behaves worse when more global updates are performed. In the case of the mapping error the suboptimal filter even seems to produce better results than the optimal instance of the CKF.

Nevertheless, we still believe that our hypothesis is valid. More global updates result in more neglected cross correlations between features belonging to different submaps. Less information on the correlation between map entities should result in less uncertain and less accurate state estimates.

To prove that, the experimental setup should be improved. In the following, we make some proposals regarding that. However, their realisation lies beyond the scope of this work.

First of all, all the features should not be pre-initialised as we have done it in this work. Instead the initialisation of landmarks during the vehicle run should be implemented. How this can be done for the CKF, is described by Guivant [15].

Suppose that the filter is initialised just with the position and velocity of the robot and that these values are known with no uncertainty. Assume, that the vehicle is moving along the one-dimensional line from the right to the left. It initialises new features as they come into the range of sight. In the beginning the robot position is known quite well. The further it moves away from its initial position the higher will be the uncertainty about its state. Depending on the uncertainty about the vehicle when it is measuring a landmark, the global location of the landmark and the corresponding uncertainty will be estimated. Thus, the uncertainties about the anchor points will be quite different from each other. The later it is measured the more uncertain it is. Considering the relative features, they will be always estimated approximately with the same uncertainty since their estimation is independent of the uncertainty about the global vehicle or anchor position. Instead it depends on the relative position between the according anchor and the vehicle.

In the experimental setup given in this work, all the anchors and relative features are initialised with the same uncertainty. The difference in their uncertainty after a considerable amount of filter cycles will not differ significantly.

The uncertainty about the vehicle position is of course not only increasing. For example, when it is re-measuring a quite certain landmark, the uncertainty about its position will drop and the estimated position will be corrected. This so-called loop closing effect is illustrated in Figure 2.1.

In our experiment no loop closing can be performed. To be able to do so, the environment should be periodic such that the landmark firstly measured can be re-measured after one round. In order to represent a periodic environment, the motion and measurement model have to be adapted. Then, we could analyse if there might be difficulties to track this firstly measured feature again. Regarding the second scenario, global updates are forced more frequently. Thus, more cross correlation information between features belonging to different submaps are lost than in the first scenario. This could decrease the ability of the filter to re-find landmarks since their predicted location might be quite offset from their real position.

To summarise, the implementation of feature initialisation and a periodic environment will in our opinion produce experimental results which support the mentioned hypothesis.

4.3 Towards a Hierarchical Kalman Filter

The aim of this work is to find an approach that reduces the quadratic time and space complexity of the SLAM algorithm based on the full Kalman Filter and maintains at the same time its accuracy as much as possible. As already shown in the related work section, this topic is as widely regarded in the SLAM community as the Kalman Filter approach itself.

One of the main directions in this subject is referred to as submapping where the whole map is subdivided into smaller parts. These submaps are used more or less independently of each other depending on the particular method. By updating just a part of the state, significant computational savings can be made. The main issue is to maintain a correlation structure between the landmarks itself and between the features and the robot which does not lead to a diverging state.

In the previous sections we analysed the Compressed Kalman Filter by Guivant and Nebot [16, 18] which is due to its theoretical soundness one of the most promising submapping algorithms. As shown in Section 4.2 it performs close to optimal when the condition of rare region transitions holds. These regions are quite rigidly established based on the scope of the applied sensor.

In our previous work [2], we illustrated the advantages when using a camera and especially a stereo camera for solving the SLAM problem. Our long term goal is therefore to use a stereo camera as observation sensor. When applying the CEKF algorithm, simply exchanging the laser range finder as used by Guivant and Nebot [16, 18] will impose problems on the division of the environment into regions. Since the scope of a camera is not that clearly limited up to certain length, the size of the regions can be just determined rather uncertainly. It is then likely that measurements of landmarks situated outside the local area around the vehicle are obtained. Since they cannot be matched with one of the landmarks within this local area, these measurements are either discarded or, even worse, used to initialise new landmarks. These landmarks could be already contained in the passive part of the state. Then we would have two entities in the state corresponding to the same beacon in the real world.

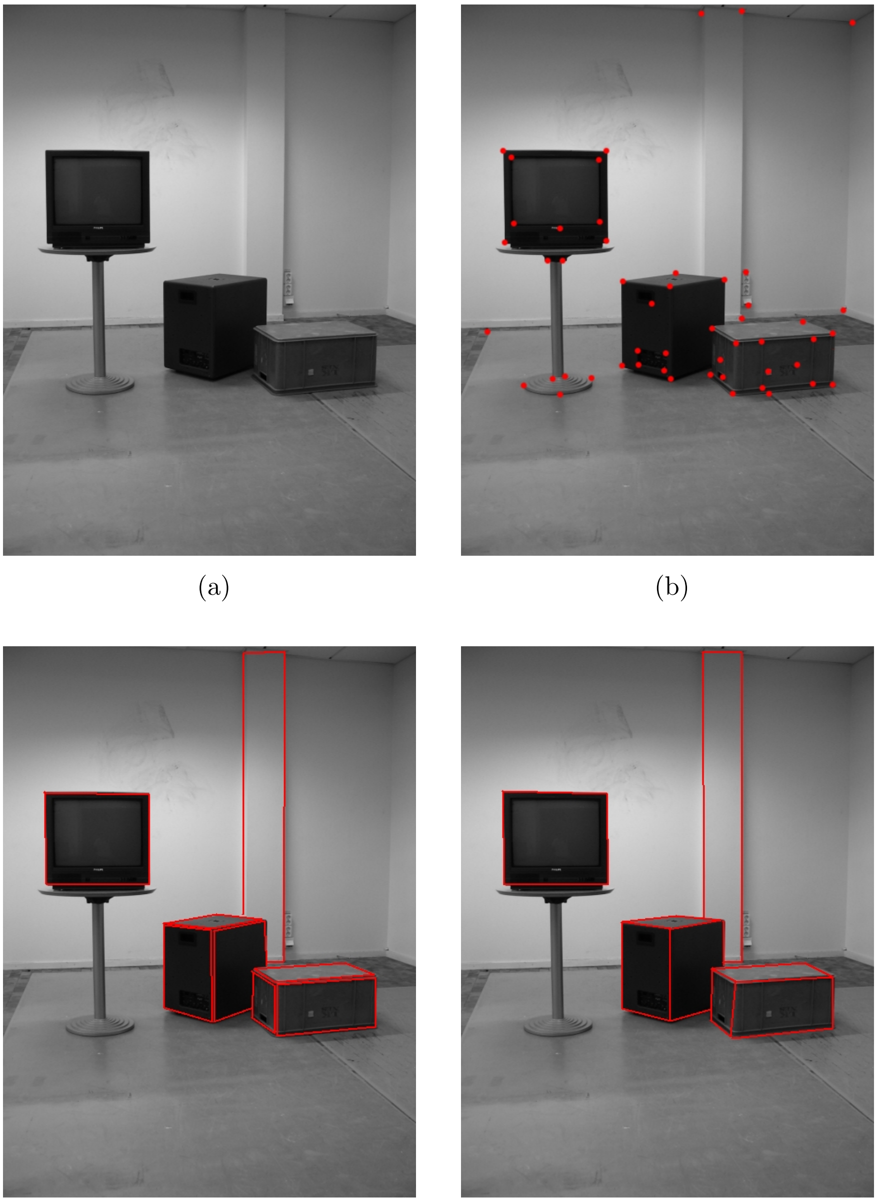
The idea proposed in this work can also be classified as a submapping technique. It takes the clearly advantageous relative landmark representation into account but does away with the regular subdivision of the environment into regions. Please recall, RLR ensures low cross correlations between features belonging to different submaps. Additionally to disregarding those low values when updating the map, we like to establish a hierarchical filter structure. Incorporating semantic information allows us to abstract from lower level to higher level features. Instead of tracking dozens of point features, we would subsume subsets of them to surfaces or even surfaces to geometrical primitives, etc. The amount of maintained features is thus decreased. However, the known features are more complex than just simple points. This makes them easier to distinguish and track since there are more attributes to identify, e.g, a surface feature has an orientation and a bounded area.

An exemplified subdivision of an indoor environment is shown in Figure 4.10. The map consists of three hierarchical levels. The lowest level contains point features. A group of points is subsumed to a surface. For this specific example, the highest hierarchical level contains primitives defined by a number of surfaces. The underlying structure of one primitive is illustrated in Figure 4.11. Although, the highest level in this example are primitives, we could think of even higher levels representing rooms, storeys, buildings, etc.

When facilitating this idea we have three problems to solve. First, we need to apply the Kalman Filter to this hierarchical structure. Second, we have to choose which subset of features of the lower level belongs to one feature of the upper level. And third, the point in time has to be found when it becomes sufficient to track just the complex feature instead of refining and measuring the underlying subset.

To find appropriate heuristics for solving the two latter issues lies beyond the scope of this work. Instead we will concentrate on the first problem which needs to be solved before we can approach the others. We will answer the specific question how to handle two hierarchical levels with the EKF without letting the learned map diverge from the real environment.

In this section, we firstly describe our idea to represent the hierarchical structure. This is followed by analysing the special requirements that emerge when



(a)

(b)

(c)

(d)

Figure 4.10: *Scribble to illustrate a potential subdivision of an indoor environment. (a) The original photo of the environment.(b) A set of point features. (c) Subsets of point features subsumed to surfaces.(d) Surfaces subsumed to primitives.*

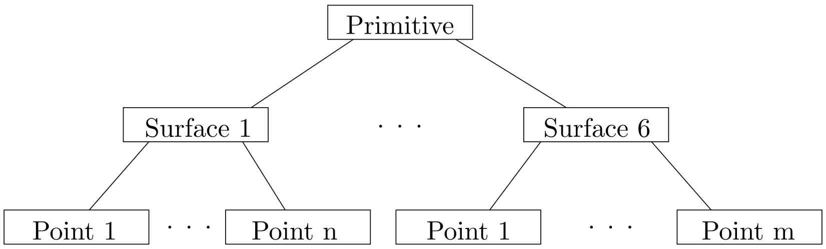


Figure 4.11: *The internal structure of one of the primitives shown in Figure 4.10(d).*

using this representation. After that we will propose potential solutions for this problem and discuss them regarding the mentioned requirements.

4.3.1 A Hierarchical Map Representation

Our goal is to develop a Kalman Filter based algorithm that can handle a hierarchically structured map consisting of a number of submaps in order to lower the quadratic time and space complexity of the full filter. For the sake of simplification, the maps considered in this section are of depth two.

As shown in Section 2, arbitrarily omitting cross correlations in the covariance matrix built by the standard Kalman Filter, will lead to a distorted estimated map in the bast case and catastrophic failure in the worst. The CKF facilitates the relative landmark representation to ensure that the correlation between features belonging to different submaps are close to zero. When the filter is neglecting these low values it will still produce close to optimal estimation results.

By splitting the whole problem hierarchically, we want to exploit the RLR and at the same time be independent of the inconvenient region division of the environment. One possible hierarchical subdivision of the map is quite closely related to the structure in Figure 4.11. The origin of each surface would be an anchor considered in the global reference frame. The lower level feature points situated on that surface would be contained in according submap states. These submaps are addressed with respect to the local reference frame whose origin is fixed to the anchors point.

The connection between the two levels is established by duplicating the robot state across the levels. On the global level, the global position of the robot with respect to the world coordinate system is included. On the local level the vehicle is represented relative to the local coordinate system defined by the corresponding anchor. Why we have to introduce that, will become clear when observing how the covariance matrix is represented hierarchically.

In Figure 4.12 the states of the full filter and the CKF are juxtaposed with the hierarchical representation.

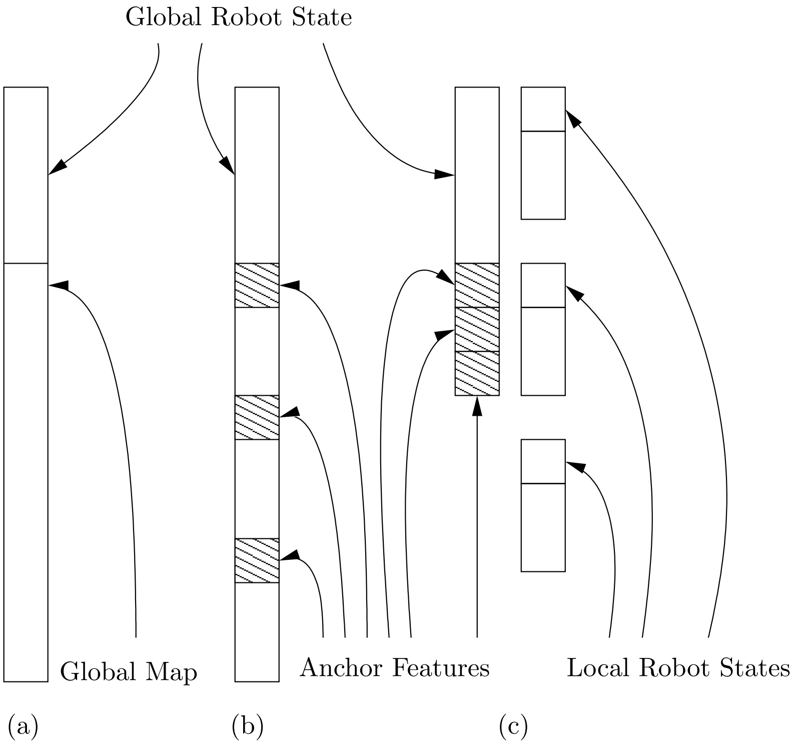


Figure 4.12: *The structure of the state of the full filter (a) and the CKF (b) compared with our proposed hierarchical representation (c).*
In every state you can find the global robot state where indicated. The map maintained by the full filter contains features which are all referenced to the global coordinate frame. The map of the CKF contains globally referenced anchor features (hatched boxes) and submaps containing relative features (white boxes). In each of the latter representations, the map and the vehicle state are gathered in one vector.
In the hierarchical state representation we have a global state (left box of (c)) which contains the global robot state and the anchors. Depending on the amount of anchors we have the same number of submap states containing the relative features and a local robot state (right boxes of (c)).

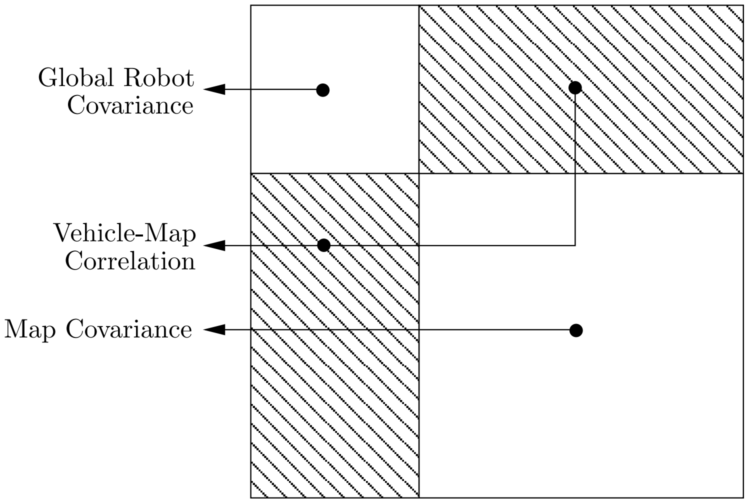


Figure 4.13: *The structure of the covariance matrix of the full filter.*

4.3.2 Requirements of a Hierarchical Kalman Filter

According to the fundamental requirements of shortcut methods mentioned in Section 4.1.1, the estimated state vector has to converge to a true value in the limit. Practically, this means that the error between the real and estimated map should approach zero in the limit. The error between the real and estimated vehicle should stay within the error bounds imposed by the corresponding uncertainty value. This criteria is summarised by the term consistency.

