Extraction, Localization, and Fusion of Collective Vehicle Data

Dissertation

zur Erlangung des Grades eines

Doktors der Ingenieurwissenschaften

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Sebastian Skibinski

Dortmund 2019

Tag der mündlichen Prüfung: 28.02.2019

Dekan: Prof. Dr.-Ing. Gernot A. Fink

Gutachter: Prof. Dr. Heinrich Müller
Prof. Dr.-Ing. Uwe Schwiegelshohn

"Study and in general the pursuit of truth and beauty is a sphere of activity in which we are permitted to remain children all our lives."

— Albert Einstein (1921)

Acknowledgments

At first, I would like to express my exceptional gratitude to my advisor Prof. Dr. Heinrich Müller and Dr. Frank Weichert for their continuous guidance, motivation, support, and aim for perfection during the genesis of my Ph.D. thesis.

Furthermore, I would like to thank Prof. Dr.-Ing. Uwe Schwiegelshohn for being the second reviewer of my thesis and his many valuable comments. I would also like to thank the other members of my committee, Prof. Dr. Jens Teubner and Prof. Dr. Heiko Krumm, for their numerous exciting and challenging questions during the defense of my thesis.

My further gratitude goes to Nils Oppermann, who was an exceptional supervisor and colleague at Audi.

I am incredibly grateful to my parents, family, and friends for their moral and emotional support. Thank you a lot!

Contents

Mathematical Nomenclature

	Description
$\mathbb{N}$	Set of non-negative integer numbers
$\mathbb{N}_{>0}$	Equal to $\mathbb{N}$ but excluding zero
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^d$	d-dimensional space
$\boldsymbol{p}$	Vector
$\boldsymbol{A}$	Matrix
$\boldsymbol{I}_{d\times d}$	Identity matrix of dimension d
(a, b)	Open interval a, b
$[a, b),\ (a, b]$	Half-open interval a, b
$[a, b]$	Closed-interval a, b
$\Vert\boldsymbol{x}\Vert_{\mathrm{p}}$	Length of a vector $\boldsymbol{x}$ by utilizing the Minkowski distance $\left(\sum_{i=1}^{n}\vert\boldsymbol{x}_i\vert^{p}\right)^{1/p}$ with $p \in \mathbb{N}_{>0}$. For $p = 1$ it is equal to the Manhattan distance, while for $p = 2$ it is equal to the Euclidean distance.
$\Vert\boldsymbol{x}\Vert_{\boldsymbol{\Sigma}}$	Length of a vector $\boldsymbol{x}$ by utilizing the Mahalanobis distance $\sqrt{\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}}$
$\Vert\boldsymbol{x}\Vert_{\boldsymbol{\Sigma}}^{n}$	Exponentiated length of a vector $\boldsymbol{x}$ by utilizing the Mahalanobis distance $\left(\sqrt{\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x}}\right)^n$
$a \sim b$	a is approximate to b
$a \propto b$	a is proportional to b
$\mathcal{X}$	Set
$\Vert\mathcal{X}\Vert$	Cardinality of the set $\mathcal{X}$
$\langle\boldsymbol{x}, \boldsymbol{y}\rangle$	Dot product of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$

$\det(\boldsymbol{X})$	Determinant of the matrix $\boldsymbol{X}$
$\operatorname{trace}(\boldsymbol{X})$	Trace of the matrix $\boldsymbol{X}$
$\exp(x), e^x$	Natural exponential function
$\log(x)$	Natural logarithm
$\log_{10}(x)$	Common logarithm
$P(x), P_x$	Probability of the random variable x
$P(x\|y), P_{x\|y}$	Conditional probability of the random variable x given variable y
$\mathrm{E}(x), \overline{x}$	Expected value of the random variable x
$\operatorname{Cov}(x, y), \boldsymbol{C}_{x,y}$	Covariance of the random variables x and y
$\widehat{x}, \widehat{\boldsymbol{x}}, \widehat{\boldsymbol{X}}$	Estimate of a variable x, vector $\boldsymbol{x}$ or matrix $\boldsymbol{X}$
$\frac{\partial f(x)}{\partial x}$	Derivative of the function $f(x)$
$\left.\frac{\partial f(x)}{\partial x}\right\|_{x_0}$	Derivative of the function $f(x)$ that is evaluated at x_0
$\frac{\partial f(\boldsymbol{x})}{\partial x_i}$	Partial derivative of the function $f(\boldsymbol{x})$ with respect to the variable $x_i \in \boldsymbol{x}$
$\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}, \begin{pmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\boldsymbol{x})}{\partial x_n} \end{pmatrix}$	Jacobian matrix of all partial derivatives of a vector-valued function $\boldsymbol{f}(\boldsymbol{x})$ with respect to the variables $x_i \in \boldsymbol{x}$
$T_n\ f_a(x)$	n-th order Taylor series of the function $f(x)$ for a given development point a
$\arg\max\limits_x\ f(x)$	Argument for which the function $f(x)$ attains its maximum value (argumentum maximi)
$\arg\min\limits_x\ f(x)$	Argument for which the function $f(x)$ attains its minimum value (argumentum minimi)

Abstract

Maps representing the detailed features of the road network are becoming more and more important for self-driving vehicles and next generation driver assistance systems. The mapping of the road network, by specially equipped vehicles of the well-known map providers, leads to usually quarterly map updates, which might result in problems encountered by self-driving vehicles in the case that the road information is outdated. Furthermore, the provided maps could lack details, such as precise landmark geometries or data known to exhibit a fast temporal decay rate, which might be, nevertheless, highly relevant, such as friction data.

As an alternative, extensive amounts of information about the road network can be acquired by common vehicles, which are, nowadays, commonly equipped with manifold types of sensors. Subsequently, this type of gathered data is referred to as CVD[1]. The process of map creation requires, at first, the extraction of relevant sensor data at the vehicle-side and its accurate localization. Unfortunately, sensor data is typically affected by measurement uncertainties and errors. A minimization of both can be achieved by means of an appropriate sensor data fusion.

This work aims for a holistic view of a three-staged pipeline, consisting of the extraction, localization, and fusion of CVD, intended for the derivation of large-scale, high-precision, real-time maps from collective sensor measurements acquired by a common vehicle fleet. The vehicle fleet is assumed to be solely equipped with commercially viable sensors. Concerning the processing at the back-end-side, general approaches that are applicable in a straightforward manner to new types of sensor data are strictly favored. For this purpose, a novel distinction of CVD into areal, point-shaped landmark, and complex landmark data is introduced. This way, the similarities between different types of environmental attributes are exploited in an overall highly beneficial manner; and the proposed algorithms can be adapted to new types of data that appertain to these categories by appropriately adjusting their parameterizations.

To achieve the above mentioned goals, both novel approaches, where the research lacks established ways, and relevant extensions/adaptations of existing ones are suggested to fulfill the very specific, automotive requirements. To briefly sum up, the major aspects tackled within this thesis are

[1] Collective Vehicle Data (CVD)

- the precise vehicle localization by utilizing GNSS[2]-postprocessing at the back-end-side, which is shown to both significantly improve the localization accuracy and to be feasible for the use case of CVD,
- an extension to global submapping, referred to as adaptive partitioning, which allows the subdivision of large-scale problems into small chunks in an adaptive, consistent, and resource efficient manner,
- a novel approach for determining temporal weighting in an adaptive manner, which allows to appropriately derive the landmark and localization specific temporal decay rates in a highly efficient way by means of preaggregation,
- a highly efficient, multi-hypothesis clustering based upon JCBB[3] data association, which involves i. a. a novel multi-criteria cost-function that allows a goal-oriented selection of the most probable hypotheses from a principally exponential search space,
- a novel approach to the simultaneous fusion of possibly only partially observed complex and non-complex landmark observations, which are allowed to exhibit tight attribute interdependencies, and are, therefore, provided in the shape of a novel, parametric description, the so-called ADG[4],
- a novel, double-staged approach to areal data fusion, which is both scalable and incrementally computable, and exploits the FGT[5] in the case of batch inserts or updates to enhance the computational efficiency by up to a factor of ten,
- a novel, highly flexible, and modular instantiation of the suggested approaches at both the vehicle- and back-end-side, and
- a novel, highly flexible, and efficient data storage, which utilizes hierarchically cascaded templates, and which can be, moreover, considered to be the foundation stone for the definition of generalized processing functions.

All in all, this thesis condenses a broad and manifold research concerning the deduction of large-scale and high-precision map data grounded on preprocessed sensor measurements that have been acquired by common vehicles, the so-called CVD. The focus is put on the utilization of commercially viable sensors. Additionally, besides its broad perspective, this thesis also emphasizes highly relevant details, such as the efficient, adaptive temporal weighting of sensor data at the back-end-side and the template-based hierarchical data storage. A complete pipeline, consisting

[2] Global Navigation Satellite System (GNSS)
[3] Joint Compatibility Branch and Bound (JCBB)
[4] Attribute Dependency Graph (ADG)
[5] Fast Gauss Transform (FGT)

of the extraction, localization, and fusion of CVD, is presented and evaluated, as each component is known to have a direct impact on the quality of the deduced map data. Approaches to the fusion of areal and point-shaped/complex landmark data are either invented from scratch or significantly enhanced according to the state of the art, always bearing in mind the highly specific needs of the automotive context.

1 Introduction

The chapter starts with a motivation of the guiding theme of the thesis, the automatic generation of up-to-date maps from sensor data acquired by a fleet of vehicles. Afterwards, it illuminates three key steps to achieve this goal, "extraction", "localization", and "fusion" of collective vehicle data (CVD), in the automotive context. Then the concept of solution presented in the thesis is outlined, and the most important contributions are pointed out. The chapter ends with surveys of the structure of the thesis and of publications by the author.

Contents

1.1 Motivation

Digital maps representing the road network and its current state are becoming more and more important for self-driving vehicles and next generation driver assistance systems. Digital maps can be considered as an additional, *virtual* sensor with an extensive foresight, that extends the typical sensor set of modern vehicles in a sensible manner. A particular strength of this virtual sensor is that it is not exposed to occlusions or adverse weather conditions. However, it is directly affected by the quality and freshness of the map data it is grounded on. Therefore, maps of a high quality and up-to-dateness are mandatory, especially if incorporated in safety-critical systems, such as piloted driving ones.

In today's times, the mapping of the road network by specially equipped vehicles of the well-known map data providers leads to usually quarter-wise map updates. This low frequency of updates is in particular an issue for self-driving vehicles.

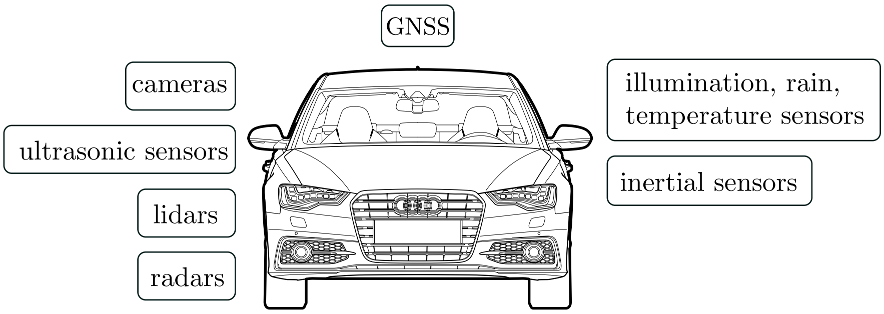


Figure 1.1: Modern vehicles incorporate manifold types of sensors which can be utilized for the acquisition of CVD and, this way, for the collaborative mapping of the road network and for gaining information about its current state by means of sensor data fusion.

Furthermore, the maps usually lack specific details, such as the exact geometry of roundabouts, or information about the road network that has a rapid temporal decay, for example that a road segment is slippery.

An approach to cope with this problem is to utilize the manifold types of distinct sensors incorporated within today's common vehicles (figure 1.1), such as cameras, ultrasonic sensors, lidars, radars, illumination/rain/temperature sensors, inertial sensors, and GNSS[1]-receivers. The road network and information about its current state can be collaboratively gathered by common vehicles, using those sensors, and transmitted to a common back-end where they are subsequently fused to detailed up-to-date maps with supplementary information which are back-propagated to vehicles for further use. The sensors of common vehicles do not always achieve the quality of the costly, highly specialized equipment of mapping vehicles, for instance due to commercial constraints. However, as common vehicles are able to collaboratively acquire vast amounts of corresponding sensor data, the hope is that imperfections of the data may be compensated in practice by exploiting the redundancy of the measurements made. This way of derivation of high-precision and highly up-to-date map data and supplemental information based upon CVD[2] can be considered as a very promising approach that will empower next generation piloted and driver assistance functions. The thesis makes novel contributions to this approach.

1.2 Extraction, Localization, and Fusion

Three key steps are crucial for the collaborative mapping of the road network and gaining information about its current state by common vehicles: the extraction of sensor data from the vehicle fleet, its accurate localization, and its precise and robust fusion.

[1] Global Navigation Satellite System (GNSS)
[2] Collective Vehicle Data (CVD)

Extraction refers to the process of acquiring CVD from the vehicles, and propagating it to a common back-end. The data reaching the back-end can directly originate from sensors, or may be preprocessed at the vehicle-side by ECU[3]s. However, due to bandwidth limitations that are imposed by the cellular network and limitations of computational resources at the back-end-side, the propagation of sensor raw data can be considered as commonly infeasible. Instead, alternative approaches that represent this data more concisely are highly desirable.

Localization means the detection of the current geographic locations of the vehicles. An accurate localization can be considered as vital for the precise geographic referencing of the acquired sensor data. The quality of localization is commonly regarded to directly affect the quality of the fusion. Highly accurate approaches to localization are known from geodesy and usually provide an absolute accuracy within a centimeter range. However, those approaches usually cannot be directly incorporated within common vehicles due to economic constraints. Unfortunately, this by now excludes especially dual- and multi-frequency GNSS receivers, and high-grade IMU[4]s. Therefore, other ways of gaining a precise vehicle navigation are required.

Fusion is concerned with combining the collaboratively acquired and subsequently referenced sensor data in a way that highly-precise and large-scale maps are gained. Different types of real-world artifacts, such as traffic signs or friction data, usually require distinct approaches. On the other hand, generic approaches to the fusion of CVD would minimize the adaptation time for new use cases and types of artifacts. It is a challenge to combine both principally converse requirements.

An illustrative scenario including acquisition, fusion, and back-propagation of CVD, is visualized in figure 1.2. Detected roundabouts, slippery road segments, stop-and-go traffic, and traffic sign observations are exemplary reported to the back-end-side by means of the cellular network. At the back-end-side, the CVD received is fused and provided as a-priori data to subsequent vehicles.

1.3 Automotive Requirements

From the automotive point of view, universality, scalability, incrementality, high-precision, robustness, and compensation for temporal decay are considered as crucial properties of a pipeline intended for the fusion of CVD. In the following, these aspects are explained in detail and, furthermore, a fundamental set of requirements considered within this thesis is constituted.

[3] Electronic Control Unit (ECU)

[4] Inertial Measurement Unit (IMU)

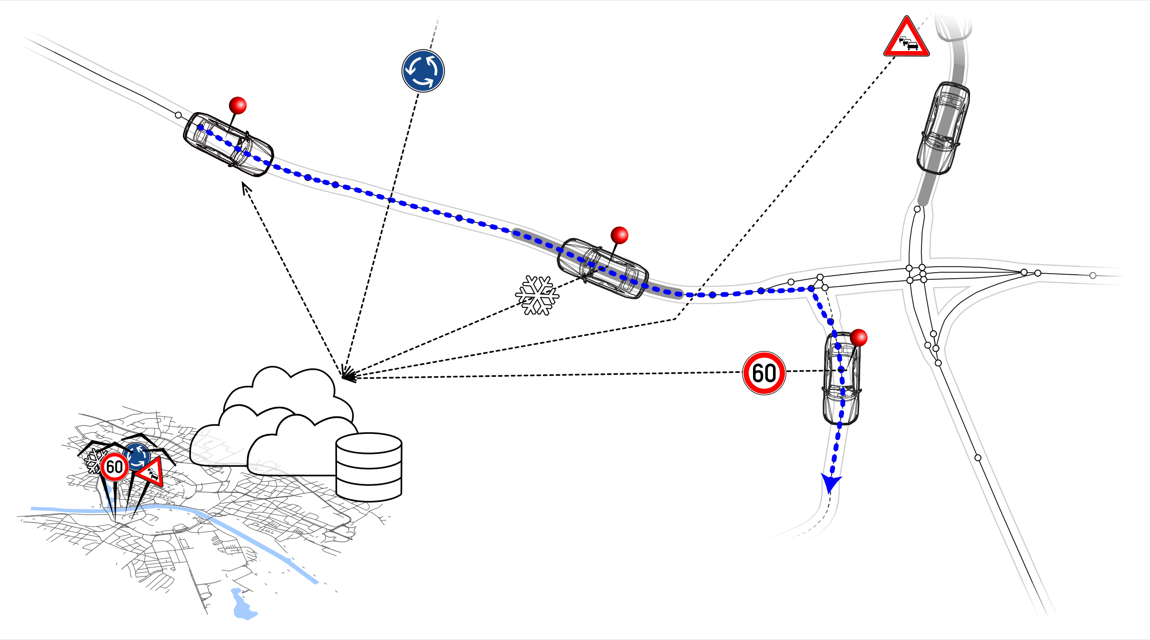


Figure 1.2: An exemplary scenario visualizing the processing cycle of CVD, consisting of data acquisition, subsequent data propagation to a common back-end, and the back-propagation of the fused data to successive vehicles.

Universality

Universality means that the incorporated acquisition, propagation, and processing steps are applicable to distinct types of CVD. This way, development and maintenance costs can be significantly reduced, and the adaptation to new use cases and types of CVD can be eased. The requirement of universality can be considered to affect the full CVD chain.

Scalability

The processing of CVD is required to tackle large scale. For this purpose, a pipeline for processing CVD needs to be scalable. Horizontal scaling is of special interest, as it enables to scale out the usually computationally expensive processing to a theoretically indefinite amount of computing units. In comparison, the impact of vertical scaling against the backdrop of huge amounts of acquired CVD is limited, however, should still be not neglected.

Incrementality

Generally, incrementality allows, in the case of new CVD input, to reutilize former computational results for determining subsequent aggregates. Usually, this way valuable computational resources can be saved. Especially in the case of applications with huge amounts of data, incrementality significantly matters.

High-Precision and Robustness

As the derived map data and information is intended to be principally utilized within self-driving vehicles and next-generation driver assistance systems, the precision and robustness of the fusion has to be particularly considered. The precision can usually be measured by determining the deviation from the ground truth. Measuring robustness is usually more complicated and commonly requires to be defined for each application individually. For example, the robustness of a fusion algorithm can be determined based upon its risk of divergence and the robustness of a clustering algorithm upon its risk of undesired cluster splits.

Compensation for Temporal Decay

The road network and information about its current state are exposed to environmental dynamics. Therefore, CVD is significantly affected by temporal decay and should be only fused in an appropriately weighted manner. However, temporal weighting can be considered as a challenging task, as the actual decay rate is actually influenced by manifold aspects.

1.4 Approach of the Thesis

The thesis considers the extraction, localization, and fusion of CVD in a holistic manner and presents solutions under consideration of the requirements of the previous section on the following basis. The CVD is collected by a fleet of common vehicles equipped with usual sensors, and transmitted to the back-end-side. At the back-end-side the fusion of the collectively acquired artifacts of the environment is performed. Subsequently, the back-end-side provides the fused artifacts as a-priori information to consecutive vehicles.

A central, novel idea of the thesis is to assign artifacts of the environment to be mapped to one of the three classes: *point-shaped landmarks*, *complex landmarks*, and *areal properties*. Each of those classes is treated by specific methods.

Point-shaped landmarks are artifacts with such a minor extent that their geographic representation by a point is sufficient. Examples are traffic signs, traffic lights, reflector posts, and other pole-like objects.

Complex landmarks are artifacts whose geometric shape is relevant, such as crossroads, roundabouts, traffic islands, pedestrian crossings, or sign gantries. The class of complex landmarks is considered as an extension of the class of point-shaped landmarks. This makes algorithms for the fusion of complex landmark data suitable for the fusion of point-shaped landmark data as well.

Areal properties denote properties which exist at every geographic location, such as temperatures, illuminations, frictions, or signal strengths.

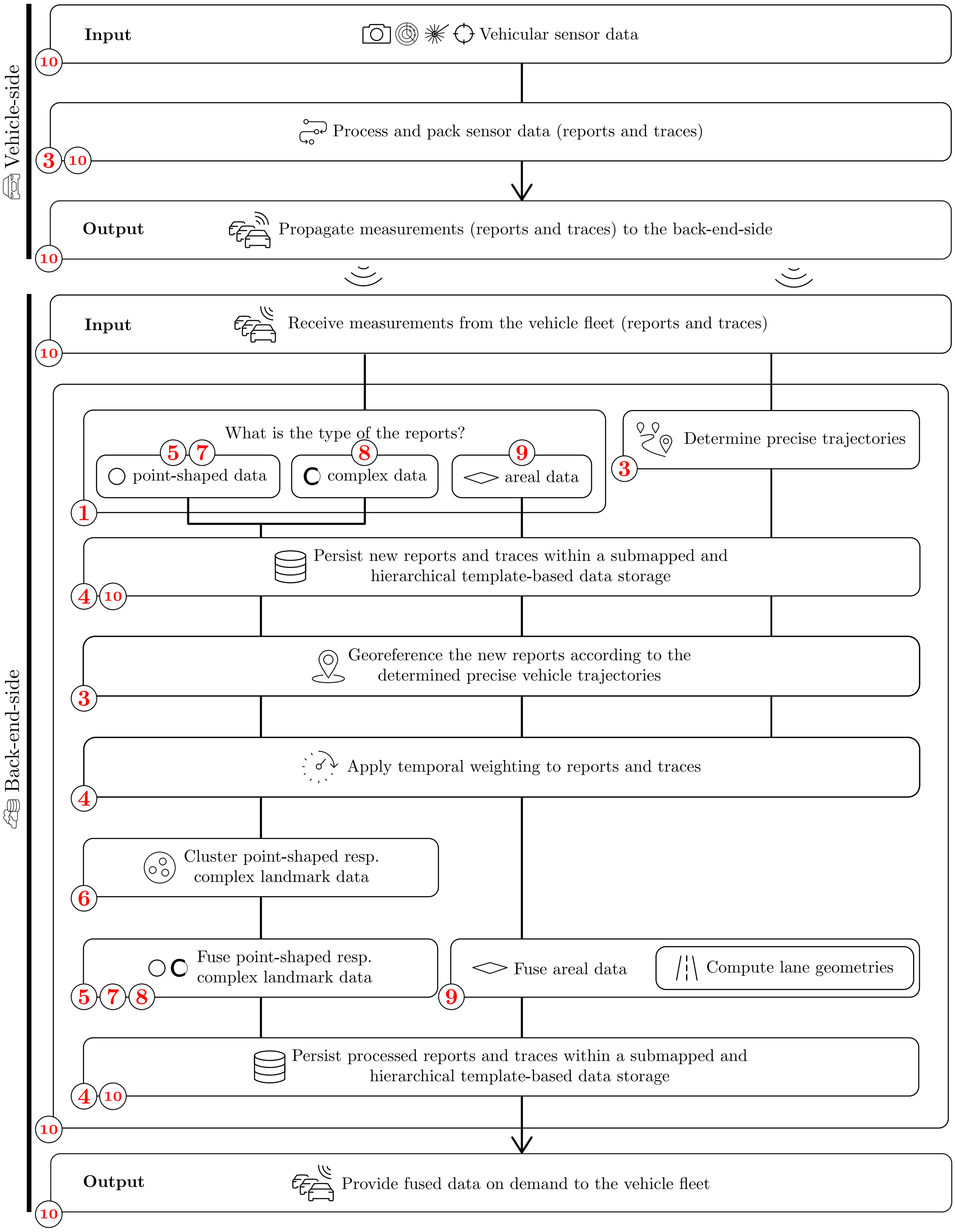

Vehicle-side
Input
Vehicular sensor data
10
Process and pack sensor data (reports and traces)
3 10
Output
Propagate measurements (reports and traces) to the back-end-side
10
Back-end-side
Input
Receive measurements from the vehicle fleet (reports and traces)
10
What is the type of the reports?
5 7
point-shaped data
8
complex data
9
areal data
1
Determine precise trajectories
3
Persist new reports and traces within a submapped and hierarchical template-based data storage
4 10
Georeference the new reports according to the determined precise vehicle trajectories
3
Apply temporal weighting to reports and traces
4
Cluster point-shaped resp. complex landmark data
6
Fuse point-shaped resp. complex landmark data
5 7 8
Fuse areal data
Compute lane geometries
9
Persist processed reports and traces within a submapped and hierarchical template-based data storage
4 10
10
Output
Provide fused data on demand to the vehicle fleet
10

Figure 1.3: The processing chain for the generalized extraction, localization, and fusion of CVD. The chapters corresponding to the individual steps of the processing chain, concerning both the vehicle- and back-end-side, are indicated by encircled ○ numbers.

The core of the approach is a hybrid *processing chain* located both at the vehicle- and the back-end-side (figure 1.3).

The processing chain at the vehicle-side is responsible for extracting and localizing artifacts of the environment, transforming them in a concise, parametric representation, and propagating them to the back-end-side for collaborative fusion.

Every measurement is related to an individual vehicle and has two components: a report and a trace. A *report* contains observations acquired by the vehicle sensors and a *trace* gives the corresponding vehicle trajectory. The data items are stochastic and usually represented by a single- or multi-dimensional value and variance or covariance information. Each report is associated with one of the three artifact classes.

The processing chain at the back-end-side combines the specific methods of the classes of artifacts with general steps shared by all methods. Its input consists of measurements sent by the vehicle fleet. The back-end-side first persists the received reports and traces according to their class in a hierarchical, template-based data storage. Subsequently, the observations in a report are *geographically referenced* with regard to the corresponding vehicle trajectory of the associated trace. Geographical referencing is performed by a known method of stochastic compounding which yields location estimates with covariances. The further steps are depending on whether the class of artifact is "point-shaped" or "complex" on the one hand, and "areal" on the other hand.

Data of point-shaped and complex landmarks, in the following denoted as "*point-shaped landmark data*" and "*complex landmark data*" for short, first undergo a step of *clustering*. The purpose of clustering is to group corresponding observations, i. e. observations which belong to the same point-shaped/complex landmark. Three well-known approaches to data association, ICNN[5], SCNN[6], and JCBB[7], are extended to clustering approaches by allowing dynamic map updates and are enhanced with respect to the automotive requirements. Afterwards, the fusion of point-shaped landmark data is performed by adapting several approaches from the fields of stochastic sensor data fusion and robotics: (F)CI[8], EKF[9]-based Online-SLAM[10], and BA[11]-based Full-SLAM. For complex landmark data EKF-based Online-SLAM and BA-based Full-SLAM are adapted. For this purpose, parametric representations of the landmark data are used, in particular the newly introduced ADG[12] for complex

[5] Individual Compatibility Nearest Neighbor (ICNN)
[6] Sequential Compatibility Nearest Neighbor (SCNN)
[7] Joint Compatibility Branch and Bound (JCBB)
[8] Covariance Intersection (CI)
[9] Extended Kalman Filter (EKF)
[10] Simultaneous Localization and Mapping (SLAM)
[11] Bundle Adjustment (BA)
[12] Attribute Dependency Graph (ADG)

landmark data. The ADG makes all components of information and their relation simultaneously available to the incorporated fusing procedure. The output of the fusing procedure are estimates with covariances of landmarks.

The fusion of "*areal data*", i. e. data of areal artifacts, consists in calculating an approximating function represented by function values at the vertices of a regular grid. The data function is represented as a weighted sum of elliptic basis functions located at the data samples. In addition to the data function, an uncertainty function is introduced which evaluates the uncertainty of the values of the data function. The FGT[13] is employed for efficient updating of the sums of basis functions in a batch mode. As an additional application, it is shown how the uncertainty function can be used to derive lane centerline geometries. In this case, the vehicle positions from the trace data are considered as areal data. The uncertainty function provides information about the density of the vehicle positions, and centerlines of lanes are assumed at locations of a high density.

The final step of the processing chain persists the point-shaped/complex landmark estimates and approximated areal properties, and makes them available on demand to the vehicle fleet.

1.5 Main Contributions

Within the thesis, extensions and adaptations of existing approaches, which are known to the state of the art, as well as novel solutions, where the research lacks established ways, are presented for the specific use case of the extraction, localization, and fusion of CVD. The main contributions are as follows.

An effective categorization of environment artifacts

By distinguishing environment artifacts to be mapped into just three big categories, i. e. point-shaped landmarks, complex landmarks, and areal properties, an effective generic treatment instead of an ineffective individual treatment of many sorts of artifacts has been made possible [117].

An efficient and generalized processing chain for dynamic mapping

Based on the categorization of environment artifacts, an efficient and generalized processing chain comprehending the extraction, localization, and fusion of CVD has been designed. For its proof-of-concept, a software architecture has been evolved and implemented [115, 116, 117]. The universal representation of CVD based on hierarchical templates can be regarded as its foundation [107, 112, 117].

[13] Fast Gauss Transform (FGT)

Employing generalized methods for efficient and quality-increasing processing

Two essential concepts, adaptive partitioning and temporal weighting, are employed (chapter 4). Those concepts allow for an efficient scalable processing of large-scale data sets by spatial and temporal locality. Additionally, locality can be used to achieve more precise linearizations and, therefore, more precise estimates of artifacts.

Adaptive partitioning is an extension of the method of global submapping, which allows to scale the processing of CVD in a computationally efficient and consistent manner for large-scale maps [115, 116].

A novel approach to temporal weighting allows determining appropriate temporal weights in a highly efficient and adaptive manner by means of preaggregation [109, 115, 116, 117]. Temporal weights are assigned to the current and past observations and the corresponding trajectory information. The reason is that the road network and information about its current state are exposed to environmental dynamics which decays its relevance for the future. This fact is taken into account by the weights which decay with increasing history of observations and which specify the amount to which past observations are considered at the fusion step. The weights are efficiently determined from a circular-buffer-based data structure which stores condensed information about the past and which is permanently updated.

A method for improving the precision of vehicle localization

An adaptation of PPP[14]-based GNSS vehicle localization by means of postprocessing is suggested for the use case of precisely georeferencing CVD [115, 116, 117] (chapter 3). The novelty is that the vehicles are required to propagate GNSS *raw data* to the back-end-side. This way, in fact, the computational effort is shifted from the vehicle to the back-end-side. However, significantly more precise localization results can be gained. By providing more accurately referenced CVD to the actual fusion algorithms, their convergence rate and accuracy is positively affected.

A novel extension of JCBB for data clustering

A novel extension of JCBB from a pure data association to a clustering algorithm by utilizing a generalization of BB[15], the so-called GBB[16], in combination with the BFS[17] search strategy is introduced [113, 116]. As the number of hypotheses does principally evolve exponentially with every new observation, a novel, multi-criteria cost-function for limiting the search to the η most promising hypotheses at each level

[14] Precise Point Positioning (PPP)
[15] Branch and Bound (BB)
[16] General Branch and Bound (GBB)
[17] Breadth-First Search (BFS)

of the GBB tree is suggested. This way the algorithm can be also deployed under computational resource limitations and/or for real-time applications.

An approach to the fusion of point-shaped landmark observations

The applicability of several state-of-the-art approaches to the fusion of point-shaped landmark data is investigated (chapter 7). For this purpose, own derivations of the suggested algorithms are provided, which aim to fulfill the imposed requirements of the large-scale point-shaped CVD fusion. The reader may also refer to the previous work of the author [104, 114], which focuses purely on the robust determination of the longitudinal position of point-shaped landmarks by utilizing map-matching and supervised learning. This previous work can be regarded as influential concerning the decisions made within this thesis.

An approach to the fusion of complex landmark observations

A novel approach to the fusion of complex landmark observations is presented [108, 110, 111, 112, 116] (chapter 8). Complex landmarks are described in a novel and highly generic manner by means of an ADG. This way, the fusion can be easily adapted to new types of complex landmarks. Furthermore, the algorithm is able to fuse multiple types of point-shaped/complex landmarks in one single pass and is, therefore, able to exploit tight correlations between them.

An efficient double-staged approach to the fusion of areal data

A novel, double-staged approach to the incremental fusion of areal data is presented [105, 115], which is tailored to suit the specific requirements of the large-scale CVD fusion (chapter 9). Additionally, it is evolved, in the case of a batch insert or update, how the areal data fusion can be sped up by more than ten times by means of the FGT. For this purpose, a state-of-the-art variant of the FGT, the so-called FIGTree[18] algorithm, is extended in a novel way to simultaneously handle distinct multivariate Gaussian distributions. Furthermore, a novel approach for the derivation of lane centerline geometries from the uncertainty function of the double-staged areal data fusion is proposed [106, 115].

1.6 Organization of the Thesis

Chapter 2 gives a survey of the state-of-the-art of CVD. Furthermore, distinct use cases/benefits and limitations/challenges known to the state of the art concerning CVD are pointed out, cross-references concerning the topics and aspects tackled within the thesis are established.

[18] Fast Improved Gauss Transform with Tree Data Structure (FIGTree)

Chapter 3 addresses aspects of localization of vehicles and CVD: precise localization of vehicles, modeling of vehicle motion for SLAM, and stochastic compounding of relative measurements for geographic referencing of sensor data.

Chapter 4 is devoted to adaptive submapping and temporal weighting.

Chapter 5 introduces the parametric description of point-shaped landmarks. It gives an overview of the approach to the fusion of point-shaped landmark data, whose details will be presented in the following two chapters.

Chapter 6 is concerned with data association respectively clustering. It presents measures of correspondence, extended versions of the existing approaches "Individual Compatibility Nearest Neighbor (ICNN)", "Sequential Compatibility Nearest Neighbor (SCNN)", and "Joint Compatibility Branch & Bound (JCBB)", and the results of experimental evaluations.

Chapter 7 elaborates the fusion of point-shaped landmark data based on the results of the clustering. It comprehends the fusion via (F)CI, fusion via EKF-based Online-SLAM, and fusion via BA-based Full-SLAM, and related evaluations.

Chapter 8 is devoted to the fusion of complex landmark data. It defines the parametric description of complex landmarks, presents approaches of fusion via an extended formulation of EKF-based Online-SLAM and fusion via an extended formulation of BA-based Full-SLAM and reports on an evaluation of BA-based Full-SLAM for two scenarios.

Chapter 9 presents the double-staged areal data fusion and the approach to derivation of lane centerline geometries.

Chapter 10 elaborates the instantiation of the processing chain for CVD at both the vehicle- and back-end-side, and introduces the hierarchical template-based data storage.

Chapter 11 briefly summarizes the central aspects of the thesis. Furthermore, recommendations for extensions thereof are provided.

1.7 Publications

Contents of the thesis have been published in several conference papers and patents.

The paper with the title "Large-Scale Fusion of Collective, Areal Vehicle Data" at the *IEEE International Conference on Multisensor Fusion and Integration (MFI) 2015* [115] presents the approach of fusion of areal data (chapter 9) and the PPP-based GNSS vehicle localization by means of postprocessing (chapter 3.2.2).

The paper “Parametric Fusion of Complex Landmark Observations Present within the Road Network by Utilizing Bundle-Adjustment-based Full-SLAM” at the *International Conference on Information Fusion (FUSION) 2016* [116] presents the fusion of complex landmark data by BA-based Full-SLAM (chapter 8). It comprehends the extension of global submapping (chapter 4.2), the approach to determining appropriate temporal weights by means of preaggregation (chapter 4.3), methods of data association (chapter 6), and PPP-based GNSS vehicle localization by means of postprocessing (chapter 3.2.2).

The paper “Selected Aspects Important from an Applied Point of View to the Fusion of Collective Vehicle Data” at the *IEEE International Conference on Multisensor Fusion and Integration (MFI) 2016* [117] gives a survey of different aspects and contributions of the thesis: the generalized pipeline for the extraction, localization, and fusion of distinct types of CVD (chapters 1.4 and 10), the PPP-based GNSS vehicle localization by means of postprocessing (chapter 3.2.2), the approach for determining appropriate temporal weights by means of preaggregation (chapter 4.3), and the concept of universal storage of CVD based on hierarchical templates (chapter 10.4).

The results presented in those publications and in the thesis are original contributions by the author. The co-authors supported in their role of supervisors, to the usual extent, by discussions of specific technical aspects, and gave hints with respect to the presentation of the results.

Furthermore, several contributions of the thesis have been transferred into patents of the author:

DE102015000394: Double-staged approach to the fusion of areal data (chapter 9).

DE102015000399: Derivation of lane centerline geometries from the intermediate results of the double-staged areal data fusion (chapter 9).

CN107077784, DE102015001247, EP3158295, US9983307, WO2016120001:
Approach to the fusion of complex landmark observations (chapter 8).

DE102015001193: Adaptive temporal weighting by preaggregation (chapter 4.3).

DE102015001194, WO2016120004: Universal storage of CVD based on hierarchical templates (chapter 10.4).

Finally, the master’s thesis of the author entitled “Dynamic georeferencing of location-related fleet data for smart map services” [104] and the related patent DE102013009856 [114] have to be mentioned. This work focuses purely on the robust determination of the longitudinal position of point-shaped landmarks by map-matching and supervised learning. It can be regarded as influential with respect to decisions made within this thesis.

2 State of the Art

In this chapter, the origins of the acquisition of CVD[1] are briefly summarized. Potential use cases/benefits and limitations/challenges are illuminated. Furthermore, topics and aspects treated in this thesis are cross-referenced and motivated.

Contents

2.1 Introduction

As stated by Bishop [5] common vehicles, such as private cars, taxis, and lorries, can be valuable sources of information concerning the road network and its current state. It is usually sufficient that a small percentage of these vehicles, according to Huber et al. [57] approximately 1-5 %, has to be equipped with the functionality of gathering and propagating sensor data, the so-called CVD, to a central instance. The percentage required is known to be dependent on multiple factors, such as the type of the data acquired, the frequenting of the road, and the incorporated data aggregation algorithms (section 2.3.4). The acquisition of CVD has a long history in Europe [5]. In Germany, for example, the Gesellschaft für Verkehrsdaten mbH is acquiring and processing CVD since 1997 [57].

For the purpose of gathering CVD, it is highly beneficial to equip modern vehicles with diverse sensors. The collectively acquired data can be subsequently fused and provided to other vehicles as a-priori information. Such data is not acquirable for large-scale by conventional measuring methods, such as camera or induction loop

[1] Collective Vehicle Data (CVD)

based ones [12, 73]. The fused data can be incorporated into navigation systems, next generation comfort, assistance, or piloted driving functions [12, 57]. Nevertheless, manifold technical challenges have to be tackled for the large-scale utilization of CVD (section 2.3).

As alternatives to the gathering of information about the road network by common vehicles, satellites and mobile phones are exemplified by Bishop [5]. Apple and Google are prominent examples of gathering information by means of mobile phones. The acquired data is processed and, for instance, incorporated into their map services. However, current vehicles are able to perceive their environments usually more comprehensively than mobile phones. This is due to the extensive sensor set of today's cars. The acquisition of comparable sensor data by satellites has been i. a. evaluated by the European Space Agency [5]. The shadowing by large buildings has been identified as the major restriction of satellite-based systems, compared to the acquisition of CVD by common vehicles. For those reasons, the acquisition of information about the road network by common vehicles can be regarded as the (currently) most favorable way.

2.2 Use Cases and Benefits of CVD

In the past, diverse projects of acquisition and processing of CVD have been initiated. The acquisition of CVD by common vehicles is a cost-effective way of gathering information about the road network when comparing it to current state-of-the-art camera or induction loop based monitoring systems [73]. Furthermore, CVD allows for principally deriving estimates of large-scale, high-accuracy, timeliness, and fine-granularity [28]. In the following, a survey of potential use cases for the utilization of CVD and the related benefits is given.

Lorkowski et al. [74] have evaluated the elicitation of CVD by taxis, which was motivated by their above-average mileage. As stated by Reinthaler et al. [98] and Brockfeld et al. [12] experiments have revealed that CVD acquired by taxis is representative for privately owned vehicles. After filtering the provided data, approximately 85 % of the data was identified to be suitable for the subsequent processing [13, 74]. According to Brockfeld et al. traffic congestions could be identified based upon this data with both a high reliability and a low latency of just a few minutes. However, real-time applications, such as controlling traffic lights, were classified as not achievable in this way. This was mainly due to bandwidth limitations of the cellular network and the, therefore, intentionally reduced data acquisition frequencies (section 2.3.2).

Furthermore, Lorkowski et al. [74] made a first attempt to derive digital road data solely from CVD. It is assessed that the GNSS[2] self-localization of common vehicles is principally not accurate enough for this type of application, due to commercial constraints of the utilized hardware. However, by combining localization data acquired by multiple vehicles, the inaccuracies induced by the utilized sensors can be (partially) compensated, as emphasized by the authors. Within this thesis, the aspect of deriving digital road data is also addressed in detail (chapter 7, chapter 8, and section 9.10.4). The challenge of gaining a precise vehicle localization in a commercially viable manner is tackled by means of postprocessing GNSS raw data at the back-end-side (section 3.2).

De Fabritiis et al. [28] have investigated the short-time prediction, ranging from 15 min to 30 min, of vehicle travel speeds by applying ANN[3]s to CVD that was gathered by approximately 600 000 privately owned vehicles in Italy. The CVD was propagated with a frequency of 12 min by the vehicles to the back-end-side. An accuracy of 90 % for short-time predictions of vehicle travel speeds was achieved.

It can be noticed that the past research on CVD has been mainly focused on improving route planning by identifying congested traffic and, in general, by determining current and predicting future vehicle travel speeds. Additionally, first attempts of deriving road geometries can be observed. However, CVD has a much greater potential because it can provide maps with multiple layers of attributes that have a fast temporal decay rate, such as slippery or dirty road segments, potholes, lost objects, or broken down vehicles. Furthermore, common vehicles can be as well employed to map the more static parts of the road network, such as lane centerline geometries, traffic lights, traffic signs, crosswalks, stop lines, roundabouts, crossroads, speed bumps, roadside structures, and curbstones, to just name a few. Those attributes can support next generation driver assistance and/or piloted driving functions. Furthermore, current improvements of the exteroperceptive sensors and GNSS-based localization existing in common vehicles allow for gaining higher sensing accuracies than in the past. The huge, redundant amount of data acquired by common vehicles allows to alleviate sensor errors so that results comparable with that acquired by high-grade sensors, as used e. g. by the well-known map providers, can be (nearly) gained. Therefore, the current trend of forming alliances between automobile manufacturers and map providers can be regarded as highly useful [1, 14].

[2] Global Navigation Satellite System (GNSS)

[3] Artificial Neural Network (ANN)

2.3 Limitations and Challenges of CVD

This section illuminates the current limitations and challenges of the elicitation of CVD. Especially the topics of precise vehicle localization (section 2.3.1), required bandwidths (section 2.3.2) and processing capabilities (section 2.3.3), penetration rates (section 2.3.4), and, furthermore, privacy and security (section 2.3.5) objections are examined.

2.3.1 Localization

A usual approach to alleviate the inaccuracies of GNSS-based localization (section 3.2) in the context of CVD is map matching [13, 70, 98]. Map matching allows to project vehicle trajectories to the most probable corresponding paths of the digital road network, according to an error measure. This raises multiple problems. For instance, additional errors may be introduced to the vehicle trajectories by matching with possibly inaccurate road geometries. Furthermore, missing road geometries within the digital road map can lead to wrong associations, e. g. to nearby roads.

The challenges of dealing with inaccurate GNSS localization are tackled in this thesis by postprocessing the raw GNSS data at the back-end-side under the incorporation of correction data. This way, more accurate vehicle trajectories can be gained, and the problematic map matching can be avoided (section 3.2.2).

2.3.2 Bandwidth

Bandwidth limitations of the cellular network on the one hand, and large amounts of acquired sensor data on the other hand, are a crucial challenge for the transmission of CVD to a central instance and the reception after its processing [2].

The cellular network is steadily extending, so that blank spots become less and the technological state of the art is matched. However, by now not all cellular base stations are providing the latest, fourth generation cellular technology, although the next, fifth generation can be considered as already mature. It has to be taken into account that areas of lower bandwidths will probably never completely disappear because the latest cellular technology is not always available. Furthermore, areas *without* any cellular network still exist, what typically requires CVD-enabled vehicles to pause the data transmission and retain the data.

It is known that the bandwidth required for CVD-enabled vehicles is strongly depending on the actual data transmitted and the frequency thereof [92]. To cover fast vehicle velocities, as for instance observed on German autobahns, a very sparse localization sampling rate, e. g. every few minutes [92], is considered as not being sufficient for the deduction of high-quality estimates. Instead a localization sampling rate of 10 Hz is proposed (section 3.2). This sampling rate implies approximately

0.42 MB/h if every position of the vehicle trajectory is represented by three coordinates of 4 Bytes each. For the case of postprocessing of GNSS raw data, much larger amounts of data have to be transmitted to the back-end-side. The uncompressed transmission of GNSS raw data approximately leads to 25.5 MB/h at a 10 Hz sampling rate. However, by means of dedicated compression algorithms, the required bandwidth can be considerably reduced to 2.3 MB/h (section 3.2.2).

Furthermore, it can be useful to support GNSS-based localization in shadowed areas, such as urban canyons, tunnels, by means of egomotion estimates. The required bandwidth for its transmission should be approximately equal to that for the transmission of GNSS localization data. This means that, depending on the demanded quality, the required bandwidth does range from 0.42 MB/h to 25.5 MB/h. Under the assumption that a similar compression ratio as for the GNSS raw data can be achieved by e. g. delta-based compression, between 0.5 MB/h and 5 MB/h can be estimated for the CVD that is only related to the vehicle localization.

Additionally, at least an equal amount of bandwidth is probably required for the propagation of features extracted from the on-board sensors, such as cameras, radars, or lidars. This way, the overall, estimated bandwidth ranges from 1 MB/h to 10 MB/h, depending on the intended use cases and the demanded quality.

Assuming an average required penetration rate of 2 % (section 2.3.4), 1 mio. CVD-enabled vehicles in Germany would lead to approximately 0.95 to 9.5 TB/h of CVD that needs to be propagated and processed. By multiplying those numbers by 24 h, 365 d, and considering all vehicles worldwide, one is faced with an impressive amount of data to be transmitted and processed each year.

For those reasons, approaches to reducing the amount of CVD propagated to the back-end-side, not only by means of data compression, are required. Kerner et al. [66] propose just to propagate CVD which is significantly different from the data at the back-end-side. However, a drawback of this approach is that no amplification by means of similar data is achieved. Only deviating data affect the accumulation, what is actually considered as undesirable. For this reason, Ayala et al. [2] propose a more sophisticated approach which introduces an equal probability for every vehicle to propagate CVD. The actual probability of data transmission is derived by means of the central limit theorem, so that a specified confidence is achieved. This way, the aforementioned drawbacks of the approach presented by Kerner et al. [66] are diminished, the amount of transmitted CVD is reduced, and the quality of the postponed data fusion is assured.

2.3.3 Processing

The real-time processing of CVD is a challenging task, because huge amounts of data (section 2.3.2) need to be processed in a highly robust manner within a restricted period. Furthermore, it is aggravated by the fact that the majority of the powerful approaches to sensor data fusion is computationally time-consuming [71]. Hence, the scalability of the method of processing, e. g. by means of data parallel programming or appropriate index data structures/storage solutions, is a crucial aspect [71]. Due to those reasons, all CVD-processing algorithms presented in this thesis are evaluated with the main focus on robustness and large-scale feasibility (section 1.3).

2.3.4 Penetration Rates

Within the context of CVD, "penetration rate" denotes the amount of CVD-enabled vehicles that is required to fulfill a given objective, such as gaining a certain timeliness or precision of the CVD fusion. For this purpose, several estimates have been presented by the research community. To briefly sum up these estimates, the required penetration rates usually vary between between 1 % and 5 % for highways, and between 2 % and 10 % for urban roads [131]. However, it shall be stressed that a general estimate is *not* appropriate, because the actually required penetration rate is highly dependent on multiple factors, such as the algorithms utilized for sensor data fusion, the quality of the acquired sensor data, and the desired confidence level/timeliness.

Kerner et al. [66] suggest a penetration rate between 1.5 % and 2 % for the identification of traffic congestions. A penetration rate of 1.5 % leads to a probability of 65 % that a traffic congestions is appropriately detected, and a penetration rate of 2 % to a probability of 85 %.

As commonly supposed within the research community, those rates could be easily fulfilled if multiple car manufacturers would cooperate [10, 66]. Therefore, it has been an important step for the large-scale acquisition and fusion of CVD that the German premium car manufacturers decided to form an alliance to tackle this challenge collaboratively [1].

2.3.5 Privacy and Security

The collection of data is widely seen as critical in public perception [33]. While the research community has mainly focused on the technical challenges of acquiring and processing CVD, both privacy and security issues have been mostly neglected [93]. Usually CVD-enabled vehicles are authorized by the back-end-side by means of a *unique* identifier. While this approach reduces the risk of fraud, it neglects the privacy of customers. Therefore, approaches for anonymizing customer data, while simultaneously minimizing the risk of exploitations, are strongly desirable.

For this purpose, Eichler and Jeske [61] suggest a ticket-based system. The so-called "Get-Dispenser protocol" still requires that a vehicle authenticates itself periodically at the back-end-side. However, the data is anonymized by means of tickets that have a limited temporal validity. Although the proposed protocol provides good privacy, some corner cases have to be additionally considered in practice. For example, in sparsely inhabited areas it might be possible to directly associate vehicle trajectories with certain persons or families.

3
Localization

This chapter addresses several aspects of geographical localization of vehicles and CVD[1] which are in particular relevant for the quality of the mapping achieved and its assessability: precise localization of vehicles, modeling of vehicle motion for SLAM[2], and stochastic compounding of relative measurements for geographic referencing of sensor data. Suitable methods are selected and analyzed with respect to their precision.

Contents

3.1 Introduction

Geography-related data collected by a vehicle is usually referring to an individual vehicle reference frame. However, a precondition for the fusion approaches presented in the remainder of the thesis is geographical referencing of the CVD to one global frame. Global geographical referencing is achieved by combining the location data provided by the traces of the vehicle trajectories, and of the location data being part of the observations captured by the vehicle sensors.

An important influential factor on the quality of the fusion results is the precision of vehicle localization. Vehicles are usually localized by a global navigation satellite system (GNSS). However, multiple error sources are diminishing the localization quality of such systems. An approach to GNSS[3]-based vehicle localization with

[1] Collective Vehicle Data (CVD)
[2] Simultaneous Localization and Mapping (SLAM)
[3] Global Navigation Satellite System (GNSS)

Error Source	Impact on GPS	Impact on DGPS
Ephemerides	2.1 m	0.1 m
Satellite Clocks	2.1 m	0.1 m
Ionosphere	4.0 m	0.2 m
Troposphere	0.7 m	0.2 m
Multipath	1.4 m	1.4 m
Receiver	0.5 m	0.5 m
Σ	10.8 m	2.5 m

Table 3.1: Typical error sources and impacts of (D)GPS-based vehicle localization by means of a single-frequency GNSS receiver [15, 138].

high precision, adapted to the processing chain of this work, will be proposed in section 3.2. It employs state-of-the-art technology and makes use of the application context which does not necessarily require online-processing of the captured data.

A helpful information to cope with measurement errors is the restricted kind of motion a vehicle can perform. Approaches like SLAM, which will be employed for the fusion of point-shaped/complex landmark data, incorporate a vehicle motion model. Several motion models have been proposed in the past. The pros and cons of utilizing a linear versus a circular vehicle motion model will be elaborated in detail in section 3.3.

Another aspect of geographical referencing of CVD is the accumulation of the measurement errors of vehicle localization when referring sensed observations to the global frame by compounding the vehicle location and locations of observations which are given relatively to the vehicle. This requires a method for estimating the location of observations relatively to the global frame including an assessment of their quality. An existing method of stochastic compounding of a sequence of relationships between objects, which yields means and covariances of geographically referenced locations, will be adopted in section 3.4.

3.2 Precise Vehicle Localization

Precise vehicle localization and capturing of vehicle trajectories are crucial for the accurate referencing of CVD and its subsequent fusion [40, 69]. Vehicles are commonly localized by satellite-based methods. However, a precise satellite-based vehicle localization is known to be challenging since multiple error sources are reducing the localization quality. The ionosphere, imprecise ephemerides, and drifting satellite clocks, which principally impose 4.0 m respectively 2.1 m of error *each*, can be regarded as the major error sources (table 3.1). Additionally, multipath effects by conceiving reflected and, therefore, delayed signals can lead, especially in (urban) canyons, to a further localization error of approximately 1.4 m. When summing up these error sources, and including the receiver's imprecision, an overall error of 10.8 m does result (table 3.1). For the use case of collaboratively building highly precise maps, this is regarded as obviously insufficient.

By means of DGPS many of these errors, such as imprecise ephemerides, drifting satellite clocks, and ionospheric/tropospheric errors, can be alleviated so that an overall localization error of approximately 2.5 m results. However, DGPS is grounded upon correction data which is usually derived from a network of terrestrial reference stations, and that needs to be propagated to the vehicle-side prior to its processing. For the large-scale propagation of correction data, the cellular network is predestinated. However, the cellular network is well-known to exhibit white spots, preventing a gapless precise vehicle localization by means of DGPS.

Therefore, current research has been mainly focusing on the suitability of other channels for providing correction data to the vehicle-side. For instance, the data channel of DAB[4] has been identified as a promising alternative [78]. However, the DAB network is affected by similar coverage restrictions as the cellular one. An ideal solution is expected to work world-wide, to provide a (nearly) geodetic-grade localization precision, and to be commercially viable for common vehicles.

The approach to overcome those issues in the processing chain of this thesis is offline-processing. The GNSS raw data is propagated by the vehicles to the back-end-side as part of the acquired CVD. At the back-end-side, the received GNSS raw data is postprocessed under consideration of GNSS correction data. This approach features two crucial benefits. Firstly, cellular white spots affecting the online-availability of correction data can now be regarded as unproblematic. Secondly, by means of postprocessing, a more precise localization can be achieved, as the full trajectory can be inspected at once, and ambiguities can be, this way, more appropriately resolved.

The rest of this section is organized as follows. First, the principle of multilateration is recalled (section 3.2.1). Subsequently, the offline approach favored in the thesis is presented (section 3.2.2). Then the offline approach is experimentally evaluated and compared with the online solution of immediately using the locations obtained by multilateration on the data provided by the GNSS receiver (section 3.2.3). Finally, the insights gained are discussed and suggestions for further optimizations are made (section 3.2.4).

3.2.1 Multilateration

Multilateration yields coordinates of location from measured signals. A vehicle trajectory is immediately available online by applying multilateration to a sequence of measurements taken with an appropriate sampling rate.

[4] Digital Audio Broadcasting (DAB)

Multilateration determines the distances between a GNSS receiver and several GNSS satellites, the so-called pseudoranges $\varphi_i(\boldsymbol{r}, \Delta t)$, $i = 1, \dots n$, n the number of satellites, by

$$\begin{aligned} \varphi_i(\boldsymbol{r}, \Delta t) &= \|\boldsymbol{s}_i - \boldsymbol{r}\|_2 \; + \; c \cdot \Delta t \\ &\overset{\text{3D}}{=} \underbrace{\sqrt{(s_{i,x} - r_x)^2 + (s_{i,y} - r_y)^2 + (s_{i,z} - r_z)^2}}_{\varphi_i(\boldsymbol{r})} + c \cdot \Delta t. \end{aligned} \tag{3.1}$$

$\boldsymbol{s}_i$ denotes the i-th satellite position, $\boldsymbol{r}$ the receiver's position, c the speed of light, and Δt the difference between the satellite's and the receiver's clocks [65].

Usually, the non-linear pseudorange equation (equation 3.1) is linearized via 1st-order Taylor series (appendix A.3) for solving it in an efficient manner, such as by means of QR[5] (section 7.4.5):

$$\begin{aligned} T_1 \; \varphi_i(\boldsymbol{r}) + c \cdot \Delta t &= \underbrace{\varphi_i(\boldsymbol{r}_0) + \frac{\partial \varphi_i(\boldsymbol{r})}{\partial \boldsymbol{r}} \cdot (\boldsymbol{r} - \boldsymbol{r}_0)}_{T_1 \; \varphi_i(\boldsymbol{r})} + c \cdot \Delta t \\ &\overset{\text{3D}}{=} \varphi_i(\boldsymbol{r}_0) + \frac{\partial \varphi_i(\boldsymbol{r})}{\partial r_x} \cdot (r_x - r_{0,x}) + \frac{\partial \varphi_i(\boldsymbol{r})}{\partial r_y} \cdot (r_y - r_{0,y}) + \\ &\quad \frac{\partial \varphi_i(\boldsymbol{r})}{\partial r_z} \cdot (r_z - r_{0,z}) + c \cdot \Delta t \\ &= \varphi_i(\boldsymbol{r}_0) + \frac{r_{0,x} - s_{i,x}}{\varphi_i(r_{0,x})} \cdot (r_x - r_{0,x}) + \frac{r_{0,y} - s_{i,y}}{\varphi_i(r_{0,y})} \cdot (r_y - r_{0,y}) + \\ &\quad \frac{r_{0,z} - s_{i,z}}{\varphi_i(r_{0,z})} \cdot (r_z - r_{0,z}) + c \cdot \Delta t. \end{aligned} \tag{3.2}$$

If the position $\boldsymbol{r}$ of the receiver has to be determined in three dimensions, at least four satellite observations $\boldsymbol{s}_i$ are required, since additionally the time-difference between the receiver's and the satellite's clocks Δt needs to be resolved. The resulting system

[5] QR Factorization (QR)

Type		Accuracy	Latency	Sample rate
Broadcast	orbits clocks	100 cm 5 ns	real time	1 d
Ultra-Rapid	orbits clocks	3 cm 150 ps	3 h - 9 h	15 min
Rapid	orbits clocks	2.5 cm 75 ps	17 h - 41 h	15 min 5 min
Final	orbits clocks	2.5 cm 75 ps	12 d - 18 d	15 min 30 s

Table 3.2: Different orbit and clock products that are provided by the IGS [58] are opposed to the broadcast ones according to their accuracies, latencies, and sample rates.

of equations for the receiver's absolute position $\boldsymbol{r}$ in the three-dimensional case is then given by

$$\begin{pmatrix} \frac{r_{0,x}-s_{1,x}}{\varphi_1(\boldsymbol{r}_0)} & \frac{r_{0,y}-s_{1,y}}{\varphi_1(\boldsymbol{r}_0)} & \frac{r_{0,z}-s_{1,z}}{\varphi_1(\boldsymbol{r}_0)} & c \\ \frac{r_{0,x}-s_{2,x}}{\varphi_2(\boldsymbol{r}_0)} & \frac{r_{0,y}-s_{2,y}}{\varphi_2(\boldsymbol{r}_0)} & \frac{r_{0,z}-s_{2,z}}{\varphi_2(\boldsymbol{r}_0)} & c \\ \frac{r_{0,x}-s_{3,x}}{\varphi_3(\boldsymbol{r}_0)} & \frac{r_{0,y}-s_{3,y}}{\varphi_3(\boldsymbol{r}_0)} & \frac{r_{0,z}-s_{3,z}}{\varphi_3(\boldsymbol{r}_0)} & c \\ \frac{r_{0,x}-s_{4,x}}{\varphi_4(\boldsymbol{r}_0)} & \frac{r_{0,y}-s_{4,y}}{\varphi_4(\boldsymbol{r}_0)} & \frac{r_{0,z}-s_{4,z}}{\varphi_4(\boldsymbol{r}_0)} & c \end{pmatrix} \cdot \begin{pmatrix} r_x - r_{0,x} \\ r_y - r_{0,y} \\ r_z - r_{0,z} \\ \Delta t \end{pmatrix} = \begin{pmatrix} \varphi_1^{\text{meas.}} - \varphi_1(\boldsymbol{r}_0) \\ \varphi_2^{\text{meas.}} - \varphi_2(\boldsymbol{r}_0) \\ \varphi_3^{\text{meas.}} - \varphi_3(\boldsymbol{r}_0) \\ \varphi_4^{\text{meas.}} - \varphi_4(\boldsymbol{r}_0) \end{pmatrix}. \quad (3.3)$$

$\varphi_i^{\text{meas.}}$ denote the by the GNSS receiver measured pseudoranges. The system of equations is allowed to be overdetermined. This is relevant in the case when more than four satellites $\boldsymbol{s}_i$ are simultaneously observed. A solution to the linearized system of equations can be determined in an efficient and robust manner, for instance, by means of QR. For further details, see Kaplan and Hegarty [65].

In practice, the online solution to GNSS-based localization is robustified by incorporating a limited number of former measurements. For this purpose, filtering techniques, such as (E)KF[6]s, are commonly utilized. One component of such methods is the motion model. Aspects of modeling vehicle motions will be the topic of section 3.3.

[6] Kalman Filter (KF)

3.2.2 Offline Processing

As already mentioned, the first step of the approach of offline processing is the transmission of the GNSS raw data from the vehicles to the back-end-side as a part of the acquired CVD. At the back-end-side, the received GNSS raw data is postprocessed under consideration of GNSS correction data. Figure 3.1 provides a top-level view of the GNSS offline processing approach. Its core is the postprocessing procedure with the following interfaces:

Solution 3.1: GNSS Postprocessing

Input: GNSS raw data (pseudoranges, doppler, signal strengths, ...), correction data (ephemerides, clocks, ionosphere, troposphere, ...), and optionally egomotion data.

Output: A postprocessed trajectory estimate.

The postprocessing employs the so-called PPP[7] which emerged in the recent decade for the precise GNSS-based localization, and which is regarded as highly suitable for postprocessing GNSS raw data [54, 55, 138]. PPP-based GNSS localization simultaneously considers GNSS code and phase measurements, and does utilize precise ephemerides and clocks instead of the ones broadcasted by the satellites [65]. Both adjustments are intended for significantly improving the accuracy and robustness of the localization.

Precise ephemerides and clocks can be, for instance, obtained from the IGS[8] which derives them from a globally distributed network of reference stations [56, 58, 138]. By utilizing precise satellite ephemerides and clocks instead of the broadcast ones, the overall localization error can be reduced by approximately 4.2 m (table 3.1). An overview of different products provided by the IGS, and the corresponding accuracies and latencies, is given in table 3.2. *Un*predicted ultra-rapid correction data is provided by the IGS with a latency of approximately 3-9 h, and final correction data after 12-18 d.

The postprocessing determines a stream of system states based on sub-sequences of the raw data stream and the correction data, and optionally supported by egomotion data to compensate for potential outages and errors. It is performed by filtering on sub-sequences of the raw data stream and correction data [55, 123]. Filtering can be applied in a forward, backward, and combined manner.

[7] Precise Point Positioning (PPP)
[8] International GNSS Service (IGS)

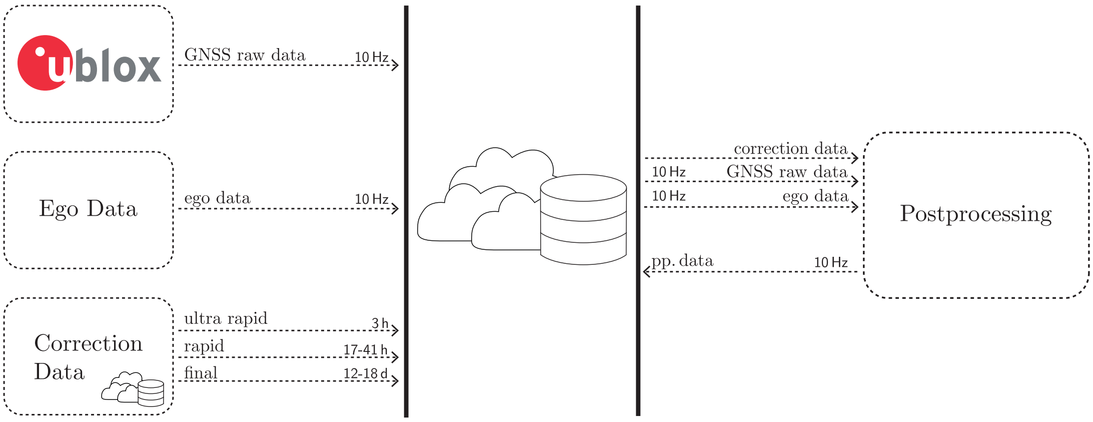


Figure 3.1: For the precise referencing of CVD, GNSS raw data is transmitted as part of the CVD to the back-end-side. At the back-end-side, the received data is processed under consideration of correction data. The postprocessing of the GNSS raw data can be supported by egomotion data to compensate for potential GNSS outages and errors.

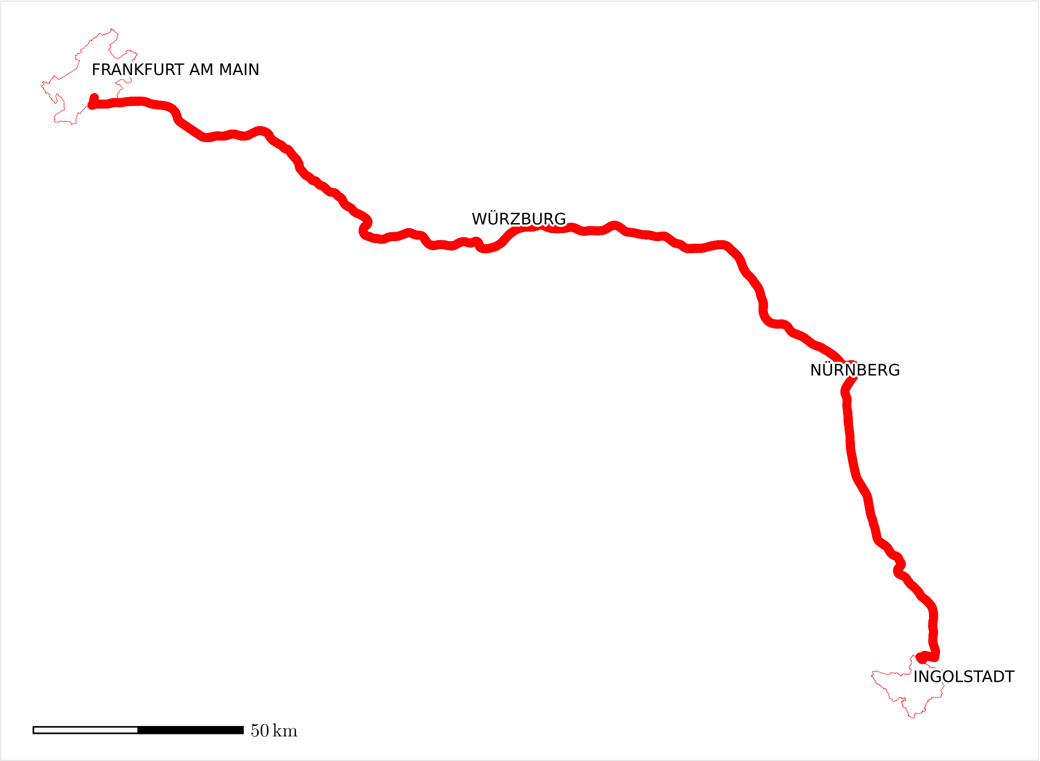


Figure 3.2: The approximately 316 km-long route from Ingolstadt to Frankfurt that has been utilized for evaluating the accuracy of the PPP-based GNSS postprocessing.

	PPP	SBAS	Proc.-Dir.	PPP-AR	Average Dev.	Std. Dev.
Offline	yes	yes	combined	no	1.301 m	0.963 m
	yes	yes	forward	no	1.301 m	0.974 m
	yes	yes	forward	yes, 1-it.	1.301 m	0.974 m
	yes	yes	backward	no	1.324 m	0.983 m
	yes	yes	backward	yes, 5-it.	1.327 m	0.981 m
	yes	no	forward	no	1.374 m	0.981 m
	yes	no	combined	no	1.376 m	0.981 m
	yes	no	backward	no	1.380 m	0.984 m
Online	no	yes	–	–	1.852 m	2.281 m

Table 3.3: Accuracies of distinct kinematic PPP-configurations that have been gained by GNSS postprocessing, using ionospheric correction data provided by the SBAS-satellites and processing the trajectory in a forward/backward/combined manner, and with(out) using PPP-AR. For this purpose, the RTKLIB and final products provided by the IGS were utilized. For comparison, the online solution is opposed.

3.2.3 Evaluation

In the following, a PPP-based offline solution to GNSS-based localization is evaluated and opposed to an GNSS-based online-localization by multilateration for the use case of precisely referencing CVD.

For the evaluation, a commercially viable u-blox 6 single frequency GNSS receiver has been incorporated at the vehicle-side for both determining the GNSS online solution and acquiring the GNSS raw data. At the back-end-side, for PPP-based postprocessing the RTKLIB [122, 123, 124, 125] was utilized. The raw observation and navigation data has been acquired by enabling the proprietary `UBX`-messages `RXM-RAW` and `RXM-SFRMB` of the u-blox 6 GNSS receiver. Furthermore, the GNSS receiver has been parametrized to a sampling rate of 10 Hz (section 3.3). The acquired GNSS raw data was propagated as part of the CVD to the back-end-side and subsequently postprocessed under the consideration of correction data. The final correction data was provided by the IGS.

For the actual evaluation, a route from Ingolstadt to Frankfurt of 316 km was utilized (figure 3.2). The route consists mainly of German Autobahn sections and can be considered to feature mainly ideal, open sky conditions for GNSS. The ground truth for this route was acquired by a deeply coupled, geodetic-grade, localization system, the iMAR iTraceRT-F400, and can be considered to be accurate, concerning its absolute position, to within a few centimeters [59].

In table 3.3 the actual accuracies achieved by means of distinct parameterizations of PPP are provided and opposed to the online solution gained by the GNSS receiver. It can be noticed that the best average deviation, which is approximately 1.30 m,

	Ratio
RINEX 3.0	1.0
Text Compression	4.2
C-RINEX 3.0	3.8
C-RINEX 3.0 + Text Compression	11.1

Table 3.4: Compression *ratios* achievable for RINEX 3.0 observation data according to Hatanaka [53]. RINEX 3.0 is set as the reference value. C-RINEX does refer to the compressed RINEX-format introduced by Hatanaka.

and the worst average deviation, which is approximately 1.38 m, differ by 8 cm on average. The best offline solution is by approximately 0.55 m more accurate on average than the online one. The improvement is accompanied by a significantly decreased standard deviation of 1.318 m. In figure 3.3 the absolute average deviation probabilities corresponding to the means and standard deviations previously provided in table 3.3 for offline localization and in figure 3.4 for online localization are visualized. The absolute average deviations are provided in a combined manner for both the lateral and longitudinal direction.

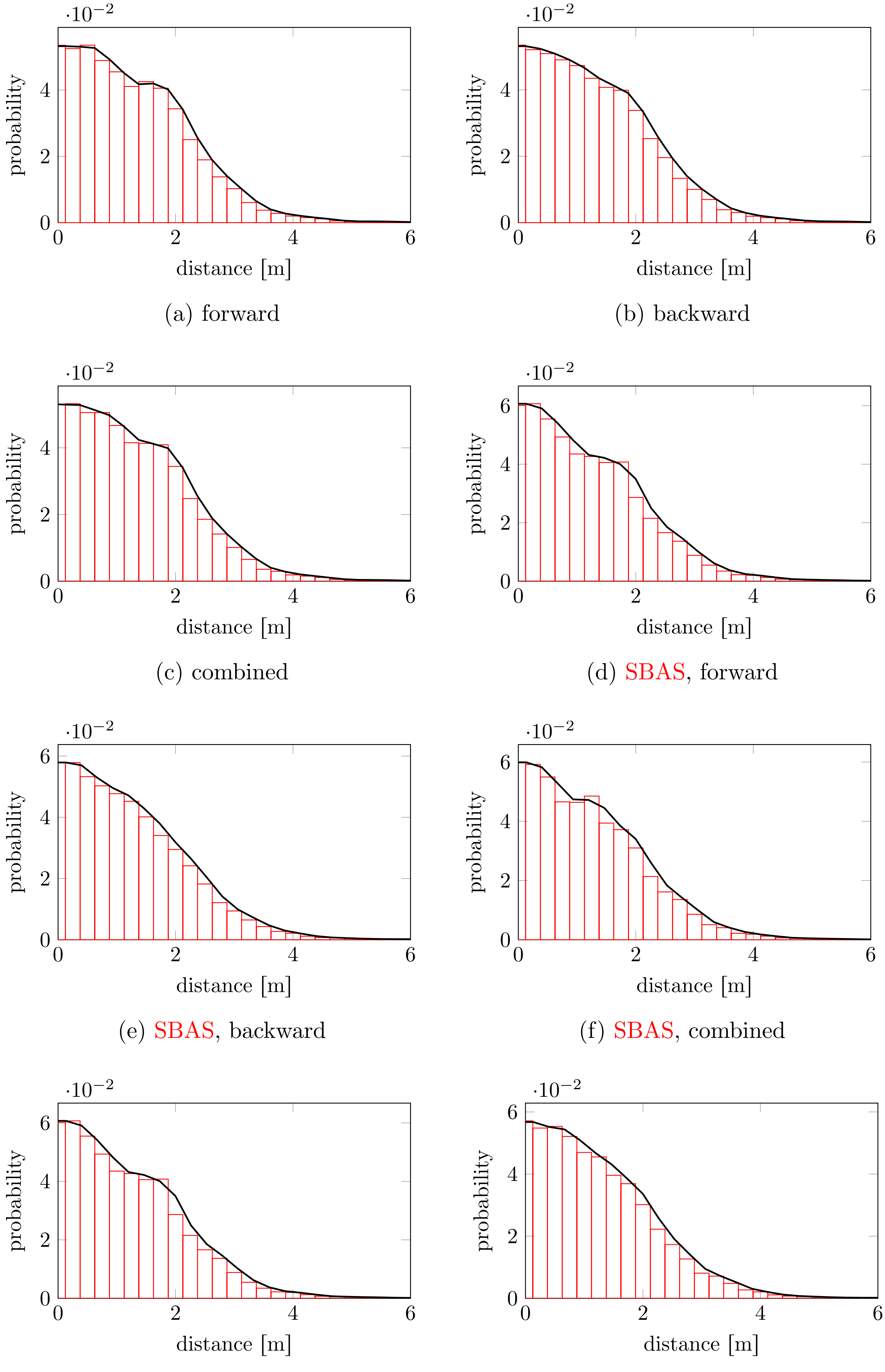
For the transmission of the raw GNSS observation data from the vehicle to the back-end-side, a protocol based upon the well-known RINEX[9] 3.0 data format has been used. On average, one measurement of raw GNSS observation data, which is gathered every $\frac{1}{10}$-th of a second, is expected to require approximately 730 B on average. By employing specialized compression algorithms for GNSS raw data, such as the one introduced by Hatanaka [53], the size of the raw data can be reduced to approximately 9 % of the original size (table 3.4). This results in an average compressed observation size of 66 B, leading to approximately 2.3 MB of raw GNSS observation data per hour and vehicle at a 10 Hz sampling rate for a single-frequency receiver.

3.2.4 Discussion

In this section, a commercially viable solution for the precise referencing of CVD by means of PPP-based postprocessing has been suggested. This approach deviates from the state of the art in that the precise vehicle position is computed at the back-end-side instead of the vehicle-side. This may be considered as unusual, but is highly beneficial concerning the achieved accuracy, robustness, and availability.

The results of 1.38 m average deviation at 0.984 m standard deviation obtained by PPP-based postprocessing under mostly open-sky conditions and by utilizing a single frequency receiver can be considered as very good. In comparison to the online solution, the offline solution has an average deviation reduced by 0.55 m and

[9] Receiver Independent Exchange Format (RINEX)



(a) forward

(b) backward

(c) combined

(d) SBAS, forward

(e) SBAS, backward

(f) SBAS, combined

(g) SBAS, forward, PPP-AR i=1

(h) SBAS, forward, PPP-AR i=5

Figure 3.3: The average deviation probabilities of distinct PPP-based GNSS post-processing parameterizations.

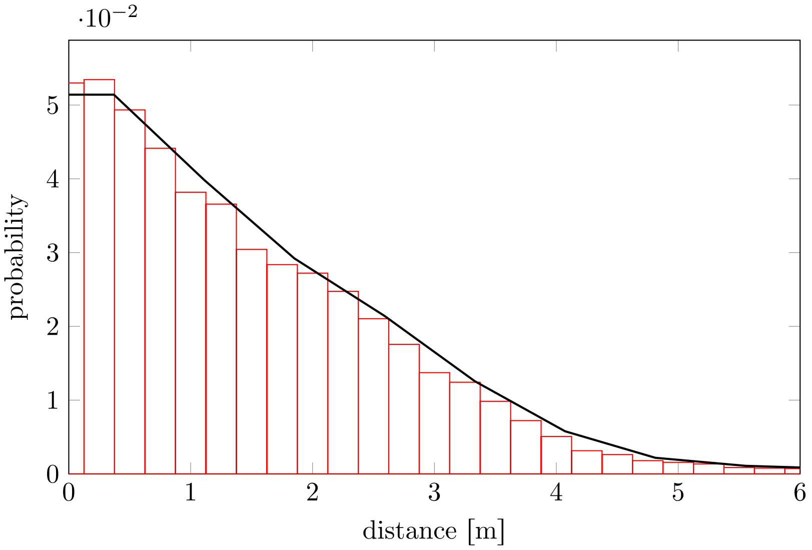


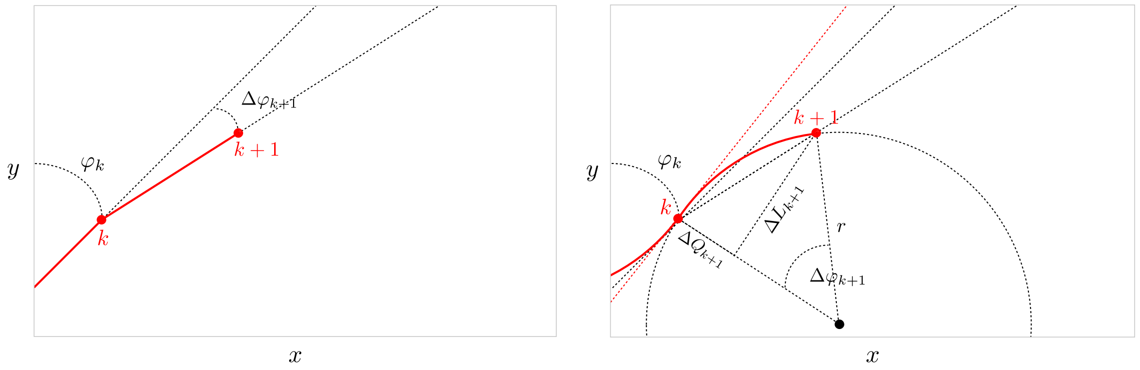
Figure 3.4: The average deviation probabilities gained by online localization.

a standard deviation reduced by 1.318 m. This improvement can be considered as significant. When comparing the suggested offline solution to the one achieved by the same GNSS receiver and SAPOS[10] EPS[11] correction data, as suggested by Mickler [78], the standard deviation of the presented offline solution is still 0.22 m less. Additionally, concerning the use case of CVD, the solution of the thesis is neither degraded by outages of the reception of correction data due to white spots of the cellular nor the DAB network. Cellular white spots can be regarded as unproblematic for the transmission of GNSS raw data to the backend-side due to the feasibility of caching. Thus the comprehensive PPP-based postprocessing of GNSS raw data can be regarded as highly promising for the use-case of CVD.

On the downside, the postprocessing of GNSS raw data at the back-end-side requires the propagation of higher amounts of data via the cellular network. However, the amount of data can be significantly reduced by means of dedicated compression algorithms, so that they become commercially viable with approximately 2.3 MB/h per vehicle.

Further improvements concerning the precise referencing of CVD are achievable by utilizing a dual- or multi-frequency GNSS receiver. However, this causes much higher expenses for the receiver and can be considered, therefore, as currently commercially not viable for common vehicles. In the future, this might of course change. Furthermore, as a vehicle cannot be expected to encounter everywhere ideal GNSS conditions, it is suggested to support the postprocessing by vehicle egomotion data, for example by data from an IMU[12] and odometry. This way, the localization

[10] Satellite Positioning Service of the German National Survey (SAPOS)
[11] Realtime Positioning Service (EPS)
[12] Inertial Measurement Unit (IMU)



(a) Linear Motion

(b) Circular Motion

Figure 3.5: Comparison of a linear (equation 3.4) and a circular (equation 3.5) vehicle motion model [103]. The linear vehicle motion model assumes rotations to be instantly performed at each time increment k. On the contrary, the circular motion model assumes rotations to be performed in a continuous manner.

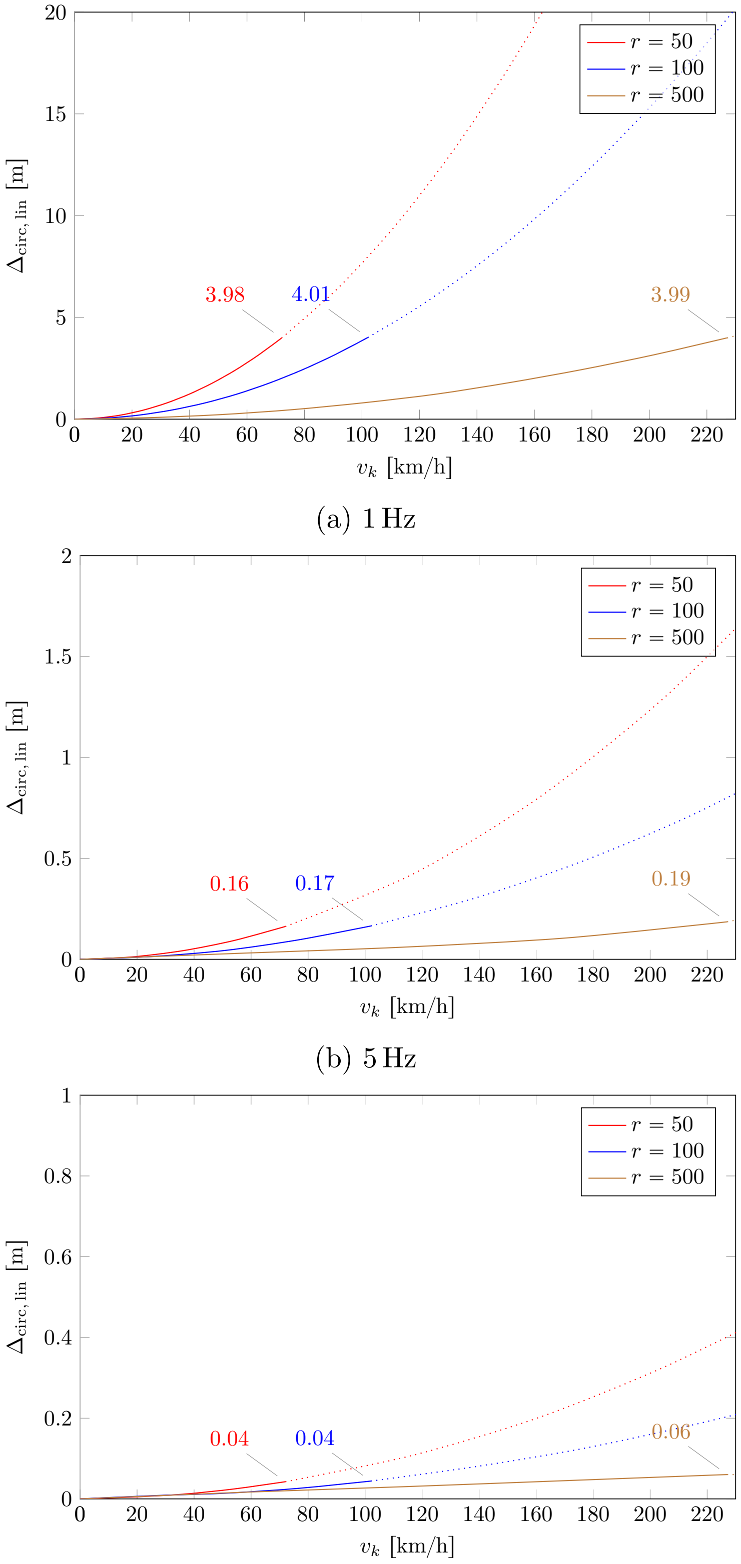
is able to compensate considerably better for GNSS shadowing and multipath effects as e. g. observed in urban canyons.

3.3 Modeling Vehicle Motion

A further approach to robustified vehicle localization besides the GNSS raw data filtering in chapter 3.2.2 is to filter the stream of location data calculated from the GNSS sensor data. For this purpose, filtering techniques like the (E)KF are commonly utilized. One component of such techniques is the assumption of a vehicle motion model. A *vehicle motion model* is defined by a class of mathematical trajectories which can represent the possible real trajectories of a vehicle sufficiently well. A vehicle motion model helps to improve the prediction of the position of a vehicle in motion by restricting the possible locations. An improved prediction in turn improves the precision of vehicle localization which is of high relevance for the fusion of point-shaped/complex landmark data.

Existing vehicle motion models can be categorized in linear models, such as CV[13] or CA[14], and curvilinear models, such as CTRV[15], CTRA[16], CSAV[17], or CCA[18] [102, 103].

[13] Constant Velocity (CV)
[14] Constant Acceleration (CA)
[15] Constant Turn Rate and Velocity (CTRV)
[16] Constant Turn Rate and Acceleration (CTRA)
[17] Constant Steering Angle and Velocity (CSAV)
[18] Constant Curvature and Acceleration (CCA)



(a) 1 Hz

(b) 5 Hz

(c) 10 Hz

Figure 3.6: Visualization of the vehicle-speed-dependent deviations of a linear (equation 3.4) and a circular (equation 3.5) vehicular motion model. It is distinguished between different localization sampling-rates $f_s \in \{1\,\text{Hz}, 5\,\text{Hz}, 10\,\text{Hz}\}$ and different curve radii $r \in \{50\,\text{m}, 100\,\text{m}, 500\,\text{m}\}$. The maximal lateral acceleration for dry road conditions is assumed to be limited by $10\,{}^{\text{m}}\!/_{\text{s}^2}$.

Linear motion models lead to simpler algorithms than circular motion models, but the question is whether or at what costs they can achieve the approximation behavior of circular models which can be expected as better. Schweitzer [103] has analyzed the influence of different rates of location sampling and different curve radii on the deviation between a linear motion model, which is comparable to CV, and a circular motion model, which is comparable to CSAV:

Definition 3.1: Linear Motion Model

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} \overset{\text{lin}}{=} \begin{pmatrix} x_k + v_k \cdot \Delta t \cdot \cos\left(\varphi_k + \Delta\varphi_{k+1}\right) \\ y_k + v_k \cdot \Delta t \cdot \sin\left(\varphi_k + \Delta\varphi_{k+1}\right) \end{pmatrix} \tag{3.4}$$

with v_k being the vehicle's velocity, φ_k its current heading, φ_{k+1} its intended heading, $\Delta\varphi_{k+1}$ the resulting heading increment, and Δt the time increment.

Definition 3.2: Circular Motion Model

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} \overset{\text{circ}}{=} \begin{pmatrix} x_k + \cos\left(\varphi_k\right) \cdot \Delta L_{k+1} - \sin\left(\varphi_k\right) \cdot \Delta Q_{k+1} \\ y_k + \sin\left(\varphi_k\right) \cdot \Delta L_{k+1} + \cos\left(\varphi_k\right) \cdot \Delta Q_{k+1} \end{pmatrix} \tag{3.5}$$

with

$$\begin{aligned} \Delta L_{k+1} &= r \cdot \sin\left(\Delta\varphi_{k+1}\right), \\ \Delta Q_{k+1} &= r \cdot \left(1 - \cos\left(\Delta\varphi_{k+1}\right)\right), \\ \Delta\varphi_{k+1} &= \frac{180^\circ \cdot v_k \cdot \Delta t}{\pi \cdot r}, \end{aligned}$$

and v_k being the vehicle's velocity, φ_k its current heading, φ_{k+1} its intended heading, $\Delta\varphi_{k+1}$ the resulting heading increment, and Δt the time increment.

Both motion models are visualized in figure 3.5.

Figure 3.6 depicts the deviation of the linear from the circular vehicle motion model for three different sampling rates, $f_s \in \{1\,\text{Hz}, 5\,\text{Hz}, 10\,\text{Hz}\}$, and three different curve radii, $r \in \{50\,\text{m}, 100\,\text{m}, 500\,\text{m}\}$. For the comparison of both motion models the Euclidean distance metric

$$\Delta_{\text{circ, lin}} = \sqrt{(x_{\text{circ}} - x_{\text{lin}})^2 + (y_{\text{circ}} - y_{\text{lin}})^2} \tag{3.6}$$

is employed. The maximal lateral acceleration a on a dry road surface is assumed to be limited by $10\,{}^{\text{m}}/_{\text{s}^2}$. The maximal velocity of a vehicle without loosing traction is then given by $v_{\max} = \sqrt{a \cdot r}$, with a being the maximal lateral acceleration and r the curve radius.

The results show that the deviations are significantly larger for lower than for higher localization sampling rates. The average error decreases from 3.99 m at 1 Hz to 0.046 m at 10 Hz. One can conclude that higher sampling rates lead to less benefit of the principally more accurate curvilinear motion models. Therefore, at higher sampling rates, a less computationally intensive, linear motion model can be safely incorporated without a considerable loss of accuracy. However, at lower sampling rates a curvilinear motion model should be the first choice.

Within this work, a localization sampling rate of 10 Hz and a linear vehicle motion model are used. This choice affects the computational efficiency positively, what is crucial for large-scale applications, and the analysis of this chapter shows that the resulting error can be regarded as negligible.

3.4 Compounding Measurements

The sensor data originating from exteroceptive sensors is typically provided by coordinates in a sensor-related reference frame. For cooperative fusion, it is reasonable to refer all the data to one global reference frame. It can be regarded as common use to refer the data of all sensors to a common vehicular reference frame.

By utilizing the timestamp of the sensor readings for determining the vehicular position at this specific point in time, it can be referenced precisely, even in the case when the sampling points/rates of the sensors involved are distinct. The current location of the vehicular reference frame in turn can be expressed by the position $\boldsymbol{x}_i$ and heading $\varphi_i^{\text{azimuth}}$ of the vehicle i, e. g. as provided by a GNSS receiver, and the respective reference frame, such as WGS84[19], may take over the role of the global reference frame. Figure 3.7 shows an exemplary visualization of the three involved reference frames, the sensory ①, the vehicular ②, and the global one ③.

The sensor data can be transferred into the global reference frame by compounding the sequence of relative reference frame locations. However, the measurements of every relationship, e. g. of the location of a captured object with respect to a sensor reference frame, of the location of a sensor with respect to the vehicle reference frame, or of the location of the vehicle with respect to the global reference frame, are affected by distinct uncertainties. Therefore, the corresponding uncertainties along the chain of transformations should be appropriately considered with the aim to assign a total uncertainty to the resulting absolute representation of the sensor data. In this work, the method for stochastic compounding developed by Smith et al. [120] is employed.

The basic compounding operation is defined as follows:

[19] World Geodetic System 1984 (WGS84)

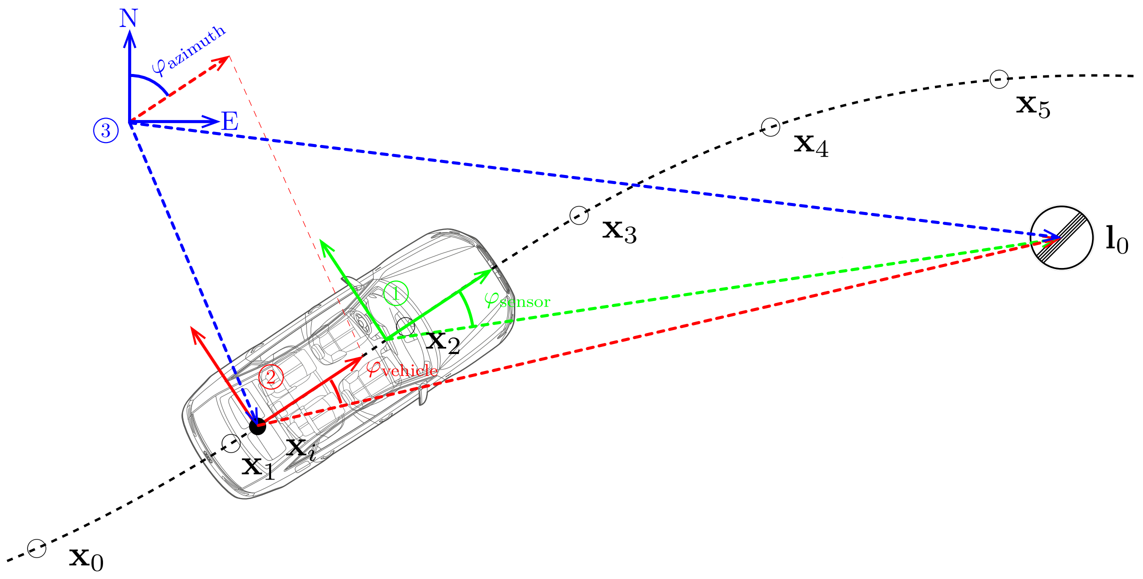


Figure 3.7: In this example, a traffic sign $\boldsymbol{l}_0$ is observed by means of a TSR-camera. As usual, it is provided within the sensor's own reference frame ①. In order to combine the sensor readings of multiple sensors, all individual sensor reference frames are transformed into a common vehicular frame ②. By transforming the common vehicular reference frame by means of an absolute position $\boldsymbol{x}_i$ and heading $\varphi_i^{\text{azimuth}}$ of a vehicle i, the acquired CVD can be referenced in a global, cross-vehicular manner ③, such as by the WGS84 reference frame. The distinct sampling points/rates of the utilized sensors need to be taken into account. This can be, for instance, achieved by means of interpolation. The interpolated vehicular position is denoted by $\boldsymbol{x}_i$.

Definition 3.3: Compounding

Input: A sequence of objects i, j, k with relationships of object j relative to object i and of object k relative to object j, represented by real relationship vectors $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{j,k}$.

Output: The spatial relationship of object k relative to object i, represented by a real relationship vector $\boldsymbol{x}_{i,k} = \boldsymbol{x}_{i,j} \oplus \boldsymbol{x}_{j,k}$.

$\oplus$ is the so-called *compounding operator* introduced by Smith et al. A relationship vector specifies the parameters of a class of the possible motions by which the reference frame describing the location of object j may result from the reference frame of object i. An example of relationship vector is $\boldsymbol{x}_{i,j} = (x_{i,j},\, y_{i,j},\, \varphi_{i,j})^T$ which specifies that a two-dimensional object j is translated by $(x_{i,j},\, y_{i,j})^T$ and rotated by an angle $\varphi_{i,j}$ relative to object i. The class of motions of this example are rigid ones in the plane.

In this work, motivated by the use of the linear vehicle motion model (section 3.3), the relation $\boldsymbol{x}_{i,j}$ of a vehicle to the global reference frame is assumed to belong to the motion class of translations in the plane, i. e. $\boldsymbol{x}_{i,j} = (x_{i,j},\, y_{i,j})^T$. Furthermore, for sensors delivering just points, the *observation model*, i. e. motion model of the relation between the captured points and the sensor, can be assumed translational as well, since the orientation of a reference frame associated with the captured object is irrelevant. If the sensor is rigidly connected with the vehicle, the relation $\boldsymbol{x}_{j,k}$ between the captured points and the vehicle motion belongs to the motion class of the translations in the plane, too, i. e. $\boldsymbol{x}_{j,k} = (x_{j,k},\, y_{j,k})^T$. In this case, $\oplus$ is the usual vector addition, and thus $\boldsymbol{x}_{i,k} = \boldsymbol{x}_{i,j} + \boldsymbol{x}_{j,k}$ holds for the relation $\boldsymbol{x}_{i,k}$ of the captured object within the global reference frame.

The related stochastic version is

Definition 3.4: Stochastic Compounding

Input: A sequence of objects i, j, k with stochastic spatial relationships of object j relative to object i and of object k relative to object j, under assumption of a Gaussian noise model, represented by means $\overline{\boldsymbol{x}}_{i,j}$ and $\overline{\boldsymbol{x}}_{j,k}$ and covariances $\mathrm{Cov}(\boldsymbol{x}_{i,j})$ and $\mathrm{Cov}(\boldsymbol{x}_{j,k})$ of the relationship vector distributions.

Output: The stochastic spatial relationship of object k relative to object i, represented by a mean $\overline{\boldsymbol{x}}_{i,k}$ and a (co)variance $\mathrm{Cov}(\boldsymbol{x}_{i,k})$ of the relationship vector distribution.

By linear approximation of the compounding operator by the first-order Taylor expansion around the means, Smith et al. have derived the following first-order approximations of the mean and the covariance of $\boldsymbol{x}_{i,k}$:

$$\overline{\boldsymbol{x}}_{i,k} \approx \overline{\boldsymbol{x}}_{i,j} \oplus \overline{\boldsymbol{x}}_{j,k} \tag{3.7}$$

$$\mathrm{Cov}(\boldsymbol{x}_{i,k}) \approx \boldsymbol{J}_{\oplus} \cdot \begin{bmatrix} \mathrm{Cov}(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,j}) & \mathrm{Cov}(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{j,k}) \\ \mathrm{Cov}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{i,j}) & \mathrm{Cov}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{j,k}) \end{bmatrix} \cdot \boldsymbol{J}_{\oplus}^T, \tag{3.8}$$

where

$$\boldsymbol{J}_{\oplus} = \left. \frac{\delta(\boldsymbol{x}_{i,j} \oplus \boldsymbol{x}_{j,k})}{\delta(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{j,k})} \right|_{\overline{\boldsymbol{x}}_{i,j}, \overline{\boldsymbol{x}}_{j,k}} .$$

is the Jacobian of the compounding operator $\oplus$.

In the case that the measurements $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_{j,k}$ are not cross-correlated, the compounded co-variance simplifies to

$$\mathrm{Cov}(\boldsymbol{x}_{i,k}) \overset{\text{no cc}}{\approx} \boldsymbol{J}_{\oplus} \cdot \begin{bmatrix} \mathrm{Cov}(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,j}) & 0 \\ 0 & \mathrm{Cov}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{j,k}) \end{bmatrix} \cdot \boldsymbol{J}_{\oplus}^{T}. \tag{3.9}$$

In practice, this is for instance the case when determining the vehicle position via GNSS and the relative vector to a point-shaped landmark, such as a traffic sign, by a TSR[20]-camera (figure 3.7). Both sensors are independent of each other and, therefore, uncorrelated.

In the case of the two-dimensional translational vehicle motion and observation models from above, the Jacobian $\boldsymbol{J}_{\oplus}$ of the compounding operator $\oplus$ is

$$\boldsymbol{J}_{\oplus} \overset{\text{transl.}}{=} \left.\frac{\delta(\boldsymbol{x}_{i,j} \oplus \boldsymbol{x}_{j,k})}{\delta(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{j,k})}\right|_{\overline{\boldsymbol{x}}_{i,j}, \overline{\boldsymbol{x}}_{j,k}} = \left.\frac{\delta(\boldsymbol{x}_{i,j} + \boldsymbol{x}_{j,k})}{\delta(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{j,k})}\right|_{\overline{\boldsymbol{x}}_{i,j}, \overline{\boldsymbol{x}}_{j,k}} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ \underbrace{0 \quad 1}_{\boldsymbol{J}_{\oplus}^{0}} & & \underbrace{0 \quad 1}_{\boldsymbol{J}_{\oplus}^{1}} & \end{bmatrix}. \tag{3.10}$$

Hence, the compounding of the measurement uncertainties $\mathrm{Cov}(\boldsymbol{x}_{i,j})$ and $\mathrm{Cov}(\boldsymbol{x}_{j,k})$ in this case is gained by means of pure matrix addition

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{x}_{i,k}) &\approx \boldsymbol{J}_{\oplus}^{0} \cdot \mathrm{Cov}(\boldsymbol{x}_{i,j}, \boldsymbol{x}_{i,j}) \cdot \boldsymbol{J}_{\oplus}^{0} + \boldsymbol{J}_{\oplus}^{1} \cdot \mathrm{Cov}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{j,k}) \cdot \boldsymbol{J}_{\oplus}^{1} \\ &= \boxed{\mathrm{Cov}(\boldsymbol{x}_{i,k}) + \mathrm{Cov}(\boldsymbol{x}_{j,k})}\,. \end{aligned} \tag{3.11}$$

3.5 Discussion

This chapter has addressed several aspects of localization of vehicles and CVD with the aim of treating them in an appropriate way in the processing chain: localization of vehicles, modeling of vehicle motion, and stochastic compounding of relative relationships of localization.

An approach for precise GNSS vehicle localization has been presented which calculates the localization at the back-end instead of the vehicle-side (chapter 3.2). The calculation is PPP-based and is performed by offline-postprocessing. This approach can be considered as unusual but it is highly beneficial concerning the gained accuracy, robustness, and availability. It is used in the processing chain proposed in the thesis.

An analysis of the difference between linear and curvilinear vehicular motion models has been presented. It shows that for sampling rates greater or equal to 10 Hz the error introduced by linear vehicular motion models can be regarded as minor. Therefore, a linear vehicular motion model is subsequently utilized for reducing

[20] Traffic Sign Recognition (TSR)

the computational load, while the prediction quality is only affected in a negligible manner (section 3.3)

A condition for the fusing approaches presented in the remainder of the thesis is that the location data captured by vehicles in motion and referring to individual vehicle frames, is referred to one universal global frame. For this purpose, a method of compounding spatial relations has been picked up which takes into account uncertainties of measurements and which is used in the processing chain of the thesis (section 3.4).

4

Submapping and Temporal Weighting

Data streams delivered by a fleet of vehicles are usually covering a large area and long periods. Therefore, their processing requires computationally efficient solutions. This chapter introduces two approaches to meet this requirement: submapping and temporal weighting. A variant of submapping is designed which includes advantages of known approaches. For temporal weighting a novel method is presented which allows an efficient dynamic adaption of the weights of observations within an arbitrarily specifiable interval back from the current moment.

Contents

4.1 Introduction

At any time, the processing chain usually acts on just a spatially and temporally local subset of vehicle locations and observations. Efficiency can be gained by supporting the processing chain by structures for those data, which take their specific properties into account in order to store or to deliver the required information more efficiently.

A common approach of e. g. robotics is not to store spatial information during processing in one global map, but in local submaps. An operation of data manipulation acts on just those submaps which are relevant for it. The advantage is that the time requirement for manipulating a submap is only depending on the size of information in the submap, and not on the total information of the map. A data structure for submapping will be presented in section 4.2 which includes advantages of known approaches.

A reason that operations should act just on temporally local information is that observations and other information of the past may lose relevance over time. This can be taken into account by assigning weights to information items of the past which decrease with increasing history. The information is considered by an operation only to an extent given by its weight, i. e. information items with higher weights are more important. As the computation of temporal weights has to be performed very frequently, its efficient evaluation is fundamental. Furthermore, the temporal decay of CVD[1] is strictly depending on its type and acquisition frequency. A novel approach for determining appropriate temporal weights in both an efficient and adaptive manner will be proposed in section 4.3.

Submapping and temporal weighting will be used in later chapters, in particular for the fusion of point-shaped/complex and areal CVD (chapters 7, 8, and 9).

4.2 Submapping

Submapping is a common way for extending algorithms that are feasible for limited scale to large scale [9]. The idea is to use several local maps instead of just a global one, each of which just cover a part of the overall region considered, but which are sufficiently large for the operations to be executed. The submapping problem is defined according to the requirements of the thesis as follows:

Problem 4.1: Submapping

Providing a data structure which represents a finite set of submaps covering the area of possible locations and which efficiently supports the following operations on the set of submaps:

Insertion: Inserts a location $\boldsymbol{x}$ with attribute values into the corresponding submap.

Deletion: Removes a location $\boldsymbol{x}$ with attribute values from the corresponding submap.

k-NN Search: Determines k locations, $k > 0$, nearest to a query location $\boldsymbol{x}$, limited to the submap of location $\boldsymbol{x}$ and possibly neighboring submaps.

r-Range Search: Determines all other locations $\boldsymbol{x}'$ in the range of radius $r > 0$ around a query location $\boldsymbol{x}$, limited to the submap of location $\boldsymbol{x}$ and possibly neighboring submaps.

Function Evaluation: Evaluates a function f on a submap and possibly neighboring submaps.

[1] Collective Vehicle Data (CVD)

Function Updating: Updates a function f on a submap and possibly neighboring submaps.

Existing approaches to submapping can be roughly subdivided into global and local methods [3]. While local methods commonly incorporate a local reference frame, which is for instance relative to the vehicle's initial position, global methods utilize submaps based on a common, vehicle-independent, global reference frame.

In the following, a variant of global submapping, called *adaptive submapping*, similar to the one suggested by Guivant and Nebot [48] is introduced. The proposed variant is employed for scaling the fusion of point-shaped (chapter 7), complex (chapter 8), and areal data (chapter 9).

The method provides a data structure which covers the region to be mapped by a set of small, equally sized submaps which are cells of a grid laid over the region. The data structure is adaptive in that only submaps which contain still relevant information from the past, and submaps which might be affected soon by a vehicle, are stored. The latter submaps are determined in a predictive manner when vehicles are located near to the boundary of the region covered up to now. The submaps are globally aligned, i. e. they all are referred to the same, global reference frame. The submaps can be processed independently from each other, and, in particular, in parallel.

Updating is performed by considering the submap cell which is directly addressed by the updating operation, but also the submap cells in its so-called 8-neighborhood, i. e. those submaps which are neighboring along a shared grid edge or a shared grid vertex (figure 4.1). This way, the consistency between neighboring submap cells can be more appropriately assured. Even globally optimal results can be achieved when certain constraints are fulfilled (chapter 9). The search for the submaps relevant for execution of an operation and the search for locations within submaps is supported by a combination of hash tables and R-trees [49] as dynamic index structures.

This data structure of global submapping has several advantages. The global approach to submapping exhibits the advantage that all vehicles share a common coordinate reference frame, what significantly simplifies the collaborative fusion of CVD. Storing only submaps which have been relevant in the past takes account to the principally sparse nature of the road network. Furthermore, the index data structures utilized for a fast grid lookup are not bloated with unused entries. The redundant calculation and comparison of overlapping areas of distinct 8-neighborhoods and allowing none to only minor deviations, due to numeric inaccuracies, between the gained results provides an additional layer of computational safety, what can be considered as crucial when the fused data shall be incorporated within safety critical automotive

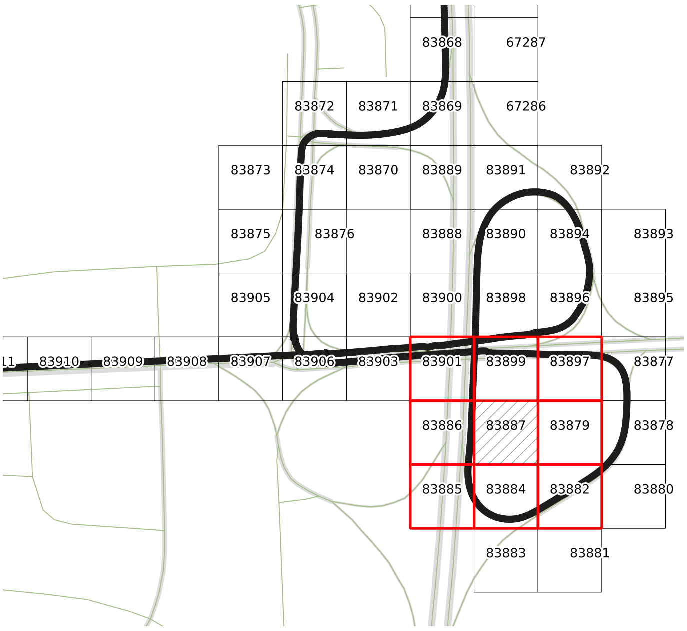


Figure 4.1: Global submapping by means of adaptive partitioning. The locality of computations, which apply to exactly one grid (hachured in gray ●), is limited to their eight direct neighbors (highlighted in red ●, the so-called 8-neighborhood). The grids are assumed to be square and have, in this example, an equal side length of 64 m.

functions. Finally, the submapping data structure allows effective local linearizations of functions in a canonical way by choosing the linearization point of the 1st-order Taylor series (appendix A.3) in the submap cell where the function is evaluated, e. g. by taking its center point.

4.3 Temporal Weighting

An important aspect of the fusion of CVD is the consideration of the temporal decay of acquired data. The decay rate should be chosen depending on the type of sensor data. For example, temperature measurements are expected to decay much faster than road sign observations or pothole detections. Additionally, the decay of measurements should depend on their acquisition frequencies. This way, the number of measurements can be reduced to the actually required ones, e. g. to gain a specific fusion accuracy, and valuable processing capacities can be saved, what can be considered as being crucial for large-scale applications.

In this work, the temporal decay is taken into account by weights assigned to information items of the past and decreasing with increasing history. In the case of areal data (chapter 9), temporal weighting will be performed for every submap, and in the case of point-shaped/complex landmark data (chapters 7 and 8) for every cluster of corresponding landmark observations.

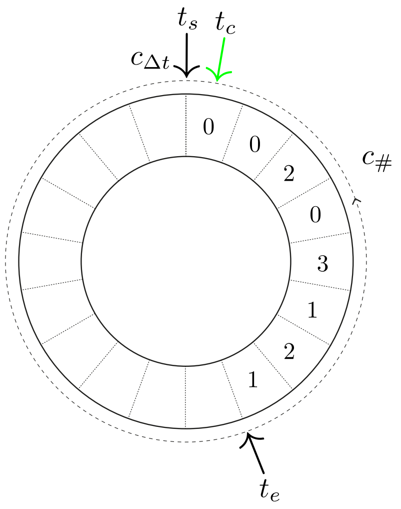


Figure 4.2: An exemplary circular buffer that is utilized for determining appropriate measurement weights in both an adaptive and computationally efficient manner.

The problem of temporal weighting is as follows:

Problem 4.2: Temporal Weighting

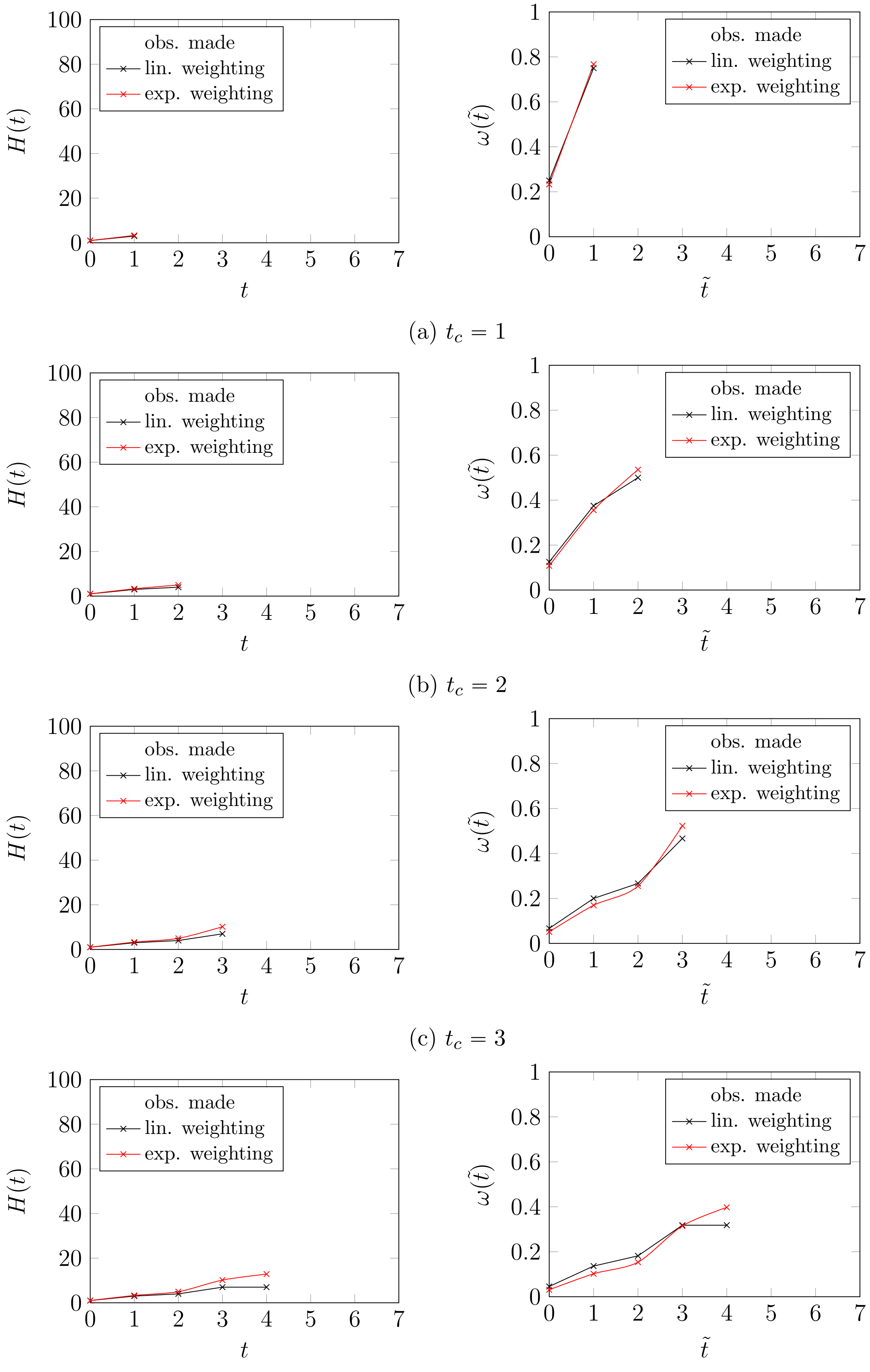
Input: A stream of data items with timestamps of their acquisition.

Output: A list of non-negative weights, which is updated whenever a new data item appears.

The problem is solved by using a *circular buffer* as the basic data structure. An exemplary instantiation of a circular buffer is visualized in figure 4.2.

The circular buffer consists of so-called *bins* which are circularly arranged. $c_{\#}$ denotes the number of bins provided by the data structure; it is set according to the targeted maximal temporal extent of the fusion. Every bin represents a time interval of width $c_{\Delta t}$ which defines the temporal discretization granularity of the weight computation. $c_{\Delta t}$ is set according to the targeted frequency of the fusion process. A sequence B of consecutive bins b represents a period whose start time and end time are denoted by t_s and t_e, representing the most actual and the least actual timestamp under consideration, respectively, i. e. $t_s > t_e$. Initially, B consists of one bin covering the time between $t_e = 0$ and $t_s = c_{\Delta t}$.

The observations are processed in ascending temporal order. Let t_c denote the timestamp of the current observation o. If $t_c \leqslant t_s$, o is assigned to the bin b of B whose $c_{\Delta t}$-period contains t_c. The number of observations assigned to a bin is denoted by $h(b)$. If the number of observations stored in bins of B exceeds a given upper bound c_{cutoff} of observations that is being considered for computing the temporal weights, the bin covering the least recent period is removed from B, and t_e is updated,

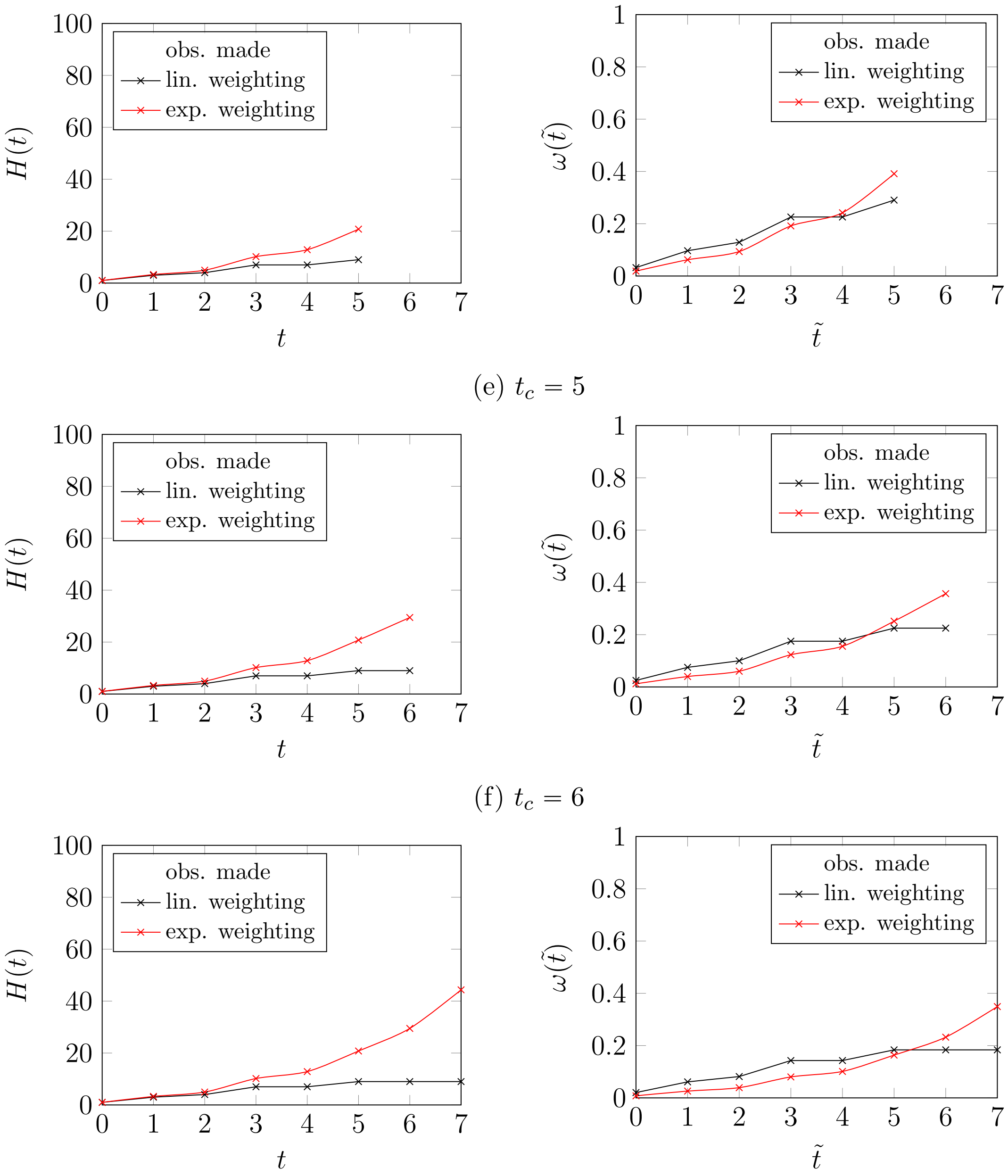


(a) $t_c = 1$

(b) $t_c = 2$

(c) $t_c = 3$

(d) $t_c = 4$



(e) $t_c = 5$

(f) $t_c = 6$

(g) $t_c = 7$

Figure 4.3: Visualization of an exemplary linear and exponential temporal weight computation according to the suggested scheme, which is applied step-by-step based on the exemplary data provided in figure 4.2. All timesteps that feature observations are hachured in gray ●. The weighting factor ν has been set to 1.06 for the exponential weighting.

i. e. $t_e = t_e + c_{\Delta t}$. By setting an upper bound, the required processing capacities at the back-end-side can be adjusted. If $t_c > t_s$, new bins are inserted into B in front of the bin covering the most recent period, until the corresponding t_s satisfies $t_c \leqslant t_s$. t_s is updated by $t_s = t_s + j \cdot c_{\Delta t}$, with j the number of bins added. Then o is inserted in the bin whose time period covers t_c.

The weight of an observation $\tilde{o}$ with timestamp $\tilde{t}$ is calculated as

$$\omega(\tilde{t}) = \frac{H(\tilde{t})}{\sum_{b \in B} H(t_s(b))}, \; t_e \leqslant \tilde{t} \leqslant t_s, \tag{4.1}$$

with

$$H(t) = \sum_{b \in B_t} h(b), \; t_e \leqslant t \leqslant t_s,$$

where $t_s(b)$ is the start time of a bin b, i. e. the bin covers the period between $t_e(b) = t_s(b) - c_{\Delta t}$ and $t_s(b)$, and B_t is the subset of bins $b \in B$ with $t_e(b) \leqslant t$. $H(t)$ is the cumulative histogram of the binned observation counts $h(b)$, covering the bins between the least recent one and the last bin whose period precedes t.

This approach leads to a linear weighting scheme, i. e. the growth of the weights is proportional to the accumulated counts in histogram bins. If an exponential weighting is required, it can be accomplished in a straightforward manner by augmenting the histogram function $h(b)$, for instance by $h_{\exp}(b_i) = (h_{\exp}(b_{i-1}) + h_{\exp}(b_i) \cdot \nu^i) \cdot \nu^i$ for $i \geqslant 1$, and $h_{\exp}(b_0) = h(b_0)$ for the least recent bin. By setting the weighting parameter $\nu > 1$ an exponential oblivion of former observations is achieved. In the case of $\nu = 1$ the oblivion is linear.

Figure 4.3 shows a step-by-step temporal weight computation according to the exemplary data visualized in figure 4.2. The pointer indicating the current timestamp t_c is moved bin by bin from t_e to t_s. At every time step, the resulting cumulative histogram and the resulting weights for the linear and exponential weighting scheme are visualized.

The suggested approach for determining temporal weights has several favorable properties. First, the most actual observations always gain the major weight, as the function describing the cumulative histogram is monotonically non-decreasing for $\nu \geqslant 1$. Second, the resulting weights $\omega(t)$ are, per definition, always normalized and, hence, within the range $[0, 1]$. Furthermore, an absence of observations leads to a temporal decay with the slope defined by the cumulative histogram of past observations, what is induced by the monotonically non-decreasing function describing the cumulative histogram. This property is crucial, as decay factors widely differ for distinct measurement types and acquisition frequencies. The presented weighting approach can adapt itself to both.

Another important feature of the presented approach is the ability to forget about past observations, what can be regarded as essential in dynamic environments. Furthermore, the circular buffer-based solution is designed in a way that it can be bidirectionally extended by moving both pointers t_s and t_e, as CVD is known to be exposed to nondeterministic latencies due to its transmission via the cellular network. However, present observations are strictly prioritized by the data structure to former ones. Therefore, the pointer t_e may never supersede the pointer t_s. However, the pointer t_s is allowed to subsume former bins and, therefore, to enforce shifting the pointer t_e to more recent ones.

The circular buffer allows for an efficient continuous updating of the histogram. This dynamic preaggregation is considerably more efficient than an aggregation of all associated timestamps every time a new weight needs to be computed or a (full) reweighting needs to be applied. This is crucial for the large-scale application of this work.

4.4 Discussion

This chapter has addressed aspects of efficiency and effectiveness relevant for several steps of the processing chain. A novel approach to global submapping has been presented, which allows scaling the processing of CVD in an adaptive and consistent manner. Even globally consistent solutions can be achieved if certain boundary conditions are fulfilled (section 4.2).

Furthermore, a novel approach for determining temporal weights in an adaptive and efficient manner by means of preaggregation has been elaborated. Based on binned observation counts, a linear as well as an exponential weighting scheme has been defined (section 4.3).

5 Mapping of Point-shaped Landmark Data

This chapter specifies the concept of collaborative point-shaped landmark data mapping from observations acquired by common vehicles and gives a survey of the approach to its realization proposed in this work. It starts with a method of parametric representation of point-shaped landmarks. Next, it specifies the task of point-shaped landmark data mapping. Then it outlines the steps of the approach of precise point-shaped landmark data mapping proposed in this work. The details of two of its major steps, i. e. clustering and fusion of point-shaped landmark data, will be presented in the subsequent chapters 6 and 7.

Contents

5.1 Introduction

The road network context features many different types of point-shaped landmarks, such as traffic signs, traffic lights, and reflector posts. Point-shaped landmarks are, on the one hand, of interest because of the information they provide to the driver/the vehicle. On the other hand, they also are a valuable, salient type of road network features which can be utilized for the objective of achieving a precise vehicle localization. In this sense, they, and all pole-like landmarks, constitute an additional, valuable source for precise vehicle localization.