The second criteria mentioned in Section 4.1.1 is conservatism. Conservative uncertainty estimates are considered as unlikely to corrupt the basic consistency requirement. However, they are not proven to do that in every instance of the problem.

From these general requirements, specific requirements can be derived for the usage of the hierarchical representation. When dividing the state into a global and a number of local parts the covariance has to be structured equivalently. In the following we will compare the covariance matrices related to the states presented in Figure 4.12 and analyse them in terms of their correlation structure. In Figure 4.13 the covariance structure corresponding to the standard full filter is shown. Generally spoken, there are three kinds of entries. We have the covariance of the robot state in the left upper corner. In the right lower part, we have the covariance of the whole globally referenced map. The third kind of entries is the correlation between the map and the vehicle.

In contrast to that, Figure 4.14 shows the covariance built up by the CKF. Here, we do not just have globally referenced map features as in the full filter, but additionally relative map features. Due to that, there are more kinds of covariance entries. You can observe a subdivision of the map into three submaps. In the picture, the corresponding covariance is bounded by a thick line. Each submap contains an anchor and a number of relative features. The covariance of an anchor is indicated by a vertical hatching. They are correlated to each

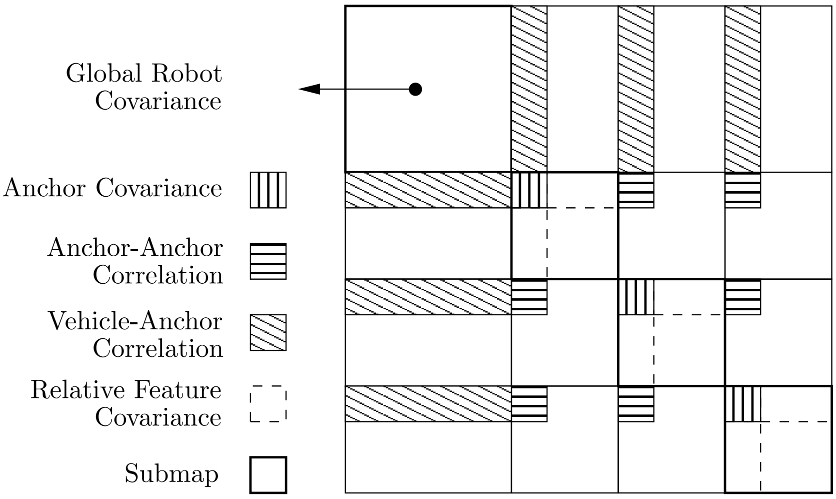

Global Robot Covariance
Anchor Covariance
Anchor-Anchor Correlation
Vehicle-Anchor Correlation
Relative Feature Covariance
Submap

Figure 4.14: *The structure of the covariance matrix of the CKF.*

other and to the vehicle. In Figure 4.14, these correlations are indicated by the horizontally or the diagonally hatched boxes, respectively. The covariance matrices of the relative features belonging to one submap are bounded by dashed lines.

Compared to that, Figure 4.15 shows the hierarchically structured covariance. The covariance referring to the global state as shown in Figure 4.12 (c) is depicted by the large box on the left. It contains the covariance of the global robot state and of the anchors, the correlations of the anchors to each other and the correlations between the anchors and the vehicle.

On the right side of Figure 4.15 the covariances referring to the submap states are shown. They contain the covariance matrix of the according set of relative features. Additionally, the covariance of the local robot state is introduced in each local covariance as well as its correlation with the relative features.

We clarified which blocks of the CKF covariance matrix are maintained when splitting it up into such an hierarchical representation. On closer examination, you can realise that there are also blocks which are not kept.

In Figure 4.16 we depicted the covariance matrix as used by the decorrelated CKF and the global covariance of the hierarchical filter temporally containing all the submap features. Neglected correlations are labelled as hatched areas. The omitted correlations for the right covariance are more than for the covariance of the CKF since there is no information available about the connection between anchors and unrelated submaps. Particularly, there are four kinds of correlations which are either totally neglected or not maintained directly. In order to avoid a diverging state, we need to develop the algorithm in such a way that these correlations are compensated or reconstructed. In the following section, the different kinds are explained and potential solutions how to handle them are presented.

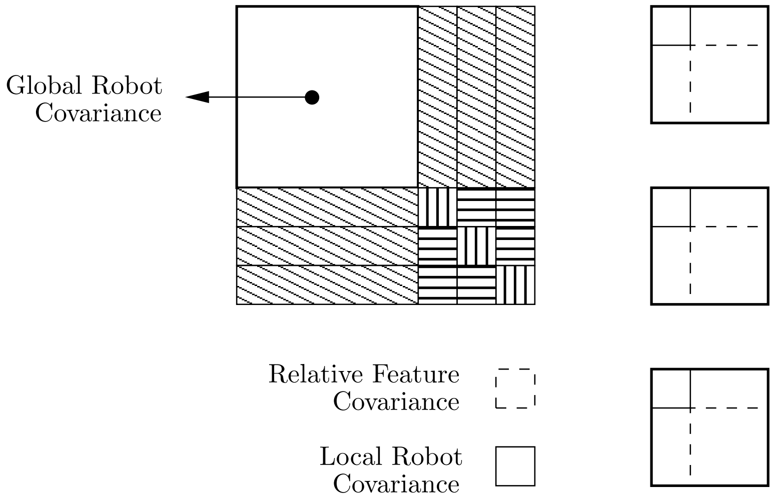


Figure 4.15: *The hierarchical representation of the covariance matrix. The hatching is adapted from Figure 4.14.*

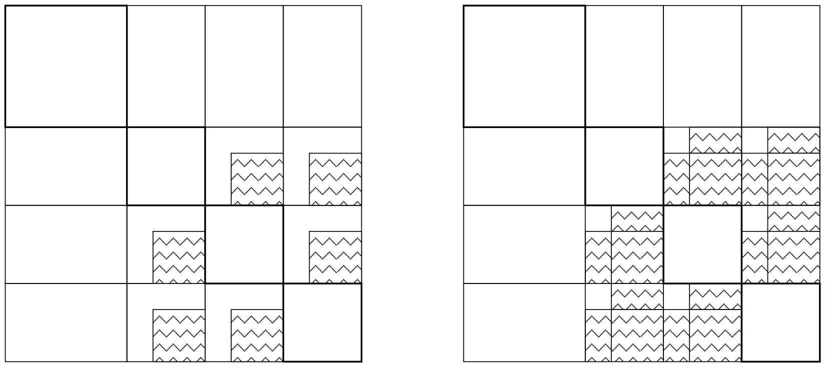


Figure 4.16: *The global covariances of the decorrelated CKF (left) and the hierarchical representation when the submaps are introduced to the global level (right).*
The structure is adapted from Figure 4.14. Hatched boxes label the correlations which are neglected. Uniform boxes label correlations or covariances as maintained.
In the left covariance the correlations between features of different submaps are omitted. The same correlations are also neglected in the right covariance. But additionally we do not have any information on the correlations between submaps and not related anchors.

4.3.3 Approaches to Fulfil these Requirements

For each block of correlations that is neglected when using the hierarchical representation, an approach to compensate the arising loss of information is presented. Generally spoken, we deeply rely on the duplicated vehicle state across the levels to propagate information between them. A rough algorithm applying that concept is given after analysing the correlation blocks. At last, two attempts to apply that algorithm are explained.

Correlation Between the Global Vehicle and Submaps

The first group of neglected covariances when using a hierarchical state and covariance representation are the blocks between the global vehicle and all submaps. These are values which become large and therefore cannot be simply neglected. Our solution to to compensate these missing correlations is to introduce a local vehicle in each local state which is referred to the related local reference frame. There will be correlations between this local robot and the relative features. We impose a constraint on this local robot state: For each submap a transformation T_{lg} should exist which maps the global robot x_g to the considered local vehicle x_l depending on the related anchor a_l.

$$\mathsf{x}_l = \mathsf{T}_{lg} \begin{pmatrix} \mathsf{x}_g \\ \mathsf{a}_l \end{pmatrix} \tag{4.7}$$

To put it in another way, the local vehicle x_l is the result of a function T_{lg} having the global robot state x_g and the corresponding anchor a_l as its parameters. In our case the transformation describes that the local vehicle position is the global position measured in the reference frame attached to the considered anchor.

T_{lg} can also be applied to the covariance of the global robot state Σ to obtain the local covariance ζ.

$$\zeta = \mathsf{T}_{lg} \Sigma \mathsf{T}_{lg}^{\top} \tag{4.8}$$

The global and the local level of the hierarchy are related to each other by this transformation. It has to be valid before and after every filter cycle.

Suppose that we transform the local states into the global level. In particular, this means to transform the local vehicle into global coordinates and leave the relative features as they are. The correlation between the transformed local vehicle and the submap features should be close to the correlations between the global vehicle and the relative features as they are existing in the CKF. This is the second constraint we impose on the hierarchical representation.

Exploiting these relations between the levels, will allow us to propagate the information obtained locally to the global level. We will further discuss this later in this work.

Correlations between Anchors and the According Submaps.

The second group of correlations, which is not maintained, relate relative features to their according anchor. This is due to the fact that the anchor is not as the vehicle included in the local state. Suppose that we incorporate an anchor in each local state by transforming it to the related local coordinate frame. Each anchor is the origin of the corresponding local frame. An anchor in local coordinates will thus be always equal to the referred origin. Also its covariance and correlation to the submap will be zero. And this will never change. Thus, we do not get any further information by incorporating the anchor in the local state.

But we do gain more information by enabling the filter to measure the anchor! Remember, the local vehicle x_l is referred to the local reference frame. Thus, the inverse $-\mathsf{x}_l$ of its position is equal to the distance between the local robot and the anchor. In the case of our one-dimensional example from Section 4.2, this would be at the same time the prediction of the measurement. Just the vehicle position is used in this measurement model.

Due to that, an update of the local state after obtaining a measurement of the anchor will strongly affect the position estimate of the local vehicle and its uncertainty. Considering correlation between the local vehicle and the relative features, it might also influence the submap estimation.

The omitted correlations between the anchors and their related submaps are thus still indirectly exploited. The exact knowledge of the local position of each anchor allows us to.

Correlations between Features of Different Submaps.

The next group of correlations which is not maintained connect the features of different submaps to each other. Due to the usage of the relative landmark representation, these value are close to zero. Guivant and Nebot [16] not just neglect but added them to the diagonal of the covariance matrix. This ensures a conservative state estimation. Please recall, conservatism means that the new covariance is equal or larger than the covariance derived by the standard full filter. In Section 4.1.1, it is also mentioned that conservatism is not a guarantee for the consistency of the state estimator. According to Julier [21], this inflation could even endanger the stability of the algorithm by constantly rising the lower error bound of the covariance matrix. Therefore, we consider it as reasonable to not establish these correlations.

In the hierarchical representation, the submap states are therefore treated as independent of each other. An update of one submap will not affect another one directly. In the suboptimal CKF, submaps contained in the active part are affected when one of the submaps is updated, since their correlation is maintained. To approximate that in the hierarchical representation, we make a detour over the global state.

Suppose that we gained some information by a measurement in one of the submaps. This will probably change the estimate of the local vehicle. In order to fulfil the constraints presented in Equation (4.7) and (4.8), we have to propagate the improvement of the local robot state to the global level. From the updated global level, the information has to be further propagated to all the other submaps. In Figure 4.17 this procedure is depicted.

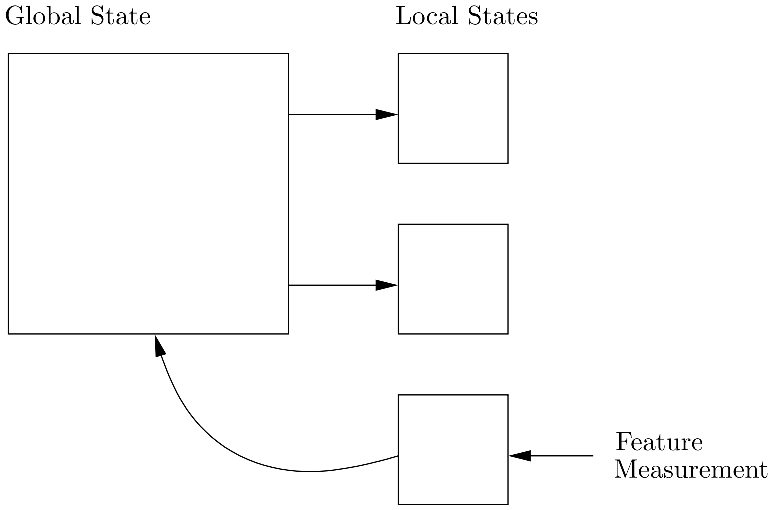


Figure 4.17: *The propagation of information between the levels. Submap 3 was updated after obtaining a measurement. The improvement of the map is propagated to the global level and from there to the other 2 local states.*

Submaps contained in the passive part of the CKF covariance might be slightly affected by an update of another submap since just the relative features of different submaps are decorrelated. The correlations with the anchor and the global vehicle are still maintained and can influence the estimate of a passive submap.

Considering the hierarchical representation, the correlations with the vehicle are as already mentioned approximated by including the local robot state. How to deal with the relation between the anchors and the unrelated submaps is discussed in the following.

Correlations between Anchors and Unrelated Submaps

The last group of correlations which is not kept when splitting the covariance of the CKF into a hierarchical structure, relates each anchor to the submaps not using it as the origin of the local reference frame. These correlations are strong in the covariance of the CKF and cannot be simply omitted. In the case of the hierarchical representation, we also approximate these values by propagating the information between the levels as shown in Figure 4.17.

The General Algorithm Handling the Hierarchical Structure.

We need to develop an algorithm that is able to compensate the missing correlation as listed in the previous subsections. From Figure 4.17 the general structure of the Kalman Filter based algorithm that is able to deal with a hierarchical representation is already recognisable. It is summarised in Algorithm 3.

The first step shown in Line 2 is to predict the global state by applying the global motion model of the vehicle. Secondly, every local state is predicted with

Algorithm 3: Kalman Filter based Algorithm for a Hierarchical Structure

```
   Input  : Belief about Global and Local States and Covariances at Time 0
   Output: Belief about Global and Local States and Covariances at Time $t$
 1 for ( $i = 0;\ i < t;\ i++$ ) do
       // Predict Step
 2     Predict Global State
 3     for All Local States do
 4         Predict Local State
 5     end
       // Update Step
 6     Update one Local State
 7     Update Global State
 8     for All Local States except from the one already updated do
 9         Update Local State
10     end
11 end
```

the corresponding local motion model (Line 4).The process noise for each case can be determined by analysing the vehicle properties.