Point-shaped landmarks are usually not shown within generally available maps. For instance, speed limits and other restrictions imposed by traffic signs are usually only provided as attributes of road segments. Other types of point-shaped landmarks, such as traffic lights or reflector posts, are not represented at all. However, induced by the crucial requirement of a highly accurate vehicle localization, even under difficult conditions, these formerly neglected types of point-shaped landmarks gain significantly in importance.

Furthermore, as the road network context is exposed to steady changes, the usual quarterly updating of maps by the well-known map data providers is not sufficient anymore for point-shaped landmark based localization and further use cases.

An approach to cope with those issues is the mapping of these point-shaped landmarks by common vehicles in a collaborative manner. As today's vehicles are usually equipped with manifold sensors, the detection of these landmarks does *not* constitute any significant additional effort.

The purpose of this chapter is to specify the mapping problem and give a survey of the approach of the thesis to its solution. Section 5.2 introduces a parametric representation of point-shaped landmarks. Section 5.3 is devoted to the problem of point-shaped landmark mapping and section 5.4 to the approach of its solution.

5.2 Parametric Representation of Point-shaped Landmarks

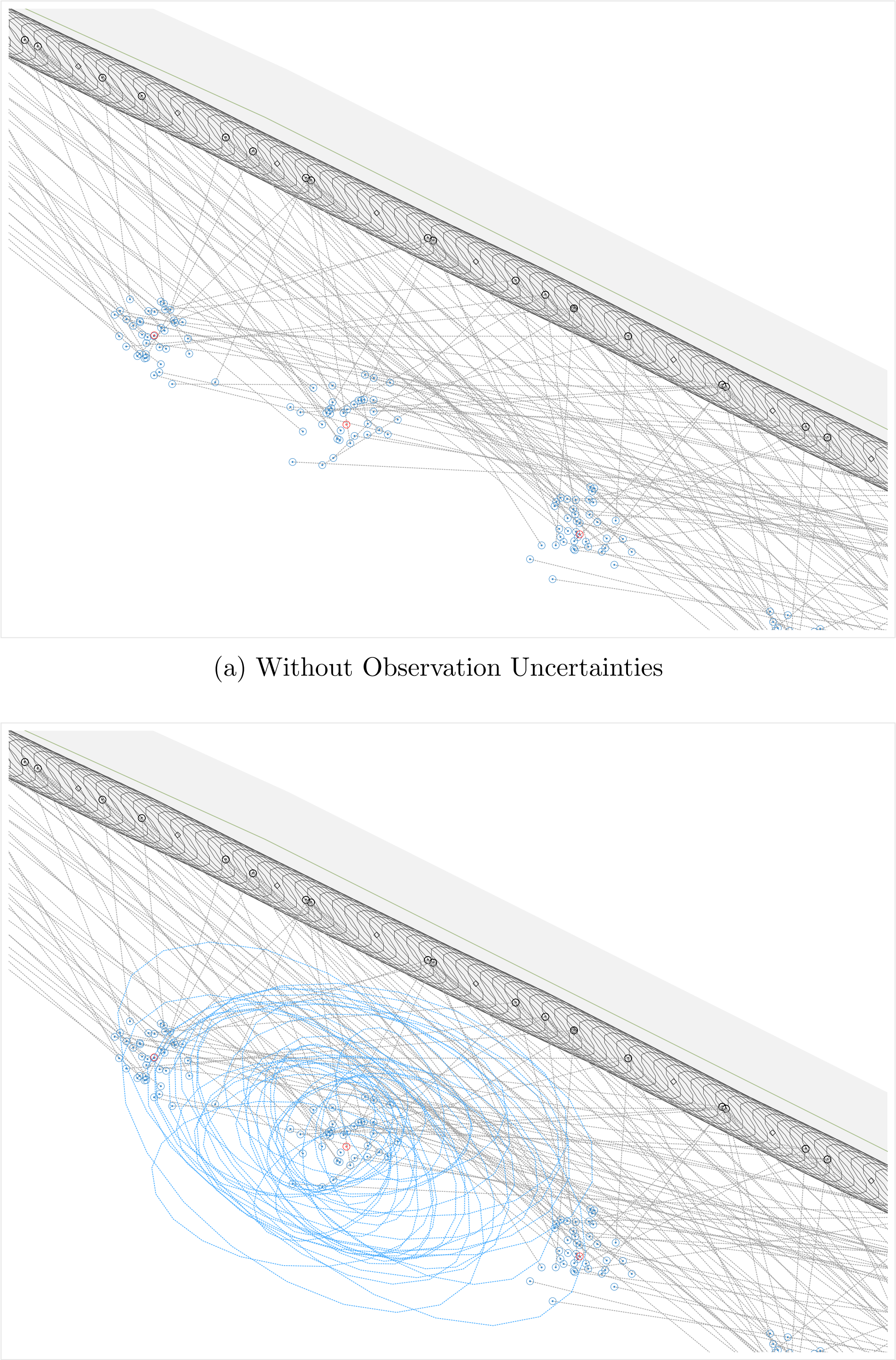
In this thesis, point-shaped landmarks are represented by a set of parameters, as follows:

Definition 5.1: Representation of a Point-shaped Landmark

A point-shaped landmark is represented by *a vector specifying its geographic location* which is usually two-dimensional (latitude, longitude) or three-dimensional (latitude, longitude, altitude), and *additional optional attributes* like the property of being a traffic sign and its direction and indication.

The process of landmark mapping uses observations of landmarks acquired by vehicle-mounted sensors. An introductory example of point-shaped landmark observations is provided in figure 5.1. The figure shows landmark observations acquired by a single vehicle which is driving along a short road excerpt. Both the trajectory and the landmark observations are exposed to measurement uncertainties which are visualized by the uncertainty ellipses of the 3σ-range of a Gaussian probability distribution (appendix A.2).

Point-shaped landmark observations consist of two parts, the observation of location and the observation of attributes:



(a) Without Observation Uncertainties

(b) With Observation Uncertainties

Figure 5.1: An exemplary road section with point-shaped landmark observations acquired by a single vehicle. The acquired landmark observations are highlighted in blue ●, their ground truths in red ●, and the individual vehicle positions of the trajectory in gray ●. The measurement uncertainties are visualized according to their 3σ-ranges by error ellipses (appendix A.2).

Definition 5.2: Representation of a Point-shaped Landmark Location Observation

A point-shaped landmark location observation is represented by a vectorial random variable representing its observed geographic location, which is assumed to be (approximately) Gaussian. It is represented by a vector $\boldsymbol{z} = (z_0, \ldots, z_d)^T$.

Definition 5.3: Representation of a Point-shaped Landmark Attribute Observation

A point-shaped landmark attribute observation is represented by *a vector of the observed additional attributes.*

The locations and the attributes may be acquired by different sensors. The location is modeled by a random variable in order to take into account the uncertainties and errors of measurement.

During the mapping process, estimations of the landmark locations and attributes are derived:

Definition 5.4: Representation of a Point-shaped Landmark Location Estimate

A point-shaped landmark location estimate is represented by *a vectorial random variable* specifying the estimated geographic location of the landmark, represented by a vector $\boldsymbol{l} = (l_0, \ldots, l_d)^T$.

Definition 5.5: Representation of a Point-shaped Landmark Attribute Estimate

A point-shaped landmark attribute estimate is represented by a *vector of the estimated additional attributes.*

5.3 Problem of Point-shaped Landmark Mapping

The task of point-shaped landmark mapping is defined as follows:

Problem 5.1: Point-shaped Landmark Mapping

Input:

1. A sequence of vehicle trajectories.
2. For every vehicle trajectory: a sequence of landmark observations, represented with respect to relative vehicular frames.
3. A temporal decay model.

Output: A sequence of submaps which contain estimates of the locations of the acquired landmarks, represented with respect to a global reference frame, taking into account the temporal decay of the acquired data.

5.4 Approach to Point-shaped Landmark Mapping

Landmark mapping denotes the process of updating a map which is represented by a submapping data structure, cf. section 4.2, by means of landmark estimates derived from the landmark observations provided by a fleet of vehicles, and taking into account the temporal decay of the acquired sensor readings, cf. section 4.3:

Solution 5.1: Point-shaped Landmark Mapping

Initialization:

1. Setup of mapping data structures.
2. Setup of temporal decay models.

Procedure of map updating:

1. Clustering: Assignment of landmark observations to clusters corresponding to distinct landmarks.
2. Landmark estimation: Estimation of landmark locations and, optionally, vehicle locations by sensor data fusion.

The process of point-shaped landmark mapping consists of two phases. The first phase, “clustering”, generates clusters of the landmark observations that may be related to distinct vehicle trajectories and that belong to the same landmark. For this purpose, methods to pure data association according to Neira and Tardós [85] are evolved to clustering ones. Further important modifications to these methods, besides manifold performance and robustness optimizations, are the localized execution of data association supported by a submapping data structure (section 4.2) and the incorporation of temporal decay by temporal weighting (section 4.3). The details will be presented in chapter 6.

The second phase, “fusion”, derives a landmark estimate for each of the previously identified clusters. It can be grounded upon one of the several methods of data fusion, such as (F)CI[1] [63] or SLAM[2] [29, 120, 126]. While the (F)CI-based methods only require compounded landmark observations, this means observations transformed into a global reference frame, the SLAM-based methods require both the sequence

[1] Covariance Intersection (CI)
[2] Simultaneous Localization and Mapping (SLAM)

of vehicle locations and the acquired landmark observations relative to them. The details of these distinct approaches will be presented in chapter 7.

Both phases process the input data incrementally from a sequential stream. The phases are executed in succession, i. e. phase 1 is performed before phase 2 is performed on the output of phase 1. Alternatively, phase 1 followed by phase 2 could be executed for every observation, before the next observation is processed. The first variant is preferred for reasons of computational efficiency. In the case of computationally demanding approaches to sensor data fusion, such as EKF[3]-based Online-SLAM or BA[4]-based Full-SLAM, the second variant can be considered as still infeasible for large-scale applications [85].

Complex landmarks will be mapped in the same two-phase manner. For this purpose, both phases have to be adapted, cf. chapter 8.

[3] Extended Kalman Filter (EKF)

[4] Bundle Adjustment (BA)

6

Clustering of Landmark Data

Prior to their fusion, the observations of point-shaped/complex landmarks have to be grouped into clusters of corresponding observations according to the landmarks to which they actually belong. Frequently utilized algorithms for the task of non-Bayesian data association are ICNN[1], SCNN[2], and JCBB[3]. In this chapter, those algorithms are extended to EICNN[4], ESCNN[5], and EJCBB[6] to satisfy the needs of large-scale and high-precision fusion of point-shaped/complex CVD[7]. Furthermore, the enhanced algorithms are evaluated and compared with each other.

Contents

[1] Individual Compatibility Nearest Neighbor (ICNN)
[2] Sequential Compatibility Nearest Neighbor (SCNN)
[3] Joint Compatibility Branch and Bound (JCBB)
[4] Extended Individual Compatibility Nearest Neighbor (EICNN)
[5] Extended Sequential Compatibility Nearest Neighbor (ESCNN)
[6] Extended Joint Compatibility Branch and Bound (EJCBB)
[7] Collective Vehicle Data (CVD)

6.1 Introduction

According to the approach of the thesis to landmark mapping, cf. section 5.4, the observations of point-shaped and complex landmark data are first grouped into clusters of corresponding observations belonging to one of the landmarks, before those clusters are used for the fusion as described in chapters 7 and 8 for point-shaped and complex landmark data, respectively.

The clustering problem will be considered as a dynamic data association problem which associates landmark observations with corresponding landmark estimates. The quality of data association is known to affect the quality of fusion significantly. A robust determination of correspondences is crucial if a high-quality point-shaped/complex landmark fusion is strictly demanded [21, 72]. Various algorithms intended for tackling data association in a robust and computationally feasible manner have been proposed in the past as approximating alternatives to exhaustive search which is impractical because of its computational requirements. Each of these approaches has distinct strengths and weaknesses.

The chapter starts with a specification of the clustering problem and an introduction to data association which closely follows a publication of Neira and Tardós [85] (section 6.2). The introduction comprehends the definitions of the data association problem and of two types of measures of correspondence, IC[8] and JC[9], as well as exhaustive search and ICNN as two contrasting algorithms of solution with respect to the trade-off of quality and efficiency. Next, solutions to the clustering problem based on dynamic data association are presented (section 6.3). The probably most frequently utilized non-Bayesian algorithms for data association in the context of landmark fusion today, ICNN, SCNN, and JCBB [85], are extended to clustering ones, EICNN (section 6.3.2), ESCNN (section 6.3.3), and EJCBB (section 6.3.4). Furthermore, the distinction of observations by their landmark type, submapping, and temporal decay are taken into account. The chapter ends with the evaluation (section 6.3.8).

6.2 Clustering and Data Association

Clustering concerns the generation of clusters of the landmark observations, which should belong to the same landmark, from a stream of trajectories. It is achieved by processing every trajectory separately. The treatment of a trajectory updates a central map data structure which also serves as input of the clustering procedure in

[8] Individual Compatibility (IC)
[9] Joint Compatibility (JC)

order to take into account the currently achieved result. Thus the clustering problem for one trajectory can be specified as follows:

Problem 6.1: Clustering

Input:

1. A stream of landmark observations of a vehicle trajectory, represented with respect to a global frame.
2. A related temporal decay model.
3. A mapping data structure which represents the current state after the input processed up to now, in particular
   a) the current clustering,
   b) the current state of the temporal weighting data structure.

Output: A new state of the mapping data structure achieved after processing the input trajectory, taking into account the temporal decay.

The clustering problem is tackled by treatment as a data association problem based on the work of Neira and Tardós [85]. In the terminology of the thesis, the data association problem can be specified as follows:

Problem 6.2: Data Association

Input:

1. A sequence of stochastic landmark observations

$$\boldsymbol{Z} = (\boldsymbol{z}_i \mid i = 0, \ldots, m)^T$$

   affected by an approximately Gaussian error, with means $\overline{\boldsymbol{z}}_i$ and covariances $\operatorname{Cov}(\boldsymbol{z}_i) = \boldsymbol{C}_{\boldsymbol{z}_i}$.

2. A sequence of stochastic landmark estimates

$$\boldsymbol{L} = (\boldsymbol{l}_j \mid j = 0, \ldots, n)^T$$

   represented by means $\overline{\boldsymbol{l}}_j$ and covariances $\operatorname{Cov}(\boldsymbol{l}_j) = \boldsymbol{C}_{\boldsymbol{l}_j}$.

3. An *implicit measurement* or *landmark association* function

$$\boldsymbol{f}_{ji}(\boldsymbol{l}_j, \boldsymbol{z}_i)$$

   with a multi-dimensional real-valued range. An association $(\boldsymbol{l}_{j_i}, \boldsymbol{z}_i)$ is compatible if $\boldsymbol{f}_{ji}(\overline{\boldsymbol{l}}_{j_i}, \overline{\boldsymbol{z}}_i) = \boldsymbol{0}$.

Output: A *correspondence map*, also called *interpretation* or *hypothesis*,

$$\mathcal{H} = (\boldsymbol{h}_i \mid i = 1, \dots, m)^T,$$

$\boldsymbol{h}_i = \boldsymbol{l}_{j_i}$, or $\boldsymbol{h}_i = *$ if $\boldsymbol{z}_i$ is spurious. $\mathcal{H}$ pairs each non-spurious landmark observation $\boldsymbol{z}_i$ with a corresponding compatible landmark estimate $\boldsymbol{l}_{j_i}$, if any. A correspondence map is *feasible* if a compatibility constraint holds. The *compatibility constraint* uses a *compatibility measure* which is based on the landmark association functions. A correspondence map is *optimized* if its total compatibility measure value is optimized.

A *clustering of landmark observations* $\mathcal{Z}_{\boldsymbol{l}_j}$ is immediately obtained from a correspondence map by taking

$$\mathcal{Z}_{\boldsymbol{l}_j} = \{\boldsymbol{z}_i \mid \boldsymbol{l}_{j_i} \in \mathcal{H}\},\ j = 1, \dots, n, \tag{6.1}$$

as clusters.

In the application of the thesis, $\boldsymbol{Z}$ represents the sequence of observations of point-shaped or complex landmarks collected along a vehicle trajectory. While landmark estimates are assumed to be initially given for this version of the data association problem, they are not necessarily available for the clustering problem. This issue will be addressed later.

The following subsection 6.2.1 is devoted to the definition of compatibility measures which are not yet specified in the problem formulation. Afterwards, the subsections 6.2.2 and 6.2.3 present “Exhaustive Search” and ICNN, respectively, as two contrasting algorithms of solution with respect to the trade-off of solution quality and efficiency.

6.2.1 Definitions of Compatibility Measures

Because $\bar{\boldsymbol{l}}_{j_i}$ and $\bar{\boldsymbol{z}}_i$ are generally not available, the condition $\boldsymbol{f}_{ji}(\bar{\boldsymbol{l}}_{j_i}, \bar{\boldsymbol{z}}_i) = \boldsymbol{0}$ for defining compatibility of an observation $\bar{\boldsymbol{z}}_i$ and an estimate $\bar{\boldsymbol{l}}_{j_i}$ has been replaced by relaxed compatibility measures [85]. One approach is to define an individual relaxed IC measure for every association $(\boldsymbol{l}_{j_i}, \boldsymbol{z}_i)$. A tighter compatibility measure can be obtained by considering all associations of a correspondence map together and defining a JC measure between the vector $\boldsymbol{Z}$ representing the sequence of observations and $\boldsymbol{L}$ representing the sequence of estimates. Both approaches will be recapped in the following.

Individual Compatibility

One possibility to remedy the issue of using $\boldsymbol{f}_{ji}(\bar{\boldsymbol{l}}_j,\ \bar{\boldsymbol{z}}_i) = \mathbf{0}$, or, equivalently, $\|\boldsymbol{f}_{ji}(\bar{\boldsymbol{l}}_j,\ \bar{\boldsymbol{z}}_i)\| = 0$ according to some norm $\|.\|$ as compatibility check for inaccurate observations is to take

$$\|\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\| < \varepsilon \tag{6.2}$$

for some small real constant $\varepsilon > 0$. This approach has been extended by taking into account the stochastic nature of the data and its error distribution, leading to the so-called stochastic IC measure:

Definition 6.1: Stochastic Individual Compatibility Measure

$$D^2_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i) = \big(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\big)^T \cdot \mathrm{Cov}(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i))^{-1} \cdot \boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i). \tag{6.3}$$

The *total IC measure of a correspondence map* is the sum of the values of the IC measures of its correspondences.

Using this measure, a relaxed compatibility criterion is obtained as follows.

Definition 6.2: Stochastic Individual Compatibility Constraint

$\boldsymbol{z}_i$ and $\boldsymbol{l}_j$ are corresponding if and only if

$$D^2_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i) < \chi^2_{d,\alpha}, \tag{6.4}$$

where $\chi^2_{d,\alpha}$ comes from the χ^2-test [89]. A correspondence map is *feasible with respect to IC* if all its associations $(\boldsymbol{l}_{j_i}, \boldsymbol{z}_i)$ fulfill the stochastic IC constraint.

For non-linear measurement functions, the covariance $\mathrm{Cov}(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i))$ is approximated by using the linear approximation of $\boldsymbol{f}_{ji}$,

$$\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i) \approx \boldsymbol{f}_{ji}(\boldsymbol{l}^0_j,\ \boldsymbol{z}^0_i) + \boldsymbol{G}_{ji}(\boldsymbol{l}_j - \boldsymbol{l}^0_j) + \boldsymbol{H}_{ji}(\boldsymbol{z}_i - \boldsymbol{z}^0_i) \tag{6.5}$$

with

$$\boldsymbol{G}_{ji} = \left.\frac{\partial \boldsymbol{f}_{ji}(\boldsymbol{l}_j, \boldsymbol{z}_i)}{\partial \boldsymbol{l}_j}\right|_{\boldsymbol{l}^0_j, \boldsymbol{z}^0_i}, \qquad \boldsymbol{H}_{ji} = \left.\frac{\partial \boldsymbol{f}_{ji}(\boldsymbol{l}_j, \boldsymbol{z}_i)}{\partial \boldsymbol{z}_i}\right|_{\boldsymbol{l}^0_j, \boldsymbol{z}^0_i},$$

obtained by Taylor expansion (appendix A.3)

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)) &= \boldsymbol{F}_{ji}\cdot\begin{bmatrix}\mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{l}_j) & \mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{z}_i)\\ \mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{z}_i) & \mathrm{Cov}(\boldsymbol{z}_i,\boldsymbol{z}_i)\end{bmatrix}\cdot\boldsymbol{F}_{ji}^T\\ &\overset{\text{no cc}}{=} \boldsymbol{F}_{ji}\cdot\begin{bmatrix}\mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{l}_j) & 0\\ 0 & \mathrm{Cov}(\boldsymbol{z}_i,\boldsymbol{z}_i)\end{bmatrix}\cdot\boldsymbol{F}_{ji}^T\\ &= \boldsymbol{G}_{ji}\cdot\mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{l}_j)\cdot\boldsymbol{G}_{ji}^T+\boldsymbol{H}_{ji}\cdot\mathrm{Cov}(\boldsymbol{z}_i,\boldsymbol{z}_i)\cdot\boldsymbol{H}_{ji}^T,\end{aligned} \tag{6.6}$$

with

$$\boldsymbol{F}_{ji} = \left.\frac{\partial\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)}{\partial(\boldsymbol{l}_j,\ \boldsymbol{z}_i)}\right|_{\boldsymbol{l}_j^0,\boldsymbol{z}_i^0} = \begin{bmatrix}\boldsymbol{G}_{ji} & \boldsymbol{H}_{ji}\end{bmatrix}$$

and $\mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{l}_j)$, $\mathrm{Cov}(\boldsymbol{l}_j,\boldsymbol{z}_i)$, and $\mathrm{Cov}(\boldsymbol{z}_j,\boldsymbol{z}_j)$ being covariances of the corresponding random variables.

A suitable choice for $\boldsymbol{l}_j^0$ and $\boldsymbol{z}_i^0$ is $\boldsymbol{l}_j$ and $\boldsymbol{z}_i$, as it is expected that the landmark estimate and measurement are both close to the ground truth values and thus approximate the function over the relevant region close to those values well.

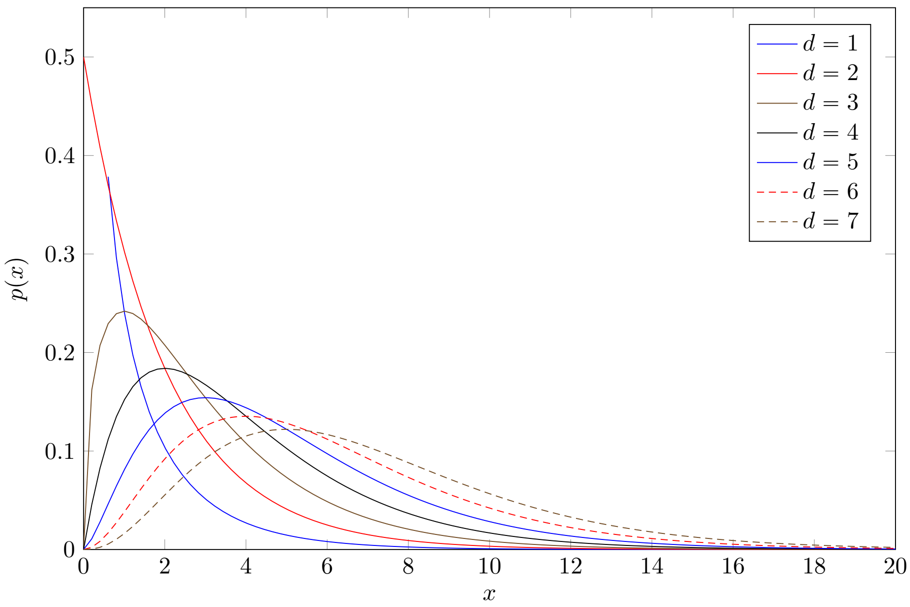
Linearization is required when non-linear vehicle motion (section 3.3) or landmark observation models (chapters 7 and 8) are utilized.

To facilitate the computational effort, it is assumed that there are no cross-correlations between landmarks $\boldsymbol{l}_j$ present within the map and their observations $\boldsymbol{z}_i$ by vehicles. This is induced by the fact that the map is built collaboratively by multiple vehicles, with principally independent observations. The fact that a vehicle may observe a landmark anew is at this neglected and justified by the fact that in practice the majority of observations is independent and the effects of self-affirmation are this way negligible. However, if self-affirmation shall be fully eliminated then anew observations of landmarks by the same vehicle have to be excluded.

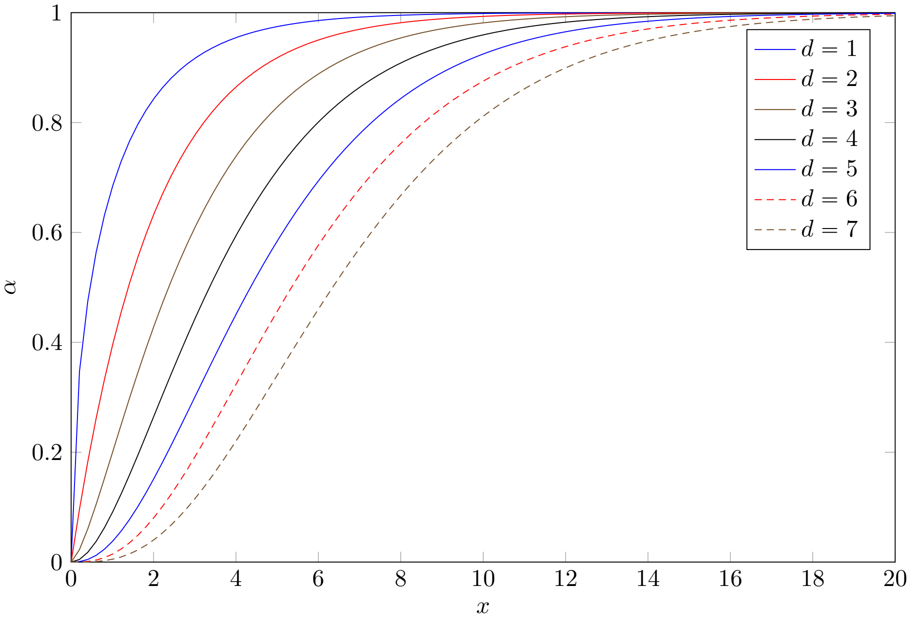
Intuitively, equation 6.4 verifies whether the observation $\boldsymbol{z}_i$ is plausible to originate from the Gaussian distribution implied by the estimated landmark $\boldsymbol{l}_j$ according to a given threshold $\chi^2_{d,\alpha}$. Both the uncertainty of the landmark observations, as well as the uncertainty of the landmark estimations, are incorporated by means of the squared Mahalanobis distance D_{ji} (appendix A.1).

The $\chi^2_{d,\alpha}$-CDF[10] allows to set the upper bound for the deviation between a landmark observation and estimate according to a desired upper error bound α, e. g. $\alpha = 0.95$. The $\chi^2_{d,\alpha}$-CDF is grounded on the accumulation of the d-dimensional χ^2_d-PDF[11]. The

[10] Cumulative Density Function (CDF)
[11] Probability Density Function (PDF)



(a)



(b)

Figure 6.1: Visualization of the χ_d^2-PDF for $d \in \mathbb{N}_{\leqslant 7}$ (subfigure a), which corresponds to the probability distribution achieved by summing random samples from a standard Gaussian distribution $\mathcal{N}(0,1)$, and the corresponding $\chi_{d,\alpha}^2$-CDF (subfigure b).

χ^2_d-PDF represents the probability distribution formed by summing random variables that have been sampled from a standard Gaussian distribution $\mathcal{N}(0,1)$. In figure 6.1 the χ^2_d-PDF and the corresponding $\chi^2_{d,\alpha}$-CDF are visualized for dimensions $d \in \mathbb{N}_{\leqslant 7}$. In the case of point-shaped landmarks, the dimensions $d = 2$ and $d = 3$ are of course of special interest. However, higher dimensions become relevant when secondary attributes shall be incorporated within the measure of correspondence (chapter 8).

On the downside, IC is restricted due to the Mahalanobis distance metric and the χ^2-test to Gaussian errors. Furthermore, the Mahalanobis distance metric is known to be sensitive to outliers. However, at least the sensitivity to outliers can be tackled by computing the Mahalanobis distance metric more robustly, e. g. by means of the MCD[12] [39].

Joint Compatibility

A disadvantage of IC is that it does only determine the degree of association quality between observed and estimated landmarks in an individual and not combined manner. In the case of ambiguities, this is known to lead to erroneous associations. Therefore, it has been suggested to jointly consider associations between multiple landmark observations and estimates. For this purpose, an extension to IC by Neira and Tardós [85], which is referred to as JC, is used:

Definition 6.3: Joint Implicit Measurement Function / Landmark Association Function

Let $\mathcal{H} = (\boldsymbol{l}_{\cdot i} \mid i = 1, \dots, m)$ be a map with the landmark estimate index wildcard $\cdot$ and $\boldsymbol{f}_{ji} = \boldsymbol{f}_{ji}(\boldsymbol{l}_j, \boldsymbol{z}_i)$, $i = 1, \dots, m$, $j = 1, \dots, n$, implicit measurement respectively landmark association functions. Then the function

$$\boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}) = \begin{pmatrix} \boldsymbol{f}_{\cdot 1} \\ \vdots \\ \boldsymbol{f}_{\cdot m} \end{pmatrix} \tag{6.7}$$

is called a *joint implicit measurement function / landmark association function.*

Definition 6.4: Stochastic Joint Compatibility Measure

The stochastic joint compatibility measure is taken as the *total joint compatibility measure of the correspondence map* $\mathcal{H}$:

$$D^2_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}) = (\boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}))^T \cdot \mathrm{Cov}(\boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}))^{-1} \cdot \boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}). \tag{6.8}$$

[12] Minimum Covariance Determinant (MCD)

Definition 6.5: Stochastic Joint Compatibility Constraint

A correspondence map $\mathcal{H}$ is *stochastically jointly compatible* if and only if

$$D^2_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}) < \chi^2_{d,\alpha}. \tag{6.9}$$

A correspondence map is *feasible with respect to joint compatibility* if it is stochastically jointly compatible.

For non-linear measurement / landmark association functions, the corresponding covariance of an assignment hypothesis $\mathcal{H}$ can be approximated under assumption of no cross-correlation by

$$\begin{aligned}
\operatorname{Cov}(\boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z})) &= \boldsymbol{F}_{\mathcal{H}} \cdot \begin{bmatrix} \operatorname{Cov}(\boldsymbol{L},\boldsymbol{L}) & \operatorname{Cov}(\boldsymbol{L},\boldsymbol{Z}) \\ \operatorname{Cov}(\boldsymbol{L},\boldsymbol{Z}) & \operatorname{Cov}(\boldsymbol{Z},\boldsymbol{Z}) \end{bmatrix} \cdot \boldsymbol{F}_{\mathcal{H}}^T \\
&\overset{\text{no cc}}{=} \boldsymbol{F}_{\mathcal{H}} \cdot \begin{bmatrix} \operatorname{Cov}(\boldsymbol{L},\boldsymbol{L}) & 0 \\ 0 & \operatorname{Cov}(\boldsymbol{Z},\boldsymbol{Z}) \end{bmatrix} \cdot \boldsymbol{F}_{\mathcal{H}}^T \\
&= \boldsymbol{G}_{\mathcal{H}} \cdot \operatorname{Cov}(\boldsymbol{L},\boldsymbol{L}) \cdot \boldsymbol{G}_{\mathcal{H}}^T + \boldsymbol{H}_{\mathcal{H}} \cdot \operatorname{Cov}(\boldsymbol{Z},\boldsymbol{Z}) \cdot \boldsymbol{H}_{\mathcal{H}}^T
\end{aligned} \tag{6.10}$$

with

$$\boldsymbol{F}_{\mathcal{H}} = \left.\frac{\partial \boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z})}{\partial(\boldsymbol{L},\ \boldsymbol{Z})}\right|_{\boldsymbol{L}_0,\,\boldsymbol{Z}_0} = \begin{bmatrix} \boldsymbol{G}_{\mathcal{H}} & \boldsymbol{H}_{\mathcal{H}} \end{bmatrix},$$

using the Taylor expansion up to the linear term

$$\boldsymbol{F}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}) \ \approx\ \boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L}_0,\ \boldsymbol{Z}_0) + \boldsymbol{G}_{\mathcal{H}} \cdot (\boldsymbol{L} - \boldsymbol{L}_0) + \boldsymbol{H}_{\mathcal{H}} \cdot (\boldsymbol{Z} - \boldsymbol{Z}_0),$$

with

$$\begin{aligned}
\boldsymbol{G}_{\mathcal{H}} &= \left.\frac{\partial \boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z})}{\partial \boldsymbol{L}}\right|_{\boldsymbol{L}_0,\,\boldsymbol{Z}_0} = \begin{pmatrix} \boldsymbol{G}_{\cdot 1} \\ \vdots \\ \boldsymbol{G}_{\cdot m} \end{pmatrix}, \\
\boldsymbol{H}_{\mathcal{H}} &= \left.\frac{\partial \boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z})}{\partial \boldsymbol{Z}}\right|_{\boldsymbol{L}_0,\,\boldsymbol{Z}_0} = \begin{pmatrix} \boldsymbol{H}_{\cdot 1} \\ \vdots \\ \boldsymbol{H}_{\cdot m} \end{pmatrix}.
\end{aligned}$$

The wildcard · denotes the index of a landmark estimate.

Neira and Tardós [85] have shown that the inverse of the approximation of $\operatorname{Cov}(\boldsymbol{f}_{\mathcal{H}}(\boldsymbol{L},\ \boldsymbol{Z}))$ can be incrementally calculated. This reduces the computational effort from $\mathcal{O}(m^3)$ to $\mathcal{O}(m^2)$ [85]. The incremental calculation uses the recursive representation

$$\boldsymbol{f}_{\mathcal{H}_t}(\boldsymbol{L},\ \boldsymbol{Z}) \ = \begin{pmatrix} \boldsymbol{f}_{\mathcal{H}_{t-1}} \\ \boldsymbol{f}_{\cdot\, t} \end{pmatrix},\ t > 1,\ \boldsymbol{f}_{\mathcal{H}_1}(\boldsymbol{L},\ \boldsymbol{Z}) \ = \ \boldsymbol{f}_{\cdot\, 1}, \tag{6.11}$$

$$\boldsymbol{G}_{\mathcal{H}_t} \ = \begin{pmatrix} \boldsymbol{G}_{\mathcal{H}_{t-1}} \\ \boldsymbol{G}_{\cdot\, t} \end{pmatrix},\ \boldsymbol{H}_{\mathcal{H}_t} \ = \begin{pmatrix} \boldsymbol{H}_{\mathcal{H}_{t-1}} \\ \boldsymbol{H}_{\cdot\, t} \end{pmatrix}. \tag{6.12}$$

where $\mathcal{H}_t$ denotes the submap of $\mathcal{H}$ restricted to the observations $\boldsymbol{z}_i \in \boldsymbol{Z}$, $i = 1, \ldots, t \leqslant m$, in particular $\mathcal{H} = \mathcal{H}_m$.

6.2.2 Data Association by Exhaustive Search

Data association can be considered as a search problem for a feasible or optimized correspondence map in the space of observation-landmark correspondence maps. The search space can be represented as a tree, the so-called *interpretation tree* [46]. Figure 6.2 shows an exemplary interpretation tree for three given landmark observations $\boldsymbol{z}_0$, $\boldsymbol{z}_1$, and $\boldsymbol{z}_2$. Each path from the root to a leaf represents a candidate correspondence map. With each incorporated landmark observation $\boldsymbol{z}_k$, which corresponds to a distinct layer within the interpretation tree, one additional possibility, besides the assignment to an already known landmark estimate $\textcircled{$\boldsymbol{l}_j$}$, that an observation is spurious $\textcircled{$*$}$ is considered. Exhaustive search checks all candidate correspondence maps for feasibility, i. e. whether the selected compatibility constraint is fulfilled, and returns those. If an optimized feasible correspondence map is desired, the total feasibility measure of the feasible correspondence maps is calculated and those with the optimal value are returned.

The number of tree paths for data association can be expressed by

$$(n+1)^m \tag{6.13}$$

with $n = |\boldsymbol{L}|$ being the number of landmark estimates and $m = |\boldsymbol{Z}|$ the number of landmark observations.

This analysis shows that the size of the search space may unfortunately grow exponentially in the size of the correspondence map. This prohibits an exhaustive search for large scale applications. For this reason, less computationally intensive, but approximating approaches have been proposed [85]: ICNN, SCNN, and JCBB. ICNN is purely based on IC, while SCNN and JCBB are based on both IC and JC.

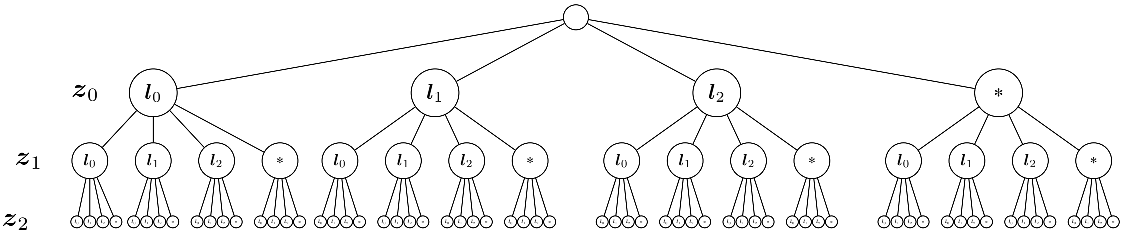


Figure 6.2: An exemplary interpretation tree for three landmark observations $\boldsymbol{z}_0$, $\boldsymbol{z}_1$, and $\boldsymbol{z}_2$. Every path between the root and the leaf nodes represents a data correspondence map, indicating the potentially exponential number of correspondence maps. For each observation, the assignment to a landmark estimate l_0, l_1, l_2, or the classification as noise * is possible.

Those algorithms deliver results of different quality with different computational requirements. The next subsection is devoted to ICNN.

6.2.3 Data Association by Individual Compatibility Nearest Neighbor

The data association algorithm ICNN is the simplest, but most efficient approximate solution of the data association problem among the alternatives for exhaustive search presented in [85]. Algorithm 6.1 shows a pseudo-code in the terminology of this thesis. ICNN uses the IC constraint (equation 6.4) for feasibility and the IC measure for optimization. ICNN has a set $\boldsymbol{L}$ of known landmark estimates, and a set of landmark observations $\boldsymbol{Z}$ as input and returns one correspondence map $\mathcal{H}$ as output.

ICNN processes the landmark observations $\boldsymbol{z}_i$ sequentially and assigns a landmark estimate $\boldsymbol{l}_i \in \boldsymbol{L}$ which is closest to $\boldsymbol{z}_i$ according to the IC measure among all given landmark estimates and which satisfies the IC constraint. If no compatible landmark exists, the version of ICNN presented here marks the observation as “spurious” in the observation map.

The runtime complexity of ICNN is bounded by $\mathcal{O}(n \cdot m)$, $n = |\boldsymbol{L}|$, $m = |\boldsymbol{Z}|$ [85].

6.3 Solutions of the Clustering Problem

“Exhaustive Search” and ICNN as presented in the preceding sections need to be extended in order to solve the clustering problem, cf. problem 6.1. A mandatory extension is that the set of landmark estimates can be extended during processing of a trajectory. The algorithm EICNN extends ICNN in this aspect, thus making ICNN basically suitable for the clustering problem. Several ideas are combined in order to achieve high quality and efficiency of the dynamic generation of landmark estimates. EICNN will be presented in subsection 6.3.2.

Besides ICNN two further algorithms from [85] are extended in this way. The first one is SCNN. While ICNN uses IC, SCNN additionally considers JC. ESCNN extends SCNN, analogously EICNN extends ICNN, cf. subsection 6.3.3. The second one is JCBB. By employing the BB[13] search paradigm, JCBB is able to correct erroneous associations, in contrast to the other approaches mentioned before. The extension EJCBB differs from JCBB by utilizing a generalized approach to BB, the so-called GBB[14] [84], in combination with the search strategy BFS[15] instead of DFS[16], cf. subsection 6.3.4. This way it is able to dynamically update the landmark map $\boldsymbol{L}$. Additionally, at each layer of the GBB tree the current set of hypotheses can be pruned to the most promising ones, e. g. for limiting the required computational resources and/or assuring real-time feasibility.

All those algorithms are combined with a separation of the observations of the input trajectory into substreams of observations of landmarks of the same type, based on landmark type attributes, cf. section 6.3.6. Each substream is processed separately. All algorithms are implemented based on a map data structure which employs submapping according to the approach of section 4.2, cf. subsection 6.3.7. Furthermore, temporal decay of landmark observations and estimates is considered by using the temporal weighting approach of section 4.3, cf. subsection 6.3.5. All this improves the computational efficiency of clustering.

The performance of the algorithms presented in this section have been experimentally analyzed. The design of the experiments and the results will be presented in subsection 6.3.8.

6.3.1 Clustering by Exhaustive Search

Clustering can be as well considered as a search problem for a feasible or optimized correspondence map in the space of observation-landmark correspondence maps. The search space can be again represented as a tree. Figure 6.3 shows an exemplary tree for three given landmark observations $\boldsymbol{z}_0$, $\boldsymbol{z}_1$, and $\boldsymbol{z}_2$. Each path from the root to a leaf represents a candidate correspondence map. With each incorporated landmark observation $\boldsymbol{z}_k$, which corresponds to a distinct layer within the tree, two additional possibilities, besides the assignment to an already known landmark estimate $\textcircled{\scriptsize $\boldsymbol{l}_j$}$, are considered:

- an observation is constituting a new landmark estimate $\textcircled{\scriptsize $\boldsymbol{l}_k$}$,
- an observation is spurious $\textcircled{\scriptsize $*$}$.

[13] Branch and Bound (BB)
[14] General Branch and Bound (GBB)
[15] Breadth-First Search (BFS)
[16] Depth-First Search (DFS)

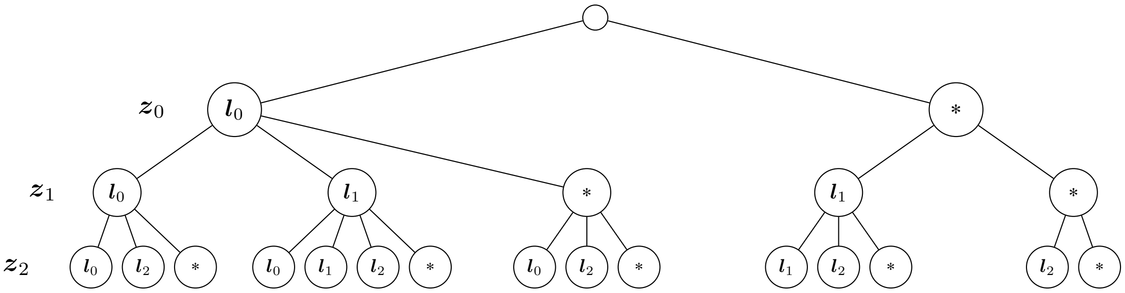


Figure 6.3: An exemplary clustering via exhaustive search for three landmark observations $\boldsymbol{z}_0$, $\boldsymbol{z}_1$, and $\boldsymbol{z}_2$. Every path between the root and the leaf nodes represents a data correspondence map, indicating the potentially exponential number of correspondence maps. For each observation, the assignment to a previous/new landmark $\textcircled{l_0}$, $\textcircled{l_1}$, $\textcircled{l_2}$, or the classification as noise $\textcircled{*}$ is possible.

The number of tree paths for clustering can be recursively expressed by

$$|\mathcal{X}_k| = \sum_{\Lambda_{\mathcal{H}},\, \mathcal{H} \in \mathcal{X}_{k-1}} \left(|\{\boldsymbol{l}_j \in \Lambda_{\mathcal{H}}\}| + 2\right). \tag{6.14}$$

$\mathcal{X}_k$ denotes the set of hypotheses $\mathcal{H}$ existing after incorporating observation $\boldsymbol{z}_k$, and $\Lambda_{\mathcal{H}}$ the set of landmark estimates occurring in $\mathcal{H}$. Hence the expression $|\{\boldsymbol{l}_j \in \Lambda_{\mathcal{H}}\}|$ denotes the number of identified landmark estimates within the hypothesis $\mathcal{H}$.

This analysis shows that the size of the search space may unfortunately grow exponentially in the size of the correspondence map. This prohibits a clustering via exhaustive search for large scale applications.

6.3.2 Extended Individual Compatibility Nearest Neighbor

The algorithm EICNN, cf. algorithm 6.2, extends ICNN in two major aspects: dynamic landmark estimation and improved efficiency and robustness. Algorithm 6.2 shows a pseudocode formulation of EICNN.

Adding new landmark estimates has two aspects, both under consideration of computational efficiency and robustness: *conditions to be fulfilled for adding a new landmark estimate*, and *calculation of the representation of a new landmark*.

During processing of the input stream of landmark observations, a landmark is either associated with an already existing landmark estimate, or a new landmark estimate is created. The decision is taken depending on *three conditions* on the currently processed landmark observation which have to be fulfilled for adding a new landmark estimate:

Solution 6.1: Clustering Conditions

IC-Condition: The landmark observation $\boldsymbol{z}_j$ does not fulfill the IC-test (definition 6.2). The reason for this condition is that it could be that a landmark estimate is among the already existing ones which can be associated with the landmark observation.

Minimal Support Condition: The number of other landmark observations $\boldsymbol{z}_k$ in the input sequence $\boldsymbol{Z}$, which are supporting the landmark observation, exceeds a given threshold $\kappa \geqslant 0$. $\boldsymbol{z}_k$ supports $\boldsymbol{z}_j$ if the bounding boxes of their uncertainty ellipses (appendix A.2 and A.4) have a non-empty intersection. The reason of this condition is to make the algorithm more robust against spurious landmark observations.

Distance Condition: The observation has a minimal distance to existing landmark estimates of the same type and with the same attributes. This condition limits the risk of undesired cluster splits.

If all conditions are fulfilled, a new landmark estimate is created. Its representation by a location and a covariance is calculated as the first (mean) and second moment (covariance) of the locations of the landmark observations of its cluster by means of averaging. This simple approach avoids the execution of a fusion algorithm like the one of chapter 7 in this phase, what usually would be computationally too expensive [72].

Three approaches to increasing the *efficiency of calculations* are used:

Solution 6.2: Improving the Efficiency of Clustering Calculations

Efficient Exclusion of Associations of High Uncertainty: An efficient pre-test of the compatibility of a landmark observation and a landmark estimation candidate is introduced. The test consists in checking the bounding boxes of the uncertainty ellipses (appendix A.2) of the two items for non-empty intersection. The uncertainty ellipses are determined from the mean and covariance information of the items and for a given sigma range, e. g. 3σ. The reason of this extension is to reduce the number of executions of computationally expensive matrix inversions for the calculation of the Mahalanobis distance in the case of low compatibility.

Efficient Analytical Calculation of the Bounding Box of an Arbitrary Ellipse: This is employed for the intersection tests incorporated. The details are provided in appendix A.4.

Efficient Search for Intersecting Bounding Boxes: An R-tree [49] is used to find those bounding boxes of a set of bounding boxes which intersect a given

```
input  : $(\boldsymbol{L},\ \boldsymbol{Z})$
output : $\mathcal{H}$
① Iterate over all landmark observations
for $\boldsymbol{z}_i \in \boldsymbol{Z}$ do
    $\chi^{\text{min}}_{\text{val}} = \infty$ ;
    $\chi^{\text{min}}_{\text{idx}} = -1$ ;
    ② Iterate over current map
    for $\boldsymbol{l}_j \in \boldsymbol{L}$ do
        — Compute Mahalanobis distance and apply $\chi^2$-test to pairing
        $\chi^{\text{min}}_{\text{current}} = \left(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\right)^T \cdot \text{Cov}(\boldsymbol{l}_j,\ \boldsymbol{z}_i) \cdot \boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)$ ;
        if $\left(\chi^{min}_{current} < \chi^{min}_{val}\right)$ and $\left(\chi^{min}_{current} < \chi^2_{d,\alpha}\right)$ then
            $\chi^{\text{min}}_{\text{val}} = \chi^{\text{min}}_{\text{current}}$ ;
            $\chi^{\text{min}}_{\text{idx}} = j$ ;
        end
    end
    if $\chi^{min}_{idx} = -1$ then
        — Mark observation as spurious
        $\mathcal{H} = \mathcal{H} \cup (*)$ ;
    else
        — Assign observation to an existing landmark
        $\mathcal{H} = \mathcal{H} \cup \left(\chi^{\text{min}}_{\text{idx}}\right)$ ;
    end
end
```

Algorithm 6.1: Detailed procedure of ICNN according to Neira and Tardós [85]. As input, the algorithm requires a set of known landmarks $\boldsymbol{L}$ and a set of landmark observations $\boldsymbol{Z}$. As output, it returns one greedy assignment hypothesis $\mathcal{H}$. ICNN performs data associations by assigning observations to their nearest neighbors by means of the Mahalanobis distance metric, if and only if the associations successfully fulfill the χ^2-test. The runtime complexity is bounded by $\mathcal{O}(|\boldsymbol{L}| \cdot |\boldsymbol{Z}|)$ [85].

bounding box. This reduces the average lookup time of the intersection test to $\mathcal{O}(M \cdot \log_M(n))$ instead of $\mathcal{O}(n)$, where n is the number of tree entries and M a constant denoting the maximal chunk size of the R-tree data structure.

The resulting EICNN is more robust, cf. solution 6.1 and efficient, cf. solution 6.2, than ICNN. However, its weaknesses of testing associations only individually and its greedy approach have not been alleviated this way.

6.3.3 Extended Sequential Compatibility Nearest Neighbor

The only difference between ESCNN, cf. algorithm 6.3, and EICNN, cf. algorithm 6.2, is that an additional test for the JC is incorporated after IC has been successfully passed. As noted in section 6.2.1, JC is tighter than IC, thus possibly avoiding inappropriate associations.

```
input  : $(\boldsymbol{L},\ \boldsymbol{Z})$
output : $\mathcal{H}$
① Iterate over all landmark observations
for $\boldsymbol{z}_i \in \boldsymbol{z}$ do
    $\chi^{\text{min}}_{\text{val}} = \infty$ ;
    $\chi^{\text{min}}_{\text{idx}} = -1$ ;
    ② Selectively iterate over current map
    for $\boldsymbol{l}_j \in \{\boldsymbol{l}_j \mid \boldsymbol{l}_j \in \boldsymbol{L},\ \mathit{fulfillsConstraints}(\boldsymbol{l}_j,\ \boldsymbol{z}_i) \wedge \mathit{intersects}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\}$ do
        — Test for individual compatibility
        $\chi^{\text{min}}_{\text{current}} = \left(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\right)^T \cdot \operatorname{Cov}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)^{-1} \cdot \boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)$ ;
        if $\left(\chi^{min}_{current} < \chi^{min}_{val}\right)$ and $\left(\chi^{min}_{current} < \chi^2_{d,\alpha}\right)$ then
            $\chi^{\text{min}}_{\text{val}} = \chi^{\text{min}}_{\text{current}}$ ;
            $\chi^{\text{min}}_{\text{idx}} = j$ ;
        end
    end
    if $\chi^{min}_{idx} = -1$ then
        — If enough support and minimum distance constraint fulfilled
          then assign observation to a new landmark
        if $|\{\boldsymbol{z}_k \mid \boldsymbol{z}_k \in \boldsymbol{Z},\ \mathit{intersects}(\boldsymbol{z}_i,\ \boldsymbol{z}_k)\}| \geqslant \kappa$ and $\forall\, \boldsymbol{l}_j \in \boldsymbol{L}$ $\mathit{fulfillsMinDistance}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)$
          then
            $\mathcal{H} = \mathcal{H} \cup (\chi^{\text{new}}_{\text{idx}})$ ;
        end
    else
        — Assign observation to an existing landmark
        $\mathcal{H} = \mathcal{H} \cup \left(\chi^{\text{min}}_{\text{idx}}\right)$ ;
    end
    ③ Update map
    $\boldsymbol{L} = \mathit{updateMapByHeuristic}(\boldsymbol{L},\ \boldsymbol{z}_i,\ \boldsymbol{C}(\boldsymbol{z}_i),\ \chi^{\text{min}}_{\text{idx}})$ ;
end
```

Algorithm 6.2: Outline of the extended version of ICNN, referred to as EICNN. As input, the algorithm requires a set of known landmarks $\boldsymbol{L}$ (which can also be empty) and a set of landmark observations $\boldsymbol{Z}$. As output, it returns one greedy assignment hypothesis $\mathcal{H}$. The algorithm provides a more efficient iteration over the current landmark map via an R-tree based intersection test, a minimal support (denoted by κ) test for the instantiation of new landmarks, an exclusion of observations with a high uncertainty, an incorporation of a heuristic utilizing the first and second moment for updating the current map, and the incorporation of constraints by augmenting the map iteration (solution 6.1 and 6.2). The runtime complexity is bounded by $\mathcal{O}(|\boldsymbol{L}| \cdot |\boldsymbol{Z}|)$ [85].

```
input  : $(\boldsymbol{L},\ \boldsymbol{Z})$
output : $\mathcal{H}$
① Iterate over all landmark observations
for $\boldsymbol{z}_i \in \boldsymbol{Z}$ do
    $\chi^{\text{min}}_{\text{val}} \ = \ \infty$ ;
    $\chi^{\text{min}}_{\text{idx}} \ = \ -1$ ;
    ② Selectively iterate over fused map
    for $\boldsymbol{l}_j \in \{\boldsymbol{l}_j \mid \boldsymbol{l}_j \in \boldsymbol{L},\ \textit{fulfillsConstraints}(\boldsymbol{l}_j,\ \boldsymbol{z}_i) \wedge \textit{intersects}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\}$ do
        — Test for individual compatibility
        $\chi^{\text{min}}_{\text{current}} = \left(\boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)\right)^T \cdot \text{Cov}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)^{-1} \cdot \boldsymbol{f}_{ji}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)$
        if $\left(\chi^{min}_{current} \geqslant \chi^{min}_{val}\right)$ or $\left(\chi^{min}_{current} \geqslant \chi^2_{d,\alpha}\right)$ then
            continue;
        end

        — Formulate hypothesis
        if $\chi^{min}_{idx} = -1$ then
            — If enough support and minimum distance constraint fulfilled
              then assign observation to a new landmark
            if $|\{1 \mid \boldsymbol{z}_k \in \boldsymbol{Z},\ \textit{intersects}(\boldsymbol{z}_i,\ \boldsymbol{z}_k)\}| < \kappa$ or
              $\exists\, \boldsymbol{l}_j \in \boldsymbol{L}\ \neg \textit{fulfillsMinDistance}(\boldsymbol{l}_j,\ \boldsymbol{z}_i)$ then
                continue;
            end
            $\mathcal{H}_{\text{new}} = \mathcal{H} \cup (\chi^{\text{new}}_{\text{idx}})$ ;
        else
            — Assign observation to an existing landmark
            $\mathcal{H}_{\text{new}} = \mathcal{H} \cup \left(\chi^{\text{min}}_{\text{idx}}\right)$ ;
        end

        — Test hypothesis for joint compatibility
        if $\boldsymbol{f}^T_{\mathcal{H}_{new}} \cdot \text{Cov}(\boldsymbol{L}_{\mathcal{H}_{new}}, \boldsymbol{Z}_{\mathcal{H}_{new}})^{-1} \cdot \boldsymbol{f}_{\mathcal{H}_{new}} \geqslant \chi^2_{d,\alpha}$ then
            continue;
        end
        $\mathcal{H} \quad = \mathcal{H}_{\text{new}}$ ;
        $\chi^{\text{min}}_{\text{val}} = \chi^{\text{min}}_{\text{current}}$ ;
        $\chi^{\text{min}}_{\text{idx}} = j$ ;
    end
    ③ Update fused map
    $\boldsymbol{L} = \textit{updateMapByHeuristic}(\boldsymbol{L},\ \boldsymbol{z}_i,\ \boldsymbol{C}(\boldsymbol{z}_i),\ \chi^{\text{min}}_{\text{idx}})$ ;
end
```

Algorithm 6.3: Outline of the extended version of SCNN, referred to as ESCNN. As input, the algorithm requires a set of known landmarks $\boldsymbol{L}$ (which can also be empty) and a set of landmark observations $\boldsymbol{Z}$. As output, it returns one greedy assignment hypothesis $\mathcal{H}$. The algorithm features the same improvements as introduced by EICNN. However, it does additionally incorporate JC for gaining a globally consistent hypothesis. The runtime complexity is bounded by $\mathcal{O}(|\boldsymbol{L}|\cdot|\boldsymbol{Z}|)+O(|\boldsymbol{L}|^2\cdot|\boldsymbol{Z}|)$ [85].

```
input  : $(\boldsymbol{L},\ \boldsymbol{Z})$
output : $\mathcal{H}_{\text{best}}$
$\mathcal{H}_{\text{branches}} = \{\mathcal{H}_0 = \{\}\}$ ;
① Traverse and expand tree (until maximal hypothesis length is reached)
for $k = 0; k < |\boldsymbol{Z}|; k = k + 1$ do
    $\mathcal{H}^{\text{new}}_{\text{branches}} = \{\}$ ;
    ② Iterate over all available hypotheses
    for $\mathcal{H}_i \in \mathcal{H}_{branches}$ do
        ③ Iterate over hypothesis and expand it (branch)
        for $\boldsymbol{l}_j \in \mathcal{H}_i \cup \{*,\ \boldsymbol{l}_k\}$ do
            $\mathcal{H}^{\text{new}}_i = \mathcal{H}_i \cup (\boldsymbol{l}_j)$ ;

            — Test individual compatibility of hypothesis (bound)
            if not individuallyCompatible($\boldsymbol{z}_k,\ \mathcal{H}^{\text{new}}_i$) then
                continue;
            end

            — Test joint compatibility of hypothesis (bound)
            if not jointlyCompatible($\boldsymbol{z}_k,\ \mathcal{H}^{\text{new}}_i$) then
                continue;
            end
            $\mathcal{H}^{\text{new}}_{\text{branches}} = \mathcal{H}^{\text{new}}_{\text{branches}} \cup \mathcal{H}^{\text{new}}_i$ ;
        end
    end
    $\mathcal{H}_{\text{branches}}$ = selectNBestHypotheses($\mathcal{H}^{\text{new}}_{\text{branches}},\ \eta$) ;
end
④ Return best hypothesis from $\mathcal{H}_{branches}$
$\mathcal{H}_{\text{best}}$ = selectBestHypothesis($\mathcal{H}_{\text{branches}}$) ;
```

Algorithm 6.4: Outline of the extended version of JCBB, referred to as EJCBB. As input, the algorithm requires a set of known landmarks $\boldsymbol{L}$ (which can also be empty) and a set of landmark observations $\boldsymbol{Z}$. The algorithm incorporates a BFS search strategy instead of DFS for traversing the exponential solution space via a generalization of BB, GBB [84], to allow a dynamic updating of the landmark map $\boldsymbol{L}$. Additionally, it includes the same enhancements as previously introduced to EICNN and ESCNN. It tracks η assignment hypotheses simultaneously and provides the best one as the final output $\mathcal{H}_{\text{best}}$.

The runtime of ESCNN is bounded by $\mathcal{O}(|\boldsymbol{L}| \cdot |\boldsymbol{Z}|) + O(|\boldsymbol{L}|^2 \cdot |\boldsymbol{Z}|)$ [85].

(E)SCNN can be still classified as a greedy algorithm, as data associations are never reconsidered [85].

6.3.4 Extended Joint Compatibility Branch & Bound

Both EICNN and ESCNN exhibit a crucial drawback: erroneous updates of the current landmark map $\boldsymbol{L}$, e. g. due to ambiguities, cannot be remedied anymore later on [16]. Therefore, it is very likely that both approaches induce divergence of the subsequent fusion algorithms and, therefore, deviations of the mapped landmarks according to their ground truths.

The problem can be tackled by considering multiple data association hypotheses simultaneously by exploring the search space in a BB-manner [68]. The two main search strategies to BB are DFS and BFS [118]. The algorithm JCBB by Neira and Tardós [85] employs DFS. In contrast to JCBB, EJCBB, cf. algorithm 6.4, incorporates a generalization of BB, the so-called GBB [84]. In combination with the search strategy BFS, EJCBB is able to dynamically update the landmark map $\boldsymbol{L}$ and, therefore, to provide a clustering instead of pure data association.

In the case of limited computational resources and/or real-time requirements, the number of hypotheses at each layer of the interpretation tree of EJCBB can be additionally diminished to the η most promising ones by incorporating a novel heuristic. The heuristic is based upon a multi-objective cost function:

$$c(\mathcal{H}) = \omega_{\text{IC}} \cdot \nu_{\text{IC}}(\mathcal{H}) + \omega_{\text{JC}} \cdot \nu_{\text{JC}}(\mathcal{H}) + \omega_{*} \cdot \nu_{*}(\mathcal{H}) + \omega_{\text{new}} \cdot \nu_{\text{new}}(\mathcal{H}).$$

ν_{IC} is the IC-value (section 6.2.1), ν_{JC} the JC-value (section 6.2.1), ν_{*} the noise ratio (percentage of noise classifications measured by the total observation count), and ν_{new} the cluster-ratio (percentage of landmark estimates measured by the total observation count). The non-negative weights ω_{IC}, ω_{JC}, ω_{*}, and ω_{new} are determined in a supervised manner based upon labeled ground truth data, as performed within the evaluation (section 6.3.8).

The suggested extension of the pure landmark association algorithm JCBB to a full landmark clustering algorithm, EJCBB, can be therefore regarded as well suited for large-scale scenarios under potential computational resource limitations and/or real-time constraints.

6.3.5 Temporal Decay

The road network context is exposed to more or less variation over time, for example, contingent upon roadworks. Therefore, it is crucial to compensate for temporal dynamics during data association/clustering.