In Line 6, one of the submaps, in which a sensor measurement of a relative point feature is available, is chosen to be updated. In our previous work [2] where a standard one-level Kalman Filter was used, we always decided to observe the feature whose measurement was most uncertainly predicted. Observing this one, provided us with more information about the feature and the vehicle position than measuring a landmark whose observation is very certainly predicted. To apply this concept to the two-level structure, we will propose two methods.

The first one is to predict all measurements with respect to the local vehicle and to chose to update the submap which contains the most uncertain one. Using this heuristic means to disregard the uncertainty about the corresponding global anchor position. No matter how certain the global position of the submap is, if it contains the most uncertain locally predicted point feature it will be updated.

The second method instead predicts the measurement with respect to the global vehicle instead of referred to the local robot. In order to do that the globally positioned anchors have to be included. The uncertainty about the predicted measurement will then be increased by the global uncertainty about the referred anchor. Submaps which are locally quite certain but globally unsettled are favoured over submaps which might be certain on the global level but not that confident locally.

We have not yet evaluated which method yields the best results. In our opinion, the second technique is more convenient from the conceptual point of view. The submaps are features for the global filter. Predicting a global measurement which will be made of a submap means to predict it with respect to the global vehicle and dependent on the anchor. The most uncertain global feature or submap, respectively, is chosen to be measured.

After updating one local state, the improvements made have to be propagated to the global level (Line 7). As shown in Equation (4.7), the global robot state always has to be transformable into all the local vehicle states. Since one of the local states is altered, the global state has to be changed, too, such that the constraint holds again. The idea is to let the local vehicle of the updated submap be the measurement for the global state. This measurement can be predicted by applying the transformation matrix T_{lg} to the global state and covariance. T_{lg} is then referred to as the global measurement model.

When the global state is updated, the Constraint 4.7 will not hold anymore. Thus, all the local submaps not yet considered have to be also updated. This is done in Line 9. We apply the same idea as already done for the global case. The measurement for each local state is the according local vehicle obtained by applying T_{lg} to the updated global vehicle state and covariance. The predicted measurement is the predicted local robot state.

In the case of measuring a landmark with a sensor, simply the noise model of the sensor applies. Considering the update the global state and the remaining submaps, we have a problem when trying to determine the according measurement noise covariance. We do not have a physical sensor whose noise attributes can found by experimental evaluation. Instead the measurement device in each of the mentioned cases is either a local or a global filter.

Two attempts to solve that problem are discussed in the following.

Uncertainty about Estimates as Measurement Noise In the case of the global measurement, the local state updated firstly is the "noisy" measurement device. The global measurement noise describes the possible error e_g between the real measurement $\mathsf{z}_{g,real}$ and the predicted measurement z_g.

$$\begin{aligned} \epsilon_g &= \mathsf{z}_{g,real} - \mathsf{z}_g \\ \mathsf{R}_g &= E[\ \epsilon_g \epsilon_g^\top] \end{aligned}$$

The global measurement is the local vehicle state x_l. The measurement error is then

$$\epsilon_g = \mathsf{x}_{l,real} - \mathsf{x}_l \tag{4.9}$$

Equation (4.9) is according to Welch and Bishop [44] the definition of the error in a filter estimate. In this case we regard to the local filter. Thus, our first attempt to approach the problem of determining the global measurement noise, is to set it equal to the local covariance of the vehicle ζ.

$$\mathsf{R}_g = E[\ \epsilon_g \epsilon_g^\top] = \zeta$$

Equivalently, we approach the measurement noise for updating all the submaps yet not considered in the current filter cycle. The local measurement in these cases is the local vehicle state as transformed from the global state. The measurement error ϵ_l is defined equivalently to ϵ_g as shown in Equation (4.9). The local measurement noise covariance R_l is equal to the uncertainty about the global robot estimate Σ transformed into the currently considered local frame as shown in Equation (4.8).

$$\mathsf{R}_l = E[\ \epsilon_l \epsilon_l^\top] = \mathsf{T}_{lg} \Sigma \mathsf{T}_{lg}^\top$$

Implementation We implemented this solution based on Algorithm 3 for the three-dimensional case where the robot is equipped with a stereo camera sensor. The according process and measurement model are presented in our previous work [2].

When testing the implementation, it turned out that the global and local states are diverging very fast. Our aim was to compensate the correlations which are neglected in the hierarchical representation by propagating information between the levels. Suppose that the global state is updated after a local map has been updated. Since the global measurement model is a function of the global vehicle state and the related global anchor, a global update will mostly affect these directly related parameters. Due to their correlation with the other anchors not related to the updated local state, the position for these anchors might be also updated. However, this indirect update is not capable of compensating the missing direct correlation between the submap and the unrelated anchors. The information passed is too “weak” to balance this loss.

Explicitly Involving the Transformation Constraint Tobias Pietzsch came up with the second idea how to approach this problem. He proposed to explicitly demand Equation (4.7) and (4.8) to hold before and after the predict and update steps. If this would be the case, Condition (4.7) and (4.8) will be by induction always fulfilled.

The derivation of this approach for the linear case is given in Appendix A. In the following we will summarise the main ideas and results.

As already done in the first attempt only the information duplicated across levels is propagated between them. In this case it is the robot state which occurs globally and in each local state.

By forcing Equation (4.7) and (4.8) to hold before and after the predict step on each level, constraints on the process model and the process noise covariance are imposed. Suppose that A_g is the global and A_l the local process model. The following relation has to be fulfilled:

$$\mathsf{A}_l \mathsf{T}_{lg} = \mathsf{T}_{lg} \mathsf{A}_g$$

In Appendix B it is shown that this holds for the models applied to the one-dimensional case. For more obscure cases the local process model can be explicitly calculated if $\mathsf{T}_{lg}\mathsf{T}_{lg}^\top$ is invertible.

$$\mathsf{A}_l = \mathsf{T}_{lg} \mathsf{A}_g \mathsf{T}_{lg}^\top (\mathsf{T}_{lg} \mathsf{T}_{lg}^\top)^{-1}$$

For the local and global process noise covariances the following relation has to be satisfied:

$$\mathsf{R}_l = \mathsf{T}_{lg} \mathsf{R}_g \mathsf{T}_{lg}^\top$$

In Appendix B this is proven to hold for the one-dimensional case.

After the update of the global state, Equation (4.7) and (4.8) should also hold. If one of the submaps is updated based on a feature measurement, the improvement has to be propagated to the global level. As in the previous approach, the global measurement model will be equal to the transformation T_{lg} in order to treat the local vehicle state as the global measurement. Fixing the measurement model determines the global measurement noise and innovation. They are not connected to a physical measurement device, e.g., a camera. Therefore, we cannot determine their noise attributes experimentally. In our case, a local filter is the global measurement device. Its parameters like the measurement noise covariance and innovation can be explicitly calculated from known filter parameters.

Equation (4.7) and (4.8) also have to hold between the global state and all the submaps not yet updated. As already mentioned, the global updated state, transformed by applying T_{lg} into the currently considered local state, will be the local measurement. The predicted uncertainty about the local vehicle can be calculated by transforming the global updated vehicle uncertainty into the considered local level. The local measurement model for submaps not updated by a vehicle measurement will return the predicted local vehicle state as predicted measurement.

Considering these relations, the local measurement noise covariance and innovation are also fixed. All parameters used to explicitly calculate them are known.

To summarise: By explicitly forcing Equation (4.7) and (4.8) to hold before and after each predict and update step and by fixing the measurement models, all remaining parameters of the filter are fixed. They have to be calculated explicitly.

Implementation We implemented this second solution based on the Algorithm 3 for the simple one-dimensional case. The models are given in Appendix B.

It turned out, that the global measurement noise becomes negative right in the first filter cycle. This is a pathological case and should never occur in a working Kalman Filter based system. Assuming that the equations used in the derivation presented in Appendix A are correct, the result of having a negative measurement noise indicates that only propagating the vehicle information between the levels is not sufficient to handle the missing correlations between submaps and unrelated anchors.

Instead of fixing the error that occurred in the first attempt, this method rather clarified the common weakness of both approaches.

Discussion We come to the conclusion that duplicating the vehicle across levels is not enough. The correlations between anchors and unrelated submaps are strong. Their absence within currently considered submaps cannot be compensated.

Including the local representation of the anchors in each submap would be quite inefficient. Suppose that we have a map with 5 submaps. Inserting each anchor in each submap will result in 20 additional entities that have to be maintained by the filter. If there are n submaps, we have $n \times n - 1$ additional values. This is not compatible with our goal of developing an efficient Kalman Filter based solution.

A more promising idea is to include the measured quantity of the considered submap into the global state. This is already used to simplify the derivation of the second approach presented in Appendix A. However, it is not really applied!

To follow up these ideas lies beyond the scope of this work and will be future work.

Chapter 5

Conclusions

In several research works EKF based SLAM has been proven to be robust and powerful. However, it suffers from one major drawback: its poor scalability. In this work, we tried to develop a submapping based approach that reduces the quadratic time and space complexity of the filter and maintains its accuracy as much as possible.

As an instance of an existing algorithm based on submaps we evaluated the Compressed Kalman Filter by Guivant and Nebot [16, 18] which comes in an optimal and suboptimal version. By applying it to a one-dimensional estimation problem within a simulated environment, we tried to analyse its advantages and disadvantages. A clear advantage is the usage of the relative landmark representation which ensures that certain cross correlations can be neglected from the whole covariance matrix. Based on the range of the applied sensor, the CKF facilitates an additional division of the whole environment into regions. Under the assumption that region transitions are occurring rarely, the suboptimal CKF will result in close to optimal results. In our experiments we tried to demonstrate that it behaves much worse when this assumption does not hold. Unfortunately, the restrictions we made in the experiments where too strong such that the decreasing quality of the filter estimates is hardly recognisable.

However, our long term goal is to facilitate a stereo camera as a sensor. A rigid region division as done by the CKF is due to the hardly determinable range of this sensor not convenient. Our basic idea is to use the relative landmark representation but to avoid the division into regions. Instead we introduce semantics to determine which point features can be subsumed to a submap. In our case, this can be points all situated on the same surface.

The state representation we proposed is hierarchical. On the top level we have a number of surfaces. They are each represented by an anchor. Related to each surface is a local state containing a number or relative features. This hierarchy imposes a number of implications for the related covariance matrices. Compared to the CKF, basically two strong groups of cross correlations are missed out: vehicle – submap correlation, and the correlation between anchor and unrelated submaps.

The first one is approximated by duplicating the vehicle across levels. We assumed that the loss of the second group would also be compensated by that. Two approaches to attempt that problem were presented. Each of them uses the duplicated vehicle to propagate information between the levels. It turned out that in each case the duplicated vehicle is not enough to avoid a failure in the estimation result.

Nevertheless, we believe that the chosen representation is desirable when considering a semantic division of the environment and approaching the tracking of more complex features than simple points.

Our future work will therefore be to develop this idea to a stage such that it does not result in a diverging map or even in failure. An idea to solve that is to involve the entity locally measured by the sensor in the global state estimation. First attempts to do so led to promising results.

After solving this problem, the two other tasks mentioned in the introduction can be approached. A semantic segmentation technique of the environment has to be found to divide the point feature among submaps. Eventually, the underlying structure of a higher level feature might not be improvable anymore. Then, we can stop refining the low level features and continue with tracking only the high level one.

Appendix A

Derivation of a Hierarchical Kalman Filter

The first part of this derivation as well as the extension proof in Appendix C is written by Tobias Pietzsch. Our contribution is Section A.3.5 and A.4.

A.1 Introduction

The vehicle state is represented both on the global and the local level, to exploit vehicle–feature correlations on both levels.

Each local submap is anchored on a global feature. The relevant part of the global state is

$$\mathsf{x}_{r;g} = \begin{pmatrix} \mathsf{x}_{v;g} \\ \mathsf{a}_{l;g} \end{pmatrix}$$

where $\mathsf{x}_{v;g}$ is the vehicle state and $\mathsf{a}_{l;g}$ is the anchor of submap l. The local state of the submap l does not contain an explicit representation of the anchor, rather the anchor implicitly fixes the origin of the local coordinate frame. The relevant part of the local state is

$$\mathsf{x}_{r;l} = \mathsf{x}_{v;l}$$

e.g., the local vehicle state.

Global and local states are linked by Condition (4.7), which is repeated here

$$\mathsf{x}_{r;l} = \mathsf{T}_l \mathsf{x}_{r;g}. \tag{A.1}$$

If state estimates are represented as Normal distributions $\mathcal{N}(\mu_g, \Sigma_g)$, $\mathcal{N}(\mu_l, \Sigma_l)$, this implies

$$\mu_{r;l} = \mathsf{T}_l \mu_{r;g}$$
$$\Sigma_{r;l} = \mathsf{T}_l \Sigma_{r;g} \mathsf{T}_l^\top.$$

In the following, we design the predict and update steps in such that if Equation (A.1) holds before the step it provably still holds afterwards. It follows by induction that the condition is always fulfilled then. The general idea is that we know which information is duplicated across levels, and only this information may be propagated between levels. This means that large parts of the process and measurement models are fixed. The remaining free parameters can be found using condition (A.1). In order to not complicate things further, time indices are omitted from the state variables.