This is achieved concerning EICNN, ESCNN, and EJCBB in two ways

- the processing of the observations $\boldsymbol{z}_j$ takes places in their chronological order,
- and the landmark estimates $\boldsymbol{l}_i$ and observations $\boldsymbol{z}_j$ are weighted according to their acquisition timestamps by augmenting the Mahalanobis distance metric.

By utilizing temporal weights ω_{l_i}, $\omega_{z_j} \in [0, 1]$ and augmenting the Mahalanobis distance metric based weighting factors by $(1 + \gamma \cdot (1 - \omega_{l_i})) \cdot \mathrm{Cov}(\boldsymbol{l}_i)$ respectively $(1 + \gamma \cdot (1 - \omega_{z_j})) \cdot \mathrm{Cov}(\boldsymbol{z}_i)$ instead of $\mathrm{Cov}(\boldsymbol{l}_i)$ respectively $\mathrm{Cov}(\boldsymbol{z}_j)$, EICNN, ESCNN, and EJCBB are able to consider the variation of landmarks over time. γ denotes a spreading factor, which is individually determined for each covariance $\mathrm{Cov}(\boldsymbol{l}_i)$ respectively $\mathrm{Cov}(\boldsymbol{z}_j)$ so that the cut-off value is reached when the temporal weights are equal to zero. This way, the clustering is able to adapt to altered landmarks and to forget about removed ones.

The different decay rates of distinct road network features, for which the weighting scheme needs to adapt to, can be considered as a challenge. For example, moving construction zones exhibit a higher temporal dynamic than road signs. To approach this challenge, an efficient temporal weight computation based upon preaggregation is utilized (section 4.3).

6.3.6 Landmark Type Attributes

As aforementioned, constraints constitute a crucial aspect that needs to be considered when applying data association/clustering to CVD. This is motivated by the fact that landmarks appertaining to the same class are, unfortunately, also sporadically observed nearby to each other within the road network context. To take a single example, a speed limit and an overtaking restriction sign plate, which are located close to each other, however, that do *not* share a common pole, should be in any case reflected by distinct clusters. Otherwise, the subsequent fusion of these observations (chapter 7) would lead to inaccurate results. This can be considered as highly undesirable in the case that these landmarks are e. g. utilized for a highly precise vehicle localization.

For that reason, the data association/clustering is extended to eventual additional attributes, such as the type of a sign. The data association/clustering is only performed when attributes match, e. g. the sign type, or their deviations are within an expected range, e. g. the orientation of a traffic sign. Sporadically, spurious

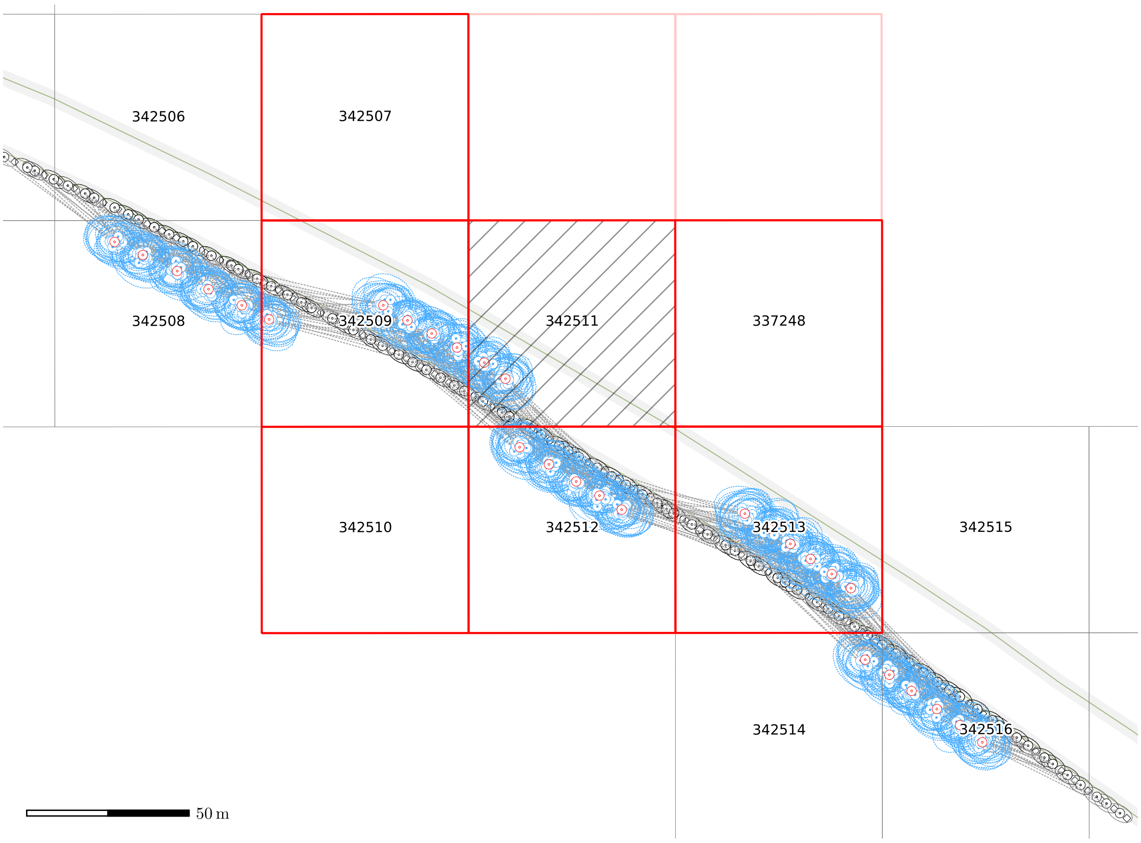


Figure 6.4: Application of global submapping to data association/clustering for an exemplary road excerpt. Data association/clustering is performed independently for every grid cell hit by an observation, but under consideration of its 8-neighborhood to achieve global consistency. The currently inspected grid cell is hachured in gray ●, while its 8-neighborhood is highlighted in red ●.

landmark observations occur, what can also affect their attributes. Therefore, it is also crucial to augment the minimal support test (sections 6.3.2, 6.3.3, and 6.3.4), which is performed by the extended approaches prior to appending an additional landmark to the map.

6.3.7 Submapping

EICNN, ESCNN, and EJCBB are principally global algorithms when utilized out-of-the-box. Since the application area of fusing CVD requires the successful tackling of large-scale regions, the scalability of the utilized algorithms is necessary. For this purpose, the global submapping by adaptive partitioning described in section 4.2 is employed. The approach partitions the landmark map by a regular grid into cells. A data association/clustering algorithm is executed on the region of the cell affected by the current landmark observation and its 8-neighborhood of cells (figure 6.4). To assure global consistency, measurements with a huge uncertainty, this means

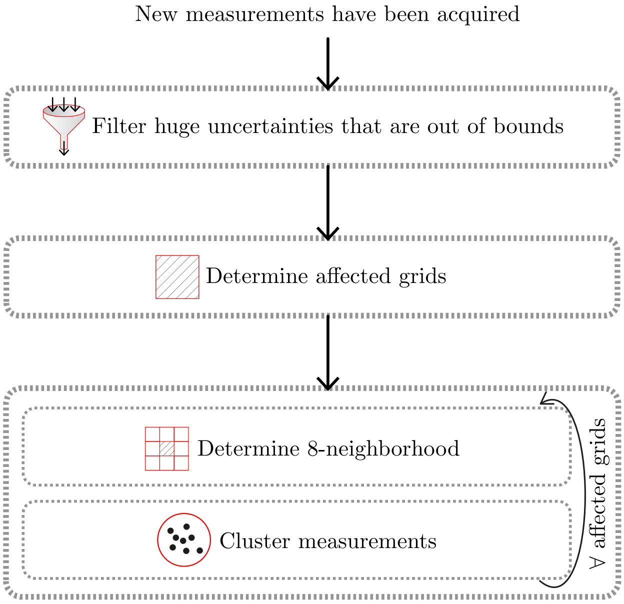


Figure 6.5: The steps of scalable data association/clustering by global submapping. First, observations with uncertainties exceeding the allowed bounds, according to the chosen parametrization of the adaptive partitioning (section 4.2), are removed. Next, the affected grid cells and the corresponding 8-neighborhood-cells are determined. Finally, data (re)association/(re)clustering is applied to the 8-neighborhood.

uncertainties that exceed the 8-neighborhood, are excluded because they would otherwise interfere with the region of processing.

Figure 6.5 gives an overview of the general steps performed by all the data association/clustering algorithms. First, the observations with a huge uncertainty are removed. Next, data association/clustering is applied to each affected 8-neighborhood. Finally, the associated/clustered observations are fused by means of algorithms intended for the aggregation of point-shaped (chapter 7) or complex landmarks (chapter 8).

6.3.8 Evaluation

In the following, the clustering by EICNN and EJCBB is evaluated based on a synthetic scenario with known ground truth. ESCNN is excluded, as it just extends EICNN by the JC-test, so its significance for the actual evaluation can be regarded as minor. For the evaluation of EICNN and EJCBB, first optimal parameterizations for both algorithms are derived by means of labeled ground truth data. Afterwards, both algorithms are evaluated according to a selection of distinct quality measures, which are regarded as highly relevant for the fusion of CVD.

Quality Measures

The quality of the data association/clustering can be principally assessed in manifold ways. However, as it is striven for a high accuracy so that fused CVD can be subsequently utilized for demanding applications, such as highly precise landmark-based vehicle localization, the following indicators are chosen:

Definition 6.6: Quality Measures for Data Association/Clustering

False positives: count of incorrect associations,

Precision: $\phi = \frac{\text{true positives}}{\text{true positives } \cup \text{ false positives}}$, and

Recall: $\rho = \frac{\text{true positives}}{\text{true positives } \cup \text{ false negatives}}$.

The classification of observations as noise does neither account for the incorrect associations count nor for the corresponding precision value ϕ, as both do not include false negatives. However, the recall value ρ does account for noise classifications.

Additionally, the commonly utilized F_β-score [130] is employed, which is based upon the precision ϕ and recall value ρ:

Definition 6.7: F_β-score

$$F_\beta = (1 + \beta^2) \cdot \frac{\phi \cdot \rho}{\beta^2 \cdot \phi + \rho}.$$

As the precision ϕ has a hight priority in the context of the fusion of CVD, $\beta = 0.1$ is chosen to weight the precision within the F_β-score ten times as high as the recall ρ. The justification thereof is that more than enough observations are propagated to the back-end-side so that the data association/clustering can be very selective. Finally, since “large-scale” is in the focus of interest, the cumulative runtime of all computing units is provided as a crucial indicator.

Optimal Parametrization

EICNN and EJCBB both have multiple parameters which need to be adjusted appropriately. Some of those parameters can be set based on system knowledge, others need to be empirically determined. The latter parameters are derived in an optimizing manner by means of a grid search. The utilization of a computationally expensive grid search, instead of other more target-oriented optimization approaches, is motivated by its comprehensive coverage of the search space to determine a global optimum. The selected parameters are evaluated according to the hereinafter evolved quality measures.

Evaluation Scenario

A partially synthetic evaluation scenario (figure 6.6), hereinafter referred to as "synthetic scenario", is defined. The scenario has 29 simulated point-shaped landmarks which were observed 2560 times in total relatively to multiple real-world trajectories. The average standard deviations of the vehicle localization and of the landmark observations are approximately 1.06 m and approximately 2.19 m, respectively. The motivation for utilizing a partially synthetic scenario for the evaluation is

- an assured knowledge about the actual ground truth and
- the possibility of designing a highly challenging environment which features many nearby landmarks, as commonly *not* observed within the road network context.

The landmark observations are sampled according to a distance-dependent two-dimensional Gaussian distribution. The Gaussian distribution is aligned so that the major axis of the corresponding error ellipse (appendix A.2) coincides with the view vector from the vehicle to the landmark. The actual standard deviations, which are represented by the major axis $f_{\text{major}}(\delta)$ and minor axis $f_{\text{minor}}(\delta)$, are determined by the following linear functions according to the length of the view vector δ:

$$\begin{aligned} f_{\text{major}}(\delta) &= 0.2 \cdot \delta + 1, \\ f_{\text{minor}}(\delta) &= 0.1 \cdot \delta + 1. \end{aligned} \tag{6.15}$$

This approach is justified by the fact that the angular error of a TSR[17]-camera is commonly minor in comparison to the distance error.

Extended Individual Compatibility Nearest Neighbor

This section reports on the evaluation of EICNN on the synthetic scenario (section 6.3.8) based upon 100, 500, 1000, 1500, and 2000 landmark observations that are randomly sampled 50 times each.

EICNN is parametrized so that a minimal support κ of new clusters by at least ten observations is required. Observations with an uncertainty exceeding 2 m are a priori classified as noise and, therefore, directly excluded from the correspondence determination. Furthermore, the cut-off value utilized by the IC-test is set to $\chi^2_{d=2,\,\alpha=0.93}$ (section 6.2.1). α has been determined via grid-search.

In figure 6.7 the average incorrect association count, precision, recall, and $F_{0.1}$-score is provided. One can notice that the incorrect association count is always equal to zero, while the precision is equal to one, from 500 observations on. 100 observations can be

[17] Traffic Sign Recognition (TSR)

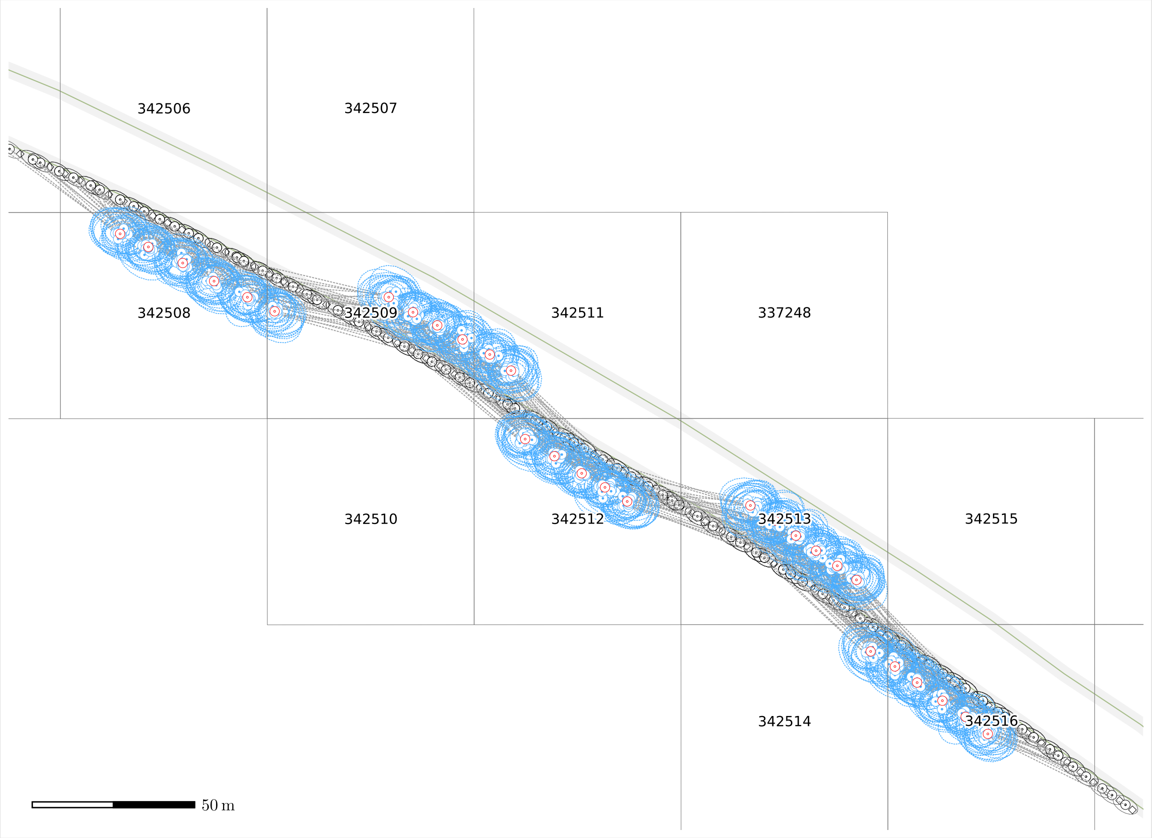
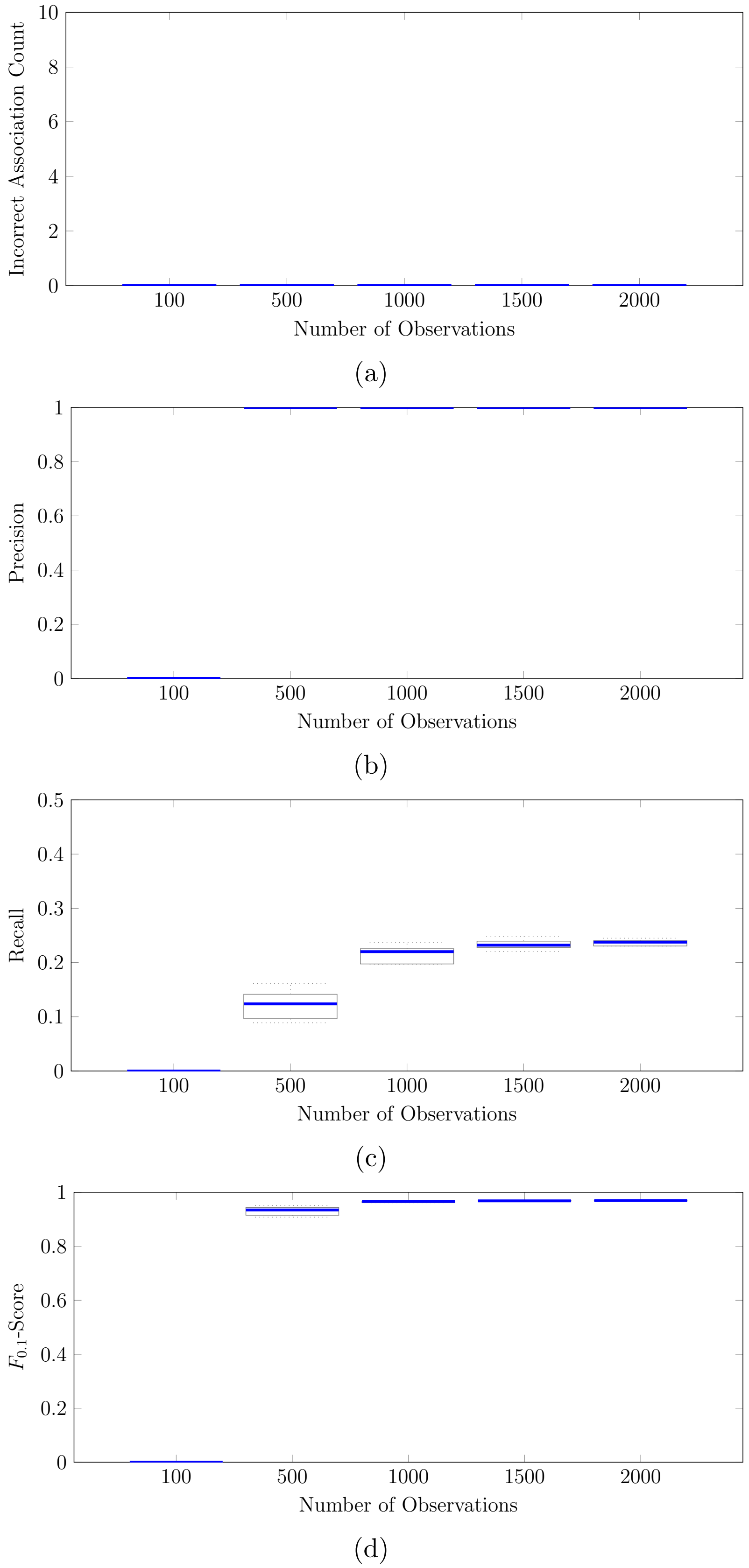


Figure 6.6: A partially synthetic scenario exhibiting 29 point-shaped landmarks and the corresponding uncertainties at 3σ. Their ground truths are visualized by red circles ○ and their observations by blue circles ○.



(a)

(b)

(c)

(d)

Figure 6.7: Accuracy evaluation of EICNN. 100, 500, 1000, 1500, and 2000 landmark observations are randomly sampled for fifty times each from the synthetic scenario (figure 6.6). The *averaged* results are then displayed by Box-Whisker-Plots (appendix A.5).

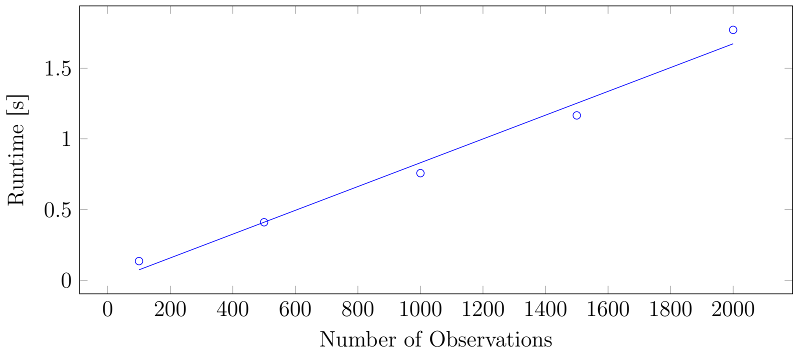


Figure 6.8: Runtime evaluation of EICNN for 100, 500, 1000, 1500, and 2000 landmark observations randomly sampled for fifty times each from the synthetic scenario (figure 6.6), and the corresponding linear regression.

considered as not sufficient for determining the clusters in a robust manner (minimal support $\kappa \geqslant 10$) and the algorithm behaves, therefore, as desired by classifying all observations as noise. Furthermore, one can observe that the $F_{0.1}$-score converges rapidly to 0.96, which is very close to the optimal score of one.

In figure 6.8 the average runtimes are provided. One can notice that EICNN scales linearly with the observation count and requires an average runtime of 1.77 s for 2000 observations (cumulative runtime of all computing units).

Extended Joint Compatibility Branch and Bound

In the following, EJCBB is evaluated on the synthetic scenario (section 6.3.8) based upon 100, 500, 1000, 1500, and 2000 landmark observations that are randomly sampled for 50 times each.

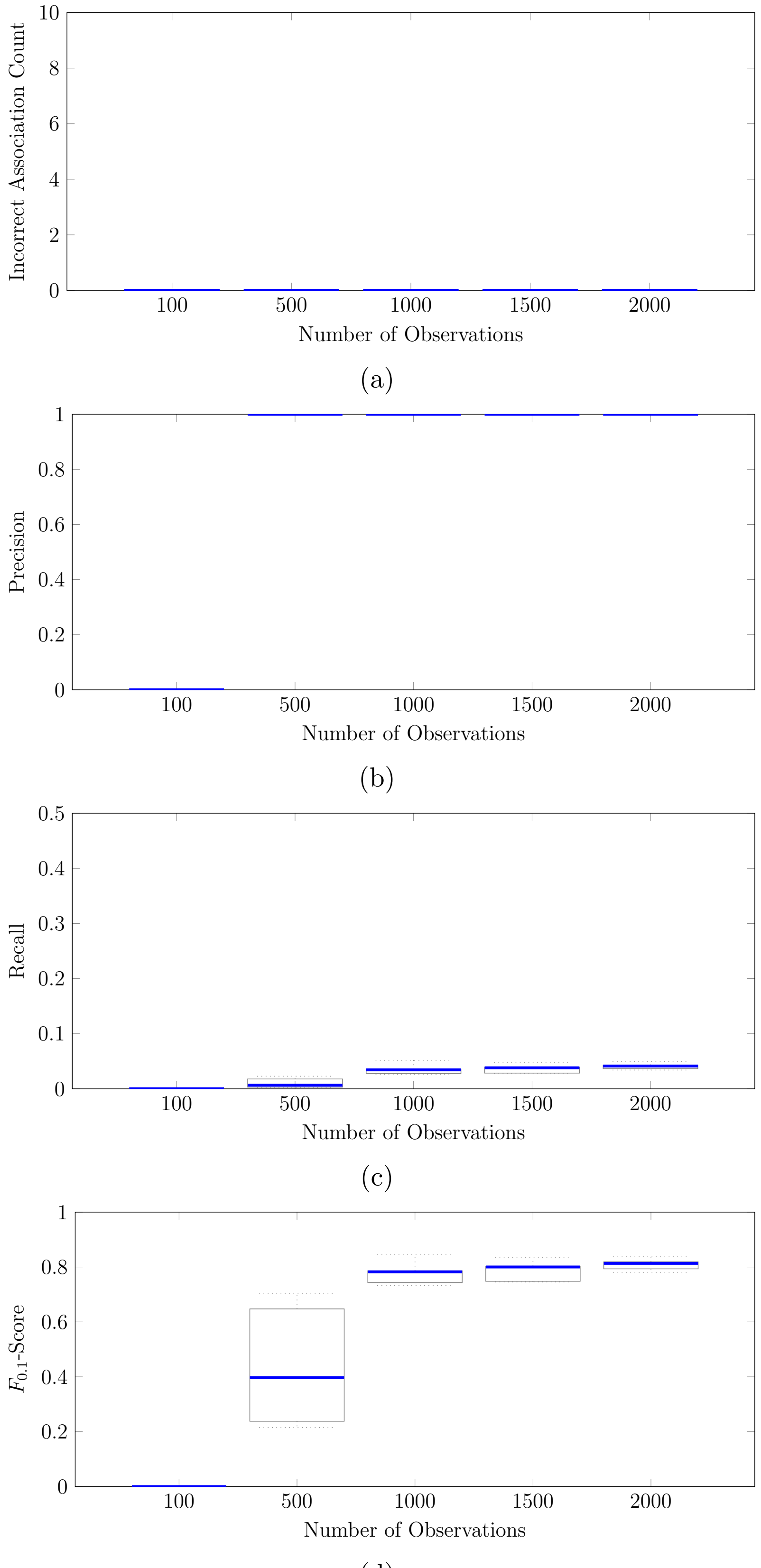
EJCBB is parametrized so that observations with an uncertainty exceeding 2 m are a priori classified as noise and, therefore, excluded from the subsequent observation fusion. Furthermore, the cut-off value utilized by the IC- and JC-tests is set to $\chi^2_{d,\,\alpha=0.93}$ (section 6.2.1).

The number of simultaneously tracked hypotheses is limited to $\eta = 10$ to assure computational feasibility. The ten most promising hypotheses are then selected by the as following parametrized cost function (section 6.3.4):

$$c(\mathcal{H}) = \underbrace{1.5}_{\omega_{\mathrm{IC}}} \cdot \nu_{\mathrm{IC}} + \underbrace{1.5}_{\omega_{\mathrm{JC}}} \cdot \nu_{\mathrm{JC}} + \underbrace{0.5}_{\omega_{*}} \cdot \nu_{*} + \underbrace{1.5}_{\omega_{\mathrm{new}}} \cdot \nu_{\mathrm{new}}.$$

The parameters ω_{IC}, ω_{JC}, ω_{*} and ω_{new} were also determined in an optimal manner by means of a grid search.

This approach deviates significantly from the one suggested by Neira and Tardós [85], as their approach assesses the quality of a hypothesis solely by a low noise count.



(a)

(b)

(c)

(d)

Figure 6.9: Accuracy evaluation of EJCBB. 100, 500, 1000, 1500, and 2000 landmark observations are randomly sampled for fifty times each from the synthetic scenario (figure 6.6). The *averaged* results are then provided in the shape of Box-Whisker-Plots (appendix A.5).

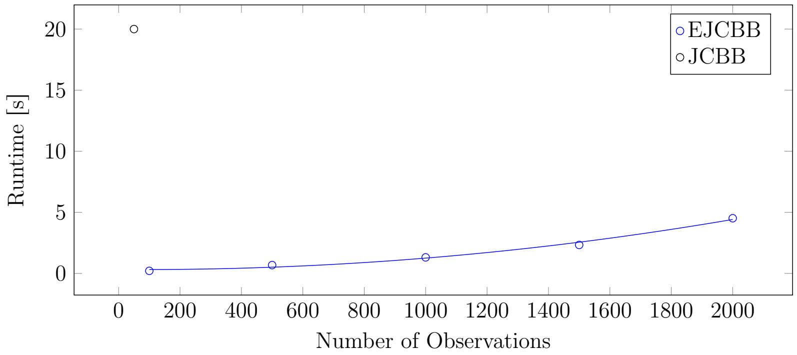


Figure 6.10: Runtime evaluation of EJCBB for 100, 500, 1000, 1500, and 2000 landmark observations that are randomly sampled for fifty times each from the synthetic scenario (figure 6.6), and the corresponding quadratic regression. Additionally, the runtime of JCBB for 50 observations is provided according to [20].

However, as the precision and not the recall is defined to be of the topmost priority for the use case of clustering CVD, this alternative selection criterion is highly favored.

Figure 6.9 visualizes the incorrect associations count, precision, recall, and $F_{0.1}$-score of EJCBB for the synthetic scenario. One can notice that the incorrect associations count is always equal to zero, while the precision is equal to 1 for 500 and more observations. 100 observations can be considered as not sufficient for determining the clusters in a robust manner (minimal support $\kappa \geqslant 10$) and the algorithm behaves, therefore, as desired by classifying all observations as noise. The reader may also notice the very low recall value, which does reflect the strong limitation of the JC-test. This has also been assessed by Neira and Tardós [85]. The recall of EJCBB is approximately 6 % for 2000 observations, while the one of EICNN is approximately 24 % (figure 6.7). The low recall value of EJCBB does of course affect the $F_{0.1}$-score, for which a value of $F_{0.1} = 0.81$ is below the value $F_{0.1} = 0.96$ of EICNN.

Figure 6.8 provides the average runtimes of EJCBB. One can notice that EJCBB scales, in this case, approximately quadratically with the observation count. As suggested by Neira and Tardós [85], the matrix inversion is performed in an incremental manner (section 6.2.1). This way, the computational effort of the matrix inversion is reduced from $O(n^3)$ to $O(n^2)$.

Overall, EJCBB can be considered as a computationally demanding algorithm, which becomes computationally tractable when the number of simultaneously considered hypotheses is bounded. An overview of the runtimes of manifold implementations of JCBB is provided by Reid [96]. Some implementations even require up to 20 s of computation time for the association of just 50 distinct observations [20], while EJCBB is able to cluster 2000 observations in approximately 4.5 s. Therefore, EJCBB can be regarded as a significant extension and enhancement both concerning the robustness and computational efficiency of the performed clustering.

6.4 Discussion

Data association/clustering is considered within the research community, as well as the subsequent fusion (chapters 7 and 8), as a highly challenging task [16, 21]. Furthermore, the results thereof are known to significantly affect the quality of subsequent processing steps. As stated by Castellanos et al. [16], large-scale, the simultaneous incorporation of multiple data sources, and the tackling of complex environments can be regarded as topics that still require further research. Besides complex environments, which are addressed later on in a dedicated manner (chapter 8), all remaining challenges and beyond, such as compensating for temporal decay, have been extensively addressed within this chapter.

Three state-of-art algorithms for data association have been extended for tackling the challenges introduced by the large-scale and high-precision fusion of CVD to the algorithms EICNN, ESCNN, and EJCBB. Not only have these algorithms been extended from pure data association to clustering ones, but also their efficiency and robustness have been significantly enhanced. In detail, large-scale has been approached i. a. by partitioning the problem into small computationally feasible chunks, and by index-based intersection tests prior to the actual IC and JC tests. As stated previously, both IC and JC can be regarded as computationally demanding due to the involved matrix inversions and, hence, constitute the dominating costs.

The extension of JCBB from a pure data association to a full clustering algorithm, EJCBB, has been approached by GBB [84] in combination with the search strategy BFS for being able to dynamically update the landmark map. For the use-case of restricted computational resources or real-time constraints, a novel, multi-objective cost function has been defined for selecting the η most probable hypotheses at each layer of the interpretation tree. The runtime efficiency of the suggested EJCBB approach (4.5 s for 2000 observations) can be regarded as impressive when compared to EICNN (1.77 s for 2000 observations) or other JCBB implementations (20 s for 50 observations [20]). If desired, the computational efficiency could be even further improved by combining EICNN with EJCBB and by utilizing EJCBB only in ambiguous cases, as suggested by Chen et al. [18].

The utilized $\chi^2_{d,\alpha}$-test can be regarded as both the strength and the weakness of (E)ICNN, (E)SCNN, and (E)JCBB. On the one hand, it allows to robustly identify correspondences in the case of Gaussian errors. On the other hand, non-Gaussian errors cannot be appropriately resolved. Fortunately, Gaussian localization and observation errors can be often assumed. However, a more universal test for errors originating from other distributions than Gaussian would be in some cases also of interest. It could be achieved by augmenting/substituting the IC- and JC-tests by alternative ones.

In summary, EICNN, ESCNN, and EJCBB can be regarded as highly valuable improvements of their original data association predecessors for the specific use case of clustering CVD, in particular with respect to robustness and computational efficiency.

7

Fusion of Point-shaped Landmark Data

In this chapter, three methods for the optimized estimation of landmark locations based on the processing state of the acquired landmark observations achieved after clustering are presented, evolved, and opposed to each other with respect to key aspects, such as accuracy and runtime. They are based on the approaches to data fusion Covariance Intersection (CI) and Simultaneous Localization and Mapping (SLAM).

Contents

7.1 Introduction

The purpose of the methods presented in this chapter is the optimized estimation of landmark locations based on the processing state of the acquired landmark observations achieved after clustering:

Problem 7.1: Estimation of Landmark and Vehicle Locations

Input:

1. A trajectory consisting of a sequence $\boldsymbol{Z}^{\mathrm{v}} = (\boldsymbol{z}_l^{\mathrm{v}} \mid l = 0, \dots, p)^T$ of vehicle location observations, a sequence $\boldsymbol{Z} = (\boldsymbol{z}_i \mid i = 0, \dots, m)^T$ of corresponding landmark observations, and $\boldsymbol{Z}|_l$ the subsequences of landmark observations acquired at the vehicle locations l.
2. A sequence $\boldsymbol{Z}^{\mathrm{l}} = (\boldsymbol{Z}_j^{\mathrm{l}} \mid j = 0, \dots, n)^T$ of clusters of landmark observations, each corresponding to a landmark. $\boldsymbol{Z}_j^{\mathrm{l}}|_l$ denotes landmark observations belonging to cluster j and acquired at vehicle location l.
3. A sequence $\boldsymbol{X}^{\mathrm{v}} = (\boldsymbol{x}_l^{\mathrm{v}} \mid l = 0, \dots, p)^T$ of vehicle location estimates, a sequence $\boldsymbol{L} = (\boldsymbol{l}_j \mid j = 0, \dots, n)^T$ of landmark estimates corresponding to the sequence of clusters. Both sequences may be incomplete, i. e. $\boldsymbol{x}_l^{\mathrm{v}} = *$, $\boldsymbol{l}_j = *$ for some values l and j, respectively. The estimates are stored in a submapping data structure.
4. Related temporal decay models.

Output: Improved estimations of landmark locations and vehicle locations, by data fusion.

Multiple approaches to the fusion of point-shaped landmark data are known to the state of the art, and the task can be tackled from different points of view. On the one hand, it can be tackled from the pure sensor fusion's point of view by utilizing approaches, such as (F)CI[1]. On the other hand, it can be approached from the robotic's point of view by utilizing one of the manifold approaches to SLAM[2] [3, 32]. However, the transition between both perspectives is regarded as fluent. Particular

[1] Covariance Intersection (CI)
[2] Simultaneous Localization and Mapping (SLAM)

	Abs. Pos.	Scalab.	Increm.	Est. Uncert.	Temp. Weight.
CI	yes	yes	yes	yes	no
FCI	yes	yes	yes	yes	no
EKF-SLAM	yes	yes	yes	yes	no
BA-SLAM solved via LM	yes	yes	yes	yes	yes
BA-SLAM solved via QR	yes	yes	yes	yes	yes

Table 7.1: A survey of the algorithms for the fusion of point-shaped landmark data presented in the thesis: (F)CI, EKF-based Online-SLAM, and BA-based Full-SLAM, which is solved both directly by utilizing LM and after linearization by utilizing QR. Additionally, the attributes crucial for the fusion of CVD are itemized.

challenges of the fusion of CVD[3] are its large-scale and temporal dynamics. Therefore, both aspects need to be taken into particular consideration.

In table 7.1 the subsequently evolved algorithms (F)CI, EKF[4]-based Online-SLAM, and BA[5]-based Full-SLAM, for the fusion of point-shaped landmark data are opposed to each other. Attributes important from the automotive point of view are particularly considered. All those algorithms can be adapted to handle absolute vehicle positions and can be effectively scaled to multiple computing units via adaptive partitioning (section 4.2). Furthermore, all algorithms can be applied in an incremental manner or extended to be incrementally computable. In the case of BA-based Full-SLAM, this aspect is dependent on the solver that is actually incorporated. In the case of LM[6], it can be achieved by utilizing the previously determined solution as an initialization. In the case of QR[7], it can be achieved by utilizing an incremental approach to QR [6, 51]. While (F)CI and EKF-based Online-SLAM provide estimation uncertainties directly, for BA-based Full-SLAM they need to be explicitly derived. Furthermore, temporal weighting, as introduced in chapter 4.3, can only be sensibly incorporated into approaches without marginalization, such as BA-based Full-SLAM.

In this chapter, the three selected approaches to the fusion of point-shaped landmark observations, cf. table 7.1, (F)CI (section 7.2), EKF-based Online-SLAM (section 7.3), and BA-based Full-SLAM (section 7.4), are evolved to suit the use case of a large-scale CVD-fusion and when possible to compensate for temporal decay. Subsequently, these approaches are opposed to each other concerning their accuracies and runtimes on the basis of two evaluation scenarios (section 7.5). Finally, the findings are discussed in detail (section 7.6).

[3] Collective Vehicle Data (CVD)
[4] Extended Kalman Filter (EKF)
[5] Bundle Adjustment (BA)
[6] Levenberg Marquardt (LM)
[7] QR Factorization (QR)

7.2 Fusion via (Fast) Covariance Intersection

Fusion by CI was initially presented by Julier and Uhlmann [63]. It tackles the task of fusing random variables with *unknown* cross-correlations. This is in contrast to the well-known KF[8] (section 7.3), which assumes *known* cross-correlations and is then able to provide optimal fusion results. However, if cross-correlations are not perfectly known, or independence is incorrectly assumed, the KF is generally known to provide suboptimal fusion results or even to diverge. On the contrary, CI is able to handle correlations of any extent.

Fusion by CI requires that consistent measurement estimates according to the definition given by Jazwinski [60] are provided for the measurements to be fused:

Definition 7.1: Consistent Estimate

Let $\boldsymbol{x}$ be a random input variable representing a measurement, $\overline{\boldsymbol{x}} = \mathrm{E}[\boldsymbol{x}]$, and $\boldsymbol{C}_{\boldsymbol{x},\boldsymbol{x}}$ its covariance. Let furthermore be $\widehat{\boldsymbol{C}}_{\boldsymbol{x},\boldsymbol{x}}$ an estimated uncertainty of $\boldsymbol{x}$. Then the estimate is *consistent* if the following consistency condition is fulfilled:

$$\widehat{\boldsymbol{C}}_{\boldsymbol{x},\boldsymbol{x}} - \boldsymbol{C}_{\boldsymbol{x},\boldsymbol{x}} \geqslant 0. \tag{7.1}$$

To put the definition into words, a consistent estimate implies that the estimated measurement uncertainty $\widehat{\boldsymbol{C}}_{\boldsymbol{x},\boldsymbol{x}}$ ist not underestimating the actual measurement uncertainty $\boldsymbol{C}_{\boldsymbol{x},\boldsymbol{x}}$.

CI is defined as follows:

Definition 7.2: Fusion by Weighted Covariance Intersection

Input:

1. Random variables $\boldsymbol{a}$ and $\boldsymbol{b}$ representing two measurements, $\overline{\boldsymbol{a}} = \mathrm{E}[\boldsymbol{a}]$, $\overline{\boldsymbol{b}} = \mathrm{E}[\boldsymbol{b}]$ their means, and $\widehat{\boldsymbol{C}}_{\boldsymbol{a},\boldsymbol{a}}$, $\widehat{\boldsymbol{C}}_{\boldsymbol{b},\boldsymbol{b}}$ consistent estimated measurement uncertainties of $\boldsymbol{a}$ and $\boldsymbol{b}$,
2. a weight ω, $0 \leqslant \omega \leqslant 1$.

Output: The mean $\overline{\boldsymbol{c}}_\omega$ and an estimated covariance $\widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega}$ of a random variable $\boldsymbol{c}_\omega$ called *fusion* of $\boldsymbol{a}$ and $\boldsymbol{b}$, which satisfy

$$\begin{aligned} \widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega} &= \left(\omega \cdot \widehat{\boldsymbol{C}}_{\boldsymbol{a},\boldsymbol{a}}^{-1} + (1-\omega) \cdot \widehat{\boldsymbol{C}}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\right)^{-1}, \\ \overline{\boldsymbol{c}_\omega} &= \widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega} \cdot \left(\omega \cdot \widehat{\boldsymbol{C}}_{\boldsymbol{a},\boldsymbol{a}}^{-1} \cdot \overline{\boldsymbol{a}} + (1-\omega) \cdot \widehat{\boldsymbol{C}}_{\boldsymbol{b},\boldsymbol{b}}^{-1} \cdot \overline{\boldsymbol{b}}\right). \end{aligned} \tag{7.2}$$

The usefulness of the concept of CI is implied by the following property:

[8] Kalman Filter (KF)

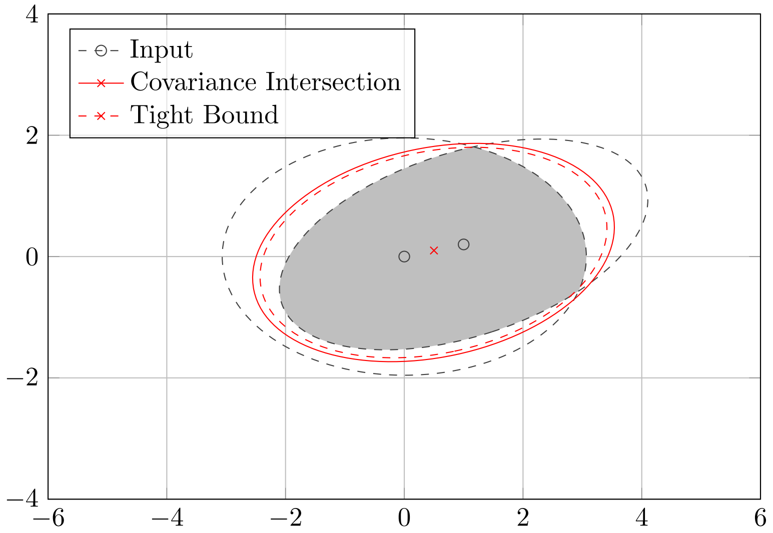


Figure 7.1: Set-theoretic interpretation of CI according to Chong and Mori [19]. CI provides a conservative error estimate that always encloses the intersected area. However, one can notice that CI does not always provide the tightest possible bound.

Property 7.1: Consistency Property of Covariance Intersection

The estimated measurement $\widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega}$ of a fusion $\boldsymbol{c}_\omega$, $0 \leqslant \omega \leqslant 1$, by CI is consistent [63]:

$$\widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega} - \boldsymbol{C}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega} \geqslant 0. \tag{7.3}$$

The interesting point of the consistency property of CI is that it is independent from the cross-correlation of the measurements $\boldsymbol{a}$ and $\boldsymbol{b}$ and of the weight ω. This implies that neither independence of $\boldsymbol{a}$ and $\boldsymbol{b}$ nor an estimation of the cross-correlation is required. Figure 7.1 depicts a set-theoretic explanation of CI given by Chong and Mori [19]. It visualizes the fusion of CI for two given, exemplary measurements. The estimated covariances are represented by covariance ellipses of range 1σ (appendix A.2). Additionally, the intersection of both covariance ellipses is shown. It can be noticed that the covariance ellipse of CI encloses the intersection. However, it can not be regarded as a tight estimate of the resulting uncertainty. Figure 7.2 shows exemplary fusions by CI for $\omega \in \{0.25, 0.5, 0.75\}$.

7.2.1 Optimal Covariance Intersection

The example of figure 7.2 shows that the tightness of the CI is depending on the parameter ω. This leads to a further variant of CI:

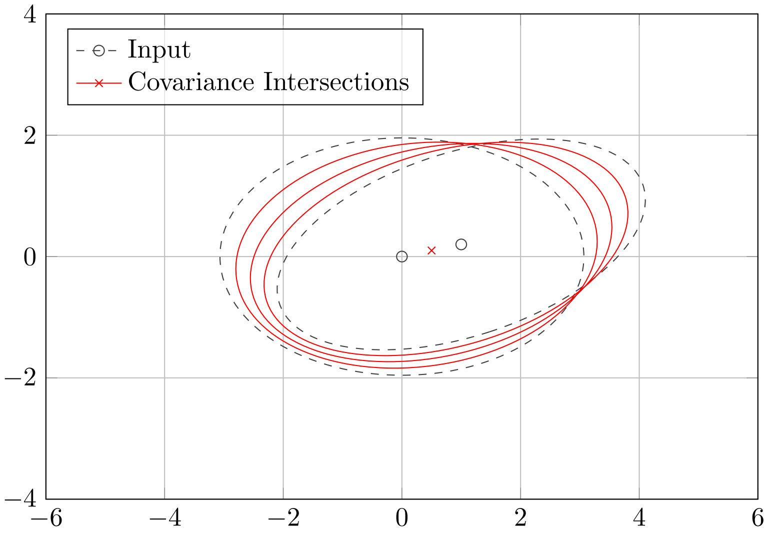


Figure 7.2: Exemplary application of CI for distinct values of $\omega \in \{0.25, 0.5, 0.75\}$.

Definition 7.3: Fusion by Optimal Covariance Intersection

Input:

1. Random variables $\boldsymbol{a}$ and $\boldsymbol{b}$ representing two measurements, $\overline{\boldsymbol{a}} = \mathrm{E}[\boldsymbol{a}]$, $\overline{\boldsymbol{b}} = \mathrm{E}[\boldsymbol{b}]$ their means, and $\widehat{\boldsymbol{C}}_{\boldsymbol{a},\boldsymbol{a}}$, $\widehat{\boldsymbol{C}}_{\boldsymbol{b},\boldsymbol{b}}$ consistent estimated measurement uncertainties of $\boldsymbol{a}$ and $\boldsymbol{b}$,
2. An objective function $f(.)$ which maps a covariance matrix to a non-negative real number.

Output: The mean $\overline{\boldsymbol{c}}_\omega$ and an estimated covariance $\widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega}$ of a weighted fusion $\boldsymbol{c}_\omega$ of $\boldsymbol{a}$ and $\boldsymbol{b}$, where

$$\omega = \arg\min_{\omega} f(\widehat{\boldsymbol{C}}_{\boldsymbol{c}_\omega,\boldsymbol{c}_\omega}). \tag{7.4}$$

Different objective functions f have been proposed to choose a suitable value of ω, such as minimizing the determinant or minimizing the trace of the matrix of $\widehat{\boldsymbol{C}}_{\boldsymbol{c},\boldsymbol{c}}$ [63]. In those cases, the objective functions are convex so that an unambiguous solution to the optimization problem in the range of $[0, 1]$ exists. Any solver for convex optimization functions can be utilized, for example, the well-known Brent algorithm [11]. By combining golden ratio search with parabolic interpolation (equivalent to a Lagrangian interpolation of degree 2) this approach is known to provide in practice superlinear convergence.

Alternatively, closed form solutions to optimal CI have been proposed by several authors [86, 97]. Those are usually referred to as FCI[9] and show an improved computational efficiency in comparison to the application of generic optimization

[9] Fast Covariance Intersection (FCI)

```
input  : $\boldsymbol{C}_1, \boldsymbol{C}_2 \in \boldsymbol{M}_{2\times 2}(\mathbb{R})$
output : $\Omega$
① Jointly diagonalize covariances
$(\boldsymbol{E}_1, \boldsymbol{V}_1) = \mathrm{eigen}(\boldsymbol{C}_1)$ ;
$\boldsymbol{T} = (\boldsymbol{V}_1 \sqrt{\boldsymbol{E}_1})^{-1}$ ;
$(\boldsymbol{E}_2, \boldsymbol{V}_2) = \mathrm{eigen}(\boldsymbol{T} \cdot \boldsymbol{C}_2 \cdot \boldsymbol{T}^T)$ ;
$\boldsymbol{e} = \begin{pmatrix} \frac{1/E_2^1}{1 - 1/E_2^1} & \cdots & \frac{1/E_2^n}{1 - 1/E_2^n} \end{pmatrix}^T$ ;
② Determine $\omega$
switch criterion do
    case determinant do
        $\Omega \cup \left\{ -\frac{1}{2} \left( \boldsymbol{e}^1 + \boldsymbol{e}^2 \right) \right\}$ ;
        break ;
    end
    case trace do
        $\boldsymbol{A} = \boldsymbol{V}_2 \cdot \boldsymbol{E}_1 \cdot (\boldsymbol{V}_2)^T$ ;
        $p = 2 \cdot \frac{\boldsymbol{A}_{11} \cdot e_2 \cdot (1 + e_1) + \boldsymbol{A}_{22} \cdot e_1 \cdot (1 + e_2)}{\boldsymbol{A}_{11} \cdot (1 + e_1) + \boldsymbol{A}_{22} \cdot (1 + e_2)}$ ;
        $q = \frac{\boldsymbol{A}_{11} \cdot (e_2)^2 \cdot (1 + e_1) + \boldsymbol{A}_{22} \cdot (e_1)^2 \cdot (1 + e_2)}{\boldsymbol{A}_{11} \cdot (1 + e_1) + \boldsymbol{A}_{22} \cdot (1 + e_2)}$ ;
        $\Omega \cup \left\{ -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q} \right\}$ ;
        break ;
    end
end
③ Handle border cases
$\Omega = \Omega \cap [0, 1]$ ;
if $\Omega = \{\}$ then
    switch criterion do
        case determinant do
            if $det(C_1) < det(C_2)$ then $\Omega = \{0\}$;
            else $\Omega = \{1\}$;
            break ;
        end
        case trace do
            if $trace(C_1) < trace(C_2)$ then $\Omega = \{0\}$;
            else $\Omega = \{1\}$;
            break ;
        end
    end
end
return $\Omega$ ;
```

Algorithm 7.1: Outline of FCI for both the determinant and the trace criterion according to Reinhardt et al. [97]. The algorithm requires two covariances $\boldsymbol{C}_1$, $\boldsymbol{C}_2 \in \boldsymbol{M}_{2\times 2}(\mathbb{R})$ as input and provides a unique solution $\omega \in \Omega$ as output in the case of a convex objective function.

algorithms. Algorithm 7.1 is an example of FCI according to Reinhardt et al. [97], for both the determinant and the trace objective function. First, both covariances of the input variables are jointly diagonalized. Afterwards, possible values of ω according to the chosen objective function are determined and inserted in a solution set Ω, utilizing the derived closed form solutions. Finally, all out-of-range solutions $\omega \notin [0, 1]$ are removed from Ω. In the case of an empty solution set, both covariances are compared to each other by the chosen objective function. Afterwards, the covariance with the smaller function value constitutes the resulting intersection, i. e. $\omega \in \{0, 1\}$. Subsequently, the determined weight can be utilized for calculation of the mean and covariance estimate of the fusion (equation 7.2).

7.2.2 Tackling Inconsistent Measurements via Union

An extension to CI is the so-called CU[10] [129]. It can be incorporated in the case of inconsistent observations, cf. definition 7.1, as follows. First, CU requires that inconsistent measurements are robustly identified. Afterwards, a consistent estimate $\boldsymbol{c}$ with the mean $\overline{\boldsymbol{c}}$ and covariance $\widehat{\boldsymbol{C}}_{\boldsymbol{c}}$ for two inconsistent measurements with the means $\overline{\boldsymbol{a}}$, $\overline{\boldsymbol{b}}$ and covariance estimates $\widehat{\boldsymbol{C}}_{\boldsymbol{a},\boldsymbol{a}}$ and $\widehat{\boldsymbol{C}}_{\boldsymbol{b},\boldsymbol{b}}$ can be derived by computing the estimate that encloses (tightly) both inconsistent measurements.

7.2.3 Extension to the Fusion of Multiple Measurements

CI can be extended to the fusion of more than two measurements. In the following, two possibilities are outlined: *incremental multiple fusion* and *batch multiple fusion*.

Incremental multiple fusion fuses original or fused measurements in pairwise manner by CI. A special case is to process the original measurements sequentially by fusing the next measurement with the result of fusion of the measurements already processed. Incremental multiple fusion is feasible due to the fact that the covariance estimate of fusion by CI is consistent.

Batch multiple fusion is defined as follows:

Definition 7.4: Batch Multiple Fusion by Weighted Covariance Intersection

Input:

1. Random variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ representing measurements, $\overline{\boldsymbol{x}}_1, \ldots, \overline{\boldsymbol{x}}_n$ their means, and $\widehat{\boldsymbol{C}}_{\boldsymbol{x}_1,\boldsymbol{x}_1}, \ldots, \widehat{\boldsymbol{C}}_{\boldsymbol{x}_n,\boldsymbol{x}_n}$ consistent estimated covariances of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$,
2. Weights ω_i, $0 \leqslant \omega_i \leqslant 1$, with $\sum_{i=1}^{n} \omega_i = 1$.

[10] Covariance Union (CU)

Output: The mean $\overline{\boldsymbol{c}}_{\omega_1,\,\dots,\,\omega_n}$ and an estimated covariance $\widehat{\boldsymbol{C}}_{\boldsymbol{c}_{\omega_1,\,\dots,\,\omega_n}}$ of a random variable $\boldsymbol{c}_{\omega_1,\,\dots,\,\omega_n}$ which is a fusion of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ and which satisfies

$$\begin{aligned} \widehat{\boldsymbol{C}}_{\omega_1,\,\dots,\,\omega_n} &= \left(\omega_1 \cdot \widehat{\boldsymbol{C}}^{-1}_{\boldsymbol{x}_1,\boldsymbol{x}_1} + \dots + \omega_n \cdot \widehat{\boldsymbol{C}}^{-1}_{\boldsymbol{x}_n,\boldsymbol{x}_n}\right)^{-1}, \\ \overline{\boldsymbol{c}} &= \widehat{\boldsymbol{C}}_{\omega_1,\,\dots,\,\omega_n} \cdot (\omega_1 \cdot \widehat{\boldsymbol{C}}^{-1}_{\boldsymbol{x}_1,\boldsymbol{x}_1} \cdot \overline{\boldsymbol{x}}_1 + \dots + \omega_n \cdot \widehat{\boldsymbol{C}}^{-1}_{\boldsymbol{x}_n,\boldsymbol{x}_n} \cdot \overline{\boldsymbol{x}}_n). \end{aligned} \tag{7.5}$$

While the batch approach for multiple variables does still provide an optimal estimate, the incremental one might provide a probably suboptimal result [129]. Fortunately, the consistency of both approaches, the incremental and the batch one, is always guaranteed.

7.2.4 Application to Point-shaped Landmarks

For point shaped landmarks, the landmark observations of a cluster resulting from the clustering step (section 6) are taken as the measurements fused by multiple CIs. The result of fusion is the desired landmark estimate.

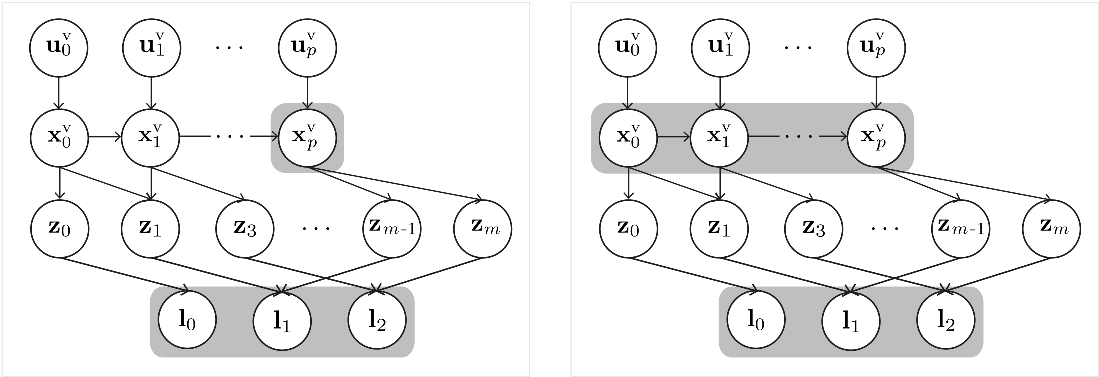
7.3 Fusion via EKF-based Online-SLAM

The section starts with a short overview of different approaches to SLAM. Then the difference between approaches to Online- and Full-SLAM is pointed out. Afterwards, feature-based EKF-Online-SLAM as initially presented by Smith et al. [120] is recapped and adapted for the fusion of point-shaped landmarks from CVD.

Approaches to SLAM aim to simultaneously solve the map building and localization task. They allow approaching the fusion of CVD in a holistic manner, as both the vehicle motion and landmark observations are jointly considered. This significantly differs from algorithms purely intended for sensor data fusion, such as CI (section 7.2).

There exist multiple variants of SLAM, which can be roughly subdivided into three main categories: grid-based, feature-based, and topological [139]. Grid-based approaches commonly utilize an occupancy-grid for the representation of the environment. Feature-based approaches use salient, often point-shaped landmarks to represent the environment. Topological approaches derive a graph representation of the environment based on the observed salient features.

Furthermore, approaches to SLAM can be distinguished into the categories of Online- or Full-SLAM [126]. Full-SLAM considers the whole vehicle trajectory at once, while approaches to Online-SLAM only consider the current vehicle position, by marginalizing out former ones. Figure 7.3 shows a graphical comparison of Online- and Full-SLAM. Hybrid approaches to SLAM which only perform a partial



(a) Online-SLAM (b) Offline-/Full-SLAM

Figure 7.3: The difference between Online- and Full-SLAM. $\boldsymbol{x}_0^v, \ldots, \boldsymbol{x}_p^v$ are the positions constituting the vehicle trajectory. $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_m$ are the observations of the landmarks $\boldsymbol{l}_0, \ldots, \boldsymbol{l}_n$, which are acquired by the vehicles. For Online-SLAM the previous vehicle positions are marginalized out, while Full-SLAM considers the *whole* vehicle trajectory [29, 126]. In both cases, the considered vehicle positions are highlighted in gray ●.

marginalization are possible, too. They are able to alleviate the computational effort while simultaneously increasing the robustness and accuracy of SLAM.

A further distinction of approaches to SLAM is between single- and multi-robot SLAM [17, 99, 137]. Multi-robot SLAM is commonly considered to be more challenging than single-robot SLAM, as the map is built in a cooperative manner by multiple vehicles. As the fusion of CVD does imply a cooperative map-building, multi-robot approaches to SLAM are basically relevant here. However, the utilization of absolute, GNSS[11]-based positions instead of relative ones partially alleviates the task.

EKF-based Online-SLAM presented in this section belongs to the feature-based category. The features are the point-shaped landmarks. Using the EKF instead of the KF allows non-linear prediction and observation models by means of linearization [64, 132]. This is favorable because in the context of SLAM the utilized model functions are commonly non-linear. Although, in this work, a linear vehicle motion model is intentionally incorporated, the observation model is still non-linear. Therefore, the extended variant of the KF is strictly required.

Subsections 7.3.1 and 7.3.2 are devoted to the representation of the estimates and observations of the vehicle and the landmark locations. Subsection 7.3.3 outlines the function of EKF-based Online-SLAM. The subsequent subsections present its steps in detail. The final subsection 7.3.8 compares EKF-based Online-SLAM to CI.

[11] Global Navigation Satellite System (GNSS)

7.3.1 Modeling the State

EKF-based Online-SLAM generates, as typical for (E)KFs [64], a sequence of states emerging from the sequence of vehicle location observations and corresponding landmark observations. A state is represented by a state vector $\hat{\boldsymbol{x}}_{k|k}$ and a state covariance matrix $\boldsymbol{P}_{k|k}$, $k = 1, \ldots$. In the case of point-shaped landmark fusion, they are defined as follows:

Definition 7.5: State Vector

$$\hat{\boldsymbol{x}}_{k|k} = \begin{bmatrix} \boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{l}_0 & \cdots & \boldsymbol{l}_n \end{bmatrix}^T, \; k = 1, \ldots \tag{7.6}$$

where $\boldsymbol{x}_l^{\mathrm{v}}$ denotes the current estimated vehicle position, which is the l-th one in the vehicle trajectory. $\boldsymbol{l}_0, \ldots, \boldsymbol{l}_n$ denote the estimated positions of all currently known landmarks. All those positions are represented by coordinates in the global frame.

Definition 7.6: State Covariance Matrix

$$\boldsymbol{P}_{k|k} = \begin{array}{c} \\ \boldsymbol{x}_l^{\mathrm{v}} \\ \boldsymbol{l}_0 \\ \vdots \\ \boldsymbol{l}_n \end{array} \begin{array}{c} \begin{array}{cccc} \boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{l}_0 & & \boldsymbol{l}_n \end{array} \\ \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_n} \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{l}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_n} \end{bmatrix} \end{array}, \; k = 1, \ldots \tag{7.7}$$

where $\boldsymbol{\Sigma}_{\cdot,\cdot}$ represents the uncertainties and cross-correlations of the vehicle and landmark positions.

The state vector stores both types of observations in an absolute manner. This deviates from the general practice. However, this definition is very advantageous for collaborative mapping as it will become apparent during the course of section 7.3. Furthermore, as typical for Online-SLAM, former vehicle positions $\boldsymbol{x}_0^{\mathrm{v}}, \ldots, \boldsymbol{x}_{p-1}^{\mathrm{v}}$ are not represented within the state vector and the state covariance matrix, as they are marginalized out (figure 7.3).

7.3.2 Representation of Observations

The current observations at every step k are represented by an observation vector $\boldsymbol{z}_k$ and the corresponding observation covariance matrix $\boldsymbol{R}_k$:

Definition 7.7: Observation Vector

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{z}_l^{\mathrm{v}} & \boldsymbol{z}_0 & \cdots & \boldsymbol{z}_m \end{bmatrix}^T , \tag{7.8}$$

where $\boldsymbol{z}_l^{\mathrm{v}}$ denotes the currently observed vehicle position, which is the l-th one in the vehicle trajectory. $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_m$ denote the current landmark observations. The landmark observations are represented relative to the vehicle, i. e. by coordinates in the vehicle frame.