A.2 The Predict Step

The estimate of the global state is

$$\mu_g = \begin{pmatrix} \mathsf{x}_{v;g} \\ \mathsf{a}_{l;g} \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathsf{x}_{r;g} \\ \vdots \end{pmatrix}$$

with covariance

$$\Sigma_g = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xa} & \cdots \\ \Sigma_{ax} & \Sigma_{aa} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} \Sigma_{rr} & \cdots \\ \vdots & \ddots \end{bmatrix}$$

The estimate of the local state is

$$\mu_l = \begin{pmatrix} \mathsf{x}_{v;l} \\ \vdots \end{pmatrix} \tag{A.2}$$

Associated covariance is

$$\zeta_l = \begin{bmatrix} \zeta_{xx} & \cdots \\ \vdots & \ddots \end{bmatrix}$$

Supposing that Equation (A.1) holds, for the local state estimate we have

$$\begin{aligned} \mathsf{x}_{v;l} &= \mathsf{T}_l \mathsf{x}_{r;g} \\ \zeta_{xx} &= \mathsf{T}_l \Sigma_{rr} \mathsf{T}_l^\top \end{aligned}$$

A.2.1 Global Prediction

The evolution of global state is characterized as function A_g, perturbed by zero-mean Gaussian process noise, i.e,

$$\overline{\mu}_g = \mathsf{A}_g \mu_g \tag{A.3}$$

$$\overline{\Sigma}_g = \mathsf{A}_g \Sigma_g \mathsf{A}_g^\top + \mathsf{R}_g \tag{A.4}$$

Assuming that all parts of the state other than the vehicle are static, we have

$$\mathsf{A}_g = \begin{bmatrix} \mathsf{A}_{rg} & 0 & \dots \\ 0 & \mathsf{I} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$
$$\mathsf{R}_g = \begin{bmatrix} \mathsf{R}_{rg} & 0 & \dots \\ 0 & 0 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

with

$$\mathsf{A}_{rg} = \begin{bmatrix} \mathsf{A}_{xg} & 0 \\ 0 & \mathsf{I} \end{bmatrix}$$
$$\mathsf{R}_{rg} = \begin{bmatrix} \mathsf{R}_{xg} & 0 \\ 0 & 0 \end{bmatrix}$$

Performing the prediction (Equations (A.3) and (A.4)) yields

$$\overline{\mu}_g = \begin{pmatrix} \overline{\mathsf{x}}_{r;g} \\ \vdots \end{pmatrix}$$
$$\overline{\Sigma}_g = \begin{bmatrix} \overline{\Sigma}_{rr} & \dots \\ \vdots & \ddots \end{bmatrix}$$

with

$$\overline{\mathsf{x}}_{r;g} = \mathsf{A}_{rg}\mathsf{x}_{r;g}$$
$$\overline{\Sigma}_{rr} = \mathsf{A}_{rg}\Sigma_{rr}\mathsf{A}_{rg}^{\top} + \mathsf{R}_{rg}$$

A.2.2 Local Prediction

The evolution of local state is characterized as function A_l, perturbed by zero-mean Gaussian process noise, i.e,

$$\overline{\mu}_l = \mathsf{A}_l\mu_l \tag{A.5}$$
$$\overline{\zeta}_l = \mathsf{A}_l\zeta_l\mathsf{A}_l^{\top} + \mathsf{R}_l \tag{A.6}$$

where R_l is the process noise covariance. Assuming that all parts of the state other than the vehicle are static, we have

$$\mathsf{A}_l = \begin{bmatrix} \mathsf{A}_{xl} & 0 & \dots \\ 0 & \mathsf{I} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$
$$\mathsf{R}_l = \begin{bmatrix} \mathsf{R}_{xl} & 0 & \dots \\ 0 & 0 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Performing the prediction (Equations (A.5) and (A.6)) yields the predicted state

$$\begin{aligned}\overline{\mu}_l &= \begin{pmatrix}\overline{\mathsf{x}}_{v;l}\\ \vdots\end{pmatrix}\\ \overline{\zeta}_l &= \begin{bmatrix}\overline{\zeta}_{xx} & \cdots\\ \vdots & \ddots\end{bmatrix}\end{aligned}$$

with

$$\begin{aligned}\overline{\mathsf{x}}_{v;l} &= \mathsf{A}_{xl}\mathsf{x}_{v;l}\\ \overline{\zeta}_{xx} &= \mathsf{A}_{xl}\zeta_{xx}\mathsf{A}_{xl}^{\top} + \mathsf{R}_{xl}\end{aligned}$$

Of course, we want that if the condition (A.1) was satisfied before the prediction, it is also satisfied afterwards, i.e.,

$$\begin{aligned}\overline{\mathsf{x}}_{v;l} &= \mathsf{T}_l\overline{\mathsf{x}}_{r;g}\\ \overline{\zeta}_{xx} &= \mathsf{T}_l\overline{\Sigma}_{rr}\mathsf{T}_l^{\top}\end{aligned}$$

This imposes some constraints on the process model. Lets look at the condition for the mean of the predicted state first

$$\begin{aligned}\overline{\mathsf{x}}_{v;l} &= \mathsf{T}_l\overline{\mathsf{x}}_{r;g}\\ \mathsf{A}_{xl}\mathsf{x}_{v;l} &= \mathsf{T}_l\mathsf{A}_{rg}\mathsf{x}_{r;g}\\ \mathsf{A}_{xl}\mathsf{T}_l\mathsf{x}_{r;g} &= \mathsf{T}_l\mathsf{A}_{rg}\mathsf{x}_{r;g}\end{aligned}$$

Obviously, this is fulfilled, if

$$\mathsf{A}_{xl}\mathsf{T}_l = \mathsf{T}_l\mathsf{A}_{rg}. \tag{A.7}$$

Considering the simple one-dimensional case, this condition is satisfied as proven in Appendix B.

For more obscure cases, the local process model can be explicitly calculated, if $\mathsf{T}_l\mathsf{T}_l^{\top}$ is invertible:

$$\begin{aligned}\mathsf{A}_{xl}\mathsf{T}_l &= \mathsf{T}_l\mathsf{A}_{rg}\\ \mathsf{A}_{xl}\mathsf{T}_l\mathsf{T}_l^{\top} &= \mathsf{T}_l\mathsf{A}_{rg}\mathsf{T}_l^{\top}\\ \mathsf{A}_{xl}\mathsf{T}_l\mathsf{T}_l^{\top}(\mathsf{T}_l\mathsf{T}_l^{\top})^{-1} &= \mathsf{T}_l\mathsf{A}_{rg}\mathsf{T}_l^{\top}(\mathsf{T}_l\mathsf{T}_l^{\top})^{-1}\\ \mathsf{A}_{xl} &= \mathsf{T}_l\mathsf{A}_{rg}\mathsf{T}_l^{\top}(\mathsf{T}_l\mathsf{T}_l^{\top})^{-1}\end{aligned}$$

Lets look at the condition for the prediction covariance now

$$\begin{aligned}\overline{\zeta}_{xx} &= \mathsf{T}_l\overline{\Sigma}_{rr}\mathsf{T}_l^{\top}\\ \mathsf{A}_{xl}\zeta_{xx}\mathsf{A}_{xl}^{\top} + \mathsf{R}_{xl} &= \mathsf{T}_l\left(\mathsf{A}_{rg}\Sigma_{rr}\mathsf{A}_{rg}^{\top} + \mathsf{R}_{rg}\right)\mathsf{T}_l^{\top}\\ \mathsf{A}_{xl}\mathsf{T}_l\Sigma_{rr}\mathsf{T}_l^{\top}\mathsf{A}_{xl}^{\top} + \mathsf{R}_{xl} &= \mathsf{T}_l\mathsf{A}_{rg}\Sigma_{rr}\mathsf{A}_{rg}^{\top}\mathsf{T}_l^{\top} + \mathsf{T}_l\mathsf{R}_{rg}\mathsf{T}_l^{\top}\end{aligned}$$

Given that Equation (A.7) holds, this translates to the condition

$$\mathsf{R}_{xl} = \mathsf{T}_l\mathsf{R}_{rg}\mathsf{T}_l^{\top} \tag{A.8}$$

This is, e.g., fulfilled for the simple one-dimensional case.

A.3 The Update Step

The predicted global state is

$$\bar{\mu}_g = \begin{pmatrix} \bar{\mathsf{x}}_{v;g} \\ \bar{\mathsf{a}}_{l;g} \\ \vdots \end{pmatrix}$$

with covariance

$$\bar{\Sigma}_g = \begin{bmatrix} \bar{\Sigma}_{xx} & \bar{\Sigma}_{xa} & \cdots \\ \bar{\Sigma}_{ax} & \bar{\Sigma}_{aa} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Predicted local state is

$$\bar{\mu}_l = \begin{pmatrix} \bar{\mathsf{x}}_{v;l} \\ \bar{\mathsf{y}}_i \\ \vdots \end{pmatrix} \tag{A.9}$$

where y_i is the feature which will be measured. Associated covariance is

$$\bar{\zeta}_l = \begin{bmatrix} \bar{\zeta}_{xx} & \bar{\zeta}_{xy} & \cdots \\ \bar{\zeta}_{yx} & \bar{\zeta}_{yy} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Let the transformation between levels be defined as

$$\mathsf{T}_l = \begin{bmatrix} \mathsf{T}_x & \mathsf{T}_a & \ldots \end{bmatrix} \tag{A.10}$$

Supposing that Equation (A.1) holds, for the predicted local state we have

$$\begin{aligned} \bar{\mathsf{x}}_{v;l} &= \mathsf{T}_x \bar{\mathsf{x}}_{v;g} + \mathsf{T}_a \bar{\mathsf{a}}_{l;g} \\ \bar{\zeta}_{xx} &= \mathsf{T}_x \bar{\Sigma}_{xx} \mathsf{T}_x^\top + \mathsf{T}_a \bar{\Sigma}_{ax} \mathsf{T}_x^\top + \mathsf{T}_x \bar{\Sigma}_{xa} \mathsf{T}_a^\top + \mathsf{T}_a \bar{\Sigma}_{aa} \mathsf{T}_a^\top \end{aligned}$$

A.3.1 Local Update

We make a measurement of feature y_i. The measurement is a function of local vehicle $\mathsf{x}_{v;l}$ and feature y_i:

$$\mathsf{z}_l = \mathsf{C}_l \mu_l = \begin{bmatrix} \mathsf{C}_{xl} & \mathsf{C}_{yl} & 0 & \ldots \end{bmatrix} \begin{pmatrix} \mathsf{x}_{v;l} \\ \mathsf{y}_i \\ \vdots \end{pmatrix}$$

The equation for the local innovation covariance is

$$\mathsf{S}_l = \mathsf{C}_l \bar{\zeta}_l \mathsf{C}_l^\top + \mathsf{Q}_l$$

with measurement noise covariance Q_l. Multiplying out yields

$$\mathsf{S}_l = \mathsf{C}_{xl}\bar{\zeta}_{xx}\mathsf{C}_{xl}^\top + \mathsf{C}_{yl}\bar{\zeta}_{yx}\mathsf{C}_{xl}^\top + \mathsf{C}_{xl}\bar{\zeta}_{xy}\mathsf{C}_{yl}^\top + \mathsf{C}_{yl}\bar{\zeta}_{yy}\mathsf{C}_{yl}^\top + \mathsf{Q}_l$$

The local innovation will be denoted by ν_l.

Relevant for the connection to the global level is the $\mathsf{x}_{v;l}$ component of the updated state and its covariance ζ_{xx}. Multiplying out the update equations we get

$$\mathsf{x}_{v;l} = \bar{\mathsf{x}}_{v;l} + \bar{\zeta}_{xx}\mathsf{m}_l + \bar{\zeta}_{xy}\mathsf{n}_l$$

$$\zeta_{xx} = \bar{\zeta}_{xx} - \left(\bar{\zeta}_{xx}\mathsf{M}_l\bar{\zeta}_{xx} + \bar{\zeta}_{xx}\mathsf{N}_l\bar{\zeta}_{yx} + \bar{\zeta}_{xy}\mathsf{L}_l\bar{\zeta}_{xx} + \bar{\zeta}_{xy}\mathsf{O}_l\bar{\zeta}_{yx}\right)$$

using a set of abbreviations similar to those defined in [26].

$$\begin{aligned}
\mathsf{m}_l &= \mathsf{C}_{xl}^\top\mathsf{S}_l^{-1}\nu_l \\
\mathsf{n}_l &= \mathsf{C}_{yl}^\top\mathsf{S}_l^{-1}\nu_l \\
\mathsf{M}_l &= \mathsf{C}_{xl}^\top\mathsf{S}_l^{-1}\mathsf{C}_{xl} \\
\mathsf{N}_l &= \mathsf{C}_{xl}^\top\mathsf{S}_l^{-1}\mathsf{C}_{yl} \\
\mathsf{L}_l &= \mathsf{C}_{yl}^\top\mathsf{S}_l^{-1}\mathsf{C}_{xl} = \mathsf{N}_l^\top \\
\mathsf{O}_l &= \mathsf{C}_{yl}^\top\mathsf{S}_l^{-1}\mathsf{C}_{yl}
\end{aligned}$$

A.3.2 Global Update

The task is now to propagate the information we gained at the local level to the global level. The local feature y_i is not part of the global map, and thus cannot be used as a global measurement directly. However, by locally measuring y_i we have learned something about the local robot state, which is defined (see Equations (A.1) and (A.10)) as a function of $\mathsf{x}_{v;g}$ and $\mathsf{a}_{l;g}$ Globally, this gain of information can be expressed as making a measurement of landmark $\mathsf{a}_{l;g}$.

The measurement is a function of global vehicle $\mathsf{x}_{v;g}$ and feature $\mathsf{a}_{l;g}$:

$$\mathsf{z}_g = \mathsf{C}_g\mu_g = \begin{bmatrix}\mathsf{C}_{xg} & \mathsf{C}_{kg} & \mathsf{0} & \ldots\end{bmatrix}\begin{pmatrix}\mathsf{x}_{v;g} \\ \mathsf{a}_{l;g} \\ \vdots\end{pmatrix} \tag{A.11}$$

The equation for the global innovation covariance is

$$\mathsf{S}_g = \mathsf{C}_g\overline{\Sigma}_g\mathsf{C}_g^\top + \mathsf{Q}_g$$

with measurement noise covariance Q_g. Multiplying out yields

$$\mathsf{S}_g = \mathsf{C}_{xg}\overline{\Sigma}_{xx}\mathsf{C}_{xg}^\top + \mathsf{C}_{kg}\overline{\Sigma}_{ax}\mathsf{C}_{xg}^\top + \mathsf{C}_{xg}\overline{\Sigma}_{xa}\mathsf{C}_{kg}^\top + \mathsf{C}_{kg}\overline{\Sigma}_{aa}\mathsf{C}_{kg}^\top + \mathsf{Q}_g \tag{A.12}$$

The global innovation will be denoted by ν_g.

The following components of the updated state are of interest for the interlevel relation:

$$\mathsf{x}_{v;g} = \bar{\mathsf{x}}_{v;g} + \overline{\Sigma}_{xx}\mathsf{m}_g + \overline{\Sigma}_{xa}\mathsf{n}_g \tag{A.13}$$

$$\mathsf{a}_{l;g} = \bar{\mathsf{a}}_{l;g} + \overline{\Sigma}_{ax}\mathsf{m}_g + \overline{\Sigma}_{aa}\mathsf{n}_g \tag{A.14}$$

$$\Sigma_{xx} = \overline{\Sigma}_{xx} - \left(\overline{\Sigma}_{xx}\mathsf{M}_g\overline{\Sigma}_{xx} + \overline{\Sigma}_{xx}\mathsf{N}_g\overline{\Sigma}_{ax} + \overline{\Sigma}_{xa}\mathsf{L}_g\overline{\Sigma}_{xx} + \overline{\Sigma}_{xa}\mathsf{O}_g\overline{\Sigma}_{ax}\right) \tag{A.15}$$

$$\Sigma_{ax}^{\top} = \Sigma_{xa} = \overline{\Sigma}_{xa} - \left(\overline{\Sigma}_{xx}\mathsf{M}_g\overline{\Sigma}_{xa} + \overline{\Sigma}_{xx}\mathsf{N}_g\overline{\Sigma}_{aa} + \overline{\Sigma}_{xa}\mathsf{L}_g\overline{\Sigma}_{xa} + \overline{\Sigma}_{xa}\mathsf{O}_g\overline{\Sigma}_{aa}\right) \tag{A.16}$$

$$\Sigma_{aa} = \overline{\Sigma}_{aa} - \left(\overline{\Sigma}_{ax}\mathsf{M}_g\overline{\Sigma}_{xa} + \overline{\Sigma}_{ax}\mathsf{N}_g\overline{\Sigma}_{aa} + \overline{\Sigma}_{aa}\mathsf{L}_g\overline{\Sigma}_{xa} + \overline{\Sigma}_{aa}\mathsf{O}_g\overline{\Sigma}_{aa}\right) \tag{A.17}$$

using the following set of abbreviations:

$$\begin{aligned}
\mathsf{m}_g &= \mathsf{C}_{xg}^{\top}\mathsf{S}_g^{-1}\nu_g \\
\mathsf{n}_g &= \mathsf{C}_{kg}^{\top}\mathsf{S}_g^{-1}\nu_g \\
\mathsf{M}_g &= \mathsf{C}_{xg}^{\top}\mathsf{S}_g^{-1}\mathsf{C}_{xg} \\
\mathsf{N}_g &= \mathsf{C}_{xg}^{\top}\mathsf{S}_g^{-1}\mathsf{C}_{kg} \\
\mathsf{L}_g &= \mathsf{C}_{kg}^{\top}\mathsf{S}_g^{-1}\mathsf{C}_{xg} = \mathsf{N}_g^{\top} \\
\mathsf{O}_g &= \mathsf{C}_{kg}^{\top}\mathsf{S}_g^{-1}\mathsf{C}_{kg}
\end{aligned}$$

Note that most expressions in the above derivation are fixed. The components which we can influence are the innovation ν_g (via the measurement z_g) and the measurement noise covariance Q_g. We also can choose the C_{xg} and C_{kg} parts of the measurement model (Equation (A.11)).

Of course, we want to choose this expressions such that Equation (A.1) holds, for the updated global and local state, i.e., we want

$$\begin{aligned}
\mathsf{x}_{v;l} &= \mathsf{T}_x\mathsf{x}_{v;g} + \mathsf{T}_a\mathsf{a}_{l;g} \\
\zeta_{xx} &= \mathsf{T}_x\Sigma_{xx}\mathsf{T}_x^{\top} + \mathsf{T}_a\Sigma_{ax}\mathsf{T}_x^{\top} + \mathsf{T}_x\Sigma_{xa}\mathsf{T}_a^{\top} + \mathsf{T}_a\Sigma_{aa}\mathsf{T}_a^{\top}
\end{aligned}$$

to hold.

Unfortunately, the information about how to choose the variable expressions is not really obvious from Equations (A.13)–(A.17). The situation is further complicated by the fact, that these equations use the *inverse* of the innovation covariance, which is an involved expression in itself (Equation (A.12)).

However, as the next section will show, we can use a trick to simplify things. We will explicitly embed the measured quantity into the global state. This embedding is purely for the purpose of deriving the correct expressions for ν_g and Q_g. It does not need to be performed directly.

Before we start, lets fix the free components of the measurement model. The most obvious choice is to measure the local state directly and thus we choose

$$\begin{aligned}
\mathsf{C}_{xg} &= \mathsf{T}_x \\
\mathsf{C}_{kg} &= \mathsf{T}_a
\end{aligned}$$

Note, that the innovation is involved nowhere in the computation of the updated covariances. Thus we can first independently find the correct measurement noise covariance Q_g, and turn to the innovation ν_g afterwards. This will be done in the next two sections.

A.3.3 Global Update: Covariance

As stated in the previous section, we proceed by incorporating the measured quantity (the local state) into the global state, in order to simplify the update equations.

This is done using the global measurement model C_g. Lets denote the "global copy" of the measurement as l_g. (With the measurement model proposed in the previous section, this means that l_g is a copy of the local vehicle state $\mathsf{x}_{v;l}$). The transformation E that takes the (predicted) global state to the *extended* global state x_e is the identity matrix, augmented with C_g for the added vehicle copy:

$$\overline{\mathsf{x}}_e = \begin{pmatrix} \overline{\mathsf{x}}_{v;g} \\ \overline{\mathsf{a}}_{l;g} \\ \vdots \\ \overline{\mathsf{l}}_g \end{pmatrix} = \begin{bmatrix} \mathsf{I} & 0 & 0 & \dots \\ 0 & \mathsf{I} & 0 & \dots \\ 0 & 0 & \mathsf{I} & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \mathsf{C}_{xg} & \mathsf{C}_{kg} & 0 & \dots \end{bmatrix} \begin{pmatrix} \overline{\mathsf{x}}_{v;g} \\ \overline{\mathsf{a}}_{l;g} \\ \vdots \end{pmatrix} = \mathsf{E}\overline{\mu}_g$$

Because the measured quantity is part of the extended state, it can be measured directly, and the measurement model is very simple:

$$\mathsf{z} = \mathsf{l}_g = \mathsf{C}_e \mathsf{x}_e$$

with

$$\mathsf{C}_e = \begin{bmatrix} 0 & \dots & 0 & \mathsf{I} \end{bmatrix}.$$

The extended (predicted) covariance is

$$\overline{\Sigma}_e = \begin{bmatrix} \overline{\Sigma}_{xx} & \overline{\Sigma}_{xa} & \dots & \overline{\Sigma}_{xl} \\ \overline{\Sigma}_{ax} & \overline{\Sigma}_{aa} & \dots & \overline{\Sigma}_{al} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\Sigma}_{lx} & \overline{\Sigma}_{la} & \dots & \overline{\Sigma}_{ll} \end{bmatrix} = \mathsf{E}\overline{\Sigma}_g \mathsf{E}^\top$$

Explicitly performing this multiplication for $\overline{\Sigma}_{ll}$ yields

$$\overline{\Sigma}_{ll} = \mathsf{C}_{xg}\overline{\Sigma}_{xx}\mathsf{C}_{xg}^\top + \mathsf{C}_{kg}\overline{\Sigma}_{ax}\mathsf{C}_{xg}^\top + \mathsf{C}_{xg}\overline{\Sigma}_{xa}\mathsf{C}_{kg}^\top + \mathsf{C}_{kg}\overline{\Sigma}_{aa}\mathsf{C}_{kg}^\top$$

Remember, that we defined C_g such that $\mathsf{C}_{xg} = \mathsf{T}_x$ and $\mathsf{C}_{kg} = \mathsf{T}_a$. Then, from the last equation obviously we have

$$\overline{\Sigma}_{ll} = \mathsf{T}_x\overline{\Sigma}_{xx}\mathsf{T}_x^\top + \mathsf{T}_a\overline{\Sigma}_{ax}\mathsf{T}_x^\top + \mathsf{T}_x\overline{\Sigma}_{xa}\mathsf{T}_a^\top + \mathsf{T}_a\overline{\Sigma}_{aa}\mathsf{T}_a^\top = \overline{\zeta}_{xx}. \tag{A.18}$$

To satisfy the condition of Equation (A.1), this, should still hold after the update, i.e., we need to have

$$\Sigma_{ll} = \zeta_{xx}.$$

In the rest of this section we will derive an expression for the measurement noise Q_g, such that this equality holds. This is done, assuming that the *extended* state is updated using the measurement model C_e. In Appendix C, we proof that equivalently we can update the *original* state with the *original* measurement model and extend it afterwards – the resulting covariances will be identical. Thus, if we use the measurement noise Q_g (as derived below) in the update of the *original* state, condition (A.1) will be fullfilled.

Using the measurement model C_e, the innovation covariance is

$$\mathsf{S}_e = \mathsf{C}_e \overline{\Sigma}_e \mathsf{C}_e^\top + \mathsf{Q}_g = \overline{\Sigma}_{ll} + \mathsf{Q}_g$$

For the Kalman gain we get

$$\mathsf{W}_e = \overline{\Sigma}_e \mathsf{C}_e^\top \mathsf{S}_e^{-1} = \begin{bmatrix} \overline{\Sigma}_{xl} \\ \overline{\Sigma}_{al} \\ \vdots \\ \overline{\Sigma}_{ll} \end{bmatrix} \mathsf{S}_e^{-1}$$

The extended covariance is updated as

$$\Sigma_e = \overline{\Sigma}_e - \mathsf{W}_e \mathsf{S}_e \mathsf{W}_e^\top$$

where for the update term we get

$$\mathsf{W}_e \mathsf{S}_e \mathsf{W}_e^\top = \begin{bmatrix} \overline{\Sigma}_{xl} \\ \overline{\Sigma}_{al} \\ \vdots \\ \overline{\Sigma}_{ll} \end{bmatrix} \mathsf{S}_e^{-1} \begin{bmatrix} \overline{\Sigma}_{lx} & \overline{\Sigma}_{la} & \dots & \overline{\Sigma}_{ll} \end{bmatrix}$$

Lets perform the update explicitly for the interesting entry of the extended covariance:

$$\Sigma_{ll} = \overline{\Sigma}_{ll} - \overline{\Sigma}_{ll} \mathsf{S}_e^{-1} \overline{\Sigma}_{ll}$$

With the value of ζ_{xx} already known from the local update, this determines our choice of Q_g:

$$\Sigma_{ll} = \zeta_{xx} \tag{A.19}$$

$$\overline{\Sigma}_{ll} - \overline{\Sigma}_{ll} \mathsf{S}_e^{-1} \overline{\Sigma}_{ll} = \zeta_{xx} \tag{A.20}$$

$$\overline{\Sigma}_{ll} \mathsf{S}_e^{-1} \overline{\Sigma}_{ll} = \overline{\Sigma}_{ll} - \zeta_{xx} \tag{A.21}$$

$$\mathsf{S}_e^{-1} = \overline{\Sigma}_{ll}^{-1} \left(\overline{\Sigma}_{ll} - \zeta_{xx} \right) \overline{\Sigma}_{ll}^{-1} \tag{A.22}$$

$$\mathsf{S}_e = \overline{\Sigma}_{ll} \left(\overline{\Sigma}_{ll} - \zeta_{xx} \right)^{-1} \overline{\Sigma}_{ll} \tag{A.23}$$

$$\overline{\Sigma}_{ll} + \mathsf{Q}_g = \overline{\Sigma}_{ll} \left(\overline{\Sigma}_{ll} - \zeta_{xx} \right)^{-1} \overline{\Sigma}_{ll} \tag{A.24}$$

$$\mathsf{Q}_g = \overline{\Sigma}_{ll} \left(\overline{\Sigma}_{ll} - \zeta_{xx} \right)^{-1} \overline{\Sigma}_{ll} - \overline{\Sigma}_{ll} \tag{A.25}$$

A.3.4 Global Update: Mean

After the update, we should have

$$\begin{aligned}
\mathsf{x}_{v;l} &= \mathsf{T}_x \mathsf{x}_{v;g} + \mathsf{T}_a \mathsf{a}_{l;g} \\
&= \mathsf{T}_x (\bar{\mathsf{x}}_{v;g} + \overline{\Sigma}_{xx} \mathsf{C}_{xg}^\top \mathsf{S}_g^{-1} \nu_g + \overline{\Sigma}_{xa} \mathsf{C}_{kg}^\top \mathsf{S}_g^{-1} \nu_g) \\
&+ \mathsf{T}_a (\bar{\mathsf{a}}_{l;g} + \overline{\Sigma}_{ax} \mathsf{C}_{xg}^\top \mathsf{S}_g^{-1} \nu_g + \overline{\Sigma}_{aa} \mathsf{C}_{kg}^\top \mathsf{S}_g^{-1} \nu_g) \\
&= \mathsf{T}_x \bar{\mathsf{x}}_{v;g} + \mathsf{T}_a \bar{\mathsf{a}}_{l;g} + \\
&\left(\mathsf{T}_x \overline{\Sigma}_{xx} \mathsf{C}_{xg}^\top + \mathsf{T}_x \overline{\Sigma}_{xa} \mathsf{C}_{kg}^\top + \mathsf{T}_a \overline{\Sigma}_{ax} \mathsf{C}_{xg}^\top + \mathsf{T}_a \overline{\Sigma}_{aa} \mathsf{C}_{kg}^\top \right) \mathsf{S}_g^{-1} \nu_g .
\end{aligned}$$

With the value of $\mathsf{x}_{v;l}$ already known from the local update, this determines ν_g. (Remember, that we defined C_g such that $\mathsf{C}_{xg} = \mathsf{T}_x$ and $\mathsf{C}_{kg} = \mathsf{T}_a$).

$$\begin{aligned}
\nu_g &= \mathsf{S}_g \left(\mathsf{T}_x \overline{\Sigma}_{xx} \mathsf{C}_{xg}^\top + \mathsf{T}_x \overline{\Sigma}_{xa} \mathsf{C}_{kg}^\top + \mathsf{T}_a \overline{\Sigma}_{ax} \mathsf{C}_{xg}^\top + \mathsf{T}_a \overline{\Sigma}_{aa} \mathsf{C}_{kg}^\top \right)^{-1} \\
&(\mathsf{x}_{v;l} - \mathsf{T}_x \bar{\mathsf{x}}_{v;g} - \mathsf{T}_a \bar{\mathsf{a}}_{l;g}) \\
&= \mathsf{S}_g \left(\mathsf{T}_x \overline{\Sigma}_{xx} \mathsf{T}_x^\top + \mathsf{T}_x \overline{\Sigma}_{xa} \mathsf{T}_a^\top + \mathsf{T}_a \overline{\Sigma}_{ax} \mathsf{T}_x^\top + \mathsf{T}_a \overline{\Sigma}_{aa} \mathsf{T}_a^\top \right)^{-1} \\
&(\mathsf{x}_{v;l} - \mathsf{T}_x \bar{\mathsf{x}}_{v;g} - \mathsf{T}_a \bar{\mathsf{a}}_{l;g}) \\
&= \mathsf{S}_g \overline{\zeta}_{xx}^{-1} (\mathsf{x}_{v;l} - \mathsf{T}_x \bar{\mathsf{x}}_{v;g} - \mathsf{T}_a \bar{\mathsf{a}}_{l;g})
\end{aligned}$$

A.3.5 Local Update II: Updating the Other Submaps

So far, we assumed that the relevant entries of the global state just consist of the global vehicle position $\mathsf{x}_{v;g}$ and one submap anchor $\mathsf{a}_{l;g}$ subsumed in the vector $\mathsf{x}_{r;g}$. Usually, we will have more than one submap in the global state. To keep the whole map consistent, the information, derived in the local map where a sensor measurement was included, has to be propagated to the remaining submaps. To exploit the updated global robot position we transfer it to each remaining submap and compare it to the local robot states.