Definition 7.8: Observation Covariance Matrix

$$\boldsymbol{R}_k = \begin{array}{c} \\ z_l^{\mathrm{v}} \\ z_0 \\ \\ z_m \end{array} \begin{array}{c} \begin{array}{cccc} z_l^{\mathrm{v}} & z_0 & & z_m \end{array} \\ \begin{bmatrix} \boldsymbol{\Sigma}_{z_l^{\mathrm{v}}} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\Sigma}_{z_0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{\Sigma}_{z_m} \end{bmatrix} \end{array} . \tag{7.9}$$

The reason for representing the observations in the context of EKF-based Online-SLAM relatively to the vehicle is that the measurements are made relatively to the vehicle at every step k. The observation covariance matrix $\boldsymbol{R}_k$, which is usually referred to as *noise matrix* in the context of (E)KF, does not feature any cross-correlations, in contrast to the state covariance matrix $\boldsymbol{P}_{k|k}$. This is implied, on the one hand, by the relative nature of the measurements, and, on the other hand, by the assumption of independence between the actual landmark observations.

The landmark observations in the observation vector and the landmark estimates in the state vector are related by an assignment by clustering according to section 6.

7.3.3 EKF-based Online-SLAM

Online-SLAM iteratively updates the system state based on new observations. Its central task of step $k-1$, $k>1$ can be specified as follows:

Problem 7.2: Online-SLAM

Input: A state vector $\hat{\boldsymbol{x}}_{k-1|k-1}$, a state covariance matrix $\boldsymbol{P}_{k-1|k-1}$, an observation vector $\boldsymbol{z}_k$, and an observation covariance matrix $\boldsymbol{R}_k$.

Output: An updated state vector $\hat{\boldsymbol{x}}_{k|k}$ and an updated state covariance matrix $\boldsymbol{P}_{k|k}$.

EKF-based Online-SLAM treats the problem in two substeps:

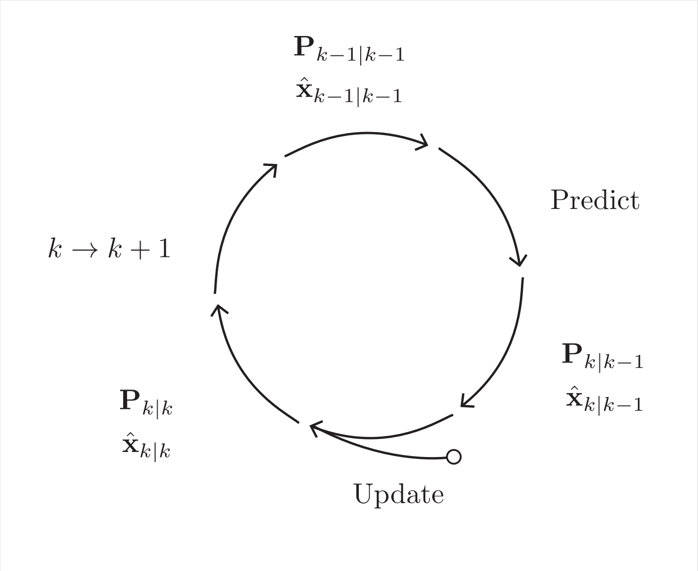


Figure 7.4: Working principle of the EKF, which consists of the iterative execution of a prediction and an update step based on new measurements (equations 7.10 to 7.14).

Solution 7.1: EKF-based Online-SLAM

Predict: Determine a prediction $\hat{\boldsymbol{x}}_{k|k-1}$ of the next state vector and a prediction $\boldsymbol{P}_{k|k-1}$ of the next state covariance matrix, based on $\hat{\boldsymbol{x}}_{k-1|k-1}$, $\boldsymbol{P}_{k-1|k-1}$.

Update: Determine the updating $\hat{\boldsymbol{x}}_{k|k}$ of the state vector and the updating $\boldsymbol{P}_{k|k}$ of the next state covariance matrix, based on $\boldsymbol{z}_k$ and $\boldsymbol{R}_k$.

Figure 7.4 illustrates the principle of iteration of this step.

The *prediction* of the state vector $\hat{\boldsymbol{x}}_{k|k-1}$ and the prediction of the state covariance matrix $\boldsymbol{P}_{k|k-1}$ by the EKF is performed according to

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k-1} &= \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}^{\mathrm{v}}_{k-1}), \\ \boldsymbol{P}_{k|k-1} &= \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1|k-1}\boldsymbol{F}^{\top}_{k-1} + \boldsymbol{Q}_{k-1}, \end{aligned} \tag{7.10}$$

with a *control vector* $\boldsymbol{u}^{\mathrm{v}}_{k-1}$ and the Jacobian $\boldsymbol{F}_{k-1}$ of $\boldsymbol{f}$ evaluated at $\hat{\boldsymbol{x}}_{k-1|k-1}$ and $\boldsymbol{u}^{\mathrm{v}}_{k-1}$. $\boldsymbol{Q}_{k-1}$ is added to the state covariance matrix to model the uncertainty that is imposed by the prediction.

The *updating* of the state vector $\hat{\boldsymbol{x}}_{k|k}$ and the updating of the state covariance matrix $\boldsymbol{P}_{k|k}$ is performed by

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\tilde{\boldsymbol{y}}_k, \\ \boldsymbol{P}_{k|k} &= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_{k|k-1}, \end{aligned} \tag{7.11}$$

where $\tilde{\boldsymbol{y}}_k$ is the so-called measurement residual and $\boldsymbol{K}_k$ the so-called gain which controls the weighting of the prediction $\hat{\boldsymbol{x}}_{k|k-1}$ and the actual observations $\boldsymbol{z}_k$.

The *measurement residual* between the observation vector $\boldsymbol{z}_k$ and the observation prediction $\hat{\boldsymbol{x}}_{k|k-1}$ is determined by

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}), \tag{7.12}$$

where $\boldsymbol{h}$ is an *observation prediction model.*

The *gain* is defined by

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^\top \boldsymbol{S}_k^{-1}, \tag{7.13}$$

where $\boldsymbol{S}_k$ is the so-called residual covariance and $\boldsymbol{H}_k$ is the Jacobian of the observation prediction model.

The *residual covariance* is calculated by

$$\boldsymbol{S}_k = \boldsymbol{H}_k\boldsymbol{P}_{k|k-1}\boldsymbol{H}_k^\top + \boldsymbol{R}_k \tag{7.14}$$

with the observation uncertainty $\boldsymbol{R}_k$.

Algorithm 7.2 presents the details of the approach in pseudocode. The algorithm first predicts a vehicle position based on a vehicle motion model (subsection 7.3.4) and then updates the vehicle position by fusing the predicted and the observed vehicle position (subsection 7.3.5). Afterwards, for every landmark observation made from the current vehicle position, the algorithm checks whether the associate landmark estimate is already in the state vector. If the landmark estimate is not yet existing, the state vector and state covariance matrix are extended (subsection 7.3.6). If the landmark estimate already exists, the landmark estimate is updated by fusing a predicted landmark estimate with the observation (subsection 7.3.7). Finally, the algorithm returns the filtered trajectory positions and the fused landmark estimates.

7.3.4 Predicting the Vehicle Position

A linear vehicle motion model is utilized. As analyzed in detail in section 3.3, the error introduced by a linear motion model (instead of a circular one), and in conjunction with a 10 Hz GNSS sampling rate, can be considered as being minor. This way, both additional linearization errors are circumvented and simultaneously the computational effort is reduced, what can be considered as being crucial for high-precision and large-scale applications.

```
input  :  $\boldsymbol{Z}^{\mathrm{v}} = (\boldsymbol{z}_0^{\mathrm{v}},\ \ldots)$ ,
          $\boldsymbol{Z} = (\boldsymbol{z}_0,\ \ldots)$ .
output :  $\hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k}$
$k = 0$;
① Iterate over distinct vehicle positions
for $\boldsymbol{z}_l^{\mathrm{v}} \in \boldsymbol{Z}^{\mathrm{v}}$ do
    $k = k + 1$;
    ② Predict vehicle position
    $(\hat{\boldsymbol{x}}_{k|k-1},\ \boldsymbol{P}_{k|k-1}) = predictVehiclePosition(\boldsymbol{x}_{l-2}^{\mathrm{v}},\ \boldsymbol{x}_{l-1}^{\mathrm{v}},\ \hat{\boldsymbol{x}}_{k-1|k-1},\ \boldsymbol{P}_{k-1|k-1})$;
    ③ Update vehicle position
    $(\hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k}) = updateVehiclePosition(\boldsymbol{z}_l^{\mathrm{v}},\ \hat{\boldsymbol{x}}_{k|k-1},\ \boldsymbol{P}_{k|k-1})$;
    ④ Iterate over currently observed landmarks
    for $\boldsymbol{z}_i \in \boldsymbol{Z}_l^{\mathrm{v}}$ do
        if $isLandmarkUnknown(\boldsymbol{z}_i)$ then
            (5a) First observation of a landmark
            $(\hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k}) = insertNewLandmark(\boldsymbol{z}_i,\ \hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k})$;
        end
        else
            (5b) Further observation of an existing landmark
            $(\hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k}) = updateLandmark(\boldsymbol{z}_i,\ \hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k})$;
        end
    end
end
return $(\hat{\boldsymbol{x}}_{k|k},\ \boldsymbol{P}_{k|k})$ ;
```

Algorithm 7.2: Fusion of point-shaped CVD via EKF-based Online-SLAM. The procedure expects a vehicle trajectory $\boldsymbol{Z}^{\mathrm{v}}$ and the corresponding landmark observations $\boldsymbol{Z}$ as input. As output, the procedure provides the fused landmark positions in the shape of the state vector $\hat{\boldsymbol{x}}_{k|k}$ and state covariance matrix $\boldsymbol{P}_{k|k}$.

The prediction $\hat{\boldsymbol{x}}_{k|k-1}$ of the state vector is performed by linear extrapolation of the vehicle motion based on the last two vehicle positions:

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1},\ \boldsymbol{u}^{\mathrm{v}}_{k-1}) = \begin{bmatrix} \boldsymbol{x}^{\mathrm{v}}_{l} & \boldsymbol{l}_0 & \cdots & \boldsymbol{l}_n \end{bmatrix}^T, \tag{7.15}$$

and

$$\boldsymbol{x}^{\mathrm{v}}_{l} = \boldsymbol{x}^{\mathrm{v}}_{l-1} + \boldsymbol{u}^{\mathrm{v}}_{k-1},$$

$$\boldsymbol{u}^{\mathrm{v}}_{k-1} = \frac{\boldsymbol{x}^{\mathrm{v}}_{l-1} - \boldsymbol{x}^{\mathrm{v}}_{l-2}}{t_{l-1} - t_{l-2}} \cdot (t_k - t_{l-1}).$$

$\boldsymbol{x}^{\mathrm{v}}_{l-1}$ resp. $\boldsymbol{x}^{\mathrm{v}}_{l-2}$ are the two last vehicle position estimates, t_k is the time of the prediction, and t_{l-1} resp. t_{l-2} are the corresponding two last timestamps of the vehicle position estimates $\boldsymbol{x}^{\mathrm{v}}_{l-1}$ resp. $\boldsymbol{x}^{\mathrm{v}}_{l-2}$. The landmark estimates $\boldsymbol{l}_0,\ \ldots,\ \boldsymbol{l}_n$ are not affected by the vehicle motion, as they are represented by their absolute positions in the global frame.

For two dimensions this explicitly yields

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1},\ \boldsymbol{u}^{\mathrm{v}}_{k-1}) \overset{\text{2D}}{=} \begin{bmatrix} \underbrace{\hat{\boldsymbol{x}}^{0}_{k-1|k-1} + \boldsymbol{u}^{\mathrm{v},x}_{k-1}}_{\boldsymbol{x}^{\mathrm{v}}_{l,x}} & \underbrace{\hat{\boldsymbol{x}}^{1}_{k-1|k-1} + \boldsymbol{u}^{\mathrm{v},y}_{k-1}}_{\boldsymbol{x}^{\mathrm{v}}_{l,y}} & \underbrace{\hat{\boldsymbol{x}}^{2}_{k-1}}_{\boldsymbol{l}^{x}_{0}} & \underbrace{\hat{\boldsymbol{x}}^{3}_{k-1|k-1}}_{\boldsymbol{l}^{y}_{0}} & \underbrace{\hat{\boldsymbol{x}}^{4}_{k-1|k-1}}_{\boldsymbol{l}^{x}_{1}} & \underbrace{\hat{\boldsymbol{x}}^{5}_{k-1|k-1}}_{\boldsymbol{l}^{y}_{1}} & \cdots \end{bmatrix}^T. \tag{7.16}$$

The predicted state covariance matrix $\boldsymbol{P}_{k|k-1}$ is obtained as

$$\begin{aligned} \boldsymbol{P}_{k|k-1} &= \boldsymbol{F}_{k-1} \cdot \boldsymbol{P}_{k-1|k-1} \cdot \boldsymbol{F}^T_{k-1} + \boldsymbol{Q}_{k-1} \\ &= \begin{array}{c} \\ \boldsymbol{x}^{\mathrm{v}}_{l} \\ \boldsymbol{l}_0 \\ \\ \boldsymbol{l}_n \end{array} \begin{array}{c} \begin{array}{cccc} \boldsymbol{x}^{\mathrm{v}}_{l} & \boldsymbol{l}_0 & & \boldsymbol{l}_n \end{array} \\ \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}_{l}} & \boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}_{l},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}_{l},\boldsymbol{l}_n} \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}^{\mathrm{v}}_{l}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{l}_n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}^{\mathrm{v}}_{l}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_n} \end{bmatrix} \end{array}, \end{aligned} \tag{7.17}$$

with the Jacobian

$$\boldsymbol{F}_{k-1} = \begin{bmatrix} \boldsymbol{F}_{\hat{\boldsymbol{x}}_{k-1}} & \boldsymbol{F}_{\boldsymbol{u}^{\mathrm{v}}{}_{k-1}} \end{bmatrix} \overset{\mathrm{2D}}{=} \begin{array}{c} \\ \boldsymbol{x}^{\mathrm{v}}_{l,x} \\ \boldsymbol{x}^{\mathrm{v}}_{l,y} \\ \boldsymbol{l}^{x}_{0} \\ \boldsymbol{l}^{y}_{0} \\ \\ \boldsymbol{l}^{x}_{n} \\ \boldsymbol{l}^{y}_{n} \end{array} \begin{array}{c} \begin{array}{ccccccccc} \boldsymbol{x}^{\mathrm{v}}_{l,x} & \boldsymbol{x}^{\mathrm{v}}_{l,y} & \boldsymbol{l}^{x}_{0} & \boldsymbol{l}^{y}_{0} & & \boldsymbol{l}^{x}_{n} & \boldsymbol{l}^{y}_{n} & \boldsymbol{u}^{\mathrm{v},x}_{k} & \boldsymbol{u}^{\mathrm{v},y}_{k} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \end{bmatrix} \end{array} \tag{7.18}$$

of the prediction model function $\boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}^{\mathrm{v}}_{k-1})$ and with the additive prediction uncertainty

$$\boldsymbol{Q}_{k-1} = \begin{array}{c} \\ \boldsymbol{x}^{\mathrm{v}}_{l} \\ \boldsymbol{l}_{0} \\ \\ \boldsymbol{l}_{n} \end{array} \begin{array}{c} \begin{array}{cccc} \boldsymbol{x}^{\mathrm{v}}_{l} & \boldsymbol{l}_{0} & & \boldsymbol{l}_{n} \end{array} \\ \begin{bmatrix} \Delta\boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}_{l}} & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \end{array}. \tag{7.19}$$

7.3.5 Updating the Vehicle Position

If the vehicle moves, and even no landmarks are observed, then the state vector $\hat{\boldsymbol{x}}_{k|k}$ and state covariance matrix $\boldsymbol{P}_{k|k}$ are nevertheless updated to filter the vehicle motion. To accomplish this, a stripped-down vehicle-selective observation vector $\hat{\boldsymbol{z}}^{\mathrm{v}}_{k}$ and observation noise matrix $\boldsymbol{R}^{\mathrm{v}}_{k}$ is utilized:

$$\begin{aligned} \hat{\boldsymbol{z}}^{\mathrm{v}}_{k} &= \begin{bmatrix} \boldsymbol{z}^{\mathrm{v}}_{l} \end{bmatrix}^{T}, \\ \boldsymbol{R}^{\boldsymbol{z}^{\mathrm{v}}_{l}}_{k} &= \begin{array}{c} \\ \boldsymbol{z}^{\mathrm{v}}_{l} \end{array} \begin{array}{c} \boldsymbol{z}^{\mathrm{v}}_{l} \\ \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{z}^{\mathrm{v}}_{l}} \end{bmatrix} \end{array}. \end{aligned} \tag{7.20}$$

Afterwards, the predicted and observed state are fused according to the EKF equations of subsection 7.3.3.

The vehicle-selective measurement residual is

$$\tilde{\boldsymbol{y}}^{\mathrm{v}}_{k} = \boldsymbol{z}^{\mathrm{v}}_{k} - \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}), \tag{7.21}$$

with the observation prediction model

$$\boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}) = \left[\boldsymbol{x}^{\mathrm{v}}\right]^{T}. \tag{7.22}$$

The vehicle-selective Jacobian of the observation prediction model is

$$\boldsymbol{H}^{\mathrm{v}}(\hat{\boldsymbol{x}}_{k|k-1}) = \begin{array}{c|ccccccc} & \boldsymbol{x}^{\mathrm{v}}_{l,x} & \boldsymbol{x}^{\mathrm{v}}_{l,y} & \boldsymbol{l}^{0}_{x} & \boldsymbol{l}^{0}_{y} & & \boldsymbol{l}^{n}_{x} & \boldsymbol{l}^{n}_{y} \\ \hline \boldsymbol{x}^{\mathrm{v}}_{l,x} & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \boldsymbol{x}^{\mathrm{v}}_{l,y} & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \end{array}. \tag{7.23}$$

The residual covariance is

$$\boldsymbol{S}^{\mathrm{v}}_{k} = \boldsymbol{H}^{\mathrm{v}}_{k}\boldsymbol{P}_{k|k-1}(\boldsymbol{H}^{\mathrm{v}}_{k})^{\top} + \boldsymbol{R}^{\mathrm{v}}_{k}. \tag{7.24}$$

The resulting vehicle-selective EKF-gain is

$$\boldsymbol{K}^{\mathrm{v}}_{k} = \boldsymbol{P}_{k|k-1}(\boldsymbol{H}^{\mathrm{v}}_{k})^{\top}(\boldsymbol{S}^{\mathrm{v}}_{k})^{-1}. \tag{7.25}$$

Finally, the vehicle-selective EKF update equations are given by

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}^{\mathrm{v}}_{k}\tilde{\boldsymbol{y}}^{\mathrm{v}}_{k}, \\ \boldsymbol{P}_{k|k} &= (\boldsymbol{I} - \boldsymbol{K}^{\mathrm{v}}_{k}\boldsymbol{H}^{\mathrm{v}}_{k})\boldsymbol{P}_{k|k-1}. \end{aligned} \tag{7.26}$$

7.3.6 First-Time Observation of a Landmark

In the case that a landmark $\boldsymbol{l}_{n+1}$ is observed for the first time, both the state vector $\hat{\boldsymbol{x}}_{k|k}$ and the state covariance matrix $\boldsymbol{P}_{k|k}$ need to be extended. The extension of the state vector $\hat{\boldsymbol{x}}_{k|k}$ is accomplished by adding one additional landmark entry $\boldsymbol{l}_{n+1}$:

$$\hat{\boldsymbol{x}}_{k|k} = \begin{bmatrix} \boldsymbol{x}^{\mathrm{v}}_{l} & \boldsymbol{l}_{0} & \cdots & \boldsymbol{l}_{n} & \boxed{\boldsymbol{l}_{n+1}} \end{bmatrix}^{T}. \tag{7.27}$$

$\boldsymbol{l}_{n+1}$ is initialized with the absolute position of the landmark observation, which is obtained by compounding according to section 3.4. Furthermore, the state covariance matrix $\boldsymbol{P}_{k|k}$ needs to be extended by an additional column and row:

$$
\begin{aligned}
\boldsymbol{P}_{k|k} &= \begin{array}{c|cccc|c|}
 & \boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{l}_0 & & \boldsymbol{l}_n & \boldsymbol{l}_{n+1} \\
\boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_{n+1}} \\
\boldsymbol{l}_0 & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{l}_{n+1}} \\
 & \vdots & \vdots & \ddots & \vdots & \vdots \\
\boldsymbol{l}_n & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{l}_{n+1}} \\
\hline
\boldsymbol{l}_{n+1} & \boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1},\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1},\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1}}
\end{array} \\
&= \begin{array}{c|cccc|c|}
 & \boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{l}_0 & & \boldsymbol{l}_n & \boldsymbol{l}_{n+1} \\
\boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} \cdot \boldsymbol{J}_{0\oplus}^T \\
\boldsymbol{l}_0 & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}_l^{\mathrm{v}}} \cdot \boldsymbol{J}_{0\oplus}^T \\
 & \vdots & \vdots & \ddots & \vdots & \vdots \\
\boldsymbol{l}_n & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}_l^{\mathrm{v}}} \cdot \boldsymbol{J}_{0\oplus}^T \\
\hline
\boldsymbol{l}_{n+1} & \boldsymbol{J}_{0\oplus} \cdot \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{J}_{0\oplus} \cdot \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_0} & \cdots & \boldsymbol{J}_{0\oplus} \cdot \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_n} & \boldsymbol{J}_{0\oplus} \cdot \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} \cdot \boldsymbol{J}_{0\oplus}^T + \boldsymbol{J}_{1\oplus} \cdot \boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1}} \cdot \boldsymbol{J}_{1\oplus}^T
\end{array} \\
&\overset{\underline{lin.}}{=} \begin{array}{c|cccc|c|}
 & \boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{l}_0 & & \boldsymbol{l}_n & \boldsymbol{l}_{n+1} \\
\boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} \\
\boldsymbol{l}_0 & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_0,\boldsymbol{x}_l^{\mathrm{v}}} \\
 & \vdots & \vdots & \ddots & \vdots & \vdots \\
\boldsymbol{l}_n & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{l}_n,\boldsymbol{x}_l^{\mathrm{v}}} \\
\hline
\boldsymbol{l}_{n+1} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_0} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}},\boldsymbol{l}_n} & \boldsymbol{\Sigma}_{\boldsymbol{x}_l^{\mathrm{v}}} + \boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1}}
\end{array} .
\end{aligned}
\tag{7.28}
$$

$\boldsymbol{J}_{0\oplus}$ and $\boldsymbol{J}_{1\oplus}$ are the Jacobians of the compounding operator $\oplus$ of section 3.4. As may be observed, the utilization of a linear vehicle motion model is again highly beneficial and leads to a pure addition of the vehicle and landmark uncertainties. The initialization of the additional column and row can be achieved either by a constant initialization or in a heuristic manner by replicating the cross-correlations of another landmark [27, 119], such as by the most converged or the closest one. Equation 7.28 reflects the first variant of the heuristic, where the entries of the most converged landmark $\boldsymbol{l}_0$ are duplicated. The heuristics are incorporated for speeding up the convergence of first-time landmark observations.

7.3.7 Further Observation of an Existing Landmark

If a landmark $\boldsymbol{l}_i$ is again observed, both the state vector $\hat{\boldsymbol{x}}_{k|k}$ and state covariance matrix $\hat{\boldsymbol{P}}_{k|k}$ need to be updated. The approach is similar to the one utilized for updating the vehicle position. It uses a stripped-down observation vector $\boldsymbol{z}_k^{\boldsymbol{z}_i}$ and stripped-down observation noise matrix $\boldsymbol{R}_k^{\boldsymbol{z}_i}$:

$$\begin{aligned} \hat{\boldsymbol{z}}_k^{\boldsymbol{z}_i} &= \begin{bmatrix} \boldsymbol{z}_l^{\mathrm{v}} & \boldsymbol{z}_i \end{bmatrix}^T, \\ \boldsymbol{R}_k^{\boldsymbol{z}_i} &= \begin{array}{c c} & \begin{array}{cc} \boldsymbol{z}_l^{\mathrm{v}} & \boldsymbol{z}_i \end{array} \\ \begin{array}{c} \boldsymbol{z}_l^{\mathrm{v}} \\ \boldsymbol{z}_i \end{array} & \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{z}_l^{\mathrm{v}}} & 0 \\ 0 & \boldsymbol{\Sigma}_{\boldsymbol{z}_i} \end{bmatrix} \end{array}. \end{aligned} \tag{7.29}$$

The observed $\boldsymbol{z}_k^{\boldsymbol{z}_i}$ and the predicted state $\hat{\boldsymbol{x}}_{k|k-1}$ are fused by means of the EKF. Let $\boldsymbol{l}_j$ be the landmark estimate of the cluster to which the landmark obeservation $\boldsymbol{z}_i$ belongs, i. e. $\boldsymbol{z}_i \in \boldsymbol{Z}_j^{\mathrm{l}}$ (section 7.1). Because $\boldsymbol{l}_j$ is represented with respect to the global frame, it is independent from the vehicle motion. Thus the prediction lets it unchanged.

The landmark-selective EKF-measurement residual is

$$\tilde{\boldsymbol{y}}_k^{\boldsymbol{z}_i} = \boldsymbol{z}_k^{\boldsymbol{z}_i} - \boldsymbol{h}^{\boldsymbol{z}_i}(\hat{\boldsymbol{x}}_{k|k-1}), \tag{7.30}$$

with the observation prediction model

$$\boldsymbol{h}^{\boldsymbol{z}_i}(\hat{\boldsymbol{x}}_{k|k-1}) = \begin{bmatrix} \boldsymbol{x}_l^{\mathrm{v}} & \boldsymbol{l}_j - \boldsymbol{x}_l^{\mathrm{v}} \end{bmatrix}^T. \tag{7.31}$$

The corresponding landmark-selective Jacobian of the observation prediction model is

$$\boldsymbol{H}^{\boldsymbol{z}_i} \overset{\text{2D}}{=} \begin{array}{c c} & \begin{array}{ccccccccccc} \boldsymbol{x}_{l,x}^{\mathrm{v}} & \boldsymbol{x}_{l,y}^{\mathrm{v}} & \boldsymbol{l}_x^0 & \boldsymbol{l}_y^0 & & \boldsymbol{l}_x^j & \boldsymbol{l}_y^j & & \boldsymbol{l}_x^m & \boldsymbol{l}_y^m \end{array} \\ \begin{array}{c} \boldsymbol{x}_{l,x}^{\mathrm{v}} \\ \boldsymbol{x}_{l,y}^{\mathrm{v}} \\ \boldsymbol{l}_x^j \\ \boldsymbol{l}_y^j \end{array} & \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ -1 & 0 & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 & 0 \end{bmatrix} \end{array}, \tag{7.32}$$

The residual covariance is

$$\boldsymbol{S}_k^{\boldsymbol{z}_i} = \boldsymbol{H}_k^{\boldsymbol{z}_i} \boldsymbol{P}_{k|k-1} (\boldsymbol{H}_k^{\boldsymbol{z}_i})^\top + \boldsymbol{R}_k^{\boldsymbol{z}_i}. \tag{7.33}$$

The resulting EKF-gain is

$$\boldsymbol{K}_k^{\boldsymbol{z}_i} = \boldsymbol{P}_{k|k-1} (\boldsymbol{H}_k^{\boldsymbol{z}_i})^\top (\boldsymbol{S}_k^{\boldsymbol{z}_i})^{-1}. \tag{7.34}$$

The landmark-selective EKF equations for the updated state vector $\hat{\boldsymbol{x}}_{k|k}$ and the updated state covariance matrix $\boldsymbol{P}_{k|k}$ are given by

$$\begin{aligned} \hat{\boldsymbol{x}}_{k|k} &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k^{\boldsymbol{z}_i} \tilde{\boldsymbol{y}}_k^{\boldsymbol{z}_i}, \\ \boldsymbol{P}_{k|k} &= (\boldsymbol{I} - \boldsymbol{K}_k^{\boldsymbol{z}_i} \boldsymbol{H}_k^{\boldsymbol{z}_i}) \boldsymbol{P}_{k|k-1}. \end{aligned} \tag{7.35}$$

7.3.8 Comparison to Covariance Intersection

As stressed by Uhlmann [129], CI and KFs should be considered for different use cases. Uhlmann differentiates between three cases for two given measurements that shall be fused: (1) both measurements are fully independent, (2) they are partially correlated, and (3) they are fully correlated. Based on those cases Uhlmann states that

- in the case of two independent measurements, the KF yields an optimal estimate, and CI a consistent one,
- in the case of partial correlation, CI yields consistent, however, suboptimal (not tight) estimates, while the KF provides inconsistent ones,
- in the case of full correlation, CI yields an optimal estimate, while the KF generates an inconsistent one.

Thus CI *always* provides a consistent estimate, and in the case of full correlation, even an optimal one. The KF provides for partially and fully correlated measurements inconsistent estimates. Only in the case of independent measurements, the KF provides optimal ones.

Those insights into the fusion of CVD imply that if correlations of multiple observations cannot be excluded, then CI should be, according to Uhlmann, preferred to KFs. CVD acquired by distinct vehicles can be assumed to be correlated concerning the visibility conditions and the current GNSS localization accuracy induced by the current satellite constellation and atmospheric effects. However, if CVD is acquired over a longer period of time by multiple vehicles, these effects are expected to cancel out. A comparison of both algorithms on simulated and real-world data will be provided in section 7.5.

7.4 Fusion via BA-based Full-SLAM

EKF-based Online-SLAM can be considered as the primal approach to SLAM [120]. In the past, approaches to Full-SLAM (figure 7.3b) have not been feasible because of their high computational demand. It results from the rapid growth of the solution space in dimensionality with an increasing amount of vehicle position measurements and corresponding landmark observations [47]. However, meanwhile BA-based Full-

SLAM has become feasible not only by Moore's law [101], but also by the advent of efficient solvers and the consideration of the specific structure of the problem. Today, solving Full-SLAM by utilizing BA can be regarded as the state of the art, especially when solving its linearization by means of factorization.

BA-based Full-SLAM as it is called in computer vision [127], or Graph-SLAM as it is denoted in robotics [7] formulates the SLAM problem as a LS[12] optimization one. The LS-based formulation to Full-SLAM was originally introduced by Lu and Milios [75] and later on refined by Dellaert [29]. In the following, the definition provided by Dellaert is recapped in detail and adapted to the fusion of point-shaped CVD.

In contrast to EKF-based Online-SLAM, Full-SLAM considers the whole vehicle trajectory at once:

Problem 7.3: BA-based Full-SLAM

Input:

1. A sequence of landmark observations,

$$\boldsymbol{Z} = (\boldsymbol{z}_i \mid i = 0, \ldots, m)^T, \tag{7.36}$$

2. A sequence of vehicle position observations,

$$\boldsymbol{Z}^{\mathrm{v}} = (\boldsymbol{z}_l^{\mathrm{v}} \mid l = 0, \ldots, p)^T, \tag{7.37}$$

   where $\boldsymbol{Z}|_l$ is the subsequence of landmark observations acquired at vehicle location l.

Output:

1. A sequence of landmark estimations,

$$\boldsymbol{L} = (\boldsymbol{l}_j \mid j = 0, \ldots, n)^T, \tag{7.38}$$

2. A sequence of vehicle position estimations,

$$\boldsymbol{X}^{\mathrm{v}} = (\boldsymbol{x}_l^{\mathrm{v}} \mid l = 0, \ldots, p)^T. \tag{7.39}$$

[12] Least Squares (LS)

BA-based Full-SLAM treats the problem as an optimization one:

$$
\begin{aligned}
\widehat{(\boldsymbol{X}^{\mathrm{v}},\, \boldsymbol{L})} &= \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\max}\; P\left(\boldsymbol{X}^{\mathrm{v}},\; \boldsymbol{L} \mid \boldsymbol{U}^{\mathrm{v}},\; \boldsymbol{Z}\right) \\
&\overset{\text{joint}}{=} \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\max}\; P\left(\boldsymbol{X}^{\mathrm{v}},\; \boldsymbol{L},\; \boldsymbol{U}^{\mathrm{v}},\; \boldsymbol{Z}\right) \\
&= \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\min}\; -P\left(\boldsymbol{X}^{\mathrm{v}},\; \boldsymbol{L},\; \boldsymbol{U}^{\mathrm{v}},\; \boldsymbol{Z}\right),
\end{aligned}
\tag{7.40}
$$

where

$$
\boldsymbol{U}^{\mathrm{v}} = \left(\boldsymbol{u}_l^{\mathrm{v}} \mid l = 0, \ldots, p\right)^T \tag{7.41}
$$

is a sequence of steering commands $\boldsymbol{u}_l^{\mathrm{v}}$ corresponding to the vehicle positions $\boldsymbol{x}_l^{\mathrm{v}}$ of the input trajectory. P denotes a probability function which will be defined in the next subsection.

7.4.1 Least Squares Formulation

Similar to EKF-based Online-SLAM, BA-based Full-SLAM is grounded upon two fundamental functions: a *vehicle motion model* and a *landmark observation model*. Under assumption of a Gaussian error distribution, the vehicle motion model is defined, by means of the probability theory, by

$$
P(\boldsymbol{x}_l^{\mathrm{v}} \mid \boldsymbol{x}_{l-1}^{\mathrm{v}},\; \boldsymbol{u}_l^{\mathrm{v}}) \propto \exp\left(-\frac{1}{2} \cdot \left\| \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v}},\; \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v}} \right\|_{\boldsymbol{\Sigma}_m}^2\right). \tag{7.42}
$$

To put the equation into words, the probability of the vehicle position estimate $\boldsymbol{x}_l^{\mathrm{v}}$, given its former position estimate $\boldsymbol{x}_{l-1}^{\mathrm{v}}$ and the corresponding steering command $\boldsymbol{u}_l^{\mathrm{v}}$, is high if its distance to its prediction $\boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v}},\; \boldsymbol{u}_l^{\mathrm{v}})$ is low. The distance is measured according to the Mahalanobis distance metric $\|.\|_{\boldsymbol{\Sigma}_m}^2$ (appendix A.1) with respect to the vehicle motion covariance matrix $\boldsymbol{\Sigma}_m$.

Similarly, the landmark observation model is, also under assumption of a Gaussian error distribution, defined by

$$
P(\boldsymbol{z}_i \mid \boldsymbol{x}_l^{\mathrm{v}},\; \boldsymbol{l}_j) \propto \exp\left(-\frac{1}{2} \cdot \left\| \boldsymbol{h}\left(\boldsymbol{x}_l^{\mathrm{v}},\; \boldsymbol{l}_j\right) - \boldsymbol{z}_i \right\|_{\boldsymbol{\Sigma}_o}^2\right). \tag{7.43}
$$

This intuitively means that the probability of the landmark observation $\boldsymbol{z}_i$, given the corresponding landmark estimate $\boldsymbol{l}_j$ and the current vehicle position $\boldsymbol{x}_l^{\mathrm{v}}$, is high if its distance to its prediction $\boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\; \boldsymbol{l}_j)$ is low. The distance is measured according to the Mahalanobis distance metric $\|.\|_{\boldsymbol{\Sigma}_o}^2$ (appendix A.1) with respect to the landmark observation covariance matrix $\boldsymbol{\Sigma}_o$.

The vehicle motion covariance matrix $\boldsymbol{\Sigma}_m$ and the landmark observation covariance matrix $\boldsymbol{\Sigma}_o$ are utilized to reflect both measurement and prediction uncertainties. The prediction uncertainties $\boldsymbol{\Sigma}_m^{\text{pred.}}$ and $\boldsymbol{\Sigma}_o^{\text{pred.}}$, respectively, are estimated by linear uncertainty models with

$$\boldsymbol{\Sigma}_m^{\text{pred.}} \overset{\text{2D}}{=} \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} \cdot \|\boldsymbol{u}_i^{\text{v}}\|_2, \quad \boldsymbol{\Sigma}_o^{\text{pred.}} \overset{\text{2D}}{=} \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} \cdot \|\boldsymbol{z}_k\|_2, \tag{7.44}$$

within which the uncertainty is expected to grow proportionally with the prediction distance. The parameter c is the so-called growth factor, and is set to $c = 0.1$.

The measurement uncertainties $\boldsymbol{\Sigma}_m^{\text{meas.}}$ and $\boldsymbol{\Sigma}_o^{\text{meas.}}$, respectively, are directly related to the sensors

$$\boldsymbol{\Sigma}_m^{\text{meas.}} \overset{\text{2D}}{=} \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}, \quad \boldsymbol{\Sigma}_o^{\text{meas.}} \overset{\text{2D}}{=} \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}. \tag{7.45}$$

The pooled covariances $\boldsymbol{\Sigma}_m$ and $\boldsymbol{\Sigma}_o$ are defined as a linear combination of the corresponding measured and predicted uncertainties:

$$\boldsymbol{\Sigma}_m = \frac{1}{2} \cdot \left(\boldsymbol{\Sigma}_m^{\text{meas.}} + \boldsymbol{\Sigma}_m^{\text{pred.}}\right), \tag{7.46}$$

$$\boldsymbol{\Sigma}_o = \frac{1}{2} \cdot \left(\boldsymbol{\Sigma}_o^{\text{meas.}} + \boldsymbol{\Sigma}_o^{\text{pred.}}\right). \tag{7.47}$$

Taking the logarithm of equation 7.40

$$\widehat{(\boldsymbol{X}^{\text{v}},\, \boldsymbol{L})} \;=\; \underset{(\boldsymbol{X}^{\text{v}},\boldsymbol{L})}{\arg\min} \; -\log\left(P\left(\boldsymbol{X}^{\text{v}},\; \boldsymbol{L},\; \boldsymbol{U}^{\text{v}},\; \boldsymbol{Z}\right)\right) \tag{7.48}$$

and discarding the constant factors $\frac{1}{2}$ leads to the following formulation of BA-based Full-SLAM:

$$\widehat{(\boldsymbol{X}^{\text{v}},\, \boldsymbol{L})} \;=\; \underset{(\boldsymbol{X}^{\text{v}},\boldsymbol{L})}{\arg\min} \left\{ \sum_{l=1}^{p} \| \underbrace{\underbrace{\boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\text{v}},\; \boldsymbol{u}_l^{\text{v}}\right)}_{\text{mot. prediction}} - \boldsymbol{x}_l^{\text{v}}}_{\text{mot. pred. deviation}} \|_{\boldsymbol{\Sigma}_m}^2 + \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j,\boldsymbol{z}_i) \\ \in \text{v}(\boldsymbol{x}_l^{\text{v}})}} \| \underbrace{\underbrace{\boldsymbol{h}\left(\boldsymbol{x}_l^{\text{v}},\; \boldsymbol{l}_j\right)}_{\text{obs. prediction}} - \boldsymbol{z}_i}_{\text{obs. pred. deviation}} \|_{\boldsymbol{\Sigma}_o}^2 \right\} \tag{7.49}$$

with $\mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}}) = \left\{(\boldsymbol{l}_j, \boldsymbol{z}_i) \mid \boldsymbol{z}_i \in \boldsymbol{Z}_j^{\mathrm{l}}|_l\right\}$, $\boldsymbol{Z}_j^{\mathrm{l}}|_l$ the sequence of observations belonging to cluster j and acquired at vehicle location $\boldsymbol{x}_l^{\mathrm{v}}$ (section 7.1), and

$$\begin{aligned}\left\|\boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v}}\right\|_{\boldsymbol{\Sigma}_m}^2 &= \left(\boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v}}\right)^T \cdot \boldsymbol{\Sigma}_m^{-1} \cdot \left(\boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v}}\right),\\ \left\|\boldsymbol{h}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j\right) - \boldsymbol{z}_i\right\|_{\boldsymbol{\Sigma}_o}^2 &= \left(\boldsymbol{h}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j\right) - \boldsymbol{z}_i\right)^T \cdot \boldsymbol{\Sigma}_o^{-1} \cdot \left(\boldsymbol{h}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j\right) - \boldsymbol{z}_i\right).\end{aligned}$$

In summary, the resulting non-linear optimization problem consists of two terms: a term concerning the vehicle motion and another one concerning the observed landmarks. Both terms are intended for determining the deviations between the actual values, $\boldsymbol{x}_l^{\mathrm{v}}$ and $\boldsymbol{z}_i$, and the predictions thereof, $\boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}\right)$ and $\boldsymbol{h}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j\right)$. Furthermore, the Mahalanobis distance metric reflects the motion prediction uncertainty $\boldsymbol{\Sigma}_m$ and the landmark observation prediction uncertainty $\boldsymbol{\Sigma}_o$.

In the following, two approaches of solution are presented. The first approach tackles the problem directly by means of the LM-algorithm (section 7.4.2). The second approach employs factorization (section 7.4.5) after linearization (section 7.4.3), with the advantage of a reduced computational load.

7.4.2 Direct Solution via Levenberg Marquardt

The LM-algorithm [41, 52, 80] is an optimization approach designed for *non-linear* objective functions. Therefore, it is suitable for directly determining a solution to BA-based Full-SLAM (equation 7.49) without the explicit need of linearization. The LM-algorithm constitutes an *iterative* procedure that does combine the Gauss–Newton method with the method of gradient descent [41, 80]. However, for fast convergence, and especially in the case of multiple local maxima/minima, an initial guess of the parameter vector has to be provided [67]. As a matter of fact, convergence can usually be assumed after approximately five iterations [47].

The basic working principle of Levenberg's approach is subsequently shown on the basis of the motion prediction deviation term of BA-based Full-SLAM (equation 7.49) the equations for the landmark observation term can be accordingly derived:

$$\begin{aligned}\mathrm{LM}(\boldsymbol{X}_k^{\mathrm{v}}) &= \sum_{\boldsymbol{x}_l^{\mathrm{v}} \in \boldsymbol{X}_k^{\mathrm{v}}} \left\|\boldsymbol{x}_{l,k}^{\mathrm{v}} - \boldsymbol{f}\left(\boldsymbol{x}_{l-1,k}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}\right)\right\|_{\boldsymbol{\Sigma}_m}^2\\ \Rightarrow\ \mathrm{LM}(\boldsymbol{X}_{k+1}^{\mathrm{v}}) &\approx \sum_{\boldsymbol{x}_l^{\mathrm{v}} \in \boldsymbol{X}_k^{\mathrm{v}}} \left\|\boldsymbol{x}_{l,k}^{\mathrm{v}} - \left(\boldsymbol{f}(\boldsymbol{x}_{l-1,k}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}) + \boldsymbol{J}_{l,k}^{\mathrm{v}} \Delta \boldsymbol{x}_{l,k}^{\mathrm{v}}\right)\right\|_{\boldsymbol{\Sigma}_m}^2,\ \boldsymbol{X}_{k+1}^{\mathrm{v}} = \boldsymbol{X}_k^{\mathrm{v}} + \Delta \boldsymbol{X}_k^{\mathrm{v}},\end{aligned} \tag{7.50}$$

with

$$\boldsymbol{J}_{l,k}^{\mathrm{v}} = \frac{\partial \boldsymbol{f}(\boldsymbol{x}_{l,k}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}})}{\partial \boldsymbol{x}_{l,k}^{\mathrm{v}}}.$$

$\boldsymbol{X}_0^{\mathrm{v}}$ is an initial estimate of the parameter vector, $\boldsymbol{X}_k^{\mathrm{v}}$, $k > 0$ the k-th step of iteration, and $\Delta\boldsymbol{X}_k^{\mathrm{v}}$ its perturbation. Taking the derivative with respect to $\Delta\boldsymbol{X}_k^{\mathrm{v}}$ and setting it to zero leads to

$$\left((\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}\boldsymbol{J}_k^{\mathrm{v}}\right)\Delta\boldsymbol{X}_k^{\mathrm{v}} = (\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{X}_k^{\mathrm{v}} - \boldsymbol{f}(\boldsymbol{X}_k^{\mathrm{v}},\,\boldsymbol{U}^{\mathrm{v}})). \tag{7.51}$$

This set of *linear* equations can now be solved for $\Delta\boldsymbol{X}_k^{\mathrm{v}}$ in a straightforward manner by methods, such as Gaussian elimination, Cramer's rule, or factorization.

Furthermore, Levenberg introduced the so-called damping factor λ, which is adjusted in each iteration:

$$\left((\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}\boldsymbol{J}_k^{\mathrm{v}} + \lambda\boldsymbol{I}\right)\Delta\boldsymbol{X}_k^{\mathrm{v}} = (\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}\left(\boldsymbol{X}_k^{\mathrm{v}} - \boldsymbol{f}(\boldsymbol{X}_k^{\mathrm{v}},\,\boldsymbol{U}^{\mathrm{v}})\right). \tag{7.52}$$

For large manifestations of λ the algorithm leads to a Gauss-Newton update, for small ones to a gradient descent one [41]. In the case of an iteration leading to a worse result, the damping factor λ is increased, otherwise, decreased.

Marquardt replaced the identity matrix of Levenberg's formulation by a diagonal matrix depending on $\boldsymbol{J}_k^{\mathrm{v}}$, i. e.

$$\left((\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}\boldsymbol{J}_k^{\mathrm{v}} + \lambda\cdot\operatorname{diag}\left((\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}\boldsymbol{J}_k^{\mathrm{v}}\right)\right)\Delta\boldsymbol{X}_k^{\mathrm{v}} = (\boldsymbol{J}_k^{\mathrm{v}})^T\,\boldsymbol{\Sigma}_m^{-1}\left(\boldsymbol{X}_k^{\mathrm{v}} - \boldsymbol{f}\left(\boldsymbol{X}_k^{\mathrm{v}},\,\boldsymbol{U}^{\mathrm{v}}\right)\right) \tag{7.53}$$

for the case considered here. Generally, this leads to a faster convergence. The resulting, improved formulation is known as the Levenberg Marquardt-algorithm.

7.4.3 Linearized Least Squares Formulation

As suggested by Dellaert [29] the non-linear equations of BA-based Full-SLAM can be linearized by computing their 1st-order Taylor expansion (appendix A.3). This allows for solving the equation system by means of factorization in a significantly more efficient manner. On the downside, the linearization introduces, of course, additional errors. However, those can be (partially) alleviated by utilizing approaches of relinearization.

The motion prediction deviation term $\boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v}},\,\boldsymbol{u}_l^{\mathrm{v}}) - \boldsymbol{x}_l^{\mathrm{v}}$, as introduced in equation 7.49, can be linearized by computing its 1st-order Taylor expansion in *one* variable. This is due to the fact that the steering commands $\boldsymbol{u}_l^{\mathrm{v}}$ can be treated as a constant. The resulting linearization is denoted by

$$\begin{aligned}
\boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}) - \boldsymbol{x}_l^{\mathrm{v}} \ &\approx\ \underbrace{\left[\boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\ \boldsymbol{u}_l^{\mathrm{v}}) + \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0})\right]}_{T_1\ \boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}})} - \underbrace{\left[\boldsymbol{x}_l^{\mathrm{v},0} + (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0})\right]}_{T_1\ \boldsymbol{x}_l^{\mathrm{v}}} \\
&= \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\ \boldsymbol{u}_l^{\mathrm{v}}\right) + \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) - \boldsymbol{x}_l^{\mathrm{v},0} - (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) \\
&= \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) - (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\ \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v},0}
\end{aligned} \tag{7.54}$$

with

$$\boldsymbol{F}_l^{l-1} \ = \ \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}})}{\partial \boldsymbol{x}_{l-1}^{\mathrm{v}}}\right|_{\boldsymbol{x}_{l-1}^{\mathrm{v},0}} .$$

The linearization point $\boldsymbol{x}_{l-1}^{\mathrm{v},0}$ is initially set to be equal to the measured vehicle position $\boldsymbol{x}_{l-1}^{\mathrm{v}}$.

Analogously, the observation prediction deviation term $\boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j) - \boldsymbol{z}_i$ can be linearized by the 1st-order Taylor expansion in two variables

$$\begin{aligned}
\boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j) - \boldsymbol{z}_i \ &\approx\ \underbrace{\left[\boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0) + \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0)\right]}_{T_1\ \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j)} - \underbrace{\left[\boldsymbol{z}_i^0 + \left(\boldsymbol{z}_i - \boldsymbol{z}_i^0\right)\right]}_{T_1\ \boldsymbol{z}_i} \\
&= \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0) + \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) - \boldsymbol{z}_i^0 - (\boldsymbol{z}_i - \boldsymbol{z}_i^0) \\
&= \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0) + \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) - \boldsymbol{z}_i \\
&= \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) + \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0) - \boldsymbol{z}_i
\end{aligned} \tag{7.55}$$

with

$$\begin{aligned}
\boldsymbol{H}_i^{lj} \ &= \ \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j)}{\partial \boldsymbol{x}_l^{\mathrm{v}}}\right|_{(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0)} \\
\boldsymbol{J}_i^{lj} \ &= \ \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j)}{\partial \boldsymbol{l}_j}\right|_{(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0)} .
\end{aligned}$$

The linearization point $\boldsymbol{l}_j^0$ is initially set to be equal to the centroid of the corresponding landmark cluster.

Substituting the linearized vehicle motion and landmark observation terms into equation 7.49, leads to the following linearized formulation of BA-based Full-SLAM:

$$
\begin{aligned}
(\widehat{\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}}) &= \underset{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L})}{\arg\min} \left\{ \sum_{l=1}^{p} \left\| \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) - (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\ \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v},0} \right\|_{\boldsymbol{\Sigma}_m}^2 + \right. \\
&\qquad \left. \sum_{i=1}^{p} \sum_{\substack{(\boldsymbol{l}_j, \boldsymbol{z}_i) \\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \left\| \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) + \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0) - \boldsymbol{z}_i \right\|_{\boldsymbol{\Sigma}_o}^2 \right\} \\
&= \underset{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L})}{\arg\min} \left\{ \sum_{l=1}^{p} \left\| \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) + \boxed{\boldsymbol{G}_l^l} \cdot \left(\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}\right) + \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\ \boldsymbol{u}_l^{\mathrm{v}}\right) - \boldsymbol{x}_l^{\mathrm{v},0} \right\|_{\boldsymbol{\Sigma}_m}^2 + \right. \\
&\qquad \left. \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j, \boldsymbol{z}_i) \\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \left\| \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) + \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0) - \boldsymbol{z}_i \right\|_{\boldsymbol{\Sigma}_o}^2 \right\}
\end{aligned}
\tag{7.56}
$$

with

$$
\boldsymbol{G}_l^l \;=\; -\mathbb{1} \;=\; \mathrm{diag}(-1, \ldots, -1).
$$

The reason for introducing the identity matrix $\boldsymbol{G}_l^l$ is to achieve conformation with the matrix notation used later on in equation 7.62.

Furthermore, the squared Mahalanobis norm can be expressed by means of the Euclidean norm by pre-multiplication of the factor $\boldsymbol{\Sigma}^{-1/2}$ which is the root of the inverse of the given covariance matrix $\boldsymbol{\Sigma}$:

$$
\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{y}\|_{\Sigma}^2 &= \left(\sqrt{(\boldsymbol{x} - \boldsymbol{y})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{y})}\right)^2 \\
&= (\boldsymbol{x} - \boldsymbol{y})^T\, \boldsymbol{\Sigma}^{-1}\, (\boldsymbol{x} - \boldsymbol{y}) && \mid\ \boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}^{-1/2} \boldsymbol{\Sigma}^{-1/2} \\
&= (\boldsymbol{x} - \boldsymbol{y})^T\, \boldsymbol{\Sigma}^{-1/2} \boldsymbol{\Sigma}^{-1/2}\, (\boldsymbol{x} - \boldsymbol{y}) && \mid\ \boldsymbol{\Sigma} = \boldsymbol{\Sigma}^T \ \Leftrightarrow\ \boldsymbol{\Sigma}^{-1} = \left(\boldsymbol{\Sigma}^{-1}\right)^T \\
&= (\boldsymbol{x} - \boldsymbol{y})^T\, (\boldsymbol{\Sigma}^{-1/2})^T \boldsymbol{\Sigma}^{-1/2}\, (\boldsymbol{x} - \boldsymbol{y}) && \mid\ (\boldsymbol{A} \cdot \boldsymbol{B})^T = \boldsymbol{B}^T \cdot \boldsymbol{A}^T \\
&= \left(\boldsymbol{\Sigma}^{-1/2}\, (\boldsymbol{x} - \boldsymbol{y})\right)^T \boldsymbol{\Sigma}^{-1/2}\, (\boldsymbol{x} - \boldsymbol{y}) && \mid\ \boldsymbol{x}^T \cdot \boldsymbol{x} = \langle \boldsymbol{x}, \boldsymbol{x} \rangle,\ \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} = \|\boldsymbol{x}\|_2 \\
&= \|\boldsymbol{\Sigma}^{-1/2}\, (\boldsymbol{x} - \boldsymbol{y})\,\|_2^2.
\end{aligned}
\tag{7.57}
$$

By utilizing this transformation, equation 7.56 can be rewritten to

$$
\begin{aligned}
(\widehat{\boldsymbol{X}^{\mathrm{v}},\, \boldsymbol{L}}) \;&=\; \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\min} \left\{ \sum_{l=1}^{p} \left\| \boldsymbol{\Sigma}_m^{-1/2} \left(\boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) + \boldsymbol{G}_l^{l} \cdot \boldsymbol{x}_l^{\mathrm{v}} + \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\; \boldsymbol{u}_l^{\mathrm{v}}\right)\right) \right\|_2^2 + \right.\\
&\qquad \left. \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j,\boldsymbol{z}_i)\\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \left\| \boldsymbol{\Sigma}_o^{-1/2} \left(\boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^{0}) + \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\; \boldsymbol{l}_j^{0}) - \boldsymbol{z}_i \right) \right\|_2^2 \right\}\\
&=\; \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\min} \left\{ \sum_{l=1}^{p} \left\| \boldsymbol{\Sigma}_m^{-1/2} \cdot \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) + \boldsymbol{\Sigma}_m^{-1/2} \cdot \boldsymbol{G}_l^{l} \cdot \boldsymbol{x}_l^{\mathrm{v}} + \boldsymbol{\Sigma}_m^{-1/2} \cdot \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\; \boldsymbol{u}_l^{\mathrm{v}}\right) \right\|_2^2 + \right.\\
&\qquad \left. \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j,\boldsymbol{z}_i)\\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \left\| \boldsymbol{\Sigma}_o^{-1/2} \cdot \boldsymbol{H}_p^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{\Sigma}_o^{-1/2} \cdot \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^{0}) + \boldsymbol{\Sigma}_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v},0},\; \boldsymbol{l}_j^{0}) - \boldsymbol{z}_i \right) \right\|_2^2 \right\}.
\end{aligned}
\tag{7.58}
$$

This pure LS-formulation is favorable since methods for solving LS-problems are now immediately applicable.

7.4.4 Linearized Least Squares Matrix Formulation

As mentioned before, the LS-formulation to Full-SLAM, as provided in equation 7.58, can be transformed into a common matrix notation:

$$
(\widehat{\boldsymbol{X}^{\mathrm{v}},\, \boldsymbol{L}}) = \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\min} \left(\|\boldsymbol{A}\boldsymbol{\delta} - \boldsymbol{b}\|_2^2 \right), \tag{7.59}
$$

with $\boldsymbol{\delta} = \begin{bmatrix} \boldsymbol{X}^{\mathrm{v}} & \boldsymbol{L} \end{bmatrix}^T$, so that it can be efficiently solved via factorization.

The following example with a trajectory of five vehicle positions, two distinct landmarks, and five landmark observations is utilized to illustrate the matrix notation of BA-based Full-SLAM:

$$
\begin{aligned}
\boldsymbol{X}^{\mathrm{v}} &= \begin{pmatrix} \boldsymbol{x}_0^{\mathrm{v}} & \boldsymbol{x}_1^{\mathrm{v}} & \boldsymbol{x}_2^{\mathrm{v}} & \boldsymbol{x}_3^{\mathrm{v}} & \boldsymbol{x}_4^{\mathrm{v}} \end{pmatrix}, \\
\boldsymbol{L} &= \begin{pmatrix} \boldsymbol{l}_0 & \boldsymbol{l}_1 \end{pmatrix}, \\
\boldsymbol{U}^{\mathrm{v}} &= \begin{pmatrix} \boldsymbol{u}_0^{\mathrm{v}} & \boldsymbol{u}_1^{\mathrm{v}} & \boldsymbol{u}_2^{\mathrm{v}} & \boldsymbol{u}_3^{\mathrm{v}} & \boldsymbol{u}_4^{\mathrm{v}} \end{pmatrix}, \\
\boldsymbol{Z} &= \begin{pmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 & \boldsymbol{z}_3 & \boldsymbol{z}_4 & \boldsymbol{z}_5 & \boldsymbol{z}_6 & \boldsymbol{z}_7 \end{pmatrix},
\end{aligned}
\tag{7.60}
$$

$$
\begin{aligned}
\mathrm{v}(\boldsymbol{x}_0^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_0)\}, \\
\mathrm{v}(\boldsymbol{x}_1^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_1), (\boldsymbol{l}_1, \boldsymbol{z}_2)\}, \\
\mathrm{v}(\boldsymbol{x}_2^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_3), (\boldsymbol{l}_1, \boldsymbol{z}_4)\}, \\
\mathrm{v}(\boldsymbol{x}_3^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_5), (\boldsymbol{l}_1, \boldsymbol{z}_6)\}, \\
\mathrm{v}(\boldsymbol{x}_4^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_7)\}.
\end{aligned}
$$

Figure 7.5 shows a graphical illustration of the example. The corresponding matrix formulation *without* incorporating the Mahalanobis norm is

$$
(\widehat{\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}}) = \underset{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L})}{\arg\min} \left(\left\| \underbrace{\begin{pmatrix}
\boldsymbol{G}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\boldsymbol{F}_1^0 & \boldsymbol{G}_1^1 & 0 & 0 & 0 & 0 & 0 \\
0 & \boldsymbol{F}_2^1 & \boldsymbol{G}_2^2 & 0 & 0 & 0 & 0 \\
0 & 0 & \boldsymbol{F}_3^2 & \boldsymbol{G}_3^3 & 0 & 0 & 0 \\
0 & 0 & 0 & \boldsymbol{F}_4^3 & \boldsymbol{G}_4^4 & 0 & 0 \\
\boldsymbol{H}_0^{0,0} & 0 & 0 & 0 & 0 & \boldsymbol{J}_0^0 & 0 \\
0 & \boldsymbol{H}_1^{1,0} & 0 & 0 & 0 & \boldsymbol{J}_0^0 & 0 \\
0 & \boldsymbol{H}_2^{1,1} & 0 & 0 & 0 & 0 & \boldsymbol{J}_1^1 \\
0 & 0 & \boldsymbol{H}_3^{2,0} & 0 & 0 & \boldsymbol{J}_0^0 & 0 \\
0 & 0 & \boldsymbol{H}_4^{2,1} & 0 & 0 & 0 & \boldsymbol{J}_1^1 \\
0 & 0 & 0 & \boldsymbol{H}_5^{3,0} & 0 & \boldsymbol{J}_0^0 & 0 \\
0 & 0 & 0 & \boldsymbol{H}_6^{3,1} & 0 & 0 & \boldsymbol{J}_1^1 \\
0 & 0 & 0 & 0 & \boldsymbol{H}_7^{4,1} & \boldsymbol{J}_0^0 & 0
\end{pmatrix}}_{\boldsymbol{A}} \cdot \underbrace{\begin{pmatrix}
\boldsymbol{x}_0 - \boldsymbol{x}_0^{\mathrm{v},0} \\
\boldsymbol{x}_1 - \boldsymbol{x}_1^{\mathrm{v},0} \\
\boldsymbol{x}_2 - \boldsymbol{x}_2^{\mathrm{v},0} \\
\boldsymbol{x}_3 - \boldsymbol{x}_3^{\mathrm{v},0} \\
\boldsymbol{x}_4 - \boldsymbol{x}_4^{\mathrm{v},0} \\
\boldsymbol{l}_0 - \boldsymbol{l}_0^0 \\
\boldsymbol{l}_1 - \boldsymbol{l}_1^0
\end{pmatrix}}_{\boldsymbol{\delta}} - \underbrace{\begin{pmatrix}
\boldsymbol{f}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{u}_0^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{u}_1^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{u}_2^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{u}_3^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_4^{\mathrm{v},0}, \boldsymbol{u}_4^{\mathrm{v}}) \\
\boldsymbol{h}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_0} \\
\boldsymbol{h}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_1} \\
\boldsymbol{h}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_2} \\
\boldsymbol{h}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_3} \\
\boldsymbol{h}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_4} \\
\boldsymbol{h}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_5} \\
\boldsymbol{h}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_6} \\
\boldsymbol{h}(\boldsymbol{x}_4^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_7}
\end{pmatrix}}_{\boldsymbol{b}} \right\|_2^2 \right).
\tag{7.61}
$$

Equation 7.62 provides the complete matrix notation, including the Mahalanobis norm.

$$(\widehat{\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}}) = \underset{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L})}{\arg\min} \left(\left\| \underbrace{\begin{pmatrix} \Sigma_m^{-1/2} \cdot \boldsymbol{G}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \Sigma_m^{-1/2} \cdot \boldsymbol{F}_1^0 & \Sigma_m^{-1/2} \cdot \boldsymbol{G}_1^1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \Sigma_m^{-1/2} \cdot \boldsymbol{F}_2^1 & \Sigma_m^{-1/2} \cdot \boldsymbol{G}_2^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma_m^{-1/2} \cdot \boldsymbol{F}_3^2 & \Sigma_m^{-1/2} \cdot \boldsymbol{G}_3^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Sigma_m^{-1/2} \cdot \boldsymbol{F}_4^3 & \Sigma_m^{-1/2} \cdot \boldsymbol{G}_4^4 & 0 & 0 \\ \Sigma_o^{-1/2} \cdot \boldsymbol{H}_0^{0,0} & 0 & 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_0^0 & 0 \\ 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_1^{1,0} & 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_0^0 & 0 \\ 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_2^{1,1} & 0 & 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_1^1 \\ 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_3^{2,0} & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_0^0 & 0 \\ 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_4^{2,1} & 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_1^1 \\ 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_5^{3,0} & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_0^0 & 0 \\ 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_6^{3,1} & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_1^1 \\ 0 & 0 & 0 & 0 & \Sigma_o^{-1/2} \cdot \boldsymbol{H}_7^{4,1} & \Sigma_o^{-1/2} \cdot \boldsymbol{J}_0^0 & 0 \end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix} \boldsymbol{x}_0^{\mathrm{v}} - \boldsymbol{x}_0^{\mathrm{v},0} \\ \boldsymbol{x}_1^{\mathrm{v}} - \boldsymbol{x}_1^{\mathrm{v},0} \\ \boldsymbol{x}_2^{\mathrm{v}} - \boldsymbol{x}_2^{\mathrm{v},0} \\ \boldsymbol{x}_3^{\mathrm{v}} - \boldsymbol{x}_3^{\mathrm{v},0} \\ \boldsymbol{x}_4^{\mathrm{v}} - \boldsymbol{x}_4^{\mathrm{v},0} \\ \boldsymbol{l}_0 - \boldsymbol{l}_0^0 \\ \boldsymbol{l}_1 - \boldsymbol{l}_1^0 \end{pmatrix}}_{\delta} - \underbrace{\begin{pmatrix} \Sigma_m^{-1/2} \cdot \boldsymbol{f}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{u}_0^{\mathrm{v}}) \\ \Sigma_m^{-1/2} \cdot \boldsymbol{f}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{u}_1^{\mathrm{v}}) \\ \Sigma_m^{-1/2} \cdot \boldsymbol{f}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{u}_2^{\mathrm{v}}) \\ \Sigma_m^{-1/2} \cdot \boldsymbol{f}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{u}_3^{\mathrm{v}}) \\ \Sigma_m^{-1/2} \cdot \boldsymbol{f}(\boldsymbol{x}_4^{\mathrm{v},0}, \boldsymbol{u}_4^{\mathrm{v}}) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_0}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_1}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_1}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_2}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_2}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_3}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_3}\right) \\ \Sigma_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_4^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_4}\right) \end{pmatrix}}_{b} \right\|_2^2 \right).$$

Equation 7.62: Exemplary instantiation of the matrix formulation of linearized BA-based Full-SLAM according to the example provided in equation 7.60 and figure 7.5.