Connection Between Levels

In this section we are going to augment this relevant part of the global state with an additional submap anchor. The estimate of the global state is therefore

$$\mu_g = \begin{pmatrix} \mathsf{x}_{v;g} \\ \mathsf{a}_{1;g} \\ \mathsf{a}_{2;g} \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathsf{x}_{r;g} \\ \vdots \end{pmatrix}$$

with covariance

$$\Sigma_g = \begin{bmatrix} \Sigma_{xx} & \Sigma_{xa_1} & \Sigma_{xa_2} & \cdots \\ \Sigma_{a_1x} & \Sigma_{a_1a_1} & \Sigma_{a_1a_2} & \cdots \\ \Sigma_{a_2x} & \Sigma_{a_2a_1} & \Sigma_{a_2a_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} \Sigma_{rr} & \cdots \\ \vdots & \ddots \end{bmatrix}$$

Equivalently to Equation (A.9) the estimates of these local states are

$$\mathsf{x}_{1;l} = \begin{pmatrix} \mathsf{x}_{1;v} \\ \mathsf{y}_i \\ \vdots \end{pmatrix} \qquad \mathsf{x}_{2;l} = \begin{pmatrix} \mathsf{x}_{2;v} \\ \mathsf{y}_j \\ \vdots \end{pmatrix}$$

Associated covariances are

$$\zeta_1 = \begin{bmatrix} \zeta_{xx;1} & \zeta_{xy;1} & \cdots \\ \zeta_{yx;1} & \zeta_{yy;1} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \qquad \zeta_2 = \begin{bmatrix} \zeta_{xx;2} & \zeta_{xy;2} & \cdots \\ \zeta_{yx;2} & \zeta_{yy;2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

The transformation between the levels will be defined as

$$\mathsf{T}_1 = \begin{bmatrix} \mathsf{T}_{x_1} & \mathsf{T}_{a_1} & \mathsf{0} & \ldots \end{bmatrix} \qquad \mathsf{T}_2 = \begin{bmatrix} \mathsf{T}_{x_2} & \mathsf{0} & \mathsf{T}_{a_2} & \ldots \end{bmatrix}$$

Supposing that Equation (A.1) holds, for all local state estimates we have

$$\mathsf{x}_{1;v} = \mathsf{T}_{x_1}\mathsf{x}_{v;g} + \mathsf{T}_{a_1}\mathsf{a}_{1;g} \tag{A.26}$$

$$\zeta_{xx;1} = \mathsf{T}_{x_1}\Sigma_{xx}\mathsf{T}_{x_1}^\top + \mathsf{T}_{a_1}\Sigma_{a_1x}\mathsf{T}_{x_1}^\top + \mathsf{T}_{x_1}\Sigma_{xa_1}\mathsf{T}_{a_1}^\top + \mathsf{T}_{a_1}\Sigma_{a_1a_1}\mathsf{T}_{a_1}^\top \tag{A.27}$$

$$\mathsf{x}_{v;2} = \mathsf{T}_{x_2}\mathsf{x}_{v;g} + \mathsf{T}_{a_2}\mathsf{a}_{2;g} \tag{A.28}$$

$$\zeta_{xx;2} = \mathsf{T}_{x_2}\Sigma_{xx}\mathsf{T}_{x_2}^\top + \mathsf{T}_{a_2}\Sigma_{a_2x}\mathsf{T}_{x_2}^\top + \mathsf{T}_{x_2}\Sigma_{xa_2}\mathsf{T}_{a_2}^\top + \mathsf{T}_{a_2}\Sigma_{a_2a_2}\mathsf{T}_{a_2}^\top \tag{A.29}$$

The Predict Step

We can apply the predict step as shown in Section A.2 to every submap linked to a landmark in the global map. Equation (A.1) will then hold under the given conditions between these local states and the global one.

The predicted global state will be

$$\overline{\mu}_g = \begin{pmatrix} \overline{\mathsf{x}}_{v;g} \\ \overline{\mathsf{a}}_{1;g} \\ \overline{\mathsf{a}}_{2;g} \\ \vdots \end{pmatrix}$$

with covariance

$$\overline{\Sigma}_g = \begin{bmatrix} \overline{\Sigma}_{xx} & \overline{\Sigma}_{xa_1} & \overline{\Sigma}_{xa_2} & \cdots \\ \overline{\Sigma}_{a_1x} & \overline{\Sigma}_{a_1a_1} & \overline{\Sigma}_{a_1a_2} & \cdots \\ \overline{\Sigma}_{a_2x} & \overline{\Sigma}_{a_2a_1} & \overline{\Sigma}_{a_2a_2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

The predicted local states will be

$$\overline{\mathsf{x}}_{1;l} = \begin{pmatrix} \overline{\mathsf{x}}_{1;v} \\ \overline{\mathsf{y}}_i \\ \vdots \end{pmatrix} \qquad \overline{\mathsf{x}}_{2;l} = \begin{pmatrix} \overline{\mathsf{x}}_{2;v} \\ \overline{\mathsf{y}}_j \\ \vdots \end{pmatrix}$$

Associated covariances are

$$\bar{\zeta}_1 = \begin{bmatrix} \bar{\zeta}_{xx;1} & \bar{\zeta}_{xy;1} & \cdots \\ \bar{\zeta}_{yx;1} & \bar{\zeta}_{yy;1} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \qquad \bar{\zeta}_2 = \begin{bmatrix} \bar{\zeta}_{xx;2} & \bar{\zeta}_{xy;2} & \cdots \\ \bar{\zeta}_{yx;2} & \bar{\zeta}_{yy;2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Supposing that Equation (A.1) holds for the predicted local states, we have

$$\begin{aligned} \bar{\mathsf{x}}_{1;v} &= \mathsf{T}_{x_1}\bar{\mathsf{x}}_{v;g} + \mathsf{T}_a\bar{\mathsf{a}}_{1;g} \\ \bar{\zeta}_{xx;1} &= \mathsf{T}_{x_1}\bar{\Sigma}_{xx}\mathsf{T}_{x_1}^\top + \mathsf{T}_{a_1}\bar{\Sigma}_{a_1x}\mathsf{T}_{x_1}^\top + \mathsf{T}_{x_1}\bar{\Sigma}_{xa_1}\mathsf{T}_{a_1}^\top + \mathsf{T}_{a_1}\bar{\Sigma}_{a_1a_1}\mathsf{T}_{a_1}^\top \end{aligned}$$

$$\begin{aligned} \bar{\mathsf{x}}_{2;v} &= \mathsf{T}_{x_1}\bar{\mathsf{x}}_{v;g} + \mathsf{T}_a\bar{\mathsf{a}}_{1;g} \\ \bar{\zeta}_{xx;2} &= \mathsf{T}_{x_2}\bar{\Sigma}_{xx}\mathsf{T}_{x_2}^\top + \mathsf{T}_{a_2}\bar{\Sigma}_{a_2x}\mathsf{T}_{x_2}^\top + \mathsf{T}_{x_2}\bar{\Sigma}_{xa_2}\mathsf{T}_{a_2}^\top + \mathsf{T}_{a_2}\bar{\Sigma}_{a_2a_2}\mathsf{T}_{a_2}^\top \end{aligned}$$

The Update Step

Our goal is that the Equations (A.26)-(A.29) hold again after the update of the global map and all submaps.

We assume that the local update of the first submap and the global update according to the previous sections are already accomplished. Thus, we can state that Equations (A.26) and (A.27) are fulfilled.

For all the remaining and predicted submaps, in our case $\bar{\mathsf{x}}_{2;l}$, we introduce a new measurement model to measure the local robot position directly. We have

$$\mathsf{z}_{2;\mathsf{l}} = \mathsf{C}_{2;l}\bar{\mathsf{x}}_{2;l} = \begin{bmatrix} \mathsf{I} & \mathsf{0} & \ldots \end{bmatrix} \begin{pmatrix} \bar{\mathsf{x}}_{2;v} \\ \bar{\mathsf{y}}_j \\ \vdots \end{pmatrix}$$

The according innovation covariance matrix is

$$\mathsf{S}_{2;l} = \mathsf{C}_{2;l}\bar{\zeta}_2\mathsf{C}_{2;l}^\top + \mathsf{Q}_{2;l}$$

with measurement noise $\mathsf{Q}_{2;l}$. Multiplying out yields

$$\mathsf{S}_{2;l} = \bar{\zeta}_{xx;2} + \mathsf{Q}_{2;l}$$

Relevant for the connection to the global level is the $\mathsf{x}_{2;v}$ component of the state and its covariance $\zeta_{xx;2}$. Performing the update equations we get

$$\begin{aligned} \mathsf{x}_{2;v} &= \bar{\mathsf{x}}_{2;v} + \bar{\zeta}_{xx;2}\mathsf{S}_{2;l}^{-1}\nu_2 \\ \zeta_{xx;2} &= \bar{\zeta}_{xx;2} - \bar{\zeta}_{xx;2}\mathsf{S}_{2;l}^{-1}\bar{\zeta}_{xx;2} \end{aligned}$$

where the innovation is denoted as ν_2. Supposing that the Conditions (A.28) and (A.29) hold between the submaps and the global map, we can calculate the expressions for the measurement noise $\mathsf{Q}_{2;l}$ and the innovation.

Local Update II: Mean After the update, Equation (A.28) should be fulfilled

$$\mathsf{x}_{2;v} = \mathsf{T}_{x_2}\mathsf{x}_{v;g} + \mathsf{T}_{a_2}\mathsf{a}_{2;g}$$

The variables $\mathsf{x}_{v;g}$ and $\mathsf{a}_{2;g}$ are already known from the updated global state.Thus, the innovation can be determined

$$\begin{aligned}
\bar{\mathsf{x}}_{2;v} + \bar{\zeta}_{xx;2}\mathsf{S}_{2;l}^{-1}\nu_2 &= \mathsf{T}_{x_2}\mathsf{x}_{v;g} + \mathsf{T}_{a_2}\mathsf{a}_{2;g} \\
\bar{\zeta}_{xx;2}\mathsf{S}_{2;l}^{-1}\nu_2 &= \mathsf{T}_{x_2}\mathsf{x}_{v;g} + \mathsf{T}_{a_2}\mathsf{a}_{2;g} - \bar{\mathsf{x}}_{2;v} \\
\mathsf{S}_{2;l}^{-1}\nu_2 &= \bar{\zeta}_{xx;2}^{-1}(\mathsf{T}_{x_2}\mathsf{x}_{v;g} + \mathsf{T}_{a_2}\mathsf{a}_{2;g} - \bar{\mathsf{x}}_{2;v}) \\
\nu_2 &= \mathsf{S}_{2;l}\bar{\zeta}_{xx;2}^{-1}(\mathsf{T}_{x_2}\mathsf{x}_{v;g} + \mathsf{T}_{a_2}\mathsf{a}_{2;g} - \bar{\mathsf{x}}_{2;v})
\end{aligned}$$

Local Update II: Covariance For the covariance matrix the Condition (A.29) should hold after the update

$$\zeta_{xx;2} = \mathsf{T}_{x_2}\Sigma_{xx}\mathsf{T}_{x_2}^{\top} + \mathsf{T}_{a_2}\Sigma_{a_2x}\mathsf{T}_{x_2}^{\top} + \mathsf{T}_{x_2}\Sigma_{xa_2}\mathsf{T}_{a_2}^{\top} + \mathsf{T}_{a_2}\Sigma_{a_2a_2}\mathsf{T}_{a_2}^{\top}$$

To simplify the notation let us summarize the right side of the latter equation to $\zeta'_{xx;2}$. It can already be calculated from the updated global covariance. This determines the according local measurement noise

$$\begin{aligned}
\zeta_{xx;2} &= \zeta'_{xx;2} \\
\bar{\zeta}_{xx;2} - \bar{\zeta}_{xx;2}\mathsf{S}_{2;l}^{-1}\bar{\zeta}_{xx;2} &= \zeta'_{xx;2} \\
\bar{\zeta}_{xx;2}\mathsf{S}_{2;l}^{-1}\bar{\zeta}_{xx;2} &= \bar{\zeta}_{xx;2} - \zeta'_{xx;2} \\
\mathsf{S}_{2;l}^{-1} &= \bar{\zeta}_{xx;2}^{-1}(\bar{\zeta}_{xx;2} - \zeta'_{xx;2})\bar{\zeta}_{xx;2}^{-1} \\
\mathsf{S}_{2;l} &= \bar{\zeta}_{xx;2}(\bar{\zeta}_{xx;2} - \zeta'_{xx;2})^{-1}\bar{\zeta}_{xx;2} \\
\bar{\zeta}_{xx;2} + \mathsf{Q}_{2;l} &= \bar{\zeta}_{xx;2}(\bar{\zeta}_{xx;2} - \zeta'_{xx;2})^{-1}\bar{\zeta}_{xx;2} \\
\mathsf{Q}_{2;l} &= \bar{\zeta}_{xx;2}(\bar{\zeta}_{xx;2} - \zeta'_{xx;2})^{-1}\bar{\zeta}_{xx;2} - \bar{\zeta}_{xx;2}
\end{aligned}$$

A.4 Special Case: Locally Measuring the Anchor

By enabling a local filter to measure the anchor locally, we gain more information about the local robot position than just considering the local features. This anchor will not be explicitely included in the map. Because we always know that it is located in the origin of the submap we can still predict the according observation.

The measurement function is then a function of the local vehicle state $\bar{\mathsf{x}}_{v;l}$.

$$\mathsf{z}_{l_a} = \mathsf{C}_{l_a}\mu_l = \begin{bmatrix}\mathsf{C}_{xl_a} & 0 & \ldots\end{bmatrix}\begin{pmatrix}\mathsf{x}_{v;l} \\ \mathsf{y}_i \\ \vdots\end{pmatrix}$$

The equation for the local innovation covariance is

$$\begin{aligned}\mathsf{S}_{l_a} &= \mathsf{C}_{l_a}\overline{\zeta}_l\mathsf{C}_{l_a}^\top + \mathsf{Q}_l\\ &= \mathsf{C}_{xl_a}\overline{\zeta}_{xx}\mathsf{C}_{xl_a}^\top + \mathsf{Q}_l\end{aligned}$$

The local innovation will be denoted by ν_{l_a}. Relevant for the connection between the local and global level is the vehicle state component of the updated state and its covariance. Multiplying out the update equations we get

$$\begin{aligned}\mathsf{x}_{v;l} &= \overline{\mathsf{x}}_{v;l} + \overline{\zeta}_{xx}\mathsf{C}_{xl_a}\mathsf{S}_{l_a}^{-1}\nu_{l_a}\\ \zeta_{xx} &= \overline{\zeta}_{xx} - \overline{\zeta}_{xx}\mathsf{C}_{xl_a}^\top\mathsf{S}_{l_a}^{-1}\mathsf{C}_{xl_a}\overline{\zeta}_{xx}\end{aligned}$$

Appendix B

Hierarchical Models for a One-Dimensional Problem

In this chapter the process and measurement model are given for the CKF applied to a one-dimensional problem.

B.1 State Representation

For the case we are using submaps, we have to consider different state representations. The global state $\mu_{g,t}$ contains the globally referred position of the vehicle x_g, its translational velocity v_g and M submap anchors.

$$\mu_{g,t} = \begin{pmatrix} x_g & v_g & a_1 & \dots & a_i & \dots & a_M \end{pmatrix}^\top$$

The M local states $y_{i,t}$ contain the appropriate locally referred vehicle position x_i, its translational velocity v_i which is equal to the global velocity, and N_i features.

$$\mu_{i,t} = \begin{pmatrix} x_i & v_i & y_1 & \dots & y_j & \dots & y_{N_i} \end{pmatrix}^\top$$

In the following we will refer to the vehicle state, comprising its position and velocity, as x_{r_g} in the global case and x_{r_i} for the local case.