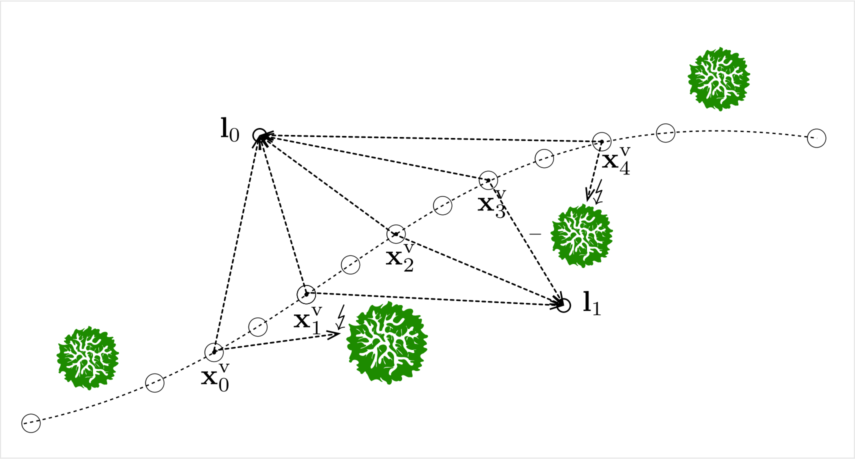


Figure 7.5: Exemplary scenario, which exhibits multiple observations of two point-shaped landmarks $\boldsymbol{l}_0$ and $\boldsymbol{l}_1$ acquired relative to a single vehicle trajectory $\boldsymbol{x}_0^{\mathrm{v}}, \ldots, \boldsymbol{x}_4^{\mathrm{v}}$ (equation 7.60). The observation of the point-shaped landmark $\boldsymbol{l}_1$ is hindered two times due to the obstacles ●.

7.4.5 Solution by QR-factorization after Linearization

In the following, the optimization problem represented by the linearized LS matrix formulation of the preceding section, equation 7.62, is solved by means of factorization.

The matrix formulation has the form

$$(\widehat{\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}}) = \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L})}{\arg\min} \left(\|\boldsymbol{A}\boldsymbol{\delta}-\boldsymbol{b}\|_2^2\right), \tag{7.63}$$

with $\boldsymbol{\delta} = \begin{bmatrix}\boldsymbol{X}^{\mathrm{v}} & \boldsymbol{L}\end{bmatrix}^T$.

At a first glance, this approach seems to be very similar to the working principle of the general-purpose LM optimization algorithm. However, by utilizing a tailored solution, the structure of the problem can be much better exploited. This has a positive effect on the performance and is reflected by the experimental comparison of the two approaches later on (section 7.5).

Cholesky and QR can be considered as two intensively explored factorization methods. When compared to Cholesky, QR is known to provide a better numerical stability, while being approximately two times more demanding concerning the computational effort [29]. However, in favor of numerical stability, QR is preferred, as the quality of the fused data has the highest priority.

Within literature, different approaches to the computation of QR exist, such as Householder Reflections, Givens Rotations, or the Graham-Schmidt method. Additionally, multiple variants of the aforementioned methods exist, for example for sparse matrices [25, 43]. However, those approaches are all grounded on the same, fundamental assumption that the matrix $\boldsymbol{A}$ can be expressed by means of two other matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, so that the equation

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}, \tag{7.64}$$

with an orthogonal matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{R}$, is fulfilled.

QR factorization by Householder Reflections is defined as

$$\underbrace{\boldsymbol{H}_n \cdot \ldots \cdot \boldsymbol{H}_2 \cdot \boldsymbol{H}_1}_{\text{Householder Reflections}} \cdot \boldsymbol{A} = \boldsymbol{Q}^T \boldsymbol{A} = \boldsymbol{R} \tag{7.65}$$

with

$$\boldsymbol{Q}^T \boldsymbol{b} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{d} \end{bmatrix}.$$

Its application to the linear LS optimization problem yields [43]:

$$\|\boldsymbol{A}\boldsymbol{\delta} - \boldsymbol{b}\|_2^2 = \|\boldsymbol{Q}^T \boldsymbol{A}\boldsymbol{\delta} - \boldsymbol{Q}^T \boldsymbol{b}\|_2^2 = \|\boldsymbol{R}\boldsymbol{\delta} - \boldsymbol{c}\|_2^2 + \|\boldsymbol{d}\|_2^2. \tag{7.66}$$

Then the solution can be obtained in straightforward manner because $\boldsymbol{R}$ is an upper triangular matrix.

The runtime complexity of the approach is dominated by the Householder Reflections. For $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ the runtime complexity is known to be bounded by

$$\mathcal{O}\left(2\left(m - \frac{n}{3}\right) n^2\right). \tag{7.67}$$

Furthermore, a well-adjusted ordering, which aims to reduce the non-zero elements of the subsequent factorization, is known to have a more significant impact on the computational efficiency than the actual choice of the factorization method [29]. Multiple general-purpose approaches to matrix preconditioning exist, such as the well-known COLAMD[13] [26]. However, also more specialized ones exist that aim to exploit the specific structure of the problem. Such a specific preconditioning, intended for BA-based Full-SLAM, is known as LX-ordering. The LX-preconditioner, firstly, aims to resolve the landmark positions $\boldsymbol{L}$ and, secondly, the vehicle positions $\boldsymbol{X}^{\text{v}}$ [67]. This way, a generally very beneficial preconditioning is achieved. However, the task of determining an optimal ordering is known to be generally NP-complete [29].

[13] Column Approximate Minimum Degree Ordering Algorithm (COLAMD)

Furthermore, approaches to QR exist that allow an incremental update when rows and/or columns are added and/or removed. This way the cost of partial map updates can be alleviated [6, 51].

Determining the Estimation Uncertainties

While the estimation uncertainties are directly provided for example by EKF-based Online-SLAM, in the shape of the state covariance matrix, they are unfortunately *not* directly provided by BA-based Full-SLAM. Therefore, they need to be explicitly derived [42]. The uncertainties can be either computed based upon the matrix $\boldsymbol{A}$ (equation 7.62) or the matrix $\boldsymbol{R}$, which is the resulting upper triangular matrix after applying QR to the matrix $\boldsymbol{A}$:

$$\boldsymbol{\Sigma} = (\boldsymbol{A}^T\boldsymbol{A})^{-1} = (\boldsymbol{R}^T\boldsymbol{R})^{-1}. \tag{7.68}$$

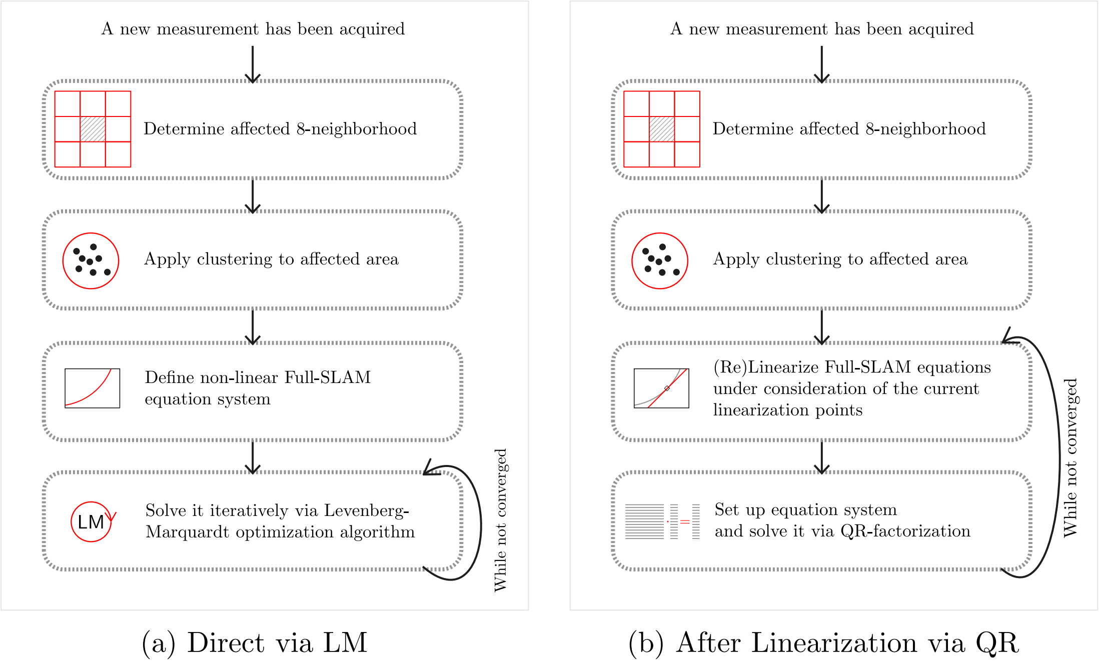
This naive approach has a runtime complexity bounded by $\mathcal{O}(n^2)$. However, there are more efficient approaches known to the state of the art [42], which aim to compute only the elements of $\boldsymbol{A}^T\boldsymbol{A}$ that are associated with the non-zero entries of the matrix $\boldsymbol{R}$. This results in a more efficient computation, which is known to feature a runtime complexity of $\mathcal{O}(n)$.

7.4.6 Compensating for Temporal Decay

The road network context is exposed to more or less variation over time, for example, contingent upon roadworks. Therefore, it is crucial to compensate for temporal dynamics during the map building process.

By utilizing temporal weights ω_m, $\omega_o \in [0,1]$ and augmenting the Mahalanobis distance metric based weighting factors by $\sqrt{\omega_m} \cdot \Sigma_m^{-1/2}$, $\sqrt{\omega_o} \cdot \Sigma_o^{-1/2}$, $\sqrt{\omega_a} \cdot \Sigma_a^{-1/2}$ instead of $\Sigma_m^{-1/2}$, $\Sigma_o^{-1/2}$, $\Sigma_a^{-1/2}$, BA-based Full-SLAM (equation 7.58) can be extended to consider the variation of landmarks over time. To be more specific, temporal dynamics can be compensated by weighting former vehicle poses and landmark positions less than more current ones. They can even be entirely excluded from the fusion process by setting ω_m respectively ω_o to 0. This way, the map computation is able to adapt to altered landmarks and to forget about removed ones.

The different decay rates of distinct road network features, for which the weighting scheme needs to adapt to, can be considered as a challenge. For example, moving construction zones exhibit a higher temporal dynamic than road signs. To approach this challenge, an efficient temporal weight computation based upon preaggregation is utilized (section 4.3).


A new measurement has been acquired
Determine affected 8-neighborhood
Apply clustering to affected area
Define non-linear Full-SLAM equation system
LM
Solve it iteratively via Levenberg-Marquardt optimization algorithm
While not converged
A new measurement has been acquired
Determine affected 8-neighborhood
Apply clustering to affected area
(Re)Linearize Full-SLAM equations under consideration of the current linearization points
Set up equation system and solve it via QR-factorization
While not converged

(a) Direct via LM

(b) After Linearization via QR

Figure 7.6: Outline of point-shaped landmark data fusion via BA-based Full-SLAM by utilizing a direct LM-based solving approach (subfigure a) and a QR-based one after linearization (subfigure b).

7.4.7 Application to Point-shaped Landmark Data

In figure 7.6 an outline of BA-based Full-SLAM, involving a direct (figure 7.6a) and a linearization-based (figure 7.6b) approach, is provided.

To put the figure into words, both approaches are basically equal until their third step. In the case that new observations are acquired, at first, the affected 8-neighborhood is determined (section 4.2). Afterwards, the determined 8-neighborhoods are (re)clustered (chapter 6). From this point on, both approaches differ.

The direct approach incorporates the non-linearized BA-based Full-SLAM equations provided in equation 7.49 and aims to determine a solution to it by utilizing the LM optimization algorithm. As the LM algorithm is an iterative procedure, usually multiple iterations are required. Furthermore, an approximation of the solution needs to be a priori provided, to prevent convergence to a local instead of a global optimum.

The linearization-based approach is based upon the linearized BA-based Full-SLAM formulation provided in equation 7.56. As the linearization requires the definition of adequate linearization points and the ideal linearization points are given by the final solution, the linearization and the subsequent solving via QR can be applied iteratively. This way the error induced by linearization can be reduced.

7.5 Evaluation

In this chapter, the previously derived approaches to the fusion of point-shaped landmark data, (F)CI, EKF-based Online-SLAM and BA-based Full-SLAM, solved directly via LM and after linearization via QR, are opposed to each other. For the evaluation, two distinct scenarios, a synthetic and a real-world one (section 7.5.1) are incorporated. For each algorithm and scenario 100, 500, 1000, 1500, and 2000 observations are randomly sampled without replacement for fifty times each. The *averaged* fusion results are then provided in the shape of a box-whisker-plot (appendix A.5).

7.5.1 Description of the Evaluation Scenarios

Subsequently, both evaluation scenarios, a synthetic and a real-world one, are described in detail.

Synthetic Scenario

The partially synthetic scenario, hereinafter referred to as just the synthetic one, as visualized in figure 7.7, is comprised of 29 simulated point-shaped landmarks, which were observed 2560 times in total relative to multiple real-world trajectories. The average standard deviation of the localization is approximately 1.06 m and of the landmark observations approximately 2.19 m.

The landmark observations are sampled based upon a distance-dependent two-dimensional Gaussian distribution. The Gaussian distribution is aligned so that the major axis of the corresponding error ellipse (appendix A.2) coincides with the sight vector from the vehicle to the landmark. The actual standard deviations represented by the major $f_{\text{major}}(\delta)$ and minor axes $f_{\text{minor}}(\delta)$ are determined by the following linear functions according to the length of the sight vector δ:

$$\begin{aligned} f_{\text{major}}(\delta) &= 0.2 \cdot \delta + 1, \\ f_{\text{minor}}(\delta) &= 0.1 \cdot \delta + 1. \end{aligned} \tag{7.69}$$

This proceeding is justified by the fact that the angular error of a TSR[14]-camera is commonly minor in comparison to the distance one.

Real-world Scenario

The real-world scenario, that shows a short road excerpt in Ingolstadt, consists of 24 point-shaped landmarks, road signs, which were observed by a TSR-camera 2490 times in total (figure 7.8). As no accuracy measure is directly provided by the sensor, an estimate of it is incorporated. The average standard deviation of the localization is approximately 1.33 m and of the landmark observations approximately 1.31 m.

[14] Traffic Sign Recognition (TSR)

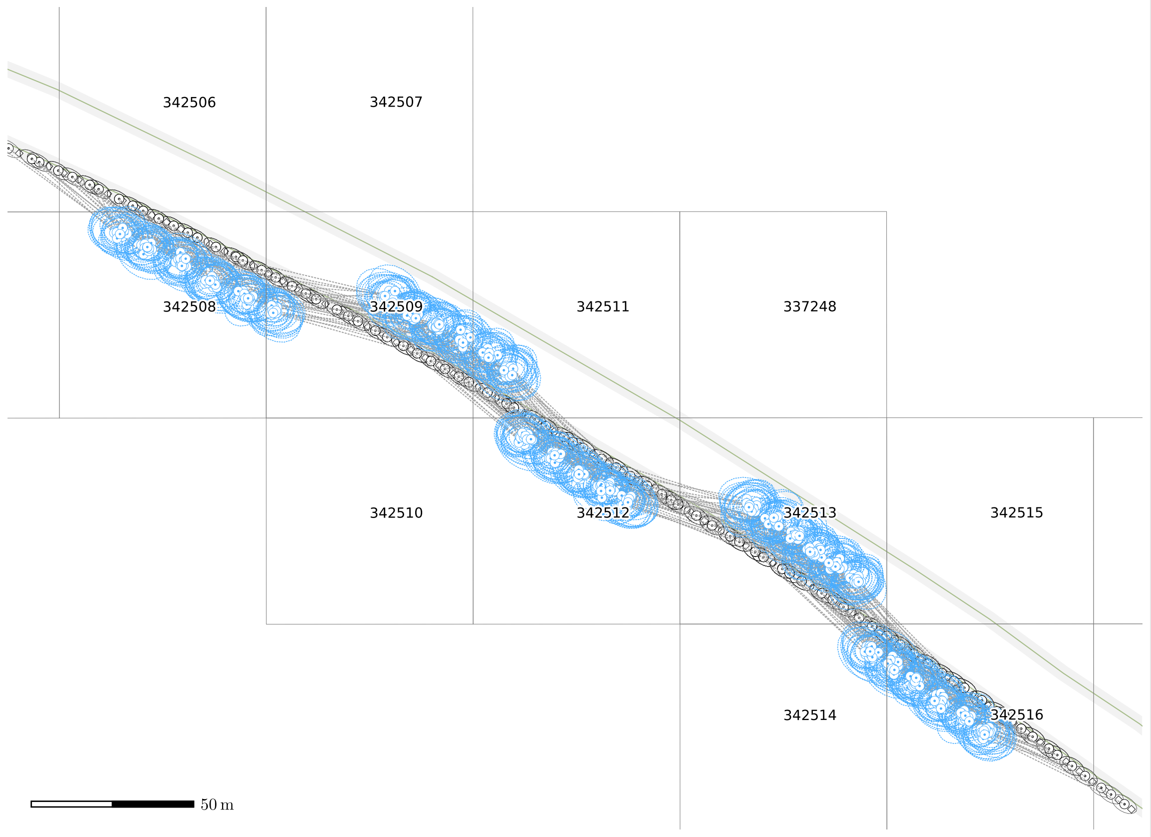


Figure 7.7: Partially synthetic scenario, which exhibits 29 point-shaped landmarks. The provided ellipses are representing the 3σ-range of the measurement uncertainties.

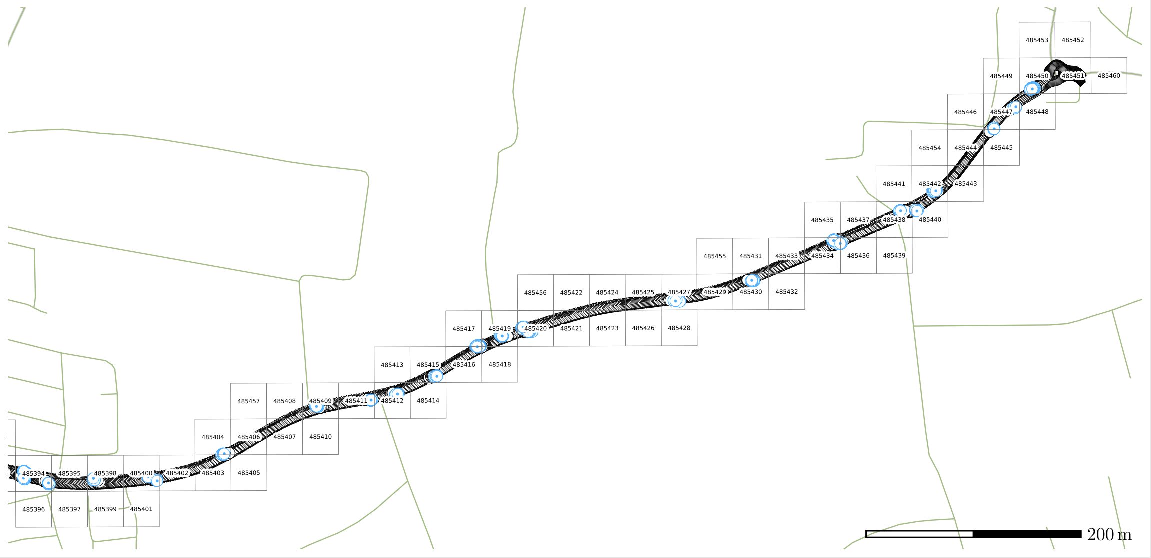


Figure 7.8: Real-world scenario that was acquired at Ingolstadt and exhibits 24 point-shaped landmarks and the corresponding uncertainties at a 3σ-range.

7.5.2 Fusion via (Fast) Covariance Intersection

Subsequently, the fusion of point-shaped landmark observations by utilizing (F)CI is evaluated. (F)CI is applied in an iterative manner (section 7.2.3). Furthermore, the determinant criterion is utilized for (F)CI (section 7.2.1). This is substantiated by the fact that the determinant of the state covariance matrix of EKF-based Online-SLAM monotonically decreases with each update [31]. Therefore, both aspects support the comparability of (F)CI and EKF-based Online-SLAM.

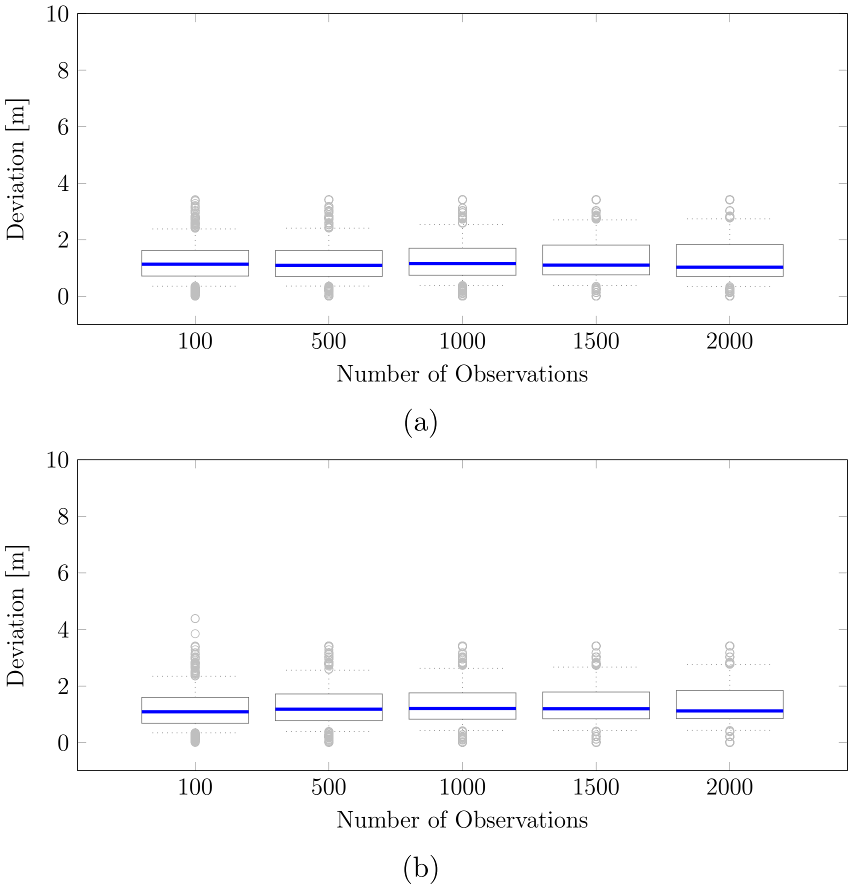
In figure 7.9 the accuracies, and in figure 7.10 the runtimes of CI and FCI gained on the synthetic scenario via CI and FCI are provided. The median accuracy is approximately 1.13 m on average after 100 observations, and approximately 1.03 m on average after 2000 observations. Hence, one can infer that the accuracy only marginally improves by 0.1 m with an increasing observation count. However, the lower and upper quartiles are tighter, and outliers are more sparse and less severe. The results concerning the accuracy of CI and FCI are, as expected, very comparable. The median accuracy of FCI on the synthetic scenario is approximately 1.13 m on average after 100 observations and approximately 1.03 m on average after 2000 observations, while the runtimes of the FCI approximately halve when compared to CI.

In figure 7.11 the accuracies and in figure 7.12 the runtimes gained via CI and FCI on the real-world scenario are provided. One can notice that the outliers are more severe than at the synthetic scenario. This is mainly due to the coarse observation error model, which partially leads to inconsistent error estimates and, therefore, to suboptimal fusion results provided by (F)CI. The median accuracy gained by CI is approximately 1.67 m on average after 100 observations and approximately 1.64 m on average after 2000 observations. While the median accuracy gained by FCI is approximately 1.71 m on average after 100 observations and approximately 2.01 m on average after 2000 observations. Therefore, one can assess that CI is able to significantly better compensate for inconsistent error estimates than FCI. Furthermore, the inconsistent error estimates also affect the computation time. Initially, FCI is even slower than CI and outperforms CI not until 800 observations. The computation time reduces then by only one fourth.

7.5.3 Fusion via EKF-based Online-SLAM

In the following, the fusion of point-shaped landmark observations by means of EKF-based Online-SLAM is evaluated according to its accuracy and runtime, for both the synthetic and real-world scenario.

In figure 7.13 the average median accuracies for EKF-based Online-SLAM are visualized for both the synthetic and real-world scenario. At the synthetic scenario,



(a)

(b)

Figure 7.9: Accuracy evaluation of CI (subfigure a) and FCI (subfigure b) for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* scenario.

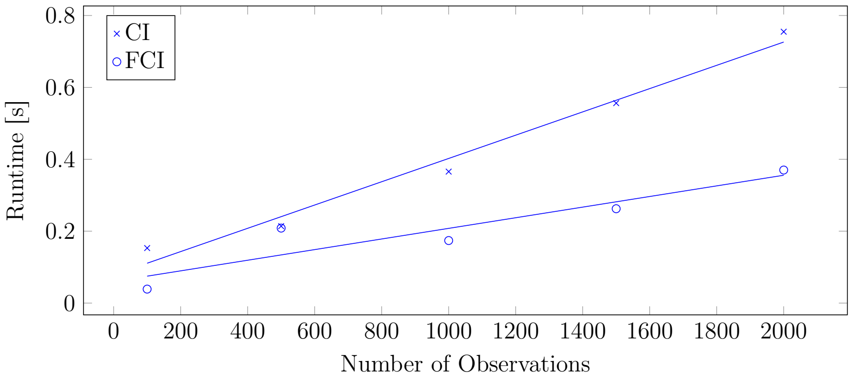
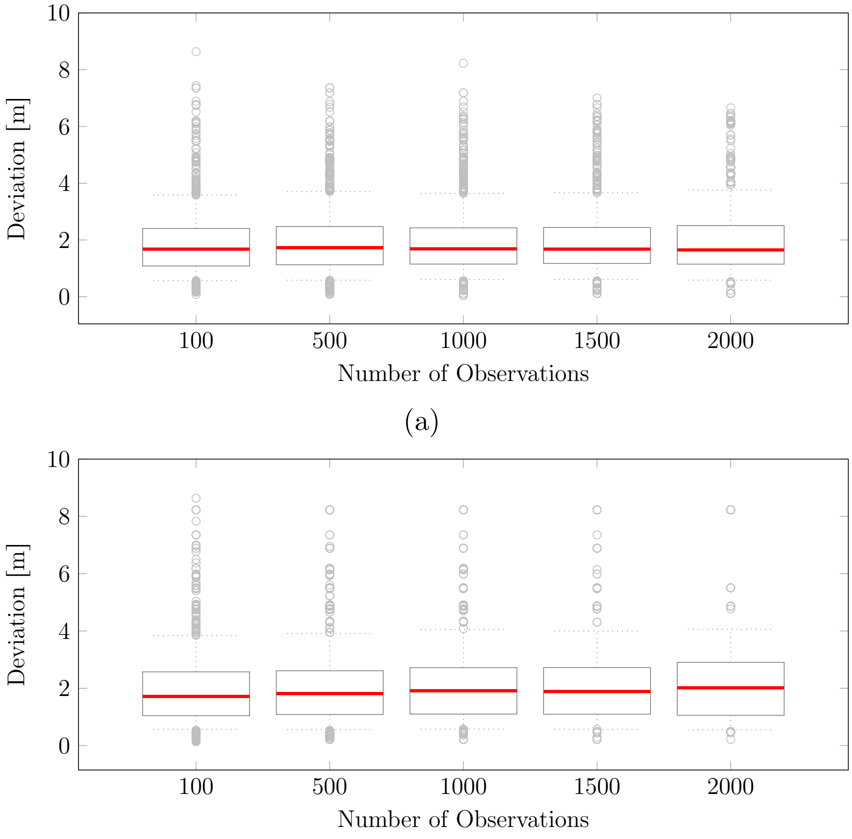


Figure 7.10: Runtime evaluation of CI and FCI for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* scenario, and the corresponding linear regressions.



(a)

(b)

Figure 7.11: Accuracy evaluation of CI (subfigure a) and FCI (subfigure b) for 100, 500, 1000, 1500, and 2000 observations from the *real-world* scenario.

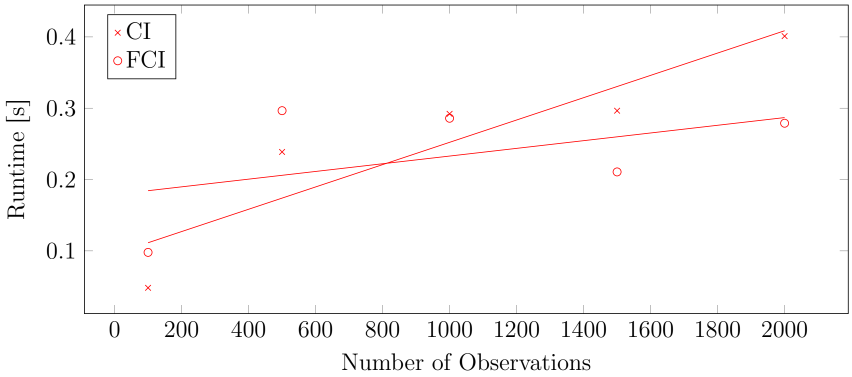


Figure 7.12: Runtime evaluation of CI and FCI for 100, 500, 1000, 1500, and 2000 observations from the *real-world* scenario, and the corresponding linear regressions.

the determined average median accuracy is 0.7 m for 100 observations, and 0.16 m for 2000 observations. This can be considered, based solely on the median values and the lower and upper quartiles, as a very good result. Regarding the frequent outliers that induce a partial divergence of the filter, the overall robustness thereof is considered as minor. Similar results emerge at the real-world scenario. EKF-based Online-SLAM provides an average median accuracy of approximately 1.67 m. The average median accuracy can be considered as nearly independent of the actual observation count. However, again severe outliers can be noticed, which lead to a low assessment of the filter's robustness.

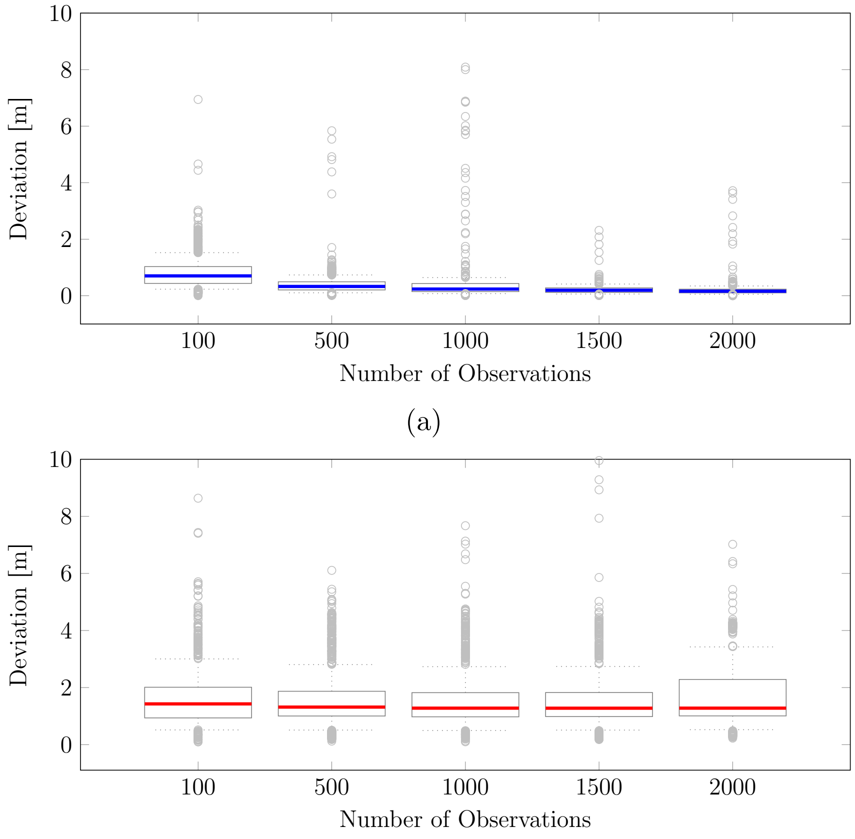
In figure 7.14 the averaged runtimes of the EKF-based Online-SLAM are visualized for both the synthetic and the real-world scenario. One can notice that both scenarios feature nearly similar runtimes.

7.5.4 Fusion via BA-based Full-SLAM

In the following, the fusion of point-shaped landmark observations via BA-based Full-SLAM, solved directly via LM, and after linearization via QR, is evaluated according to its accuracy and runtime for both the synthetic and real-world scenario. The LM-based approach is limited to at most five iterations, whereas the QR-based one is utilized without relinearization to emphasize its advantage.

In figure 7.15 the gained average accuracies for the synthetic scenario, and in figure 7.17 for the real-world one, are provided. Concerning the synthetic scenario, one can notice that both approaches to BA-based Full-SLAM monotonically converge and are able to determine tight estimates of the fused landmarks. For 100 observations the average median deviation for solving via LM is 0.65 m and 0.23 m for 2000 observations. The solving via QR leads to a slightly increased error of 1.25 m for 100 observations and 0.35 m for 2000 observations. Furthermore, the determined deviations are in both cases minor, what does reflect the superior robustness of BA-based Full-SLAM. The same can be observed at the real-world scenario. Both solving methods are able to reduce the amplitude of outliers with an increasing observation count. The average median deviation is for 100 observation approximately 1.39 m and for 2000 observations approximately 1.24 m. However, despite the coarse observation error model, in the case of the real-world scenario, both approaches to BA-based Full-SLAM provide highly robust estimates.

In figure 7.16 the runtimes of BA-based Full-SLAM solved via LM and QR for the synthetic scenario, and in figure 7.18 for the real-world one are provided. One can notice the much higher computational efficiency of the QR-based solver, which is between 8- and 20-times faster for 2000 observations, than the LM-based one.



(a)

(b)

Figure 7.13: Accuracy evaluation of EKF-based Online-SLAM for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* (subfigure a) and *real-world* (subfigure b) scenario.

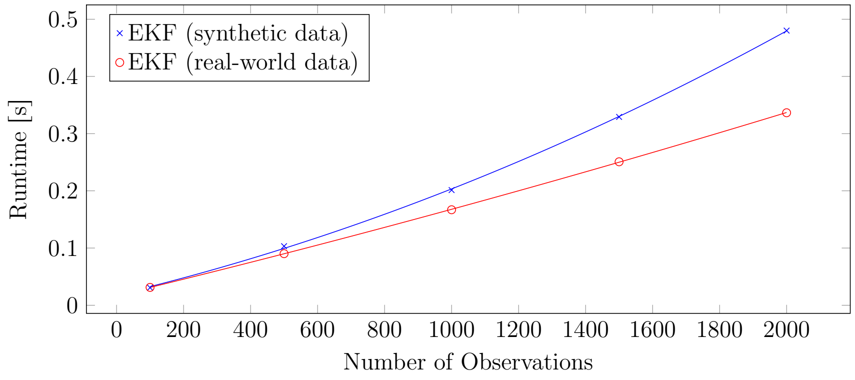
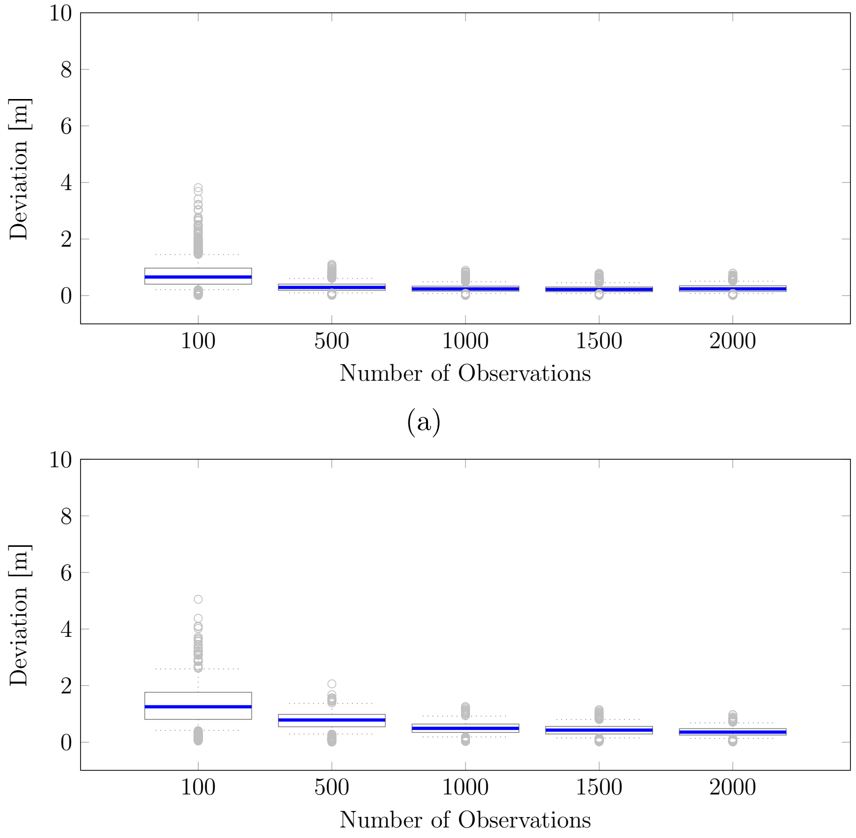


Figure 7.14: Runtime evaluation of EKF-based Online-SLAM for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* and *real-world* scenario, and the corresponding regressions.



(a)

(b)

Figure 7.15: Accuracy evaluation of BA-based Full-SLAM solved via LM (subfigure a) and QR (subfigure b) for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* scenario.

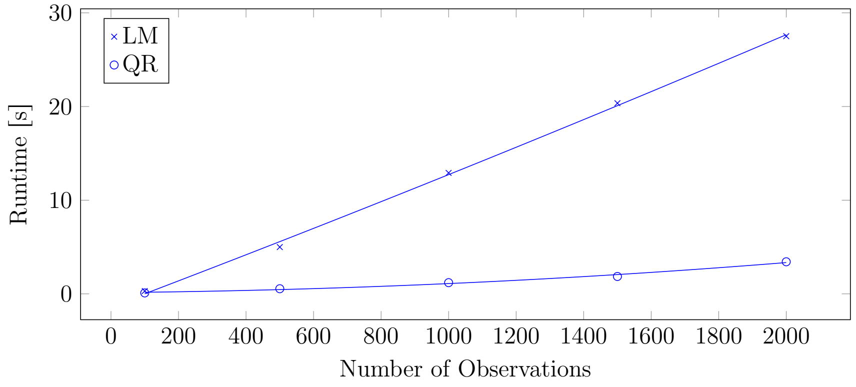
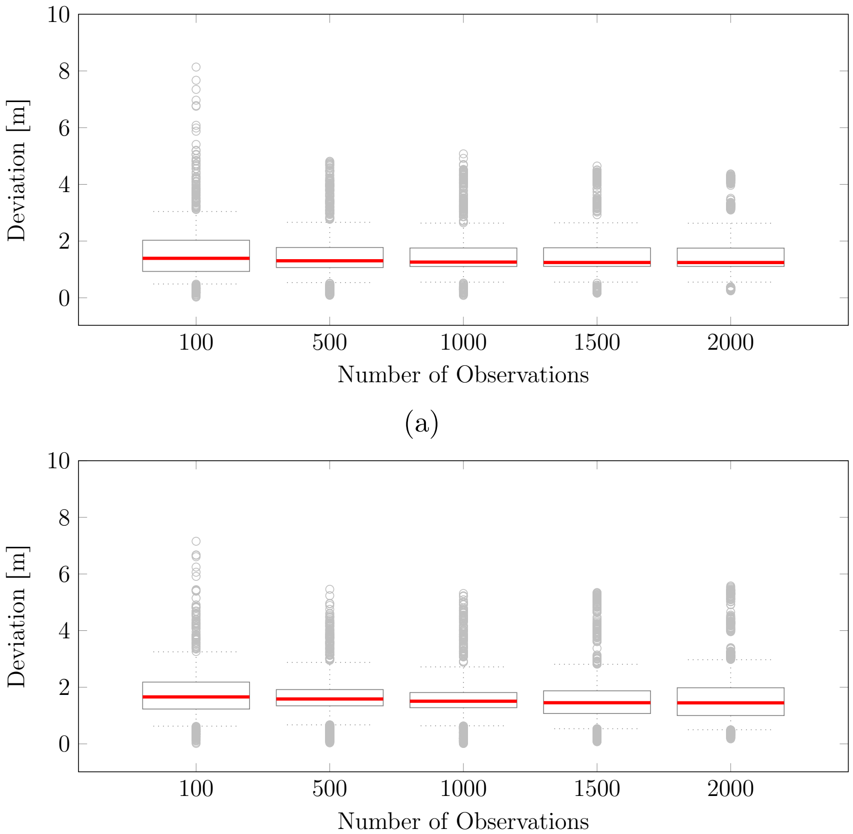


Figure 7.16: Runtime evaluation of BA-based Full-SLAM solved via LM and QR for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* scenario, and the corresponding regressions.



(a)

(b)

Figure 7.17: Accuracy evaluation of BA-based Full-SLAM solved directly via LM (subfigure a) and after linearization via QR (subfigure b) for 100, 500, 1000, 1500, and 2000 observations from the *real-world* scenario.

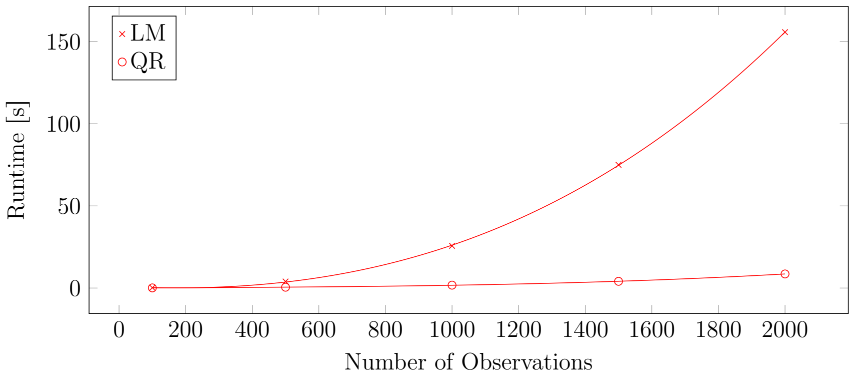
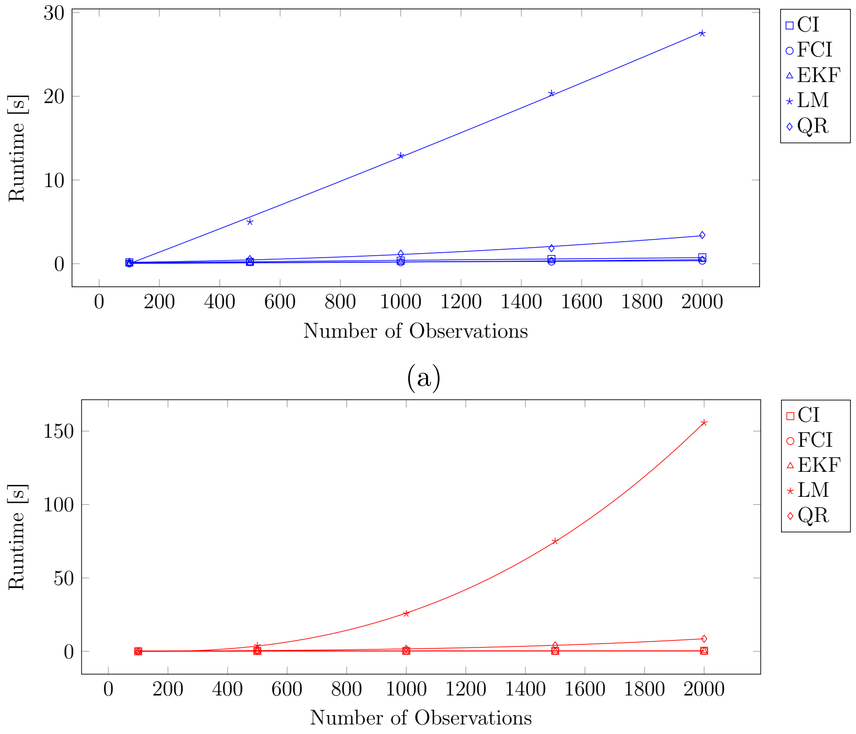


Figure 7.18: Runtime evaluation of BA-based Full-SLAM solved directly via LM and after linearization via QR for 100, 500, 1000, 1500, and 2000 observations from the *real-world* scenario, and the corresponding regressions.



(a)

(b)

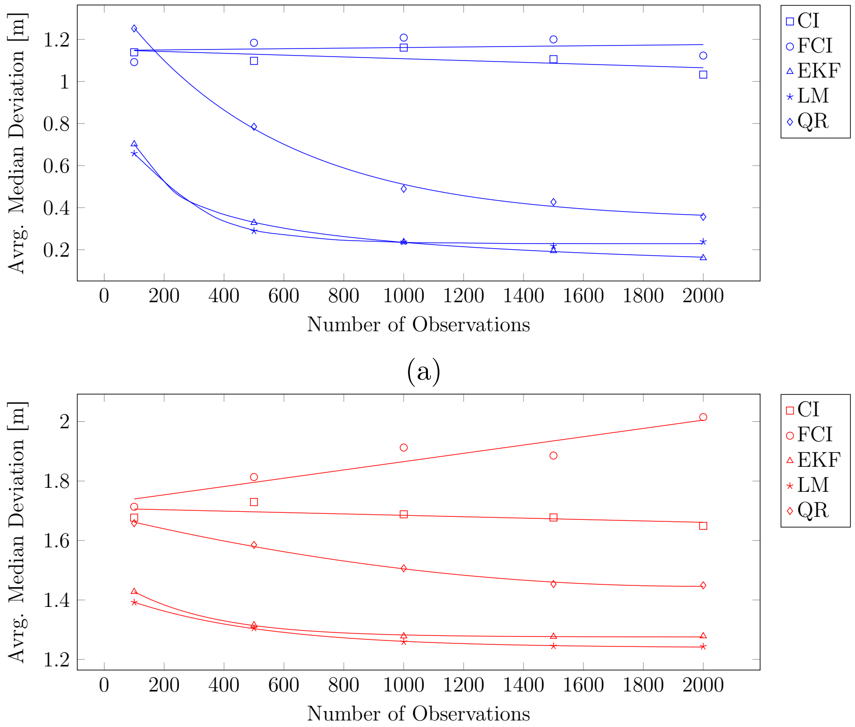
Figure 7.19: Runtime comparison of point-shaped landmark observation fusion via CI, FCI, EKF-based Online-SLAM, BA-based Full-SLAM solved directly via LM and after linearization via QR for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* (subfigure a) and *real-world* (subfigure b) scenario, and the corresponding regressions.

7.6 Discussion

In this chapter, five approaches to the fusion of point-shaped CVD, such as traffic signs, traffic lights, or reflector posts, have been adapted and evaluated in detail. To be more specific: CI, FCI, EKF-based Online-SLAM, BA-based Full-SLAM, which is solved either directly via LM or after linearization via QR.

CI and FCI apparently differ from the other presented approaches, as they cannot be directly assigned to the category of SLAM. However, they are considered as a worthwhile alternative in the case that the computational efficiency of the fusion is put at a premium. On the other side, approaches to SLAM, regardless of whether Online-SLAM or Full-SLAM, are known to be computationally demanding. Full-SLAM is known to be even more computationally challenging than Online-SLAM because of its principal lack of marginalization.

The runtimes of the evaluated approaches are contrasted with each other in figure 7.19 for both the synthetic and the real-world scenario. One can notice that BA-based



(a)

(b)

Figure 7.20: Accuracy comparison of point-shaped landmark observation fusion via CI, FCI, EKF-based Online-SLAM, BA-based Full-SLAM solved directly via LM and after linearization via QR for 100, 500, 1000, 1500, and 2000 observations from the *synthetic* (subfigure a) and *real-world* (subfigure b) scenario, and the corresponding regressions.

Full-SLAM solved directly via LM is computationally most demanding, followed by EKF-based Online-SLAM.

In figure 7.20 the algorithms are contrasted with each other concerning their average median deviation for the synthetic and the real-world scenario. One can notice that EKF-based Online-SLAM and BA-based Full-SLAM solved directly via LM provide the most accurate results, both for the synthetic and the real world scenario.

Concerning the robustness, which is *crucial* for the deduction of high-precision maps from CVD, BA-based Full-SLAM, regardless of whether solved by LM or QR, (figures 7.13, 7.15 and 7.17) should be preferred to EKF-based Online-SLAM. Both solving methods to BA-based Full-SLAM have distinct pros and cons. The direct one via LM is accompanied by a high accuracy but also by a significantly larger computational effort. The solving via QR after linearization remarkably alleviates the computational effort but is also accompanied by linearization errors, which decrease the overall estimation accuracy. However, the resulting linearization errors can be alleviated by means of relinearization. The challenge of robustifying EKF-based Online-SLAM to prevent its divergence has been tackled by several authors.

A commonly utilized approach is based upon the identification of divergence and the subsequent reinitialization of the diverging filter [139]. This would also be a sensible extension of the provided implementation. However, approaches to BA-based Full-SLAM are, grounded on their working principle, still superior concerning their robustness. Therefore, based upon the gained results, BA-based Full-SLAM is unreservedly recommended for the fusion of point-shaped CVD, when robustness and accuracy is strictly demanded and the required computational resources are just secondary.

8
Fusion of Complex Landmark Data

In the previous chapter 7, algorithms for the fusion of point-shaped landmark observations have been presented and opposed to each other. However, there are several types of landmarks within the road network context, such as crossroads, roundabouts, traffic islands, pedestrian crossings, and sign gantries, which can be considered as being complex (section 1.4). The reduction of these complex landmarks to point-shaped ones or the independent consideration of their primitive features, such as points, circles, and lines, involves a severe loss of useful information and accuracy and should be avoided. Therefore, a generalized approach to the parametric description and fusion of complex landmarks is presented and evaluated in this chapter.

Contents

8.1 Introduction

Besides many point-shaped landmarks (chapter 7), such as road signs, traffic lights, and line-shaped ones, such as lane markings, the road network context possesses also more complex landmarks, such as crossroads, roundabouts, and sign gantries. Those complex landmarks should not be just reduced to e. g. their center points, referred to as primary attributes within this thesis, as it would lead to a severe loss of useful detail information. Furthermore, it is also not feasible to consider the multiple attributes of complex landmarks independently, as one would ignore this way the usually tight attribute interdependencies. Instead, an approach should be favored that allows to fuse complex landmarks under the strict consideration of attribute interdependencies and to retain their full geometric detail.

This chapter starts with a brief summary of the state of the art of fusing more complex landmark observations (section 8.2). Afterwards, a generalized approach for the parametric description of complex landmarks based on a DAG[1], referred to as the ADG[2], is presented (section 8.3.2). It is incorporated into both EKF[3]-based Online- (section 8.5) and BA[4]-based Full-SLAM[5] (section 8.6). Finally, the extended BA-based approach to Full-SLAM, which is solved via Sparse-QR[6], is evaluated in detail (section 8.7). This is motivated by the fact that this approach to SLAM has been identified to provide a superior robustness on point-shaped landmark data while still being computationally feasible if employing linearization (section 7.5). For evaluation purposes, a hybrid scenario consisting of roundabout and traffic sign observations is selected. The roundabout, as a complex landmark, is chosen intentionally, as its parametric description provides an above average complexity and multiple attribute interdependencies. However, the approach presented is not limited to roundabout observations. It can be adapted in a straightforward manner to other types of complex landmarks by defining an appropriate parametric description of their shapes.

[1] Directed Acyclic Graph (DAG)
[2] Attribute Dependency Graph (ADG)
[3] Extended Kalman Filter (EKF)
[4] Bundle Adjustment (BA)
[5] Simultaneous Localization and Mapping (SLAM)
[6] QR Factorization (QR)

8.2 State of the Art

First attempts for tackling the challenge of complex landmark observations have been to additionally incorporate the diameter of a landmark or the description of a landmark by a polygonal shape. They have already led to higher accuracies and faster convergence, i. a. due to a more accurate determination of correspondences [3, 35], thus showing the general usefulness of fusion of complex landmarks.

Two main approaches can be distinguished: non-parametric and parametric ones. A non-parametric approach has been i. a. proposed by Nieto et al. [87]. They suggested to describe and fuse complex landmark observations of an arbitrary shape by incorporating scan-matching of raw sensor data into EKF-based SLAM. A disadvantage of this approach is that it requires to transmit huge amounts of raw sensed data via the cellular network when utilized for the fusion of CVD[7].

First steps towards the concise, parametric description of complex landmarks were done by both Emter [36] and Turan [128]. They have considered trees as complex landmarks. Both extended the description of point-shaped tree landmark observations by an additional attribute, the radius. Furthermore, Emter additionally incorporated a visual tree signature into the point-shaped landmark description. Both authors noticed a more reliable determination of correspondences, a faster convergence, and a resulting more accurate map.

Because of bandwidth restrictions of the cellular network it is (by now) not feasible to propagate large amounts of raw sensed data, such as point clouds, from common vehicles to the back-end-side. Therefore, the focus of this thesis is on a novel concise, parametric description of complex road network features. The parametric description of landmarks allows sharing the workload between the vehicle- and back-end-side. This can be considered as a crucial feature for large-scale applications.

8.3 Parametric Representation of Complex Landmarks

A parametric representation of a complex landmark separates structure and appearance. The structure is specified by its components and their arrangement depending on a finite set of parameters. An appearance is obtained by assignment of concrete, feasible values to the parameters. For example, components of circular roundabouts could be the road segment in the roundabout itself and at its junctions. Parameters may be the radii and the common center of an inner and an outer circle which define the boundary of the roundabout road, and angles which specify the locations of the junctions.

[7] Collective Vehicle Data (CVD)

The application of the fusion methods of chapter 7 requires the representation of landmark estimates and landmark observations in a way that a measurement residual in the case of EKF-based online-SLAM, cf. equation 7.12, or a landmark observation probability in the case of BA-based Full-SLAM, cf. equation 7.43, can be calculated. Both are based on some sort of difference between an observation and an observation prediction according to an observation prediction model. Thus representations of landmark estimates and observations are required which fulfill those conditions. One possibility is to use a vector consisting of the parameters of the parametric representation as a representation of landmark estimates and observations. For observations, the values of the parameters might for example be obtained by fitting the parametric model into a point cloud acquired by the sensor mounted on the vehicle, cf. [91].

A problem of this kind of approximation is that independent updating of the parameter estimates might imply inconsistent parameter values because of observation and estimation errors due to imprecision or incompleteness of measurements. In this chapter, an approach to reducing the problem is presented. It consists in introducing auxiliary attributes which may depend on several parameters and which are used for observation prediction. The attribute interdependencies are represented by a DAG, referred to as the ADG.

8.3.1 An Example: Circular Roundabouts

In the following, circular roundabout landmarks are used as main case study since roundabouts are particularly complex. The parametric representation chosen orients itself on the construction guidelines for circular roundabouts with concentric rings according to Haller [50]. The roundabout is defined by three circles which have the same center point (figure 8.1). The radii of the three concentric circles are denoted as inner radius, inner ring radius, and outer radius, respectively. The first two circles represent the so-called truck apron, and the last two the road. The height of the curbstone of the inner ring is another attribute.

A junction is defined by two circles, an inner rounding circle and an outer rounding circle. The inner rounding circle is tangential to the middle circle of the roundabout, and the outer rounding circle is tangential to the outer circle of the roundabout. Thus the two circles are uniquely specified by their center points. Their center points are given by two vectors, outer relative vector and inner relative vector, respectively, which provide their locations relatively to the center point of the roundabout. The boundary of the road of a junction close to the roundabout is induced by arc segments of the two circles. The arc segments are defined by an outer rounding arc angle and an inner rounding arc angle which are measures relative to the line between

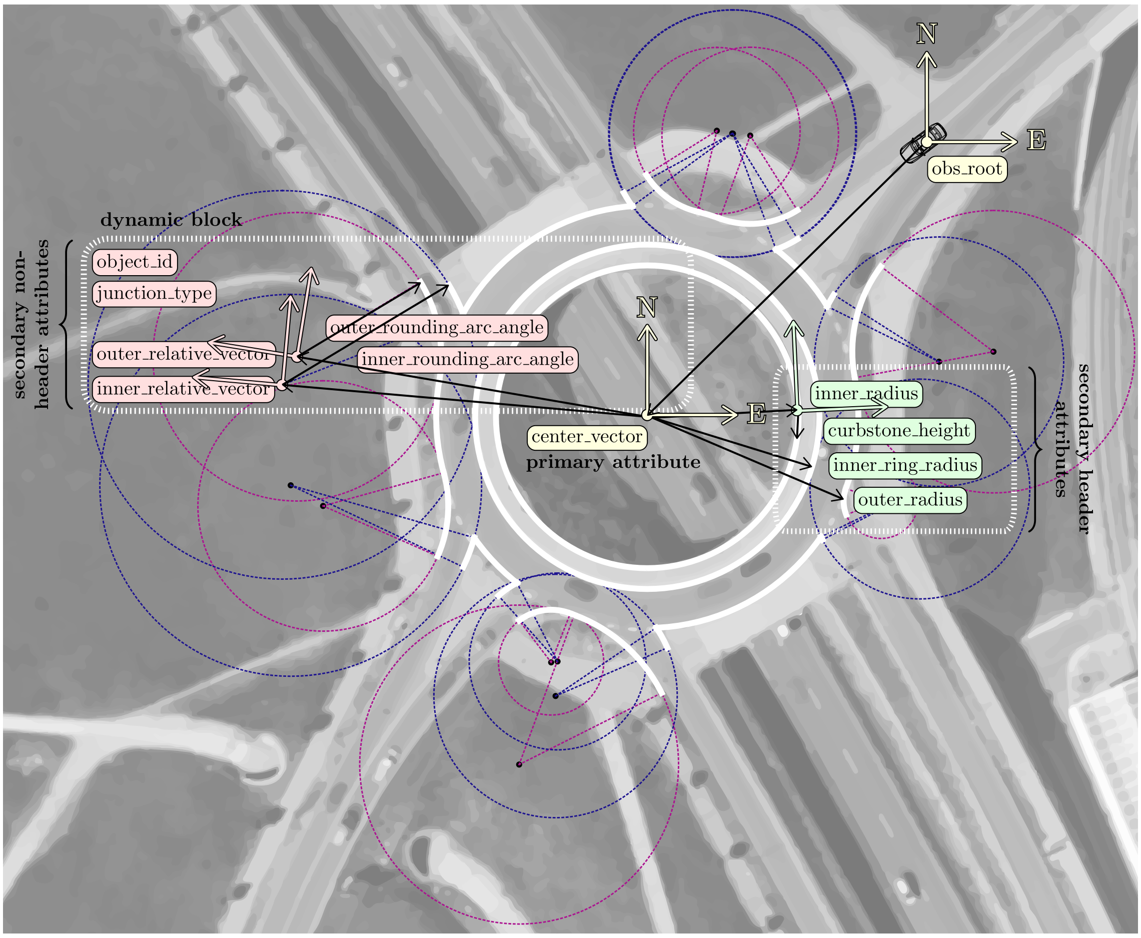


Figure 8.1: Parametric description of a roundabout in Ingolstadt according to the construction guides provided by Haller [50]. Digital orthophoto: © GeoBasis-DE / BKG 2018 (modified).

the center point of each of the two circles and the center point of the roundabout, respectively.

This informal description will be formalized later in section 8.3.6 based on the concept of ADGs which is presented in the next section. The parameters of this example are “center frame”, “inner radius”, “inner ring radius”, “outer radius”, “outer relative vector”, “inner relative vector”, “outer rounding arc angle” and “inner rounding arc angle” and will inspire the choice of the so-called LA[8]s of ADGs which play the role of parameters in this concept.