The global covariance matrix is denoted as

$$\Sigma_t = \begin{pmatrix} \Sigma_{xx} & \Sigma_{xv} & \Sigma_{xa_1} & \cdots & \Sigma_{xa_i} & \cdots & \Sigma_{xa_M} \\ \Sigma_{vx} & \Sigma_{vv} & \Sigma_{va_1} & \cdots & \Sigma_{va_i} & \cdots & \Sigma_{va_M} \\ \Sigma_{a_1x} & \Sigma_{a_1v} & \Sigma_{a_1a_1} & \cdots & \Sigma_{a_1a_i} & \cdots & \Sigma_{a_1a_M} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{a_ix} & \Sigma_{a_iv} & \Sigma_{a_ia_1} & \cdots & \Sigma_{a_ia_i} & \cdots & \Sigma_{a_ia_M} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{a_Mx} & \Sigma_{a_Mv} & \Sigma_{a_Ma_1} & \cdots & \Sigma_{a_Ma_i} & \cdots & \Sigma_{a_Ma_M} \end{pmatrix}$$

$$= \begin{pmatrix} \Sigma_{rr} & \Sigma_{ra_1} & \cdots & \Sigma_{ra_i} & \cdots & \Sigma_{ra_M} \\ \Sigma_{a_1r} & \Sigma_{a_1a_1} & \cdots & \Sigma_{a_1a_i} & \cdots & \Sigma_{a_1a_M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{a_ir} & \Sigma_{a_ia_1} & \cdots & \Sigma_{a_ia_i} & \cdots & \Sigma_{a_ia_M} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \Sigma_{a_Mr} & \Sigma_{a_Ma_1} & \cdots & \Sigma_{a_Ma_i} & \cdots & \Sigma_{a_Ma_M} \end{pmatrix}$$

The local covariance is formed equally.

$$\zeta_{i,t} = \begin{pmatrix} \zeta_{xx} & \zeta_{xv} & \zeta_{xy_1} & \cdots & \zeta_{xy_j} & \cdots & \zeta_{xy_N} \\ \zeta_{vx} & \zeta_{vv} & \zeta_{vy_1} & \cdots & \zeta_{vy_j} & \cdots & \zeta_{vy_N} \\ \zeta_{y_1x} & \zeta_{y_1v} & \zeta_{y_1y_1} & \cdots & \zeta_{y_1y_j} & \cdots & \zeta_{y_1y_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \zeta_{y_jx} & \zeta_{y_jv} & \zeta_{y_jy_1} & \cdots & \zeta_{y_iy_j} & \cdots & \zeta_{y_jy_N} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \zeta_{y_Nx} & \zeta_{y_Nv} & \zeta_{y_Ny_1} & \cdots & \zeta_{y_Ny_j} & \cdots & \zeta_{y_Ny_N} \end{pmatrix}$$

$$= \begin{pmatrix} \zeta_{rr} & \zeta_{xy_1} & \cdots & \zeta_{xy_j} & \cdots & \zeta_{xy_N} \\ \zeta_{y_1r} & \zeta_{y_1y_1} & \cdots & \zeta_{y_1y_j} & \cdots & \zeta_{y_1y_N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \zeta_{y_jr} & \zeta_{y_jy_1} & \cdots & \zeta_{y_iy_j} & \cdots & \zeta_{y_jy_N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \zeta_{y_Nr} & \zeta_{y_Ny_1} & \cdots & \zeta_{y_Ny_j} & \cdots & \zeta_{y_Ny_N} \end{pmatrix}$$

According to Appendix A there exists a transformation between the global and all local states. In this one-dimensional example it is structured as follows

$$x_{r_i} = \mathsf{T}_i \begin{pmatrix} x_{r_g} \\ a_i \end{pmatrix}$$

where

$$\begin{aligned}
\mathsf{T}_i &= \begin{pmatrix}\mathsf{T}_r & \mathsf{T}_{a_i}\end{pmatrix}\\
\mathsf{T}_r &= \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}\\
\mathsf{T}_{a_i} &= \begin{pmatrix}-1\\ 0\end{pmatrix}
\end{aligned}$$

Also the covariances are transformable.

$$\begin{aligned}
\zeta_{rr} &= \mathsf{T}_i \begin{pmatrix}\Sigma_{rr} & \Sigma_{ra_i}\\ \Sigma_{a_ir} & \Sigma_{a_ia_i}\end{pmatrix} \mathsf{T}_i^\top\\
&= \mathsf{T}_r\Sigma_{rr}\mathsf{T}_r^\top + \mathsf{T}_{a_i}\Sigma_{a_ir}\mathsf{T}_r^\top + \mathsf{T}_r\Sigma_{ra_i}\mathsf{T}_{a_i}^\top + \mathsf{T}_{a_i}\Sigma_{a_ia_i}\mathsf{T}_{a_i}^\top
\end{aligned}$$

These conditions have to hold before a filter cycle and after it.

B.2 The Predict Step

For the predict step the similar process model described in Section 4.2.1 is used. The difference is that the calculations have to be performed for the global and all local states.

As a result we obtain a predicted global state $\bar{\mu}_{g,t}$ and covariance $\bar{\Sigma}_t$ and local states and $\bar{\mu}_{i,t}$ and covariance $\bar{\zeta}_t$.

In order to ensure that

$$\bar{x}_{r_i} = \mathsf{T}_i \begin{pmatrix}\bar{x}_{r_g}\\ \bar{s}_i\end{pmatrix}$$

and

$$\bar{\zeta}_{rr} = \mathsf{T}_i \begin{pmatrix}\bar{\Sigma}_{rr} & \bar{\Sigma}_{ra_i}\\ \bar{\Sigma}_{a_ir} & \bar{\Sigma}_{a_ia_i}\end{pmatrix} \mathsf{T}_i^\top$$

hold after the predict step we have to fulfill two conditions according to Appendix A. The first is

$$\mathsf{A}_{r_i,t}\mathsf{T}_i = \mathsf{T}_i\mathsf{A}_{r_g,t}$$

where $\mathsf{A}_{r_g,t}$ and $\mathsf{A}_{r_i,t}$ denote the relevant section of the process model for the global and local states, respectively.

$$\begin{aligned}
\mathsf{A}_{r_g,t} &= \begin{pmatrix}1 & \Delta t & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\\
\mathsf{A}_{r_i,t} &= \begin{pmatrix}1 & \Delta t\\ 0 & 1\end{pmatrix}
\end{aligned}$$

Multiplying out the equations for the first condition yields

$$\mathsf{A}_{r_i,t}\mathsf{T}_i = \begin{pmatrix}1 & \Delta t & -1\\ 0 & 1 & 0\end{pmatrix} = \mathsf{T}_i\mathsf{A}_{r_g,t}$$

The second condition is

$$\mathsf{R}_l = \mathsf{T}_i \mathsf{R}'_g \mathsf{T}_i$$

where R_l is the process noise covariances for the local states. R'_g denotes the augmented process noise covariance for the global state.

$$\mathsf{R}'_g = \begin{pmatrix} \mathsf{R}_g & 0 \\ 0 & 0 \end{pmatrix}$$

Multiplying out yields

$$\mathsf{R}_l = \begin{pmatrix} \frac{1}{4}\Delta t^4 \sigma_a^2 & \frac{1}{2}\Delta t^3 \sigma_a^2 \\ \frac{1}{2}\Delta t^3 \sigma_a^2 & \Delta t^2 \sigma_a^2 \end{pmatrix} = \mathsf{T}_i \mathsf{R}'_g \mathsf{T}_i$$

Thus we can state that by using these process models and process noise covariances the transformability between the global and local states still holds after the predict step.

B.3 The Update Step

The update equations are different for both levels due to different measurement entities. On the local level, the distance between a feature and the vehicle is measured, on the global level, local vehicle states are measured.

B.3.1 Local Update

Assuming that we have an observation vector regarding features in one or more submaps, the local observation model is similar to the one decribed in Section 4.2.1. Because it is of the same structure for every submap we will refer to the local observation model as $\mathsf{C}_{l,t}$.

Additionally to the local features in a submap we can also measure the anchor locally. We know that it is always positioned in the submap's origin. Thus, the predicted measurement is equal to the negative robot position. The local anchor oberservation model is written in the form

$$\begin{aligned} z_{a_i,t} &= \mathsf{C}_{a_i,t} \bar{\mu}_{i,t} + \delta_t \\ &= \begin{pmatrix} -1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_i \\ \bar{v}_i \\ \vdots \\ \bar{y}_{N_i} \end{pmatrix} \\ &= -\bar{x}_i \end{aligned}$$

The measurement error covariance matrix Q_t is the same as used for predicting the measurements of the local features.

The output of the local update comprises estimates for those local states and according covariances where a distance measurement was available.

B.3.2 Global Update

The goal of the global update is to ensure that the transformability between the updated local states and the global level holds again. To perform the global update we use measurements of the local robot states which were updated in the previous local update. Supposing that submap i was updated, the global observation model is written in the form

$$z_{i,t} = \mathsf{C}_{i,t}\bar{\mu}_{g,t} + \delta_t$$

where δ_t is a vector of temporally uncorrelated observation errors with zero mean and covariance $\mathsf{Q}_{g,t}$, and $\mathsf{C}_{i,t}$ is the observation matrix which relates the measurement vector $z_{i,t}$ to the state vector $\bar{\mu}_{g,t}$. We yield

$$z_{i,t} = \begin{pmatrix} 1 & 0 & 0 & \dots & -1 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} \bar{x}_g \\ \bar{v}_g \\ \bar{a}_1 \\ \vdots \\ \bar{a}_i \\ \vdots \\ \bar{a}_M \end{pmatrix} + \delta_t$$

to predict the local robot state of the i^{th} submap. Note that the measurement prediction with $\mathsf{C}_{i,t}$ is similar to transform the global state with T_i into the local level. Thus, $\mathsf{C}_{i,t}$ can also be written as

$$\mathsf{C}_{i,t} = \begin{pmatrix} \mathsf{T}_r & 0 & \dots & \mathsf{T}_{a_i} & \dots & 0 \end{pmatrix}$$

According to Appendix A, the measurement noise covariance $\mathsf{Q}_{i,t}$ is not a fixed matrix but needs to be calculated for every filter cycle. Also the global measurement innovation is not simply the difference between predicted and real measurement.
The measurement noise covariance is calculated as follows

$$\mathsf{Q}_{i,t} = \bar{\Sigma}_{ll}(\bar{\Sigma}_{ll} - \zeta_{rr})^{-1}\bar{\Sigma}_{ll} - \bar{\Sigma}_{ll}$$

where

$$\begin{aligned} \bar{\Sigma}_{ll} &= \mathsf{T}_i \begin{pmatrix} \bar{\Sigma}_{rr} & \bar{\Sigma}_{ra_i} \\ \bar{\Sigma}_{a_ir} & \bar{\Sigma}_{a_ia_i} \end{pmatrix} \mathsf{T}_i^\top \\ &= \mathsf{T}_r\bar{\Sigma}_{rr}\mathsf{T}_r^\top + \mathsf{T}_{a_i}\bar{\Sigma}_{a_ir}\mathsf{T}_r^\top + \mathsf{T}_r\bar{\Sigma}_{ra_i}\mathsf{T}_{a_i}^\top + \mathsf{T}_{a_i}\bar{\Sigma}_{a_ia_i}\mathsf{T}_{a_i}^\top \\ &= \begin{pmatrix} \bar{\Sigma}_{xx} - \bar{\Sigma}_{a_ix} - \bar{\Sigma}_{xa_i} + \bar{\Sigma}_{a_ia_i} & \bar{\Sigma}_{xv} - \bar{\Sigma}_{a_iv} \\ \bar{\Sigma}_{vx} - \bar{\Sigma}_{va_i} & \bar{\Sigma}_{vv} \end{pmatrix} \end{aligned}$$

The measurement innovation ν_g is calculated by performing

$$\begin{aligned} \nu_g &= S_g\bar{\Sigma}_{ll}^{-1}(x_{r_i} - \mathsf{T}_r\bar{x}_{r_g} - \mathsf{T}_{a_i}\bar{a}_i) \\ &= S_g\bar{\Sigma}_{ll}^{-1} \begin{pmatrix} x_i & -x_g & +a_i \\ v_i & -v_g & \end{pmatrix} \end{aligned}$$

where

$$\begin{aligned} S_g &= \mathsf{T}_i \begin{pmatrix} \bar{\Sigma}_{rr} & \bar{\Sigma}_{ra_i} \\ \bar{\Sigma}_{a_i r} & \bar{\Sigma}_{a_i a_i} \end{pmatrix} \mathsf{T}_i^\top + \mathsf{Q}_{i,t} \\ &= \bar{\Sigma}_{ll} + \mathsf{Q}_{i,t} \\ &= \bar{\Sigma}_{ll}(\bar{\Sigma}_{ll} - \zeta_{rr})^{-1}\bar{\Sigma}_{ll} \end{aligned}$$

B.3.3 Local Update II

In the considered filter cycle there might be submaps which were predicted but not yet updated because none of its local features was measured. However, we need to update also these submaps to ensure that the transfomability holds between the global state and the mentioned local states. In order to do so we exploit the information we gained globally about the vehicle state in the previous global update. The measurements in this second local update will contain the local vehicle positions. Assuming that submap $i+1$ was not yet updated, as a measurement model we have

$$z_{i+1,t} = \begin{pmatrix} \bar{x}_{i+1} \\ \bar{v}_{i+1} \end{pmatrix} + \delta_t$$

Again, the according measurement noise is not a fixed covariance matrix but needs to be calculated for every update equation. According to Appendix A, we have

$$\mathsf{Q}_{i+1,t} = \bar{\zeta}_{rr}(\bar{\zeta}_{rr} - \zeta'_{rr})^{-1}\bar{\zeta}_{rr} - \bar{\zeta}_{rr}$$

where

$$\begin{aligned} \zeta'_{rr} &= \mathsf{T}_i \begin{pmatrix} \Sigma_{rr} & \Sigma_{ra_i} \\ \Sigma_{a_i r} & \Sigma_{a_i a_i} \end{pmatrix} \mathsf{T}_i^\top \\ &= \mathsf{T}_r \Sigma_{rr} \mathsf{T}_r^\top + \mathsf{T}_{a_i} \Sigma_{a_i r} \mathsf{T}_r^\top + \mathsf{T}_r \Sigma_{ra_i} \mathsf{T}_{a_i}^\top + \mathsf{T}_{a_i} \Sigma_{a_i a_i} \mathsf{T}_{a_i}^\top \\ &= \begin{pmatrix} \Sigma_{xx} - \Sigma_{a_{i+1}x} - \Sigma_{xa_{i+1}} + \Sigma_{a_{i+1}a_{i+1}} & \Sigma_{xv} - \Sigma_{a_{i+1}v} \\ \Sigma_{vx} - \Sigma_{va_{i+1}} & \Sigma_{vv} \end{pmatrix} \end{aligned}$$

The innovation is again different from just substracting the predicted from the real measurement. We have to calculate

$$\begin{aligned} \nu_{i+1,t} &= S_{i+1,t}\bar{\zeta}_{rr}^{-1}(\mathsf{T}_r x_{r_g} + \mathsf{T}_{a_{i+1}} a_{i+1} - \bar{x}_{r_{i+1}}) \\ &= S_{i+1,t}\bar{\zeta}_{rr}^{-1} \begin{pmatrix} x_g & -a_{i+1} & -\bar{x}_{i+1} \\ v_g & & -\bar{v}_{i+1} \end{pmatrix} \end{aligned}$$

where

$$\begin{aligned} S_{i+1,t} &= \bar{\zeta}_{rr} + \mathsf{Q}_{i+1,t} \\ &= \bar{\zeta}_{rr}(\bar{\zeta}_{rr} - \zeta'_{rr})^{-1}\bar{\zeta}_{rr} \end{aligned}$$

Appendix C

State Extension

Suppose we have a state vector μ of which we make measurement

$$\mathsf{z} = \mathsf{C}\mu,$$

using measurement model C.