8.3.2 Attribute Dependency Graph (ADG)

ADGs are a special kind of parametric representation of complex landmarks. They specify ADF[9]s which determine the pose of an AA[10] in dependence on one or several

[8] Landmark Attribute (LA)
[9] Attribute Dependency Function (ADF)
[10] Auxiliary Attribute (AA)

other AAs. The functions are controlled by so-called LAs which are the parameters of the representation. The ADGs are DAGs. Every vertex represents an elementary ADF and is labeled with a corresponding elementary LA. An edge represents the AA which is the output of the ADF of its start vertex and an input of the ADF of its end vertex. A concrete incarnation of a complex landmark is represented by the values of its LAs.

The ADF of an AA of the ADG is determined by a subgraph in the ADG. This so-called ACG[11] is induced by all those vertices for which a directed path exists to the vertex which calculates the AA. This vertex is the root of the subgraph. The ACGs are denoted by 🔗 $(\boldsymbol{a})$ where $\boldsymbol{a}$ is the root. The inputs of the ADF are the inputs of the ADFs of the leaves of the ACG. The ADF results by concatenating the ADFs at the vertices according to the edges of the subgraph.

An important special case is that the ADG is a tree (ADT[12]). For ADTs a vertex has at most one ingoing edge, and the ACGs become paths. The concatenation of two ADFs can be mathematically expressed as follows. Let $\boldsymbol{f}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_f)$ and $\boldsymbol{g}(\boldsymbol{a}_g^{\text{aux.}}, \boldsymbol{a}_g)$ be two ADFs with LAs $\boldsymbol{a}_f$ and $\boldsymbol{a}_g$ and input AAs $\boldsymbol{a}_f^{\text{aux.}}$ and $\boldsymbol{a}_g^{\text{aux.}}$, respectively. The result of concatenation is a function $\boldsymbol{h}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_h) = \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_f), \boldsymbol{a}_g)$ where $\boldsymbol{a}_h$ is a LA which is depending on $\boldsymbol{a}_f$ and $\boldsymbol{a}_g$. According to the terminology of section 3.4, $\boldsymbol{a}_h$ results from $\boldsymbol{a}_f$ and $\boldsymbol{a}_g$ by compounding, i. e. $\boldsymbol{a}_h = \boldsymbol{a}_f \oplus \boldsymbol{a}_g$.

In the case of an ADG with possibly multiple ingoing edges at its vertices, $\boldsymbol{a}_f^{\text{aux.}}$ represents a d-tuple, $d \geqslant 1$, of AAs which are provided by d ADFs $\boldsymbol{f}_i(\boldsymbol{a}_{f_i}^{\text{aux.}}, \boldsymbol{a}_{f_i})$, $i = 0, \ldots, d-1$. The concatenation yields $\boldsymbol{h}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_h) = \boldsymbol{g}(\boldsymbol{f}_0(\boldsymbol{a}_{f_0}^{\text{aux.}}, \boldsymbol{a}_{f_0}), \ldots, \boldsymbol{f}_{d-1}(\boldsymbol{a}_{f_{d-1}}^{\text{aux.}}, \boldsymbol{a}_{f_{d-1}}), \boldsymbol{a}_g)$ where $\boldsymbol{a}_f^{\text{aux.}} = (\boldsymbol{a}_{f_0}^{\text{aux.}}, \ldots, \boldsymbol{a}_{f_{d-1}}^{\text{aux.}})$ and $\boldsymbol{a}_h$ is depending on $\boldsymbol{a}_{f_i}$, $i = 0, \ldots, d-1$ and $\boldsymbol{a}_g$.

The LAs of the ADFs will be used for observation prediction in section 8.5.2. The attribute observation prediction functions have the form $\boldsymbol{h}^{\text{a}}\,(\boldsymbol{x}_l^{\text{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,$ 🔗 $(\boldsymbol{a}_j^m))$ where $\boldsymbol{x}_l^{\text{v}}$ denotes the vehicle position estimate, $\boldsymbol{a}_j^m$ an LA estimate of the complex landmark estimate $\boldsymbol{l}_j$, and 🔗 $(\boldsymbol{a}_j^m)$ the estimates of all attributes in the ACG for the LA $\boldsymbol{a}_j^m$. By optimization over subgraphs of concatenation the error can be uniformly distributed among the LAs which are the shape-representing parameters.

This general definition of an ADG is now made concrete in an appropriate way. The AAs are represented by frames, i. e. an origin together with two or three orthonormal basis vectors, depending on the dimension of the complex landmark. An ADF with exactly one input-AA specifies the location of the frame of its output AA relatively to the frame of the input-AA. For example, in the case of a circular roundabout (figure 8.1), the center point of every circle can be represented by a frame.

[11] Attribute Concatenation Graph (ACG)

[12] Attribute Dependency Tree (ADT)

The origin of the frame defines the center point, and the vectors the orientation of the frame. The location of the frame of the center point of the inner rounding circle can be specified relatively to the frame of the center point of the circles of the roundabout.

In this framework, an ADF $\boldsymbol{f}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_f)$ specifies the relative location of the origin and the relative orientation of the basis vectors of the output frame. The relative location of the origin is defined by a position vector $\boldsymbol{t}_f$. The relative orientation is described by the rotational difference between the basis vectors of the two frames involved. In the two-dimensional case, the rotation is defined by an angle α_f. The parameters of the position vector and the rotational difference are the LA of the ADF, that is $\boldsymbol{a}_f = (\boldsymbol{t}_f, \alpha_f)$. The concatenation $\boldsymbol{h}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_h) = \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{a}_f^{\text{aux.}}, \boldsymbol{a}_f), \boldsymbol{a}_g)$ of two such functions has the same form with $\boldsymbol{a}_h = (\boldsymbol{t}_h, \alpha_h)$. Its angle of rotation is $\alpha_h = \alpha_f + \alpha_g$. If the position vectors are in Cartesian representation, its position vector is $\boldsymbol{t}_h = \boldsymbol{R}_{\alpha_f}\boldsymbol{t}_f + \boldsymbol{t}_g$, where $\boldsymbol{R}_{\alpha_f}$ is the matrix of rotation by α_f.

An ADF with more than one input AA determines the location of its output frame relatively to all input frames. This might e. g. be performed by taking some sort of weighted mean of the input frames where the weights are LA of the ADF.

The implementation of ADGs in the thesis is restricted to ADTs so that the latter, general case does not occur. The general case is postponed to future work.

8.3.3 Dynamic Blocks and ADG-schemes

The basic idea of parametric representations is to specify a family of incarnations with a single scheme. The incarnations are defined by the values of the parameters. ADGs are limited in the sense that the number of components of a complex landmark is fixed and only their geometry and mutual location can be varied. However, a class of complex landmarks may contain incarnations with different numbers of components. Examples are roundabouts or crossroads which may have different numbers of junctions. When starting with the data acquisition the number of junctions is not yet known, and it has possibly to be adapted when more information is available.

For this purpose ADG-schemes are introduced. An ADG-scheme has the form of an ADG, but in addition disjoint, connected, vertex-induced subgraphs can be marked. Those subgraphs are called “dynamic blocks”. Every dynamic block has an attribute called “dynamic block counter”. Depending on the value of the dynamic block counter, an ADG-instance of the ADG-scheme is generated by creating n copies of every dynamic block where n is the value of the dynamic block counter of the block. The vertices of the dynamic block copies are connected to vertices outside the block by edges between all copies of vertices whose originals have been connected by an edge.

The concepts of ADGs and ADG-schemes can be summarized as follows:

Definition 8.1: Attribute Dependency Graph (ADG)

An ADG represents the dependency of the attributes of a complex landmark and is defined as the instantiation of an ADG-scheme.

Definition 8.2: Attribute Dependency Graph Scheme (ADG-scheme)

An ADG-scheme is given by a directed acyclic graph (DAG) with dynamic blocks.

Vertex Set: Represents the elementary ADFs which specify the pose of an output-AA relatively to the poses of the input-AAs. The functions are controlled by LAs.

Edge Set: Specifies the concatenations of the elementary ADFs and AAs communicated from the start-ADF to the end-ADF of the edges.

Dynamic Block: Is a connected subgraph induced by a subset of vertices together with a block counter attribute. The dynamic blocks have to be disjoint.

Definition 8.3: Instanced Attribute Dependency Graph Scheme (Instanced ADG-scheme)

An instanced ADG-scheme results by replacement of every dynamic block with a block-count-number of disjoint copies. The edges linking a dynamic block to its environment are canonically copied.

Definition 8.4: Attribute Concatenation Graph (ACG)

An ACG is defined for every LA of an instanced ADG and is induced by all those vertices for which a directed path exits to the vertex of the LA. The ACG of a LA $\boldsymbol{a}$ is denoted by 🔗$(\boldsymbol{a})$.

8.3.4 Relation to Point-shaped Landmarks

In order to treat point-shaped landmarks as a special case of complex landmarks, a convention on the LAs is introduced by defining several types of LAs:

Definition 8.5: Categories of Attributes of Complex Landmarks

Primary Attributes: Specify the location of the complex landmark.

Secondary Attributes: Specify the structure and shape of the complex landmark. The secondary attributes are distinguished in two categories:

↪ **Header Attributes** which are obligatory, and

↪ **Non-header Attributes** which are optional.

A natural choice of the primary attribute is a frame whose origin is the center point of the region covered by a complex landmark and whose basis vectors are chosen in an appropriate way, e. g. according to some global geographic reference frame. For a roundabout the center of the road-bounding circles may be used as primary attribute. The related ADF can be constant, i. e. without any input, or can have the sensor frame as an input.

Secondary header attributes of a roundabout can be the radii of the circles. The angles of the junctions can be secondary non-header attributes. One reason for distinguishing between header and non-header secondary attributes is to keep the number of attributes dynamically depending on the concrete instantiation, cf. the concept of dynamic blocks (section 8.3.3). For example, the number of junctions of the roundabout can be different so that secondary attributes specifying junctions are non-header attributes. Another reason is that secondary header attributes are required to be obligatory observed, while secondary non-header attributes are optional to be observed with every complex landmark observation.

In this framework, a point-shaped landmark is a complex landmark without secondary attributes and just one primary attribute which represents its position.

8.3.5 Treating Rotational Attributes

Rotational attributes occur in ADGs in relation with frame rotations and polar representations of position vectors. A polar position vector is denoted by (r, φ) with radius r and angle φ. Within the SLAM-approaches presented in this chapter, it is assumed that the position vector attributes are represented in Cartesian notation. One approach to treating rotational attributes in this context is by matrix representation of rotations. The central operation is matrix-vector multiplication, which e. g. occurs for the compounding of LAs of ADGs, see section 8.3.2 for the general two-dimensional case and section 8.3.6 for the example of circular roundabouts.

In the following a linearization of a rotation matrix-vector-multiplication,

$$\boldsymbol{R}_{\alpha} \cdot \boldsymbol{a} \approx \underbrace{\boldsymbol{R}^{lh}_{\alpha_0} \cdot (\alpha - \alpha_0) \cdot \boldsymbol{a}_0}_{\textit{rotation part}} + \underbrace{\boldsymbol{R}^{rh}_{\alpha_0} \cdot \boldsymbol{a}}_{\textit{vector part}}, \tag{8.1}$$

for use in the matrix formulation of linearized extended BA-based Full-SLAM is derived, where

$$\boldsymbol{R}_{\alpha} \overset{\text{2D}}{=} \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \tag{8.2}$$

α is a given angle, $\boldsymbol{a}$ is a vector, and $\boldsymbol{R}^{lh}_{\alpha_0}$, $\boldsymbol{R}^{rh}_{\alpha_0} \cdot \boldsymbol{a}$ is defined below.

Linearization of the trigonometric terms sin and cos by their 1st-order Taylor-series (appendix A.3) with respect to the linearization point α_0 yields

$$\begin{aligned}\sin\alpha &= \sin\alpha_0 + \cos\alpha_0 \cdot (\alpha - \alpha_0),\\ \cos\alpha &= \cos\alpha_0 - \sin\alpha_0 \cdot (\alpha - \alpha_0).\end{aligned} \tag{8.3}$$

Putting the linearized trigonometric terms into the rotation matrix leads to the linearized rotation matrix

$$\begin{aligned}\boldsymbol{R}_{\alpha-\alpha_0} &\overset{2\text{D}}{\approx} \begin{pmatrix}\cos\alpha_0 - \sin\alpha_0\cdot(\alpha-\alpha_0) & -(\sin\alpha_0 + \cos\alpha_0\cdot(\alpha-\alpha_0))\\ \sin\alpha_0 + \cos\alpha_0\cdot(\alpha-\alpha_0) & \cos\alpha_0 - \sin\alpha_0\cdot(\alpha-\alpha_0)\end{pmatrix}\\ &\Leftrightarrow \begin{pmatrix}\cos\alpha_0 - \sin\alpha_0\cdot(\alpha-\alpha_0) & -\sin\alpha_0 - \cos\alpha_0\cdot(\alpha-\alpha_0)\\ \sin\alpha_0 + \cos\alpha_0\cdot(\alpha-\alpha_0) & \cos\alpha_0 - \sin\alpha_0\cdot(\alpha-\alpha_0)\end{pmatrix}\\ &\Leftrightarrow \begin{pmatrix}\cos\alpha_0 & -\sin\alpha_0\\ \sin\alpha_0 & \cos\alpha_0\end{pmatrix}\cdot 1 + \begin{pmatrix}-\sin\alpha_0 & -\cos\alpha_0\\ \cos\alpha_0 & -\sin\alpha_0\end{pmatrix}\cdot(\alpha-\alpha_0)\\ &\Leftrightarrow \underbrace{\begin{pmatrix}-\sin\alpha_0 & -\cos\alpha_0\\ \cos\alpha_0 & -\sin\alpha_0\end{pmatrix}}_{\boldsymbol{R}^{lh}_{\alpha_0} = \boldsymbol{J}_{\alpha_0}}\cdot(\alpha-\alpha_0) + \underbrace{\begin{pmatrix}\cos\alpha_0 & -\sin\alpha_0\\ \sin\alpha_0 & \cos\alpha_0\end{pmatrix}}_{\boldsymbol{R}^{rh}_{\alpha_0}}\\ &\Leftrightarrow \boldsymbol{R}^{rh}_{\alpha_0} + \boldsymbol{J}_{\alpha_0}(\alpha-\alpha_0)\end{aligned} \tag{8.4}$$

Multiplying the linearized rotation matrix with the development $\boldsymbol{a} = (\boldsymbol{a} - \boldsymbol{a}_0) + \boldsymbol{a}_0$ around $\boldsymbol{a}_0$ results in

$$\begin{aligned}&\left(\boldsymbol{R}^{lh}_{\alpha_0}\cdot(\alpha-\alpha_0) + \boldsymbol{R}^{rh}_{\alpha_0}\right)\cdot\boldsymbol{a}\\ \approx\ &\boldsymbol{R}^{lh}_{\alpha_0}\cdot(\alpha-\alpha_0)\cdot\boldsymbol{a}_0 + \boldsymbol{R}^{rh}_{\alpha_0}\cdot\boldsymbol{a}.\end{aligned} \tag{8.5}$$

The left-hand part is associated with the rotational factor and a right-hand part with the vectorial one $\boldsymbol{a}$.

The rotational factor $\boldsymbol{R}^{lh}_{\alpha_0}\cdot\boldsymbol{a}_0$ can be transformed into a pure premultiplication factor for $(\alpha - \alpha_0)$:

$$\boldsymbol{R}^{lh}_{\alpha_0}\cdot(\alpha-\alpha_0)\cdot\boldsymbol{a}_0 = \begin{pmatrix}\boldsymbol{R}^{lh}_{\alpha_0;0,0}\cdot\boldsymbol{a}^0_0 & \boldsymbol{R}^{lh}_{\alpha_0;0,1}\cdot\boldsymbol{a}^0_1\\ \boldsymbol{R}^{lh}_{\alpha_0;1,0}\cdot\boldsymbol{a}^0_0 & \boldsymbol{R}^{lh}_{\alpha_0;1,1}\cdot\boldsymbol{a}^0_1.\end{pmatrix}\cdot\begin{pmatrix}\alpha-\alpha_0\\ \alpha-\alpha_0\end{pmatrix}. \tag{8.6}$$

According to section 3.4, the covariance of the linear approximation of the function $\boldsymbol{R}_\alpha$ for a stochastic α and under assumption of no cross-correlation is

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{R}_\alpha\boldsymbol{a}) &\approx \boldsymbol{J}_{\alpha_0,\boldsymbol{a}_0}\cdot\mathrm{Cov}((\alpha,\boldsymbol{a})^T)\cdot\boldsymbol{J}^T_{\alpha_0,\boldsymbol{a}_0}\\ &= \boldsymbol{J}_{\alpha_0}\cdot\mathrm{Var}(\alpha)\cdot\boldsymbol{J}^T_{\alpha_0} + \boldsymbol{J}_{\boldsymbol{a}_0}\cdot\mathrm{Cov}(\boldsymbol{a})\cdot\boldsymbol{J}^T_{\boldsymbol{a}_0}.\end{aligned} \tag{8.7}$$

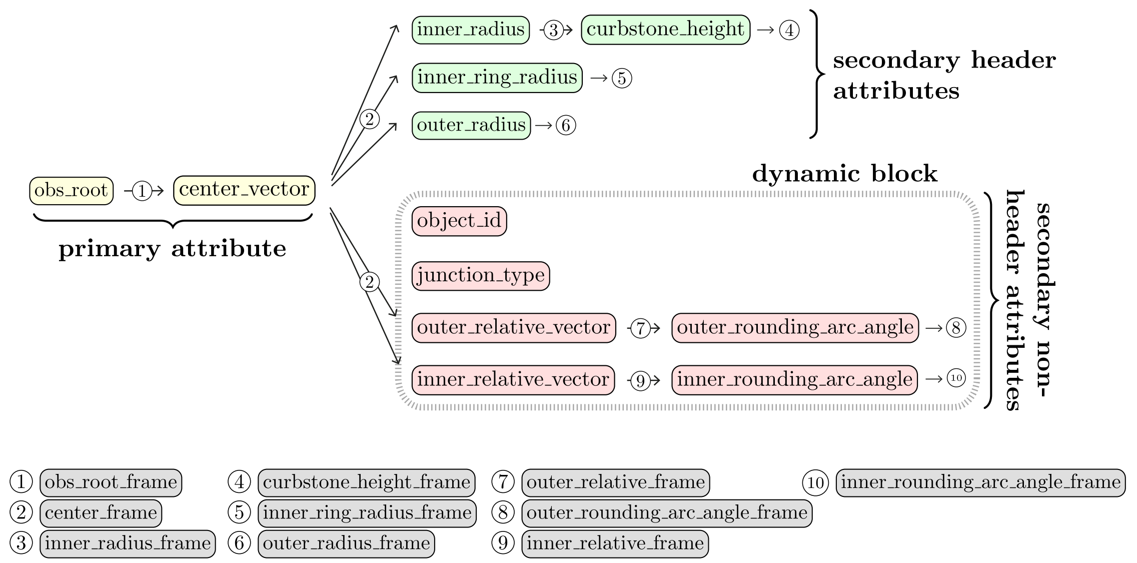


Figure 8.2: ADG of a roundabout. Primary attributes are highlighted in yellow ●, secondary header attributes in green ●, secondary non-header attributes in red ● (representing a junction each), and AAs in gray ●.

where $\boldsymbol{J}_{\alpha_0,\boldsymbol{a}_0}$ is the Jacobian matrix of the linearized function $\boldsymbol{R}_\alpha$.

The Jacobian matrix of the linearized function $\boldsymbol{R}_\alpha$ results, as well, from the concatenation of $\boldsymbol{R}^{lh}_{\alpha_0} \cdot \boldsymbol{a}_0$ and $\boldsymbol{R}^{rh}_{\alpha_0}$:

$$\boldsymbol{J}_{\alpha_0,\boldsymbol{a}_0} = (\underbrace{\boldsymbol{R}^{lh}_{\alpha_0} \cdot \boldsymbol{a}_0}_{\boldsymbol{J}_{\alpha_0}}, \underbrace{\boldsymbol{R}^{rh}_{\alpha_0}}_{\boldsymbol{J}_{\boldsymbol{a}_0}}). \tag{8.8}$$

As an alternative to the suggested approach, the small-angle approximation [8] could be used, too. However, by utilizing a variable linearization point α_0 close to the rotation angle, instead of setting the linearization point α_0 to zero as presumed by the small-angle approximation, a more accurate linearization is achieved.

8.3.6 ADG for Circular Roundabouts

This section presents an ADG scheme for circular roundabouts. Figure 8.1 shows an overlay of the LAs over a geographically represented circular roundabout, and Figure 8.2 depicts the scheme of the roundabout-ADG. The ADG scheme has one dynamic block which is used for the representation of the junctions. In the following, primary attributes are highlighted in yellow ●, secondary header attributes in green ●, secondary non-header attributes in red ●, and AAs in gray ●.

The primary attribute of the scheme for circular roundabout is the center_vector. It is the LA of an ADF which specifies the pose of an AA relatively to the AA obs_root_frame. The obs_root_frame is located on the vehicle. The center_vector

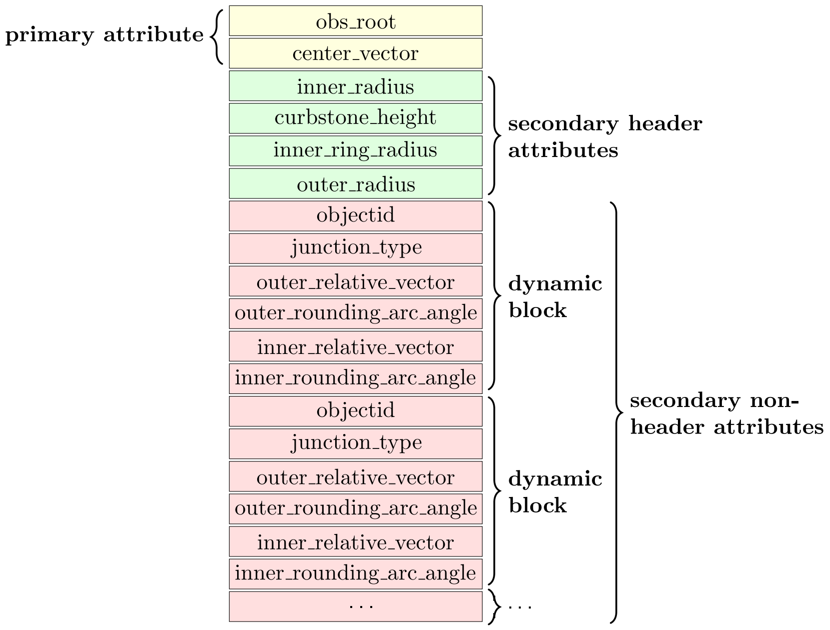


Figure 8.3: State vector representation for a parametric roundabout description. Primary attributes are highlighted in yellow ●, secondary header attributes in green ●, and secondary non-header attributes in red ● (representing a junction each).

is the position vector of the origin of the center_frame. The basis vectors of the center_frame have the same orientation as the basis vectors of the obs_root_frame. This means that the frame rotation of the related ADF is the identity. For that reason the frame rotation part of the LA of the ADF defining the pose of the AA obs_root_frame relatively to the AA center_frame is omitted, i. e. the corresponding LA is just the position vector center_vector.

The AAs inner_radius_frame, inner_ring_radius_frame, and outer_radius_frame are basically specified relatively to the center_frame by three position vectors as AAs. The orientations of the basis vectors of the frames are chosen being those of the center_frame, so that the frame rotation parts are also omitted here. The vectors are represented in polar coordinates. The lengths of the vectors given by the LAs are inner_radius, inner_ring_radius, and outer_radius. The orientation is chosen arbitrarily w. r. t. to the center frame, but constant. For that reason only the inner_radius, inner_ring_radius, and outer_radius occur as LAs of the ADF defining the poses of the AAs inner_radius_frame, inner_ring_radius_frame, and outer_radius_frame relatively to the AA center_frame. They have the role of secondary header attributes.

The AA curbstone_height_frame is located above the AA inner_radius_frame. The orientation of its position vector is in the z-direction of the inner_radius_frame, and its length is specified by the LA curbstone_height which is a further secondary header attribute. Again the orientations of both frames are chosen to be the same, so that the frame rotation part is omitted.

The main items of a junction are the inner and the outer rounding arcs. An arc can be defined by the center point and the radius of the circle of which it is a part, and two angles which define the section of the circle forming the arc. In the ADG the circles are defined w. r. t. to an inner_relative_frame and an outer_relative_frame as AAs, respectively. The poses of the frames are defined relatively to the AA center_frame. The locations of their origins are defined by the position vectors inner_relative_vector and outer_relative_vector as the position part of the LA. The basis vectors of the inner_relative_frame and outer_relative_frame are oriented so that the x-basis vectors are collinear with the position vectors inner_relative_vector and outer_relative_vector, respectively.

The four parameters of an arc are the circle center, circle radius, and two angles. By the constraints of roundabout construction according to Haller [50], outlined in section 8.3.1, the four parameters of an arc, circle center, circle radius, and two angles, cannot be chosen independently of each other. For that reason, only the center point and one angle are used as parameters. Then values of the radius and the second angle, e. g. the opening angle of the arc are uniquely determined by the tangential constraints of the construction. The angle used is the inner_rounding_arc_angle resp. outer_rounding_arc_angle which specifies the end point of the section of the junction road considered. This end point might be used as the origin of an AA inner_rounding_arc_angle_frame resp. outer_rounding_arc_angle_frame, with a position vector having the LA inner_rounding_arc_angle resp. outer_rounding_arc_angle in polar representation. However, the length of this position vector is the radius of the arc-defining circle which is not directly available. As it does not affect the optimization, the length of the position vector of the origin of the inner_rounding_arc_angle_frame resp. outer_rounding_arc_angle_frame is chosen to 1. Furthermore, the orientation of its basis vectors is chosen to be the same as of the inner_relative_frame resp. outer_relative_frame so that the rotation of the associated ADF is the identity.

To exemplify the compounding of complex roundabout LAs, both the value and covariance for a junction's inner_rounding_arc_angle are subsequently derived by applying the compounding rules (section 3.4). The dependency chain of the inner_rounding_arc_angle - attribute according to the roundabout ADG (figure 8.2) is: center_vector, inner_relative_vector, inner_rounding_arc_angle. Each of those attributes specify an ADF defining the pose of an output-AA relatively to an input-AA. An ADF consists of a position vector for the origin and a rotation for the basis vectors, cf. table 8.1.

LA	LA$^{\text{pos}}$	LA$^{\text{rot}}$	Input-AA	Output-AA
center_vector	center_vector	Identity	obs_root_frame	center_frame
inner_relative_vector	inner_relative_vector	By the angle of inner_relative_vector	center_frame	inner_relative_frame
inner_rounding_arc_angle	Unit-length vector with angle inner_rounding_arc_angle	Identity	inner_relative_frame	inner_rounding_arc_angle_frame

Table 8.1: A tabular overview of the involved translations, rotations, input-AAs, and resulting output-AAs for the compounding chain of the inner_rounding_arc_angle-attribute: center_vector ⊕ inner_relative_vector ⊕ inner_rounding_arc_angle.

The compounded ADFs have the same form. For example, the compounded ADF

$$\boxed{\text{center\_vector}} \oplus \boxed{\text{inner\_relative\_vector}} \oplus \boxed{\text{inner\_rounding\_arc\_angle}} \tag{8.9}$$

specifies the pose of the AA $\boxed{\text{inner\_rounding\_arc\_angle\_frame}}$ relatively to the AA $\boxed{\text{obs\_root\_frame}}$. The relative rotation angle of the compounded ADF is

$$\mathrm{LA}^{\mathrm{rot}} = \boxed{\text{inner\_relative\_angle}}, \tag{8.10}$$

i. e. the polar angle of the $\boxed{\text{inner\_relative\_vector}}$ relatively to the AA $\boxed{\text{center\_frame}}$. The position vector of the compounded ADF in Cartesian representation is

$$\begin{aligned} \mathrm{LA}^{\mathrm{pos}} \; = \; & \boxed{\text{center\_vector}} \; + \; \boxed{\text{inner\_relative\_vector}} \; + \\ & \boldsymbol{R}_{\boxed{\text{inner\_relative\_angle}}} \; \cdot \; \boxed{\text{inner\_rounding\_arc\_vector}}, \end{aligned} \tag{8.11}$$

with

$$\begin{aligned} \boxed{\text{inner\_relative\_vector}} &= \boxed{\text{inner\_relative\_length}} \cdot \begin{pmatrix} \cos\left(\boxed{\text{inner\_relative\_angle}}\right) \\ \sin\left(\boxed{\text{inner\_relative\_angle}}\right) \end{pmatrix} \\ &= \boldsymbol{R}_{\boxed{\text{inner\_relative\_angle}}} \cdot \begin{pmatrix} \boxed{\text{inner\_relative\_length}} \\ 0 \end{pmatrix}, \\ \boxed{\text{inner\_rounding\_arc\_vector}} &= \boxed{\text{inner\_rounding\_arc\_length}} \cdot \begin{pmatrix} \cos\left(\boxed{\text{inner\_rounding\_arc\_angle}}\right) \\ \sin\left(\boxed{\text{inner\_rounding\_arc\_angle}}\right) \end{pmatrix} \\ &= \boldsymbol{R}_{\boxed{\text{inner\_rounding\_arc\_angle}}} \cdot \begin{pmatrix} \boxed{\text{inner\_rounding\_arc\_length}} \\ 0 \end{pmatrix}. \end{aligned}$$

The corresponding (co)variances, cf. sections 3.4 and 8.3.5, are

$$\begin{aligned} &\mathrm{Var}\left(\mathrm{LA}^{\mathrm{rot}}\right) = \mathrm{Var}\left(\boxed{\text{inner\_relative\_angle}}\right), \\ &\mathrm{Cov}\left(\mathrm{LA}^{\mathrm{pos}}\right) \;\approx\; \mathrm{Cov}\left(\boxed{\text{center\_vector}}\right) \; + \\ &\boldsymbol{J}_{\boxed{\text{inner\_relative\_angle}}_0} \cdot \mathrm{Var}\left(\boxed{\text{inner\_relative\_angle}}\right) \cdot \boldsymbol{J}^T_{\boxed{\text{inner\_relative\_angle}}_0} \; + \\ &\boldsymbol{J}_{\boxed{\text{inner\_relative\_length}}_0} \cdot \mathrm{Var}\left(\boxed{\text{inner\_relative\_length}}\right) \cdot \boldsymbol{J}^T_{\boxed{\text{inner\_relative\_length}}_0} \; + \\ &\boldsymbol{J}_{\boxed{\text{inner\_relative\_angle}}_0} \cdot \mathrm{Var}(\boxed{\text{inner\_relative\_angle}}) \cdot \boldsymbol{J}^T_{\boxed{\text{inner\_relative\_angle}}_0} \; + \\ &\boldsymbol{J}_{\boxed{\text{inner\_rounding\_arc\_angle}}_0} \cdot \mathrm{Var}(\boxed{\text{inner\_rounding\_arc\_angle}}) \cdot \boldsymbol{J}^T_{\boxed{\text{inner\_rounding\_arc\_angle}}_0} \; + \\ &\boldsymbol{J}_{\boxed{\text{inner\_rounding\_arc\_length}}_0} \cdot \mathrm{Var}(\boxed{\text{inner\_rounding\_arc\_length}}) \cdot \boldsymbol{J}^T_{\boxed{\text{inner\_rounding\_arc\_length}}_0}. \end{aligned} \tag{8.12}$$

8.4 Approach to Complex Landmark Mapping

As for point-shaped landmarks, complex landmark mapping is updating a map which is represented by a submapping data structure, cf. section 4.2, by means of landmark estimates derived from the landmark observations provided by a fleet of vehicles, and taking into account the temporal decay of the acquired sensor readings, cf. section 4.3. The procedure is basically the same as for point-shaped landmarks, cf. chapter 5.4:

Solution 8.1: Complex Landmark Mapping

Setup of map updating:

1. Setup of mapping data structures.
2. Setup of temporal decay models.

Procedure of map updating:

1. Clustering: Assignment of landmark observations to clusters corresponding to distinct landmarks.
2. Landmark estimation: Estimation of landmark locations and, optionally, vehicle locations by means of sensor data fusion.

The clustering of complex landmark observations is performed on vectors representing the observations. The vector representation of a complex landmark extends the geographic information which corresponds to the point-shaped landmark, by entries of the secondary attributes. The clustering consists of two parts. The first part concerns the primary attribute and it is performed analogously to point-shaped landmarks, cf. chapter 6. The second part concerns the clustering of observations of corresponding dynamic blocks, for which the methods of chapter 6 are applied as well. The secondary header attributes do not need a separate clustering since they are immediately coupled with the primary attribute.

Sensor data fusion is performed by SLAM. Both variants employed for point-shaped landmarks in chapter 7, EKF-based Online-SLAM and BA-based Full-SLAM, will be extended to complex landmarks.

8.5 Fusion of Complex Landmarks by EKF-based Online-SLAM

In the following an extension of fusion by EKF-based Online-SLAM (section 7.3) to complex landmarks is presented. As EKF-based Online-SLAM has already been described in detail in section 7.3, this section focuses on the modifications required for complex landmark observations.

EKF-based Online-SLAM requires the definitions of a state representation, a motion prediction function, and an observation prediction function, cf. chapter 7.3.3. Given a sequence of landmark observations and motion increments, it generates a sequence of state vectors and state covariance matrices as output. The next state of the sequence is obtained from the previous one, depending on the current motion increment and the corresponding landmark observations.

8.5.1 Modeling the State

Equation 8.13 shows the extended state vector and the corresponding state covariance matrix. The state vector is comprised of the most current, estimated vehicle position $\boldsymbol{x}^{\mathrm{v}}$ and the estimated positions of all landmarks, represented by the primary LA $\boldsymbol{l}^{\mathrm{prim.}}$, as for point-shaped landmarks. Additionally, vectors of secondary header and non-header LA are part of the state vector. Figure 8.3 shows such a vectorized representation for the example of a circular roundabout landmark.

Analogously, the state covariance matrix contains the uncertainties of the vehicle/landmark positions and their cross-correlations, and, beyond those of point-shaped landmarks, the uncertainties and cross-correlations of the primary and secondary (non-)header LAs of complex landmarks. Cross-correlations between the vehicle and a complex landmark or pairs of complex landmarks are always expressed according to the primary LAs.

Primary LAs of complex landmarks are stored within the state vector $\hat{\boldsymbol{x}}_{k|k}$ according to their absolute position. In contrast, secondary header and non-header LAs are stored relatively to the primary ones or are independent.

8.5.2 Observation Prediction for Landmark Attributes

The observation prediction function predicts the observation of a landmark based on a given position and former landmark estimates (chapter 7). For complex landmarks, it consists of two subfunctions: a subfunction for the prediction of primary LAs, the so-called *observation prediction function*, and a subfunction for the prediction of secondary LAs, the so-called *attribute observation prediction function*. By this separation, filtering and smoothing algorithms for the ADG-based fusion of complex landmarks can be as well utilized for the fusion of point-shaped landmarks because the observation prediction function is equal for both.

The observation prediction function $\boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j^{\mathrm{prim.}})$ has the same form as for point-shaped landmarks and is applied to the primary LAs $\boldsymbol{l}_j^{\mathrm{prim.}}$ and the estimated vehicle position $\boldsymbol{x}_l^{\mathrm{v}}$.

$$\hat{\boldsymbol{x}}_{k|k} = \Bigg[\boldsymbol{x}^{\mathrm{v}} \quad \overbrace{\begin{bmatrix} \boldsymbol{l}_0^{\text{prim.}} & \boldsymbol{l}_0^{\text{sec. head.}} & \boldsymbol{l}_0^{\text{sec.}} \end{bmatrix}}^{\boldsymbol{l}_0} \quad \cdots \quad \overbrace{\begin{bmatrix} \boldsymbol{l}_n^{\text{prim.}} & \boldsymbol{l}_n^{\text{sec. head.}} & \boldsymbol{l}_n^{\text{sec.}} \end{bmatrix}}^{\boldsymbol{l}_n} \Bigg]^T,$$

$$\boldsymbol{P}_{k|k} = \begin{array}{c} \\ \boldsymbol{x}^{\mathrm{v}} \\ \\ \boldsymbol{l}_0 \\ \\ \\ \boldsymbol{l}_n \end{array} \left[\begin{array}{cccc} \boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_0 & & \boldsymbol{l}_n \\ \begin{matrix} \boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}} \end{matrix} & \begin{matrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{x}^{\mathrm{v}},\boldsymbol{l}_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{x}^{\mathrm{v}},\boldsymbol{l}_0} \end{matrix} & \cdots & \begin{matrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{x}^{\mathrm{v}},\boldsymbol{l}_n} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{x}^{\mathrm{v}},\boldsymbol{l}_n} \end{matrix} \\ \begin{matrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_0,\boldsymbol{x}^{\mathrm{v}}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_0,\boldsymbol{x}^{\mathrm{v}}} \end{matrix} & \underbrace{\begin{bmatrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_0} & \boldsymbol{\Sigma}^{\text{sec. head., prim.}}_{\boldsymbol{l}_0} & \boldsymbol{\Sigma}^{\text{sec., prim.}}_{\boldsymbol{l}_0} \\ \boldsymbol{\Sigma}^{\text{sec. head., prim.}}_{\boldsymbol{l}_0} & \boldsymbol{\Sigma}^{\text{sec. head.}}_{\boldsymbol{l}_0} & \boldsymbol{\Sigma}^{\text{sec., sec. head.}}_{\boldsymbol{l}_0} \\ \boldsymbol{\Sigma}^{\text{sec., prim.}}_{\boldsymbol{l}_0} & \boldsymbol{\Sigma}^{\text{sec., sec. head.}}_{\boldsymbol{l}_0} & \boldsymbol{\Sigma}^{\text{sec.}}_{\boldsymbol{l}_0} \end{bmatrix}}_{\boldsymbol{\Sigma}_{\boldsymbol{l}_0}} & \cdots & \begin{matrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_0,\boldsymbol{l}_n} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_0,\boldsymbol{l}_n} \end{matrix} \\ \vdots & \vdots & \ddots & \vdots \\ \begin{matrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_n,\boldsymbol{x}^{\mathrm{v}}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_n,\boldsymbol{x}^{\mathrm{v}}} \end{matrix} & \begin{matrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_n,\boldsymbol{l}_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_n,\boldsymbol{l}_0} \end{matrix} & \cdots & \underbrace{\begin{bmatrix} \boldsymbol{\Sigma}^{\text{prim.}}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}^{\text{sec. head., prim.}}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}^{\text{sec., prim.}}_{\boldsymbol{l}_n} \\ \boldsymbol{\Sigma}^{\text{sec. head., prim.}}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}^{\text{sec. head.}}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}^{\text{sec., sec. head.}}_{\boldsymbol{l}_n} \\ \boldsymbol{\Sigma}^{\text{sec., prim.}}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}^{\text{sec., sec. head.}}_{\boldsymbol{l}_n} & \boldsymbol{\Sigma}^{\text{sec.}}_{\boldsymbol{l}_n} \end{bmatrix}}_{\boldsymbol{\Sigma}_{\boldsymbol{l}_n}} \end{array} \right].$$

Equation 8.13: State vector $\hat{\boldsymbol{x}}_{k|k}$ and state covariance matrix $\boldsymbol{P}_{k|k}$ of EKF-based Online-SLAM extended to complex landmark observations. Primary LAs are highlighted in yellow ●, secondary header LAs in green ●, and secondary non-header LAs in red ●.

The attribute observation prediction function

$$\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j^{\mathrm{prim.}},\ \boldsymbol{l}_j^{\mathrm{sec.head/sec.},\,m},\ \text{🔗}\left(\boldsymbol{l}_j^{\mathrm{sec.head/sec.},\,m}\right)\right) \tag{8.14}$$

predicts the values of chained LAs according to given model functions and by utilizing compounding (section 3.4). $\boldsymbol{x}_l^{\mathrm{v}}$ is the estimated vehicle position, $\boldsymbol{l}_j^{\mathrm{prim.}}$ the estimated primary LA of the complex landmark, $\boldsymbol{l}_j^{\mathrm{sec.head/sec.},\,m}$ a secondary header or non-header LA, and $\text{🔗}\left(\boldsymbol{l}_j^{\mathrm{sec.head/sec.},\,m}\right)$ the dependency chain of the LA $\boldsymbol{l}_j^{\mathrm{sec.head/sec.},\,m}$.

8.5.3 First-Time Observation of a Complex Landmark

If a complex landmark $\boldsymbol{l}_{n+1}$ is observed for the first time, the state vector $\hat{\boldsymbol{x}}_{k|k}$ and the corresponding state covariance matrix $\boldsymbol{P}_{k|k}$ have to be extended in a similar manner as for point-shaped landmarks (section 7.3.6). The expansion of the state vector $\hat{\boldsymbol{x}}_{k|k}$ has to take care of the complex landmark's primary, secondary header, and secondary non-header LA:

$$\begin{aligned}
\hat{\boldsymbol{x}}_{k|k} &= \begin{bmatrix} \boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_0 & \cdots & \boldsymbol{l}_n & \boldsymbol{l}_{n+1} \end{bmatrix}^T \\
&= \begin{bmatrix} \boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_0 & \cdots & \boldsymbol{l}_n & \overbrace{\begin{bmatrix} \boldsymbol{l}_{n+1}^{\mathrm{prim.}} & \boldsymbol{l}_{n+1}^{\mathrm{sec.\ head.}} & \boldsymbol{l}_{n+1}^{\mathrm{sec.}} \end{bmatrix}}^{\boldsymbol{l}_{n+1}} \end{bmatrix}^T \\
&= \begin{bmatrix} \boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_0 & \cdots & \boldsymbol{l}_n & \overbrace{\begin{bmatrix} \boldsymbol{l}_{n+1}^{\mathrm{prim.}} & \underbrace{\begin{bmatrix} \boldsymbol{l}_{n+1}^{\mathrm{sec.\ head.},\,0} & \cdots & \boldsymbol{l}_{n+1}^{\mathrm{sec.\ head},\,m} \end{bmatrix}}_{\boldsymbol{l}_{n+1}^{\mathrm{sec.\ head}}} & \underbrace{\begin{bmatrix} \boldsymbol{l}_{n+1}^{\mathrm{sec.},\,0} & \cdots & \boldsymbol{l}_{n+1}^{\mathrm{sec.},\,m} \end{bmatrix}}_{\boldsymbol{l}_{n+1}^{\mathrm{sec.}}} \end{bmatrix}}^{\boldsymbol{l}_{n+1}} \end{bmatrix}^T .
\end{aligned} \tag{8.15}$$

Analogously, the state covariance matrix $\boldsymbol{P}_{k|k}$ has to be extended for every complex landmark observation that is perceived for the first time (equation 8.16). The initialization of the additional columns and rows can be achieved, similar as for point-shaped landmarks, either by a constant initialization or in a heuristic manner by replicating the cross-correlations of another landmark [27, 119], such as by the most converged or closest one. This approach is known to usually speed up the convergence of landmark observations that are perceived for the first time.

$$
\boldsymbol{P}_{k|k} = \left[\begin{array}{cccccc}
 & \boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_0 & & \boldsymbol{l}_n & \boldsymbol{l}_{n+1} \\
\boldsymbol{x}^{\mathrm{v}} & \boldsymbol{\Sigma}_{v} & \boldsymbol{\Sigma}_{v,l_0} & \cdots & \boldsymbol{\Sigma}_{v,l_n} & \begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{x}^{\mathrm{v}},\boldsymbol{l}_{n+1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{x}^{\mathrm{v}},\boldsymbol{l}_{n+1}} \end{array} \\
\boldsymbol{l}_0 & \boldsymbol{\Sigma}_{l_0,v} & & \cdots & \boldsymbol{\Sigma}_{l_0,l_n} & \begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_0,\boldsymbol{l}_{n+1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_0,\boldsymbol{l}_{n+1}} \end{array} \\
 & \vdots & \vdots & \ddots & \vdots & \vdots \\
\boldsymbol{l}_n & \boldsymbol{\Sigma}_{l_n,v} & \boldsymbol{\Sigma}_{l_n,l_0} & \cdots & \boldsymbol{\Sigma}_{l_n} & \begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_n,\boldsymbol{l}_{n+1}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_n,\boldsymbol{l}_{n+1}} \end{array} \\
\boldsymbol{l}_{n+1} & \begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1},\boldsymbol{x}^{\mathrm{v}}} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1},\boldsymbol{x}^{\mathrm{v}}} \end{array} & \begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1},\boldsymbol{l}_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1},\boldsymbol{l}_0} \end{array} & \cdots & \begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1},\boldsymbol{l}_n} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1},\boldsymbol{l}_n} \end{array} & \underbrace{\left[\begin{array}{ccc} \boldsymbol{\Sigma}^{\mathrm{prim.}}_{\boldsymbol{l}_{n+1}} & \boldsymbol{\Sigma}^{\mathrm{sec.\ head.,\ prim.}}_{\boldsymbol{l}_{n+1}} & \boldsymbol{\Sigma}^{\mathrm{sec.,\ prim.}}_{\boldsymbol{l}_{n+1}} \\ \boldsymbol{\Sigma}^{\mathrm{sec.\ head.,\ prim.}}_{\boldsymbol{l}_{n+1}} & \boldsymbol{\Sigma}^{\mathrm{sec.\ head.}}_{\boldsymbol{l}_{n+1}} & \boldsymbol{\Sigma}^{\mathrm{sec.,\ sec.\ head.}}_{\boldsymbol{l}_{n+1}} \\ \boldsymbol{\Sigma}^{\mathrm{sec.,\ prim.}}_{\boldsymbol{l}_{n+1}} & \boldsymbol{\Sigma}^{\mathrm{sec.,\ sec.\ head.}}_{\boldsymbol{l}_{n+1}} & \boldsymbol{\Sigma}^{\mathrm{sec.}}_{\boldsymbol{l}_{n+1}} \end{array}\right]}_{\boldsymbol{\Sigma}_{\boldsymbol{l}_{n+1}}}
\end{array}\right]
$$

Equation 8.16: Extension of the state covariance matrix $\boldsymbol{P}_{k|k}$ when a complex landmark is observed for the first time. Primary LAs are highlighted in yellow ●, secondary header LAs in green ●, and secondary non-header LA in red ●.

8.5.4 Repeated Observation of a Complex Landmark

If a complex landmark $\boldsymbol{l}_i$ is again observed, the state vector $\hat{\boldsymbol{x}}_{k|k}$ and the state covariance matrix $\boldsymbol{P}_{k|k}$ updating by EKF-based Online-SLAM is performed using a stripped-down observation vector $\hat{\boldsymbol{z}}_k^i$ and observation noise matrix $\boldsymbol{R}_k^i$ (section 7.3.7):

$$\begin{aligned}
\hat{\boldsymbol{z}}_k^i &= \begin{bmatrix}\boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_i\end{bmatrix}^T \\
&= \begin{bmatrix}\boldsymbol{x}^{\mathrm{v}} & \overbrace{\begin{bmatrix}\boldsymbol{l}_i^{\text{prim.}} & \boldsymbol{l}_i^{\text{sec. head.}} & \boldsymbol{l}_i^{\text{sec.}}\end{bmatrix}}^{\boldsymbol{l}_i}\end{bmatrix}^T, \\
&= \begin{bmatrix}\boldsymbol{x}^{\mathrm{v}} & \overbrace{\begin{bmatrix}\underbrace{\begin{bmatrix}\boldsymbol{l}_i^{\text{prim.},0} & \cdots & \boldsymbol{l}_i^{\text{prim.},m}\end{bmatrix}}_{\boldsymbol{l}_i^{\text{prim.}}} & \boldsymbol{l}_i^{\text{sec. head.}} & \boldsymbol{l}_i^{\text{sec.}}\end{bmatrix}}^{\boldsymbol{l}_i}\end{bmatrix}^T,
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{R}_k^i &= \begin{array}{cc} & \begin{array}{cc}\boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_i\end{array} \\ \begin{array}{c}\boldsymbol{x}^{\mathrm{v}} \\ \boldsymbol{l}_i\end{array} & \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}} & 0 \\ 0 & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}\end{bmatrix}\end{array} \\
&= \begin{array}{cc} & \begin{array}{cc}\boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_i\end{array} \\ \begin{array}{c}\boldsymbol{x}^{\mathrm{v}} \\ \boldsymbol{l}_i\end{array} & \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}} & 0 \\ 0 & \underbrace{\begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{prim.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{prim., sec. head.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., prim.}} \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head., prim.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., sec. head.}} \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., prim.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., sec. head.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec.}}\end{bmatrix}}_{\boldsymbol{\Sigma}_{\boldsymbol{l}_i}}\end{bmatrix}\end{array} \\
&= \begin{array}{cc} & \begin{array}{cc}\boldsymbol{x}^{\mathrm{v}} & \boldsymbol{l}_i\end{array} \\ \begin{array}{c}\boldsymbol{x}^{\mathrm{v}} \\ \boldsymbol{l}_i\end{array} & \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{x}^{\mathrm{v}}} & 0 \\ 0 & \underbrace{\begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{prim.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{prim., sec. head.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., prim.}} \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head., prim.}} & \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head.}} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head.}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head.}} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec. head.}}\end{bmatrix} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., sec. head.}} \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., prim.}} & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec., sec. head.}} & \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec.}} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec.}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec.}} & \cdots & \boldsymbol{\Sigma}_{\boldsymbol{l}_i}^{\text{sec.}}\end{bmatrix}\end{bmatrix}}_{\boldsymbol{\Sigma}_{\boldsymbol{l}_i}}\end{bmatrix}\end{array}.
\end{aligned} \tag{8.17}$$

The stripped-down observation vector $\hat{\boldsymbol{z}}_k^i$ and the stripped-down observation noise matrix $\boldsymbol{R}_k^i$ both incorporate secondary header and non-header LAs if they have been observed by the vehicle. Partial landmark observations, which do not include all secondary non-header attributes, are typical for complex landmark observations covering extensive areas and are caused by the sensor's limited field of view or by occlusion. Hence, the landmark $\boldsymbol{l}_j$ selective observation model $\boldsymbol{h}_{\boldsymbol{l}_i}(\boldsymbol{x}_k)$ and its

Jacobian $\boldsymbol{H}_{l_i}(\boldsymbol{x}_k)$ have to additionally tackle these peculiarities of complex landmarks in comparison to point-shaped ones. Afterwards, the updated state vector $\hat{\boldsymbol{x}}_{k|k}$ and state covariance matrix $\boldsymbol{P}_{k|k}$ are determined in the same manner for complex landmarks as for point-shaped ones (section 7.3.3).

8.6 Fusion via an Extended Formulation of BA-based Full-SLAM

In the following, BA-based Full-SLAM for point-shaped landmarks (section 7.4) is extended to complex landmarks represented by an ADG. At first, the required extension of its LS[13] formulation is elaborated (section 8.6.1). Subsequently, the linearization of the LS formulation (section 8.6.2) and its transformation into a matrix notation (section 8.6.3), so that it can be efficiently solved by factorization, is worked out.

8.6.1 Extended Least Squares Formulation

For complex landmarks the LS formulation of BA-based Full-SLAM of section 7.4 is extended as follows:

$$\begin{aligned}\widehat{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}, \boldsymbol{A})} = \underset{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}, \boldsymbol{A})}{\arg\min} \Bigg\{ & \sum_{l=1}^{p} \Bigg\| \underbrace{\Bigg(\underbrace{\boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}}\right)}_{\text{mot. prediction}} - \boldsymbol{x}_l^{\mathrm{v}} \Bigg)}_{\text{mot. pred. deviation}} \Bigg\|_{\boldsymbol{\Sigma}_m}^2 + \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j, \boldsymbol{z}_i) \\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \Bigg\| \underbrace{\Bigg(\underbrace{\boldsymbol{h}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j\right)}_{\text{obs. prediction}} - \boldsymbol{z}_i \Bigg)}_{\text{obs. pred. deviation}} \Bigg\|_{\boldsymbol{\Sigma}_o}^2 + \\ & \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j, \boldsymbol{z}_i) \\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \sum_{\substack{(\boldsymbol{a}_j^m, \boldsymbol{z}_i^{\mathrm{a},m}) \\ \in \mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_i)}} \Bigg\| \Bigg(\underbrace{\underbrace{\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)}_{\text{attr. obs. prediction}} - \boldsymbol{z}_i^{\mathrm{a},m}}_{\text{attr. pred. deviation}} \Bigg) \Bigg\|_{\boldsymbol{\Sigma}_{\boldsymbol{a}}}^2 \Bigg\} . \end{aligned} \tag{8.18}$$

The first two terms correspond to those of the original formulation for point-shaped landmarks. $\boldsymbol{X}^{\mathrm{v}} = (\boldsymbol{x}_0^{\mathrm{v}}, \ldots)$ denotes again the estimated, full vehicle trajectories. $\boldsymbol{L} = (\boldsymbol{l}_0, \ldots)$ is the set of all estimated primary LAs which represent the reduction of the complex landmarks to point-shaped ones. $\boldsymbol{U}^{\mathrm{v}} = (\boldsymbol{u}_0^{\mathrm{v}}, \ldots)$ represents the estimated vehicle motion increments and $\boldsymbol{Z} = (\boldsymbol{z}_0, \ldots)$ all observations of primary LAs.

The secondary LAs are considered in the additional third term. $\boldsymbol{A} = \left(\boldsymbol{a}_j^0, \ldots\right)$ denotes the set of all estimated secondary LAs, and $\boldsymbol{Z}_{\boldsymbol{A}} = \left(\boldsymbol{z}_i^{\mathrm{a},0}, \ldots\right)$ all observations of secondary LAs. The third term is structured quite similarly to the second one. It quantifies the deviations between the predictions $\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)$ (sec-

[13] Least Squares (LS)

$$
\begin{aligned}
\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right) - \boldsymbol{z}_i^{\mathrm{a},m} \;\approx\;& \underbrace{\Big[\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right) + \boldsymbol{K}_i^{l,j,m}\cdot\left(\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}\right) + \boldsymbol{L}_i^{l,j,m}\cdot\left(\boldsymbol{l}_j - \boldsymbol{l}_j^0\right) + \boldsymbol{M}_i^{l,j,m}\cdot\left(\boldsymbol{a}_j^m - \boldsymbol{a}_j^{m,0}\right)}_{T_1\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)} \\
&\underbrace{+\ \boldsymbol{N}_i^{l,j,m}\cdot\left(\text{🔗}\left(\boldsymbol{a}_j^m\right) - \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right)\Big]}_{T_1\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)} - \underbrace{\Big[\boldsymbol{z}_i^{\mathrm{a},m,0} + \left(\boldsymbol{z}_i^{\mathrm{a},m} - \boldsymbol{z}_i^{\mathrm{a},m,0}\right)\Big]}_{T_1\boldsymbol{z}_i^{\mathrm{a},m}} \\[1em]
&= \boldsymbol{K}_i^{l,j,m}\cdot\left(\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}\right) + \boldsymbol{L}_i^{l,j,m}\cdot\left(\boldsymbol{l}_j - \boldsymbol{l}_j^0\right) + \boldsymbol{M}_i^{l,j,m}\cdot\left(\boldsymbol{a}_j^m - \boldsymbol{a}_j^{m,0}\right) \\
&\quad +\ \boldsymbol{N}_i^{l,j,m}\cdot\left(\text{🔗}\left(\boldsymbol{a}_j^m\right) - \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right) + \boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right) - \boldsymbol{z}_i^{\mathrm{a},m}
\end{aligned}
$$

with

$$
\boldsymbol{K}_i^{l,j,m} = \left.\frac{\partial\,\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)}{\partial\,\boldsymbol{x}_l^{\mathrm{v}}}\right|_{\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right)}, \qquad
\boldsymbol{M}_i^{l,j,m} = \left.\frac{\partial\,\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)}{\partial\,\boldsymbol{a}_j^m}\right|_{\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right)},
$$

$$
\boldsymbol{L}_i^{l,j,m} = \left.\frac{\partial\,\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)}{\partial\,\boldsymbol{l}_j}\right|_{\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right)}, \qquad
\boldsymbol{N}_i^{l,j,m} = \left.\frac{\partial\,\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j,\ \boldsymbol{a}_j^m,\ \text{🔗}\left(\boldsymbol{a}_j^m\right)\right)}{\partial\,\text{🔗}\left(\boldsymbol{a}_j^m\right)}\right|_{\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right)}.
$$

Equation 8.19: Linearization of the additional term of the extended LS formulation of BA-based Full-SLAM, which is intended for the incorporation of secondary header and non-header attributes.

$$
\begin{aligned}
\left(\widehat{\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}, \boldsymbol{A}}\right) = \underset{(\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}, \boldsymbol{A})}{\arg\min} \Bigg\{ & \sum_{l=1}^{p} \left\| \boldsymbol{\Sigma}_m^{-1/2} \cdot \boldsymbol{F}_l^{l-1} \cdot (\boldsymbol{x}_{l-1}^{\mathrm{v}} - \boldsymbol{x}_{l-1}^{\mathrm{v},0}) + \boldsymbol{\Sigma}_m^{-1/2} \cdot \boldsymbol{G}_l^{l} \cdot \boldsymbol{x}_l^{\mathrm{v}} + \boldsymbol{\Sigma}_m^{-1/2} \cdot \boldsymbol{f}\left(\boldsymbol{x}_{l-1}^{\mathrm{v},0},\ \boldsymbol{u}_l^{\mathrm{v}}\right) \right\|_2^2 + \\
& \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j, \boldsymbol{z}_i) \\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \left\| \boldsymbol{\Sigma}_o^{-1/2} \cdot \boldsymbol{H}_i^{lj} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^{\mathrm{v},0}) + \boldsymbol{\Sigma}_o^{-1/2} \cdot \boldsymbol{J}_i^{lj} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) + \boldsymbol{\Sigma}_o^{-1/2} \cdot \left(\boldsymbol{h}(\boldsymbol{x}_l^0,\ \boldsymbol{l}_j^0) - \boldsymbol{z}_i\right) \right\|_2^2 + \\
& \sum_{l=1}^{p} \sum_{\substack{(\boldsymbol{l}_j, \boldsymbol{z}_i) \\ \in \mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})}} \sum_{\substack{(\boldsymbol{a}_j^m, \boldsymbol{z}_i^{\mathrm{a},m}) \\ \in \mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_i)}} \Big\| \boldsymbol{\Sigma}_a^{-1/2} \cdot \boldsymbol{K}_i^{l,j,m} \cdot (\boldsymbol{x}_l^{\mathrm{v}} - \boldsymbol{x}_l^0) + \boldsymbol{\Sigma}_a^{-1/2} \cdot \boldsymbol{L}_i^{l,j,m} \cdot (\boldsymbol{l}_j - \boldsymbol{l}_j^0) + \boldsymbol{\Sigma}_a^{-1/2} \cdot \boldsymbol{M}_i^{l,j,m} \cdot (\boldsymbol{a}_j^m - \boldsymbol{a}_j^{m,0}) + \\
& \qquad \boldsymbol{\Sigma}_a^{-1/2} \cdot \boldsymbol{N}_i^{l,j,m} \cdot \left(\text{🔗}\left(\boldsymbol{a}_j^m\right) - \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right) + \boldsymbol{\Sigma}_a^{-1/2} \cdot \left(\boldsymbol{h}^{\mathrm{a}}\left(\boldsymbol{x}_l^{\mathrm{v},0},\ \boldsymbol{l}_j^0,\ \boldsymbol{a}_j^{m,0},\ \text{🔗}\left(\boldsymbol{a}_j^{m,0}\right)\right) - \boldsymbol{z}_i^{\mathrm{a},m,0}\right) \Big\|_2^2 \Bigg\}
\end{aligned}
$$

with

$$
\boldsymbol{F}_l^{l-1} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}_{l-1}^{\mathrm{v}},\ \boldsymbol{u}_l^{\mathrm{v}})}{\partial \boldsymbol{x}_{l-1}}\right|_{\boldsymbol{x}_{i-1}^{\mathrm{v},0}}, \quad \boldsymbol{H}_i^{lj} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j)}{\partial \boldsymbol{x}_l^{\mathrm{v}}}\right|_{(\boldsymbol{x}_l^{\mathrm{v},0}, \boldsymbol{l}_j^0)}, \quad \boldsymbol{J}_i^{lj} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_l^{\mathrm{v}},\ \boldsymbol{l}_j)}{\partial \boldsymbol{l}_j}\right|_{(\boldsymbol{x}_l^{\mathrm{v},0}, \boldsymbol{l}_j^0)}, \quad \boldsymbol{G}_l^{l} = -\mathbb{1} = \mathrm{diag}(-1, \dots, -1),
$$

Equation 8.20: Linearized extended LS formulation of BA-based Full-SLAM. For the definition of the matrices $\boldsymbol{K}$, $\boldsymbol{L}$, $\boldsymbol{M}$, and $\boldsymbol{N}$ the reader may refer to equation 8.19.