We will add the measured quantity $\mathsf{C}\mu$ to the state as e. We create the *extended state* μ_e which incorporates e using an extension transformation E as follows.

$$\mu_e = \begin{pmatrix} \mu \\ \mathsf{e} \end{pmatrix} = \begin{bmatrix} \mathsf{I} \\ \mathsf{C} \end{bmatrix} \mu = \mathsf{E}\mu$$

We define a measurement model C_e which measures e directly

$$\mathsf{z} = \mathsf{C}\mu = \mathsf{e} = \begin{bmatrix} 0 & \mathsf{I} \end{bmatrix} \mu_e = \mathsf{C}_e\mu_e$$

We use the function symbol **update** to denote a Kalman Filter update step, i.e.

$$\mu = \text{update}(\overline{\mu}, \mathsf{C}, \mathsf{Q}, \mathsf{z})$$

where $\mu \sim \mathcal{N}(\mu, \Sigma)$ is the updated state, $\overline{\mu} \sim \mathcal{N}(\overline{\mu}, \overline{\Sigma})$ is the predicted state, C is the measurement model, Q the measurement noise covariance and z the measurement.

Applying the extension transformation E to a state estimate yields

$$\mathcal{N}(\mu_e, \Sigma_e) = \mathsf{E} \cdot \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mathsf{E}\mu, \mathsf{E}\Sigma\mathsf{E}^\top)$$

It is not too difficult to see that

$$\mathsf{E} \cdot \text{update}(\overline{\mu}, \mathsf{C}, \mathsf{Q}, \mathsf{z}) = \text{update}(\mathsf{E} \cdot \overline{\mu}, \mathsf{C}_e, \mathsf{Q}, \mathsf{z}).$$

Proof: Given a state prediction with mean $\overline{\mu}$ and covariance $\overline{\Sigma}$, the extended state prediction has mean and covariance

$$\overline{\mu}_e = \begin{pmatrix} \overline{\mu} \\ \mathsf{C}\overline{\mu} \end{pmatrix}$$

$$\overline{\Sigma}_e = \begin{bmatrix} \overline{\Sigma} & \overline{\Sigma}\mathsf{C}^\top \\ \mathsf{C}\overline{\Sigma} & \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix}$$

Before we take a closer look at update($\overline{\mu}, \mathsf{C}, \mathsf{Q}, \mathsf{z}$) and update($\mathsf{E} \cdot \overline{\mu}, \mathsf{C}_e, \mathsf{Q}, \mathsf{z}$), we note that the innovations and innovation covariances are the same for both:

$$\nu_e = \mathsf{z} - \mathsf{C}_e \overline{\mu}_e = \mathsf{z} - \mathsf{C}\overline{\mu} = \nu$$

$$\mathsf{S}_e = \mathsf{C}_e \overline{\Sigma}_e \mathsf{C}_e^\top + \mathsf{Q} = \mathsf{C}\overline{\Sigma}\mathsf{C}^\top + \mathsf{Q} = \mathsf{S}$$

Now, lets perform update($\overline{\mu}, \mathsf{C}, \mathsf{Q}, \mathsf{z}$). We have

$$\begin{aligned} \mathsf{W} &= \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1} \\ \mathsf{W}\mathsf{S}\mathsf{W}^\top &= \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma} \end{aligned}$$

For the updated state we get

$$\begin{aligned} \mu &= \overline{\mu} + \mathsf{W}\nu = \overline{\mu} + \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\nu \\ \Sigma &= \overline{\Sigma} - \mathsf{W}\mathsf{S}\mathsf{W}^\top = \overline{\Sigma} - \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma} \end{aligned}$$

We extend the updated state to get $\mathsf{E} \cdot$ update($\overline{\mu}, \mathsf{C}, \mathsf{Q}, \mathsf{z}$). To distinguish it from the updated extended state update($\mathsf{E} \cdot \overline{\mu}, \mathsf{C}_e, \mathsf{Q}, \mathsf{z}$), we denote its mean and covariance by μ'_e and Σ'_e. From the extension transformation, we get

$$\begin{aligned} \overline{\mu}'_e &= \begin{pmatrix} \mu \\ \mathsf{C}\mu \end{pmatrix} \\ \Sigma'_e &= \begin{bmatrix} \Sigma & \Sigma\mathsf{C}^\top \\ \mathsf{C}\Sigma & \mathsf{C}\Sigma\mathsf{C}^\top \end{bmatrix} \end{aligned}$$

Lets look at update($\mathsf{E} \cdot \overline{\mu}, \mathsf{C}_e, \mathsf{Q}, \mathsf{z}$). We have

$$\begin{aligned} \mathsf{W}_e &= \overline{\Sigma}_e \mathsf{C}_e^\top \mathsf{S}^{-1} = \begin{bmatrix} \overline{\Sigma}\mathsf{C}^\top \\ \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix} \mathsf{S}^{-1} \\ \mathsf{W}_e \mathsf{S}\mathsf{W}_e^\top &= \overline{\Sigma}_e \mathsf{C}_e^\top \mathsf{S}^{-1} \mathsf{C}_e \overline{\Sigma}_e \top = \begin{bmatrix} \overline{\Sigma}\mathsf{C}^\top \\ \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix} \mathsf{S}^{-1} \begin{bmatrix} \mathsf{C}\overline{\Sigma} & \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix} \\ &= \begin{bmatrix} \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma} & \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma}\mathsf{C}^\top \\ \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma} & \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix} \end{aligned}$$

For the updated extended state we get

$$\begin{aligned} \mu_e &= \overline{\mu}_e + \mathsf{W}_e\nu = \begin{pmatrix} \overline{\mu} \\ \mathsf{C}\overline{\mu} \end{pmatrix} + \begin{bmatrix} \overline{\Sigma}\mathsf{C}^\top \\ \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix} \mathsf{S}^{-1}\nu = \begin{bmatrix} \overline{\mu} + \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\nu \\ \mathsf{C}\overline{\mu} + \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\nu \end{bmatrix} = \begin{pmatrix} \mu \\ \mathsf{C}\mu \end{pmatrix} = \mu'_e \\ \Sigma_e &= \overline{\Sigma} - \mathsf{W}_e \mathsf{S}\mathsf{W}_e^\top = \begin{bmatrix} \overline{\Sigma} - \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma} & \overline{\Sigma}\mathsf{C}^\top - \overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma}\mathsf{C}^\top \\ \mathsf{C}\overline{\Sigma} - \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma} & \mathsf{C}\overline{\Sigma}\mathsf{C}^\top - \mathsf{C}\overline{\Sigma}\mathsf{C}^\top \mathsf{S}^{-1}\mathsf{C}\overline{\Sigma}\mathsf{C}^\top \end{bmatrix} \\ &= \begin{bmatrix} \Sigma & \Sigma\mathsf{C}^\top \\ \mathsf{C}\Sigma & \mathsf{C}\Sigma\mathsf{C}^\top \end{bmatrix} = \Sigma'_e \end{aligned}$$

□

Bibliography

[1] Y. Bar-Shalom. *Tracking and data association.* Academic Press Professional, Inc., San Diego, CA, USA, 1987.

[2] J. Bohg. Real-time structure from motion using kalman filtering. Prediploma, Knowledge Representation and Reasoning Group, Department of Computer Science, Dresden University of Technology, March 2005.

[3] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[4] J. A. Castellanos, M. Devy, and J. D. Tardós. Towards a topological representation of indoor environments: A landmark-based approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 23–28, 1999.

[5] J. A. Castellanos, J. D. Tardós, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *Proceedings of the 1997 IEEE International Conference on Robotic and Automation*, April 1997.

[6] K. S. Chong and L. Kleeman. Large scale sonarray mapping using multiple connected local maps. In *International Conference on Field and Service Robotics*, pages 538–545, December 1997.

[7] M. Csorba. *Simultaneous Localisation and Map Building.* PhD thesis, University of Oxford, 1997.

[8] A. J. Davison. *Mobile Robot Navigation Using Active Vision.* Phd thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, 1998.

[9] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 2003.

[10] F. Dellaert and A. Stroupe. Linear 2d localization and mapping for single and multiple robots. In *Proceedings of the IEEE International Conference on Robotics and Automation, 2002.* IEEE, May 2002.

[11] G. Dissanayake, H. F. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In *ICRA*, pages 1009–1014, 2000.

[12] G. Dissanayake, H. F. Durrant-Whyte, and P. Gibbons. Towards deployment of large scale simultaneous localisation and map building (slam) systems. Technical report, Australien Centre of Field Robotics, University of Sydney, 2000.

[13] G. Dissanayake, P. Newman, H. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. In *IEEE Transactions on Robotics and Autonomation*, volume 17, pages 229–241, June 2001.

[14] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical slam: real-time accurate mapping of large environments. In *IEEE transactions on Robotics*, 2004.

[15] J. Guivant. *Efficient Simultaneous Localisation and Mapping in Large Environments*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, 2001.

[16] J. Guivant and E. Nebot. Solving Computational and Memory Requirements of Feature Based Simultaneous Localization and Map Building Algorithms. Technical report, Australian Centre for Field Robotics, Sydney, 2002.

[17] J. Guivant, E. Nebot, and H. Durrant-Whyte. Simultaneous localisation and map building using natural features in outdoor environments. In *Proceedings of Intelligent Autonomous Systems*, pages 581–586, Italy, July 2000.

[18] J. E. Guivant and E. M. Nebot. Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation. In *IEEE Transactions on Robotics and Automation*, volume 17, pages 242–257. IEEE Computer Society Press, 2001.

[19] D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7):579–598, 2003.

[20] I.N. Bronstein and K.A. Semendjajew and G.Musiol and H. Mühlig. *Handbuch der Mathematik*. Verlag Harry Deutsch, 2001.

[21] S. J. Julier. The stability of covariance inflation methods for slam. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2749– 2754, October 2003.

[22] S. J. Julier and J. K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *The American Control Conference*, pages 27–31, San Francisco, California, 1997. Morgan Kaufmann Publishers.

[23] S. J. Julier and J. K. Uhlmann. Using multiple slam algorithms. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 200–205, 2003.

[24] R. E. Kalman. An new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[25] J. Knight. Computationally Tractable SLAM. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 2001.

[26] J. Knight. *Towards Fully Autonomous Visual Navigation.* PhD thesis, Robotics Research Group, University of Oxford Department of Engineering Science, 2002.

[27] J. Knight, A. Davison, and I. Reid. Towards constant time SLAM using postponement. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems, Maui, HI*, volume 1, pages 406–412. IEEE Computer Society Press, oct 2001.

[28] J. J. Leonard and H. Durrant-Whyte. Simultaneous Map Building and Localization for an Autonomous Mobile Robot. In *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 1442–1447, May 1991.

[29] J. J. Leonard and H. J. S. Feder. Decoupled stochastic mapping. Technical Memorandum 99-1, MIT Marine Robotics Laboratory, December 1999.

[30] J. J. Leonard and P. M. Newman. Consistent, convergent, and constant-time slam. In *IJCAI*, pages 1143–1150, 2003.

[31] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset. Hierarchical simultaneous localization and mapping. In *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, pages 448–453. IEEE/RSJ, Oct. 27-31 2003.

[32] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *AAAI/IAAI*, pages 593–598, 2002.

[33] D. Murray and J. J. Little. Environment modeling with stereo vision. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, September/ October 2004.

[34] D. Murray and J. J. Little. Segmenting correlation stereo range images using surface elements. In *3DPVT*, pages 656–663, 2004.

[35] D. Murray and J. J. Little. Patchlets: Representing stereo vision data with surface elements. In *WACV/MOTION*, pages 192–199, 2005.

[36] P. Newman. *On the Structure and Solution of the Simultaneous Localization and Mapping Problem.* PhD thesis, University of Sydney, 1999.

[37] P. M. Newman, J. J. Leonard, and R. J. Rikoski. Towards constant-time slam on an autonomous underwater vehicle using synthetic aperture sonar. In *Proceedings of the Eleventh International Symposium on Robotics Research*, Sienna, Italy, October 2003.

[38] T. Pietzsch and A. Großmann. A method of estimating oriented surface elements from stereo images. In *Proceedings of the British Machine Vision Conference (BMVC),*, volume 1, pages 320–329, Oxford, UK, September 2005.

[39] Point Grey::Home, 2006. Available via http://www.ptgrey.com.

[40] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C++*. Cambridge University Press, Cambridge, 2002.

[41] R. Sim and N. Roy. Global a-optimal robot exploration in slam. In *Proceedings of the IEEE/RSJ International Conference of Robotics and Automation (ICRA 2005)*, 2005.

[42] R. Smith, M. Self, and P. Cheesman. Estimating Uncertain Spatial Relationships in Robotics. In *Proceedings of the 1st Annual Conference on Uncertainty in Artificial Intelligence (UAI-85)*, New York, NY, 1985. Elsevier Science Publishing Company, Inc.

[43] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[44] G. Welch and G. Bishop. An Introduction to Kalman Filter, 2001. Siggraph, Course 8.

[45] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(3):193–205, 1997. ISSN 0162-8828.

[46] Y.-K. Yu, K.-H. Wong, and M. M.-Y. Chang. Recursive 3D Model Reconstruction Based on Kalman Filtering. In *IEEE Transactions on Systems, Man and Cybernetics- Part B*, 2003.

[47] Y.-K. Yu, K.-H. Wong, and M. M.-Y. Chang. A fast recursive 3d model reconstruction algorithm for multimedia applications. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, 2004.

Declaration

All the work enclosed was done by me except where otherwise stated. Every section that reports or paraphrases work by other authors indicates who the other authors are and gives a reference to the article, book, or other source. All quotations from other work are explicitly indicated as such and a reference to the original is given.

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfmittel verwendet, sowie Zitate kenntlich gemacht habe.

Dresden, Dezember 2005 Jeannette Bohg