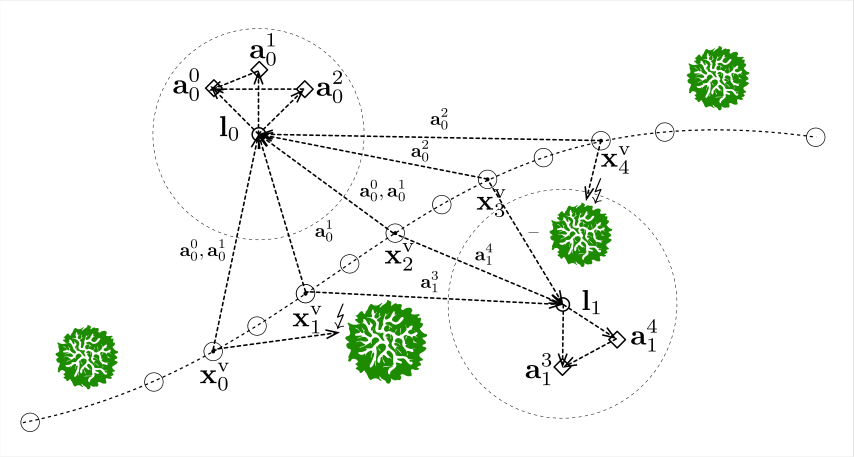

$\mathbf{a}_0^1$
$\mathbf{a}_0^0$
$\mathbf{a}_0^2$
$\mathbf{l}_0$
a_0^2
a_0^2
$\mathrm{a}_0^0, \mathrm{a}_0^1$
a_0^1
$\mathrm{a}_0^0, \mathrm{a}_0^1$
$\mathbf{x}_4^{\mathrm{v}}$
$\mathbf{x}_3^{\mathrm{v}}$
$\mathbf{x}_2^{\mathrm{v}}$
$\mathbf{x}_1^{\mathrm{v}}$
$\mathbf{x}_0^{\mathrm{v}}$
a_1^4
a_1^3
$\mathbf{l}_1$
$\mathbf{a}_1^4$
$\mathbf{a}_1^3$

Figure 8.4: Exemplary scenario which shows complex landmark observations acquired relatively to a single vehicle trajectory (equation 8.21). For visual clarity, the complex landmarks are purely exemplary. The observation of the complex landmark $\boldsymbol{l}_1$ is hindered two times due to the obstacles ●.

tion 8.5.2) and the observations $\boldsymbol{z}_i^{\mathrm{a},m}$ of secondary header and non-header LAs according to the ADG. The function $\mathrm{v}(\boldsymbol{x}_l^{\mathrm{v}})$, already used in the second term, determines all corresponding tuples $(\boldsymbol{l}_j, \boldsymbol{z}_i)$ of primary LA estimations $\boldsymbol{l}_j$ and observations $\boldsymbol{z}_i$ perceived from trajectory point $\boldsymbol{x}_i^{\mathrm{v}}$. Analogously, $\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_i)$ returns all corresponding tuples of secondary LAs $\boldsymbol{z}_i^{\mathrm{a},m}$ observed together with the primary LA $\boldsymbol{z}_i$ and their estimates $\boldsymbol{a}_j^m$.

The uncertainties of the predictions and measurements are expressed, as for point-shaped landmark data, by employing the Mahalanobis metric. They are incorporated in form of combined covariance matrices within the LS formulation: $\boldsymbol{\Sigma}_{\boldsymbol{m}}$ for the vehicle motion, $\boldsymbol{\Sigma}_{\boldsymbol{o}}$ for primary LAs, and $\boldsymbol{\Sigma}_{\boldsymbol{a}}$ for secondary LAs.

8.6.2 Linearized Extended Least Squares Formulation

The LS-based formulation of extended BA-based Full-SLAM can be linearized for efficient computational solution by factorization, analogously to the case of point-shaped landmarks (section 7.4.1). In equation 8.19 the linearization of the additional term is derived in detail. Subsequently, the complete, linearized extended formulation of BA-based Full-SLAM is provided in equation 8.20.

8.6.3 Linearized Extended Least Squares Matrix Formulation

The linearized LS-based formulation of extended BA-based Full-SLAM (equation 8.20) can be transformed into a corresponding matrix formulation. This way the linearized LS formulation can be efficiently solved by factorization. To illustrate the transformation, the example provided in section 7.4.4 for point-shaped landmarks is enriched to feature multiple complex landmark observations, see figure 8.4. The related definitions are

$$
\begin{aligned}
\boldsymbol{X}^{\mathrm{v}} &= \begin{pmatrix} \boldsymbol{x}_0^{\mathrm{v}} & \boldsymbol{x}_1^{\mathrm{v}} & \boldsymbol{x}_2^{\mathrm{v}} & \boldsymbol{x}_3^{\mathrm{v}} & \boldsymbol{x}_4^{\mathrm{v}} \end{pmatrix},\\
\boldsymbol{L} &= \begin{pmatrix} \boldsymbol{l}_0 & \boldsymbol{l}_1 \end{pmatrix},\\
\boldsymbol{A} &= \begin{pmatrix} \boldsymbol{a}_0^0 & \boldsymbol{a}_0^1 & \boldsymbol{a}_0^2 & \boldsymbol{a}_1^3 & \boldsymbol{a}_1^4 \end{pmatrix},\\
\boldsymbol{U}^{\mathrm{v}} &= \begin{pmatrix} \boldsymbol{u}_0^{\mathrm{v}} & \boldsymbol{u}_1^{\mathrm{v}} & \boldsymbol{u}_2^{\mathrm{v}} & \boldsymbol{u}_3^{\mathrm{v}} & \boldsymbol{u}_4^{\mathrm{v}} \end{pmatrix},\\
\boldsymbol{Z} &= \begin{pmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \boldsymbol{z}_2 & \boldsymbol{z}_3 & \boldsymbol{z}_4 & \boldsymbol{z}_5 & \boldsymbol{z}_6 & \boldsymbol{z}_7 \end{pmatrix},\\
\boldsymbol{Z}^{\mathrm{a}} &= \begin{pmatrix} \boldsymbol{z}_0^{\mathrm{a},0} & \boldsymbol{z}_0^{\mathrm{a},1} & \boldsymbol{z}_1^{\mathrm{a},1} & \boldsymbol{z}_2^{\mathrm{a},3} & \boldsymbol{z}_3^{\mathrm{a},0} & \boldsymbol{z}_3^{\mathrm{a},1} & \boldsymbol{z}_4^{\mathrm{a},4} & \boldsymbol{z}_5^{\mathrm{a},2} & \boldsymbol{z}_7^{\mathrm{a},2} \end{pmatrix},
\end{aligned}
\qquad
\begin{aligned}
\mathrm{v}(\boldsymbol{x}_0^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_0)\},\\
\mathrm{v}(\boldsymbol{x}_1^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_1), (\boldsymbol{l}_1, \boldsymbol{z}_2)\},\\
\mathrm{v}(\boldsymbol{x}_2^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_3), (\boldsymbol{l}_1, \boldsymbol{z}_4)\},\\
\mathrm{v}(\boldsymbol{x}_3^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_5), (\boldsymbol{l}_1, \boldsymbol{z}_6)\},\\
\mathrm{v}(\boldsymbol{x}_4^{\mathrm{v}}) &= \{(\boldsymbol{l}_0, \boldsymbol{z}_7)\},\\
\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_0) &= \left\{\left(\boldsymbol{a}_0^0, \boldsymbol{z}_0^{\mathrm{a},0}\right), \left(\boldsymbol{a}_0^1, \boldsymbol{z}_0^{\mathrm{a},1}\right)\right\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_1) &= \left\{\left(\boldsymbol{a}_0^1, \boldsymbol{z}_1^{\mathrm{a},1}\right)\right\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_2) &= \left\{\left(\boldsymbol{a}_1^3, \boldsymbol{z}_2^{\mathrm{a},3}\right)\right\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_3) &= \left\{\left(\boldsymbol{a}_0^0, \boldsymbol{z}_3^{\mathrm{a},0}\right), \left(\boldsymbol{a}_0^1, \boldsymbol{z}_3^{\mathrm{a},1}\right)\right\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_4) &= \left\{\left(\boldsymbol{a}_1^4, \boldsymbol{z}_4^{\mathrm{a},4}\right)\right\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_5) &= \left\{\left(\boldsymbol{a}_0^2, \boldsymbol{z}_5^{\mathrm{a},2}\right)\right\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_6) &= \{\},\\
\mathrm{v}^{\mathrm{a}}(\boldsymbol{z}_7) &= \left\{\left(\boldsymbol{a}_0^2, \boldsymbol{z}_7^{\mathrm{a},2}\right)\right\}.
\end{aligned}
\tag{8.21}
$$

For visual clarity, simplified complex landmarks are used which do not correspond to any real-world ones, and which do not have non-header attributes.

The resulting matrix formulation of the optimization problem is provided in equation 8.22. The Mahalanobis premultiplication factors are omitted because of visual clarity. However, they can be added in the same manner as for point-shaped landmarks (equation 7.62).

8.7 Evaluation

This chapter presents the results of an evaluation of fusion by BA-based Full-SLAM, solved via Sparse-QR. This method has been selected for evaluation because it has shown an outstanding balance between accuracy, robustness, and computational efficiency for point-shaped landmarks (chapter 7).

8.7.1 Description of the Evaluation Scenarios

The evaluation has been performed on two scenarios, a single complex landmark scenario and a hybrid landmarks scenario consisting of the same complex landmark and several point-shaped landmarks. Both are real-world scenarios and are located in Ingolstadt at the WGS84[14] coordinate (48.784906, 11.390313).

[14] World Geodetic System 1984 (WGS84)

$$
(\widehat{\boldsymbol{X}^{\mathrm{v}}, \boldsymbol{L}, \boldsymbol{A}}) = \underset{(\boldsymbol{X}^{\mathrm{v}},\boldsymbol{L},\boldsymbol{A})}{\arg\min} \left\| \underbrace{\begin{pmatrix}
\boldsymbol{G}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\boldsymbol{F}_1^0 & \boldsymbol{G}_1^1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \boldsymbol{F}_2^1 & \boldsymbol{G}_2^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \boldsymbol{F}_3^2 & \boldsymbol{G}_3^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \boldsymbol{F}_4^3 & \boldsymbol{G}_4^4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\boldsymbol{H}_0^{0,0} & 0 & 0 & 0 & 0 & \boldsymbol{J}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\boldsymbol{K}_0^{0,0,0} & 0 & 0 & 0 & 0 & \boldsymbol{L}_0^{0,0,0} & \boldsymbol{M}_0^{0,0,0} & 0 & 0 & 0 & 0 & 0 \\
\boldsymbol{K}_0^{0,0,1} & 0 & 0 & 0 & 0 & \boldsymbol{L}_0^{0,0,1} & \boldsymbol{N}_0^{0,0,0} & \boldsymbol{M}_0^{0,0,1} & 0 & 0 & 0 & 0 \\
\hline
0 & \boldsymbol{H}_1^{1,0} & 0 & 0 & 0 & \boldsymbol{J}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \boldsymbol{K}_1^{1,0,1} & 0 & 0 & 0 & \boldsymbol{L}_1^{1,0,1} & \boldsymbol{N}_1^{1,0,0} & \boldsymbol{M}_1^{1,0,1} & 0 & 0 & 0 & 0 \\
\hline
0 & \boldsymbol{H}_2^{1,1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{J}_1^1 & 0 & 0 \\
0 & \boldsymbol{K}_2^{1,1,3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{L}_2^{1,1,3} & \boldsymbol{M}_2^{1,1,3} & 0 \\
\hline
0 & 0 & \boldsymbol{H}_3^{2,0} & 0 & 0 & \boldsymbol{J}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \boldsymbol{K}_3^{2,0,0} & 0 & 0 & \boldsymbol{L}_3^{2,0,0} & \boldsymbol{M}_3^{2,0,0} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \boldsymbol{K}_3^{2,0,1} & 0 & 0 & \boldsymbol{L}_3^{2,0,1} & \boldsymbol{N}_3^{2,0,0} & \boldsymbol{M}_3^{2,0,1} & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & \boldsymbol{H}_4^{2,1} & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{J}_1^1 & 0 & 0 \\
0 & 0 & \boldsymbol{K}_4^{2,1,4} & 0 & 0 & 0 & 0 & 0 & 0 & \boldsymbol{L}_4^{2,1,4} & \boldsymbol{N}_4^{2,1,3} & \boldsymbol{M}_0^{2,1,4} \\
\hline
0 & 0 & 0 & \boldsymbol{H}_5^{3,0} & 0 & \boldsymbol{J}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \boldsymbol{K}_5^{3,0,2} & 0 & \boldsymbol{L}_5^{3,0,2} & \boldsymbol{N}_5^{3,0,0} & 0 & \boldsymbol{M}_5^{3,0,2} & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & \boldsymbol{H}_6^{3,1} & 0 & 0 & 0 & 0 & 0 & \boldsymbol{J}_1^1 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & \boldsymbol{H}_7^{4,0} & \boldsymbol{J}_0^0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \boldsymbol{K}_7^{4,0,2} & \boldsymbol{L}_7^{4,0,2} & \boldsymbol{N}_7^{4,0,0} & 0 & \boldsymbol{M}_7^{4,0,2} & 0 & 0 & 0
\end{pmatrix}}_{A} \cdot \underbrace{\begin{pmatrix}
\boldsymbol{x}_0^{\mathrm{v}} - \boldsymbol{x}_0^{\mathrm{v},0} \\
\boldsymbol{x}_1^{\mathrm{v}} - \boldsymbol{x}_1^{\mathrm{v},0} \\
\boldsymbol{x}_2^{\mathrm{v}} - \boldsymbol{x}_2^{\mathrm{v},0} \\
\boldsymbol{x}_3^{\mathrm{v}} - \boldsymbol{x}_3^{\mathrm{v},0} \\
\boldsymbol{x}_4^{\mathrm{v}} - \boldsymbol{x}_4^{\mathrm{v},0} \\
\boldsymbol{l}_0 - \boldsymbol{l}_0^0 \\
\boldsymbol{a}_0^0 - \boldsymbol{a}_0^{0,0} \\
\boldsymbol{a}_0^1 - \boldsymbol{a}_0^{1,0} \\
\boldsymbol{a}_0^2 - \boldsymbol{a}_0^{2,0} \\
\boldsymbol{l}_1 - \boldsymbol{l}_1^0 \\
\boldsymbol{a}_1^3 - \boldsymbol{a}_1^{3,0} \\
\boldsymbol{a}_1^4 - \boldsymbol{a}_1^{4,0}
\end{pmatrix}}_{\delta} - \underbrace{\begin{pmatrix}
\boldsymbol{f}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{u}_0^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{u}_1^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{u}_2^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{u}_3^{\mathrm{v}}) \\
\boldsymbol{f}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{u}_4^{\mathrm{v}}) \\
\hline
\boldsymbol{h}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_0} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{0,0}, \text{🔗}\,(\boldsymbol{a}_0^{0,0})) - \boldsymbol{z}_0^{\mathrm{a},0} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_0^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{1,0}, \text{🔗}\,(\boldsymbol{a}_0^{1,0})) - \boldsymbol{z}_0^{\mathrm{a},1} \\
\hline
\boldsymbol{h}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_1} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{1,0}, \text{🔗}\,(\boldsymbol{a}_0^{1,0})) - \boldsymbol{z}_1^{\mathrm{a},1} \\
\hline
\boldsymbol{h}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z_2} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_1^{\mathrm{v},0}, \boldsymbol{l}_1^0, \boldsymbol{a}_1^{3,0}, \text{🔗}\,(\boldsymbol{a}_1^{3,0})) - \boldsymbol{z}_2^{\mathrm{a},3} \\
\hline
\boldsymbol{h}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z_3} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{0,0}, \text{🔗}\,(\boldsymbol{a}_0^{0,0})) - \boldsymbol{z}_3^{\mathrm{a},0} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{1,0}, \text{🔗}\,(\boldsymbol{a}_0^{1,0})) - \boldsymbol{z}_3^{\mathrm{a},1} \\
\hline
\boldsymbol{h}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_1^0, \boldsymbol{a}_3^0) - \boldsymbol{z_4} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_2^{\mathrm{v},0}, \boldsymbol{l}_1^0, \boldsymbol{a}_1^{4,0}, \text{🔗}\,(\boldsymbol{a}_1^{4,0})) - \boldsymbol{z}_4^{\mathrm{a},4} \\
\hline
\boldsymbol{h}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_3^0) - \boldsymbol{z_5} \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{2,0}, \text{🔗}\,(\boldsymbol{a}_0^{2,0})) - \boldsymbol{z}_5^{\mathrm{a},2} \\
\hline
\boldsymbol{h}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_1^0) - \boldsymbol{z}_6 \\
\hline
\boldsymbol{h}(\boldsymbol{x}_4^{\mathrm{v},0}, \boldsymbol{l}_0^0) - \boldsymbol{z}_7 \\
\boldsymbol{h}^{\mathrm{a}}(\boldsymbol{x}_3^{\mathrm{v},0}, \boldsymbol{l}_0^0, \boldsymbol{a}_0^{2,0}, \text{🔗}\,(\boldsymbol{a}_0^{2,0})) - \boldsymbol{z}_7^{\mathrm{a},2}
\end{pmatrix}}_{b} \right\|_2^2 .
$$

Equation 8.22: Exemplary instantiation of the matrix formulation of linearized extended BA-based Full-SLAM according to the example of equation 8.19 and figure 8.4. Primary LAs are highlighted in yellow ●, and secondary header LAs in green ●.

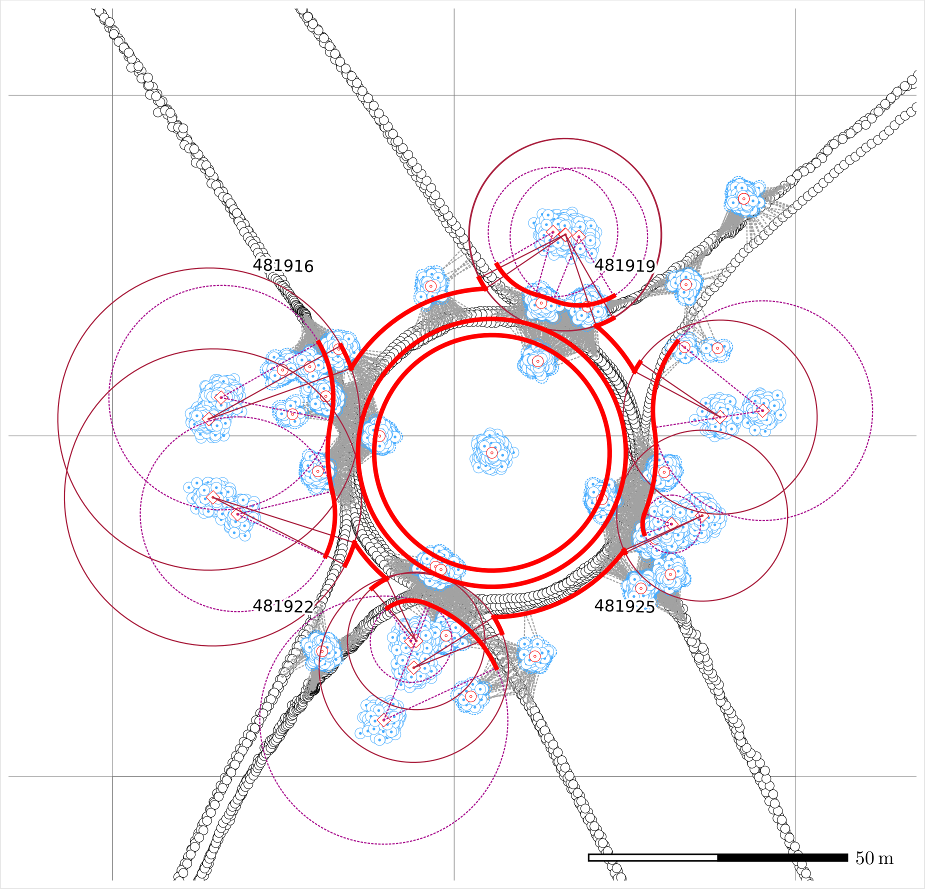


Figure 8.5: Complex landmark scenario consisting of a single roundabout landmark. Primary attributes are visualized by a red circle ○ and secondary attributes by a red diamond ◇. The point-shaped traffic sign landmarks are not used for fusion in this scenario; they are only visualized for comparability reasons.

Complex Landmark Scenario

The complex landmark scenario is visualized in figure 8.5. It consists of a single roundabout. The roundabout is observed in a partial manner 2587 times and from 12 distinct vehicle trajectories. The average standard deviations of the GNSS[15]-based vehicle localization and of the landmark observations, including primary and secondary (non-)header attributes are approximately 1.22 m and 2.8 m, respectively. Real and simulated sensor data are combined because the roundabout perception based on lidar data, especially concerning the outer roundabout ring, is at this point in time still work in progress. However, the simulated data should accurately reflect the utilized sensors and algorithms.

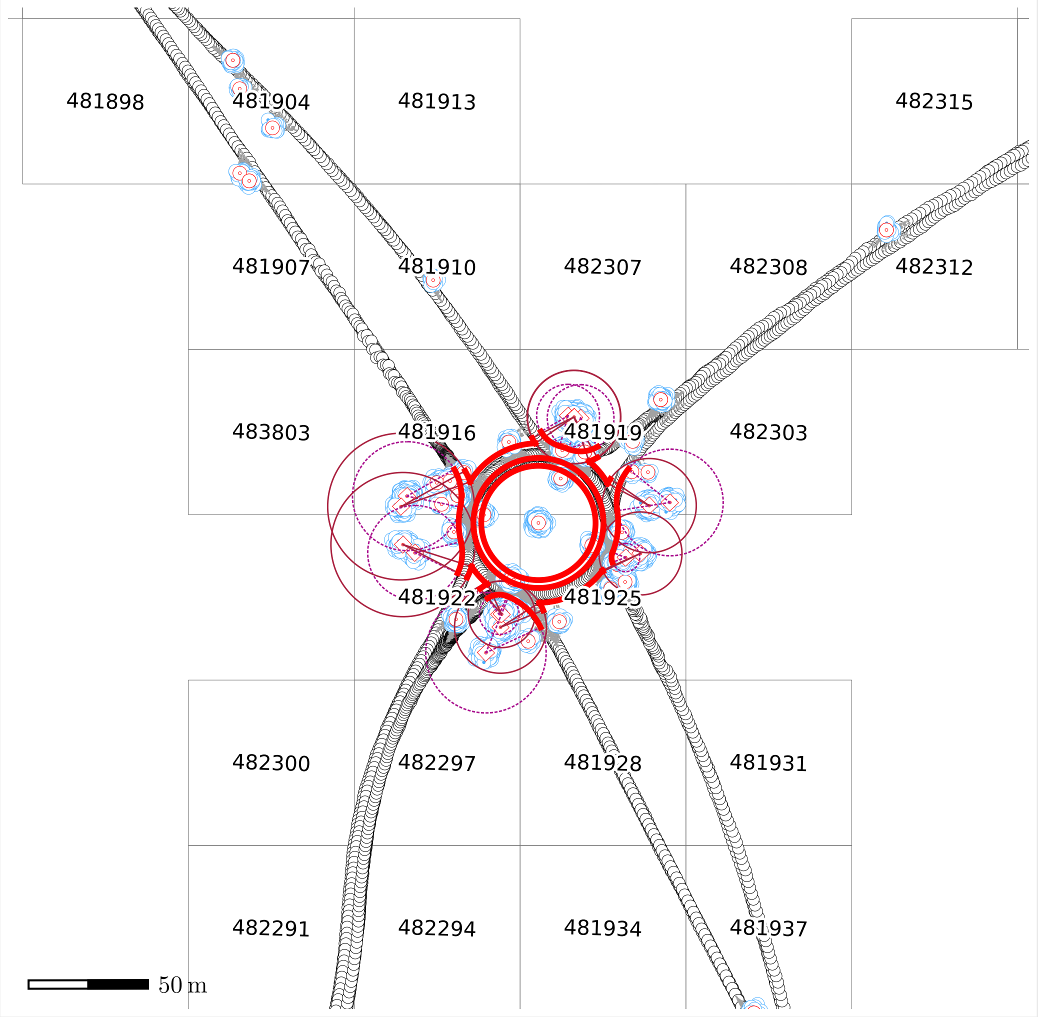


Figure 8.6: Hybrid landmarks scenario consisting of 42 point-shaped traffic signs and one roundabout. Primary attributes are visualized by a red circle ○ and secondary attributes by a red diamond ◇. The point-shaped and the complex landmarks are simultaneously used for fusion in this scenario.

Hybrid Landmarks Scenario

The hybrid landmarks scenario is visualized in figure 8.6. It consists of the roundabout landmark of the previous scenario and 42 additional point-shaped traffic sign landmarks which are observed 5517 times in total. The average standard deviations of the vehicle localization, the point-shaped traffic sign observations, and the complex roundabout landmark observations including primary and secondary (non-)header attributes, are approximately 1.24 m, 2.02 m, and 2.8 m (as before). A combination of actual and simulated sensor data is also used here. This scenario is intended for the evaluation of complex landmark fusion in a more challenging manner. Because

[15] Global Navigation Satellite System (GNSS)

of its larger-scale and hybrid nature it resembles a common urban scenario within the automotive context.

8.7.2 Fusion by BA-based Full-SLAM

The fusion of complex landmark observations by BA-based Full-SLAM solved via Sparse-QR is evaluated for both scenarios with respect to the accuracy achieved and the runtime required. For this purpose, 100, 500, 1000, 1500, and 2000 observations are used which have been randomly sampled without replacement for fifty times and which may be accompanied by secondary header and non-header attributes. The Sparse-QR-based approach is applied in the same manner as for point-shaped data (section 7.5.4), this means without relinearization. The fifty fusion results obtained are averaged to get robust estimates of the accuracies, runtimes, and the occurrence of outliers, and are visualized by box-whisker-plots, when sensible. Appendix A.5 provides the precise definition of this type of plot.

Complex Landmark Scenario

Figure 8.7 shows the fusion result of the complex landmark scenario for 100 and 500 observations. It can be noticed that the fusion result based on 100 partial observations does not incorporate all junctions and noticeably deviates from the ground truth. In contrast to that, the fusion based on 500 observations leads to an estimate that nearly exactly covers the ground truth and incorporates all present junctions.

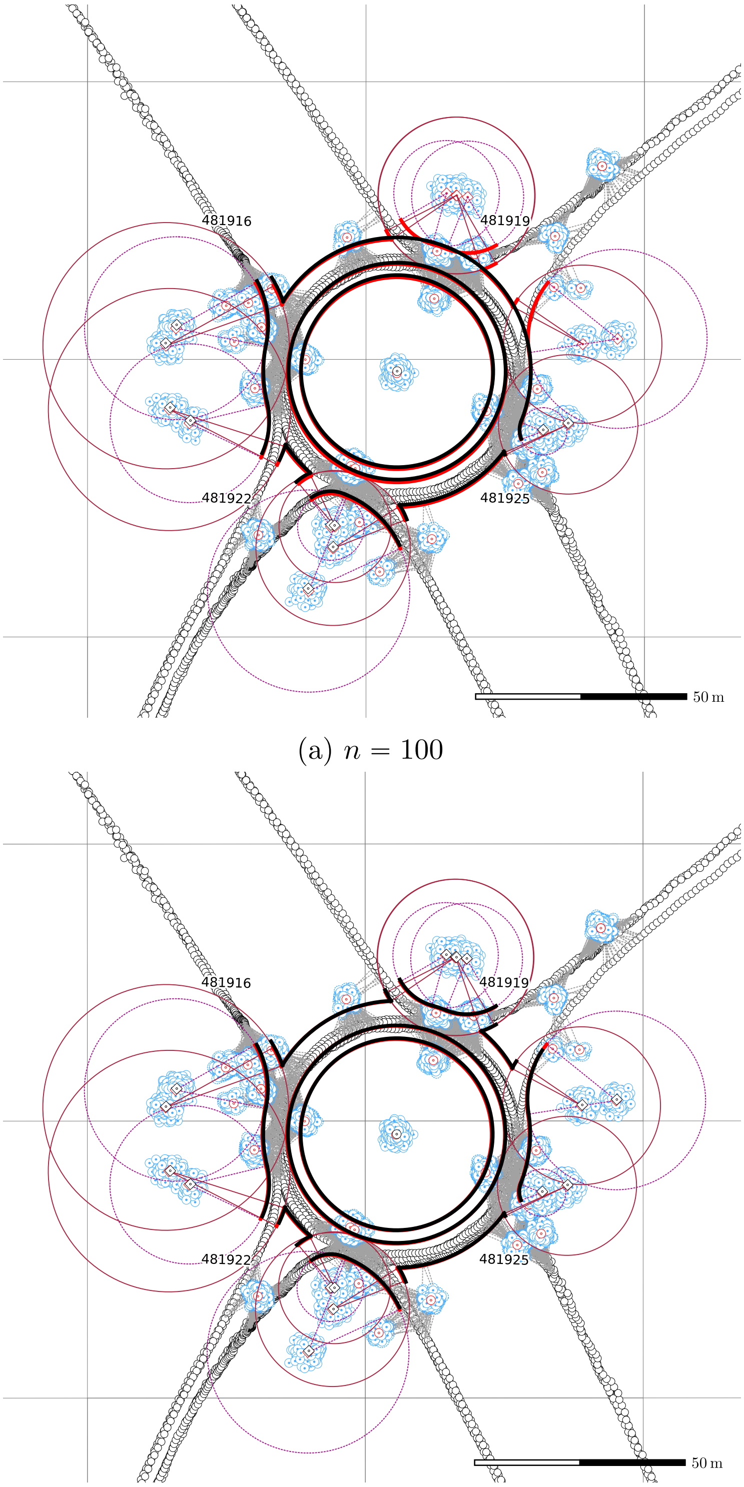
Figure 8.8 compiles the accuracies achieved on the primary, secondary header, and non-header attributes in detail for 100, 500, 1000, 1500, and 2000 partial observations in total. Figure 8.9 additionally shows the corresponding runtimes.

Hybrid Landmarks Scenario

Figure 8.11 depicts the fused result of the hybrid landmarks scenario (figure 8.6) based on 1000 partial observations. It can be noticed that the point-shaped traffic sign landmarks as well as the complex roundabout landmark are estimated accurately. Figure 8.10 shows the results of a detailed analysis of the accuracies achieved and the corresponding runtimes for 100, 500, 1000, 1500, and 2000 partial observations in total.

8.8 Discussion

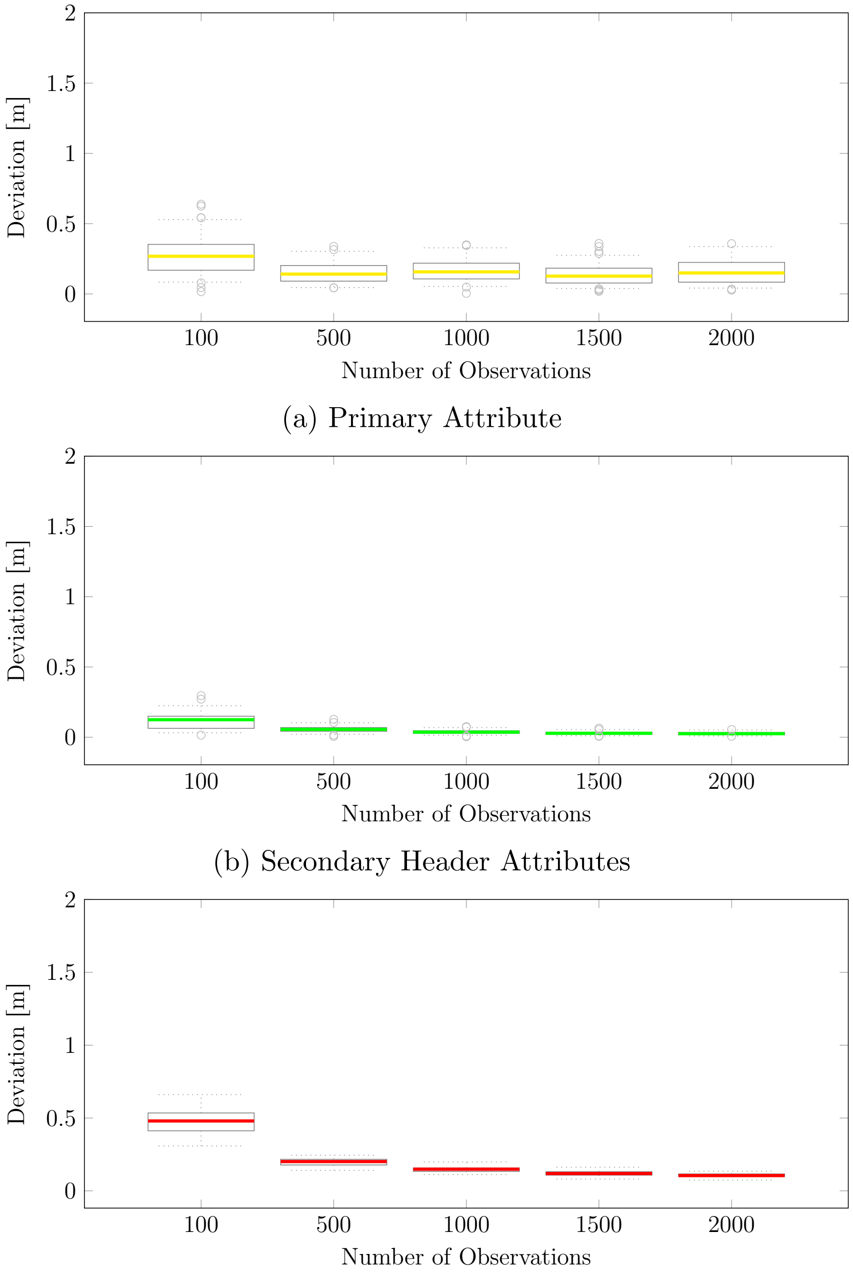
In this chapter, a generalized, parametric description of complex landmarks, the so-called ADG, has been introduced. Approaches for the fusion of complex landmarks represented by ADGs have been derived for EKF-based Online- and BA-based Full-SLAM. However, the concept proposed can also be adapted to other approaches



(a) $n = 100$

(b) $n = 500$

Figure 8.7: Visualization of the fused result of the complex landmark scenario for 100 and 500 observations. Primary attributes are visualized by circles ○ and secondary attributes by diamonds ◇. The ground truth is provided in red ● and the fusion result in black ●.



(a) Primary Attribute

(b) Secondary Header Attributes

(c) Secondary Non-Header Attributes

Figure 8.8: Accuracy evaluation of BA-based Full-SLAM solved by Sparse-QR, applied to the complex landmark scenario (figure 8.5) for 100, 500, 1000, 1500, and 2000 observations. The accuracy evaluation is divided into the accuracy achieved on primary, secondary header, and secondary non-header attributes. The evaluation is visualized by three box-whisker-plots according to the definition in appendix A.5.

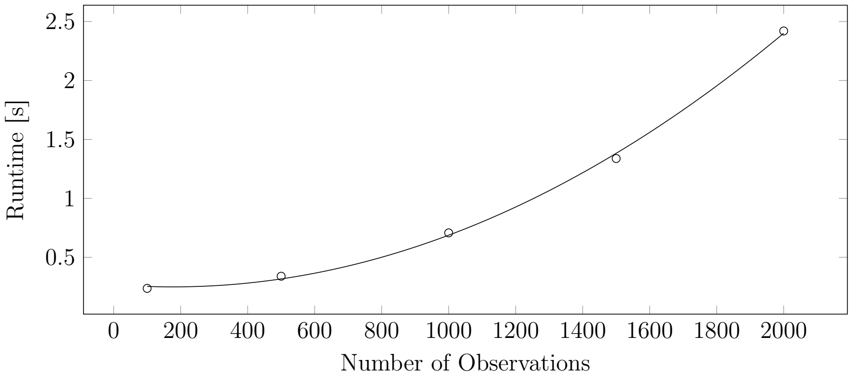
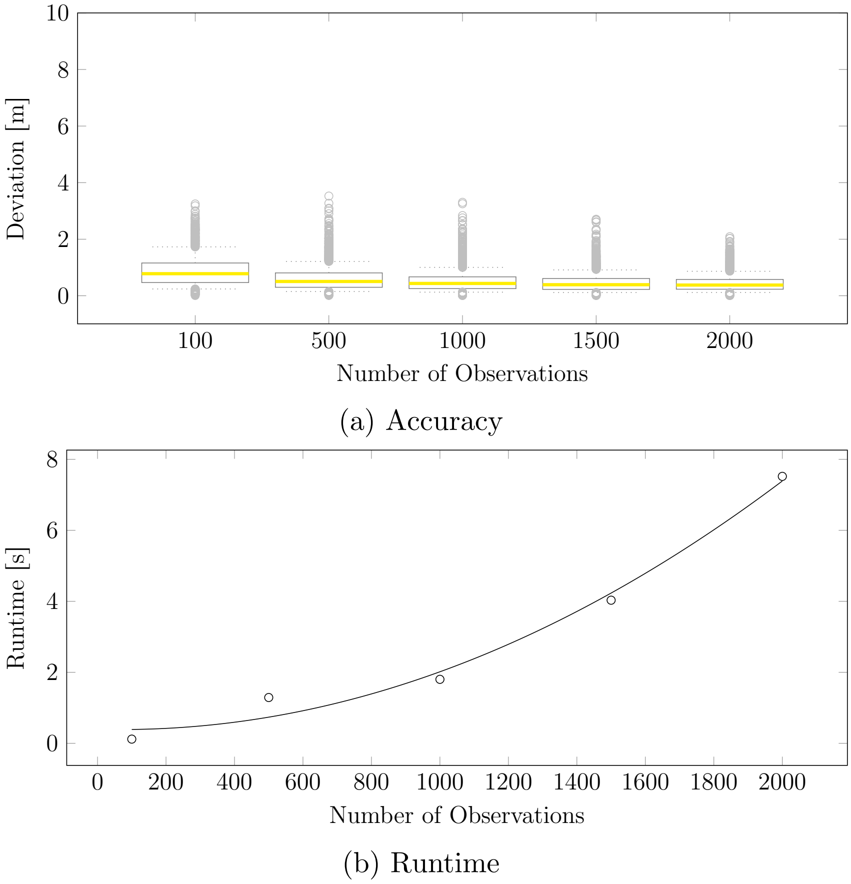


Figure 8.9: Runtime evaluation of BA-based Full-SLAM solved by Sparse-QR, applied to the complex landmark scenario for 100, 500, 1000, 1500, and 2000 observations.



(a) Accuracy

(b) Runtime

Figure 8.10: Accuracy and runtime evaluation of BA-based Full-SLAM solved via Sparse-QR, applied to the hybrid landmarks scenario (figure 8.6) for 100, 500, 1000, 1500, and 2000 observations, concerning the primary attributes. The evaluation of the accuracy is provided in the shape of a box-whisker-plot according to the definition provided in appendix A.5.

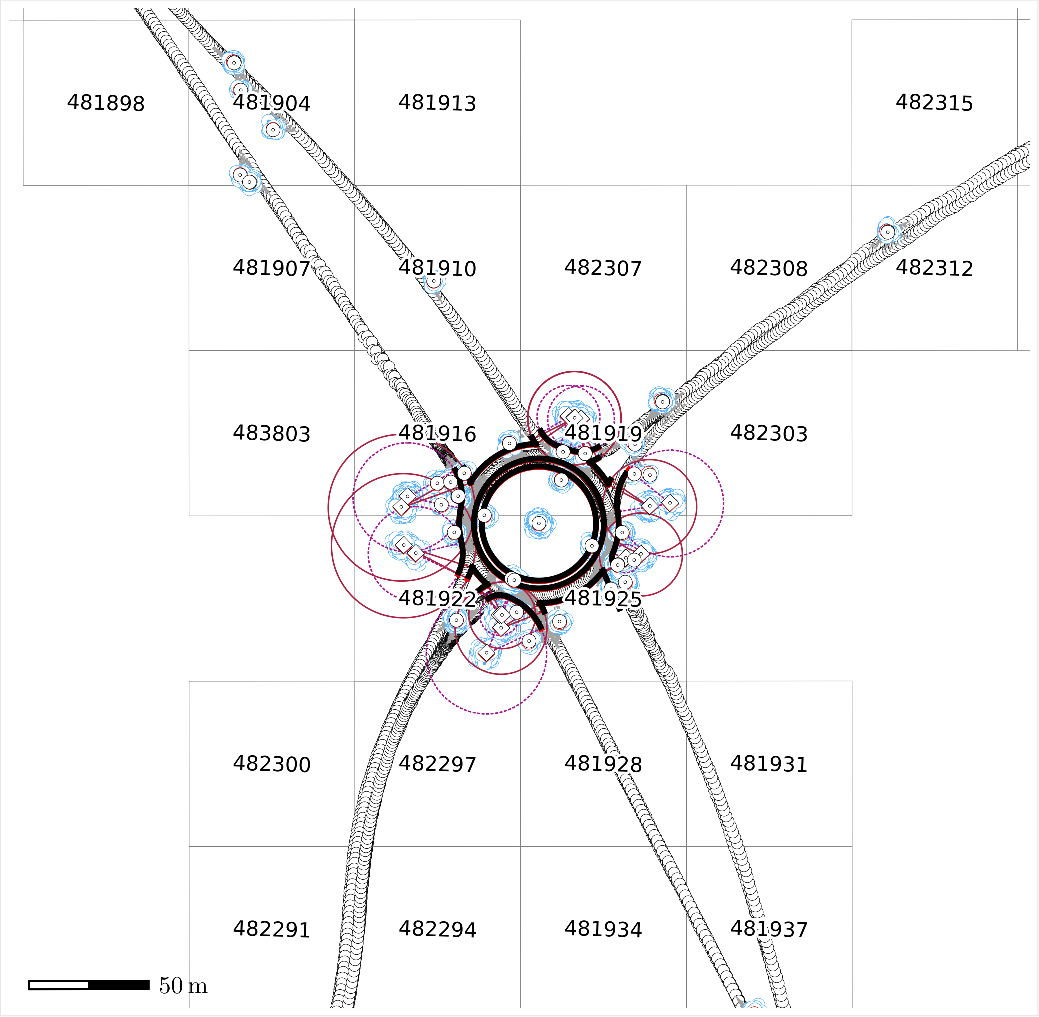


Figure 8.11: Fused result of the hybrid landmarks scenario based on 1000 partial observations. Primary attributes are visualized by circles ○ and secondary (non-) header attributes by diamonds ◇. The ground truth is shown in red ● and the fusion result in black ●.

to SLAM due to its generic structure. Because BA-based Full-SLAM solved via Sparse-QR has shown an outstanding balance between accuracy, robustness, and computational efficiency for point-shaped landmark data it has been chosen for the evaluation for complex landmarks, too.

According to the evaluation of the larger-scale hybrid landmark scenario, consisting of 43 landmarks (42 traffic signs and one roundabout), an overall accuracy concerning the primary attribute of 0.37 m (average median deviation) has been achieved for 2000 observations in total, what corresponds to approximately 46 observations per landmark. Compared to the complex landmark scenario nearly the same accuracy of 0.26 m (average median deviation), obtained from 100 partial observations, has been achieved, by only utilizing less than half of the observations. Thus it seems to be reasonable to fuse landmarks of different kind in a combined, hybrid manner because the overall accuracy and consistency is positively influenced. Furthermore, even for a larger-scale hybrid scenario the runtime of the BA-based Full-SLAM fusion has turned out as being feasible. For 2000 partial observations a runtime of 7.51 s, accumulated for all processing units, could be observed.

As mentioned earlier, the transformation of raw sensor data into a high-level, parametric landmark description takes place at the vehicle-side. This way both the required amount of cellular traffic, and the necessary computational resources at the back-end-side can be significantly reduced. However, the vehicles are expected only to propagate landmark observations strictly matching the parametric model or in the case of any doubt to mark them as invalid.

As an outlook, the approach presented could be extended by tackling both explicit constraints and cascaded subgraphs (cascaded dynamic blocks). It is expected that this way most complex landmarks in the automotive context that can be described in a parametric manner by an ADG can be appropriately fused.

Knowing the exact positions and shapes of the variety of complex landmarks in the automotive context, such as crossroads, roundabouts, sign gantries, traffic islands, and crosswalks, is of high importance to autonomous driving applications and next generation driver assistance systems. This data is by now usually not mapped by the common mapping-providers in the desired detail level. However, it could be instead mapped by common vehicles and by utilizing the suggested approach to complex landmark fusion, leading to much more up-to-date and highly detailed maps.

9 Fusion of Areal Data

This chapter is devoted to the fusion of areal CVD[1], such as temperatures, friction values, ambient light levels, traffic densities, signal strengths, or air qualities, into a continuous map. The map is represented by a data function *which represents the acquired data, and an* uncertainty function *which represents the reliability of the data function at every location of the domain of the map. The uncertainty function is obtained as some sort of non-normalized KDE[2] employing elliptical BF[3]s as kernels. The BFs are derived from the error covariances of the locations of the data samples. The data function approximates the acquired data points and is represented as a weighted sum of those BFs, inspired by the well-known RBFN[4] interpolation method. Computational efficiency is in particular gained by submapping and by utilizing the FGT[5]. An experimental analysis shows that in this manner a speed-up by more than a factor of ten can be achieved in the case of batch inserts or updates. A particular application of the concept is demonstrated by using the uncertainty function for the determination of lane centerline geometries which subsequently might be employed for referencing the fused areal data with highly up-to-date lane affiliations.*

Contents

[1] Collective Vehicle Data (CVD)
[2] Kernel Density Estimation (KDE)
[3] Basis Function (BF)
[4] Radial Basis Function Network (RBFN)
[5] Fast Gauss Transform (FGT)

9.1 Introduction

Areal CVD, such as temperatures, friction values, ambient light levels, traffic densities, signal strengths, or air qualities, can be used to collaboratively build highly up-to-date and large-scale maps thereof. The fields of application of fused areal CVD are manifold. Such fused areal data can be used as a valuable, a priori input for next generation comfort, assistance, and piloted driving functions. For example, the data can be utilized for a next generation predictive ACC[6] or a piloted driving function, where the velocity of the vehicle is additionally adjusted to maintain safety and comfort on slippery or bad road segments which were collaboratively identified.

This chapter presents an approach to the large-scale fusion of areal CVD by means of a double-staged areal data fusion. The data is given by measurements consisting of a location, a related uncertainty, and a data value, e. g. the friction at the location of measurement. “Double-staged areal data fusion” means that the areal data map is represented by two functions.

The *data function* represents the acquired data values, e. g. friction measurements. It approximates the acquired data values by a weighted sum of elliptical BFs, in contrast to RBF[7]-based representations. The elliptical BFs are derived from the error covariances of the locations of the data samples and the mutual locations of the samples.

The *uncertainty function* represents the reliability of the data function at every location. It is obtained as some sort of non-normalized KDE employing the BFs of the data function as kernels.

The areal map is calculated incrementally by augmenting the current map by a batch of further measurements. Besides measurement uncertainties, the approach takes into account requirements of large-scale fusion, such as incrementality, scalability, and compensating for the temporal decay of measurements. Computational efficiency is in particular gained by submapping according to section 4.2 and by utilizing the FGT. The approach has been experimentally evaluated on a data set of friction

[6] Adaptive Cruise Control (ACC)
[7] Radial Basis Function (RBF)

measurements, and it could be found that in this manner a speed-up by more than a factor of ten can be achieved in the case of batch inserts or updates. Furthermore, it is shown how the uncertainty function can be utilized for the determination of lane centerline geometries.

Section 9.2 introduces the concept of double-staged areal data fusion. Section 9.3 recaps interpolation by RBFNs and probability density estimation by KDE, and defines the uncertainty and data functions that are incorporated within the double-staged areal data fusion. Section 9.4 provides the details of the elliptic BFs. Section 9.5 incorporates compensation of the temporal decay of measurements by temporal weighting and section 9.6 of adaptive submapping. Section 9.7 is devoted to the application of the FGT for efficient evaluation. Section 9.8 summarizes the algorithmic approach of double-staged areal data fusion. Section 9.9 introduces the approach to the determination of lane centerline geometries. Finally, section 9.10 presents the results of the experimental evaluation.

9.2 Double-staged Areal Data Fusion

An areal data map is represented together with the measured data from which it is derived, as follows:

Definition 9.1: Double-Staged Areal Data Map

$R = (\boldsymbol{U}, \boldsymbol{D}, \mathcal{M})$,

where $\mathcal{M}$ is a finite set of measurements $(\boldsymbol{s}, \boldsymbol{\Sigma}, v)$, with $\boldsymbol{s}$ the location of the sample, $\boldsymbol{\Sigma}$ the covariance of the measurement error of the location, and $v \in \mathbb{R}$ the value of the measured sample,

$\boldsymbol{D}$ represents a data function which approximates the measurements,

$\boldsymbol{U}$ represents an uncertainty function.

$\boldsymbol{D}$ and $\boldsymbol{U}$ are defined over a region of interest and are derived from $\mathcal{M}$ by fusion. They are explicitly represented in rasterized form, i. e. on a data raster and on an uncertainty raster. A raster corresponds to a regular quad grid in the region of interest, and its elements, called pixels, have the function value at the center of the quads they represent.

Areal data fusion is performed by iteratively fusing new batches of measurements $\mathcal{M}_{\text{new}}$ with the already existing areal data map R:

Definition 9.2: Incremental Areal Data Fusion

Input:
1. An areal data map $R = (\boldsymbol{U}, \boldsymbol{D}, \mathcal{M})$.
2. A set of new measurements $\mathcal{M}_{\text{new}}$.
3. A temporal decay model.

Output: An areal data mapping $R' = (\boldsymbol{U}', \boldsymbol{D}', \mathcal{M}')$ which results from R by augmentation by $\mathcal{M}_{\text{new}}$ and that takes into account the temporal decay of the acquired data.

The basic idea to solving the problem of areal data fusion is to determine the data function as approximating function of the data values of the measurements. In literature, the fusion of spatial measurements via interpolation is usually performed either by *Kriging* or by RBFNs [22, 23, 82, 83]. However, a recent result by Fazio and Roisenberg [38] surprisingly reveals that, depending on the actual parametrization, both approaches are able to provide the same results. Furthermore, their research indicates that RBFNs are much more time efficient. Fazio and Roisenberg found a speed-up by a factor of 200. According to them, this difference even increases with more data. For those reasons, a BF-based approach has been chosen for the areal data fusion presented in the following.

Incremental areal data fusion will be performed by the following steps:

Solution 9.1: Incremental Areal Data Fusion

1. Updating the set of measurements.
2. Adapting the scales of affected old and the new measurements (optional).
3. Computing temporal weights.
4. Generating the rasterized uncertainty function for the updated set of measurements.
5. Generating the rasterized data function for the updated set of measurements.

9.3 Uncertainty and Data Functions

Interpolation by RBFNs represents the value $f(\boldsymbol{t})$ interpolated from data values $f(\boldsymbol{s}_i)$ at source points $\boldsymbol{s}_i$, $i = 1, \ldots, n$, at a target point $\boldsymbol{t}$ as a linear combination of RBFs $\varphi(\|\boldsymbol{t} - \boldsymbol{s}_i\|)$,

$$f(\boldsymbol{t}) = \sum_{i=1}^{n} \lambda_i \cdot \varphi(\|\boldsymbol{t} - \boldsymbol{s}_i\|), \tag{9.1}$$

where the weight parameters λ_i satisfy a linear equation system which expresses the approximation condition:

$$\underbrace{\begin{pmatrix} \varphi(r_{11}) & \varphi(r_{12}) & \cdots & \varphi(r_{1n}) \\ \varphi(r_{21}) & \varphi(r_{22}) & \cdots & \varphi(r_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(r_{n1}) & \varphi(r_{n2}) & \cdots & \varphi(r_{nn}) \end{pmatrix}}_{\boldsymbol{\varphi}} \underbrace{\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{pmatrix}}_{\boldsymbol{\lambda}} = \underbrace{\begin{pmatrix} f(\boldsymbol{s}_1) \\ f(\boldsymbol{s}_2) \\ \vdots \\ f(\boldsymbol{s}_n) \end{pmatrix}}_{\boldsymbol{f}}. \tag{9.2}$$

φ determines the shape of the RBF, e. g.

$$\varphi(r_{ij}) = \exp\left(-(a \cdot r_{ij})^2\right) \tag{9.3}$$

for Gaussian-RBFs. r_{ij} denotes the distance, e. g. Euclidean distance, between two corresponding source points $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, and a is a scale parameter.

The runtime complexity of interpolation by RBFNs is depending on the method used for solving the underlying equation system (equation 9.2). For QR[8], the time complexity is bounded by $\mathcal{O}\left(2\left(m - \frac{n}{3}\right)n^2\right)$ for a $m \times n$ matrix.

The double-staged areal data fusion presented in the following is an alternative to interpolation by RBFNs, which uses a basis representation as well. It is computationally more efficient and takes into account the uncertainty of the measurements.

The basic idea to defining the uncertainty function is a non-normalized KDE. The KDE is a non-parametric method to estimate the PDF[9] from a set of samples $\boldsymbol{s}_i$, $i = 1, \ldots, n$, of a distribution with unknown density function by

$$\hat{p}_h(\boldsymbol{t}) = \sum_{i=1}^{n} k_h(\boldsymbol{t} - \boldsymbol{s}_i), \tag{9.4}$$

where k_h is a kernel function scaled by h, i. e.

$$k_h(\boldsymbol{x}) = \frac{1}{h^{\dim(\boldsymbol{x})}} \cdot k\left(\frac{\boldsymbol{x}}{h}\right). \tag{9.5}$$

The kernel k is a density function, e. g. a radial Gaussian kernel

$$k(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^{\dim(\boldsymbol{x})}}} \cdot \exp\left(-\frac{\|\boldsymbol{x}\|^2}{2}\right). \tag{9.6}$$

[8] QR Factorization (QR)

[9] Probability Density Function (PDF)

The idea behind considering KDE is that the certainty is low in regions where a minor number of data samples has been acquired. This is immediately reflected by the PDF estimated by KDE from the locations of the measured samples.

The basic concept of KDE is considerably modified for the definition of uncertainty functions. The uniform kernel shapes are replaced with individual Gaussian kernel shapes k_i for every measured sample $\boldsymbol{s}_i$, $i = 1, \ldots, n$, whose covariance $\boldsymbol{\Sigma}_i$ is the covariance of the measurement error of the location of the sample. The uniform scaling factor h is replaced with an individual scaling factor h_i at every sample $\boldsymbol{s}_i$. The individual scaling factors allow for coping with varying sampling densities which make it difficult to choose an appropriate uniform scaling factor. For reasons of efficiency, the evaluation of the scaled kernels is restricted to their truncations at a 3σ range (equation A.2), i. e. covering 99.73 % of their values. The normalization by division of the sum by n is omitted since it is more convenient for deriving the data function and computation time is saved. This together leads to

Definition 9.3: Uncertainty Function

$$\begin{aligned} U(\boldsymbol{t}) &= \sum_{i=1}^{n} \varphi_i(\boldsymbol{t})|_{\mathcal{E}_{i,3}}, \\ \varphi_i(\boldsymbol{t}) &= \frac{1}{h_i} k_i \left(\frac{\boldsymbol{t} - \boldsymbol{s}_i}{h_i} \right), \end{aligned} \tag{9.7}$$

where k_i is a Gaussian density function with the covariance Σ_i of the measurement at location $\boldsymbol{s}_i$, h_i an individual scaling factor, $\mathcal{E}_{i,3}$ the truncation of the BF φ_i at a 3σ range, $i = 1, \ldots, n$.

The choice of the scaling factors is implied by the requirements of the data function. The details, in particular the calculation of the scaling factors, will be presented in section 9.4.

The data function calculates the data value v at an arbitrary location $\boldsymbol{t}$ as a weighted sum of the measurement values v_i at data points $\boldsymbol{s}_i$. The weight $w_i(\boldsymbol{t})$ expresses the degree of influence of the data point on $\boldsymbol{t}$. The influence is depending on the variance Σ_i. It is taken into account by employing the BFs φ_i and performing a normalization by using the uncertainty function so that the sum of weights at every location is equal to 1:

Definition 9.4: Data Function

$$\begin{aligned} D(\boldsymbol{t}) &= \sum_{i=1}^{n} w_i(\boldsymbol{t}) \cdot v_i, \\ w_i(\boldsymbol{t}) &= \frac{\varphi_i(\boldsymbol{t})|_{\mathcal{E}_{i,3}}}{U(\boldsymbol{t})}, \end{aligned} \tag{9.8}$$

where $U(\boldsymbol{t})$ is the uncertainty function, φ_i is the BF, $\mathcal{E}_{i,3}$ is the truncation of the BF φ_i at a 3σ range, and v_i is the value of the measurement at location $\boldsymbol{s}_i$, $i = 1, \ldots, n$.

9.4 Elliptical Kernels and Adaptive Adjustment of Scales

The task treated in this section is:

Problem 9.1: Elliptical Kernels and Adaptive Adjustment of Scales

Input: A finite set of locations $\boldsymbol{s}_i$ with corresponding covariance matrices $\boldsymbol{\Sigma}_i$, $i = 1, \ldots, n$.

Output: For every location $\boldsymbol{s}_i$ an elliptic Gaussian function with center $\boldsymbol{s}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$ so that the Gaussian functions cover the region of interest "well" and have a "moderate" overlap.

The approach of solution is to formulate those requirements as an LP[10] problem [24] which is based on covariance ellipses (appendix A.2) induced by the given covariances. Then the LP problem is solved by existing methods.

The LP problem is first derived for $n = 3$ locations for illustration purposes, and will then be formulated for an arbitrary number of locations. Figure 9.1 shows such an example.

The basic idea is to find scaling factors so that initially disjoint neighboring covariance ellipses with a common range c of the kernels k_i induced by the $\boldsymbol{s}_i$ and the covariances $\boldsymbol{\Sigma}_i$ do not intersect, but are expanded to their maximal extent. By closing these gaps, the homogeneity of the approximation is increased.

[10] Linear Program (LP)

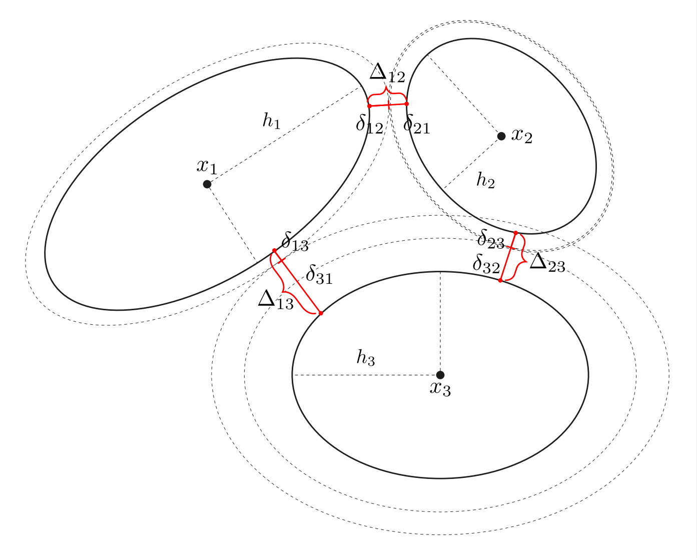


Figure 9.1: Adaptive adjustment of the scales h_i for three given uncertainty ellipses (appendix A.2). The sum of the corresponding deltas δ_{ij}, δ_{ji} is *not* allowed to exceed the shortest distance between two ellipses Δ_{ij}.

The optimization problem is

$$\begin{aligned} &\textbf{maximize:} \\ &\qquad h_1 + h_2 + h_3 \\ &\textbf{subject to:} \\ &\qquad \delta_{12}(h_1) + \delta_{21}(h_2) \leqslant \Delta_{12} \\ &\qquad \delta_{13}(h_1) + \delta_{31}(h_3) \leqslant \Delta_{13} \\ &\qquad \delta_{23}(h_2) + \delta_{32}(h_3) \leqslant \Delta_{23}. \end{aligned} \tag{9.9}$$

h_i denotes the wanted scales, $\delta_{ij}(h_i)$ the shortest distance between two uncertainty ellipses (appendix A.2), where the i-th ellipse is scaled by the parameter h_i and the j-th ellipse remains unscaled, and Δ_{ij} the shortest distance between the unscaled i-th and the unscaled j-th ellipse. The analytical computation of the shortest distance between two (uncertainty) ellipses is stressed by Zheng [136] as a surprisingly sophisticated problem. For details how to approach it analytically, the reader may refer to the literature provided.

Since the scale dependent shortest-distance function $\delta_{ij}(h_i)$ is non-linear, it is approximated by means of its 1st-order Taylor series (appendix A.3)

$$\delta_{ij}(h_i) = \delta_{ij}^m h_i + \delta_{ij}^b. \tag{9.10}$$

This way, it can be incorporated in a straightforward manner into the LP [24]. The scale dependent shortest-distance function is linearized through sampling, and the application of linear regression to the determined sample points. Evaluations have shown that the error introduced by this approach is negligible. The resulting LP is

$$\begin{aligned}
&\textbf{maximize:} \\
&\qquad h_1 + h_2 + h_3 \\
&\textbf{subject to:} \\
&\qquad \delta_{12}^m h_1 + \delta_{21}^m h_2 \leqslant \Delta_{12} - \delta_{12}^b - \delta_{21}^b \\
&\qquad \delta_{13}^m h_1 + \delta_{31}^m h_3 \leqslant \Delta_{13} - \delta_{13}^b - \delta_{31}^b \\
&\qquad \delta_{23}^m h_2 + \delta_{32}^m h_3 \leqslant \Delta_{23} - \delta_{23}^b - \delta_{32}^b .
\end{aligned} \tag{9.11}$$

The solution of the LP is usually not unique, i. e. multiple, according to the objective function, equally good solutions exist. Among those solutions are preferred which scale the ellipses of close-by measurements in a similar manner. This preference is expressed by augmenting the LP by additional constraints limiting the maximal deviation between neighboring adaptive scales h_i, h_j:

$$\begin{aligned}
&\textbf{maximize:} \\
&\qquad h_1 + h_2 + h_3 \\
&\textbf{subject to:} \\
&\qquad \delta_{12}^m h_1 + \delta_{21}^m h_2 \leqslant \Delta_{12} - \delta_{12}^b - \delta_{21}^b \\
&\qquad \delta_{13}^m h_1 + \delta_{31}^m h_3 \leqslant \Delta_{13} - \delta_{13}^b - \delta_{31}^b \\
&\qquad \delta_{23}^m h_2 + \delta_{32}^m h_3 \leqslant \Delta_{23} - \delta_{23}^b - \delta_{32}^b \\
&\\
&\qquad -\Omega_{12} \leqslant h_1 - h_2 \leqslant \Omega_{12} \\
&\qquad -\Omega_{13} \leqslant h_1 - h_3 \leqslant \Omega_{13} \\
&\qquad -\Omega_{23} \leqslant h_2 - h_3 \leqslant \Omega_{23} .
\end{aligned} \tag{9.12}$$

Ω_{ij} is modeled by a logistic, distance-dependent function

$$\Omega(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2) = \Omega_{ij} = G \cdot \frac{1}{1 + e^{-k \cdot G \cdot \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2} \left(\frac{G}{\Omega(0)} - 1 \right)}, \tag{9.13}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the centers of the ellipses i and j, respectively.

The values of the parameters have been empirically determined and are chosen as $G = 1.0$, $\Omega_0 = 0.1$, and $k = 0.4$ (figure 9.2).

This approach can be regarded as more natural than a common, constant limit for all given pairs of measurements. The two main reasons for incorporating a logistic function are that

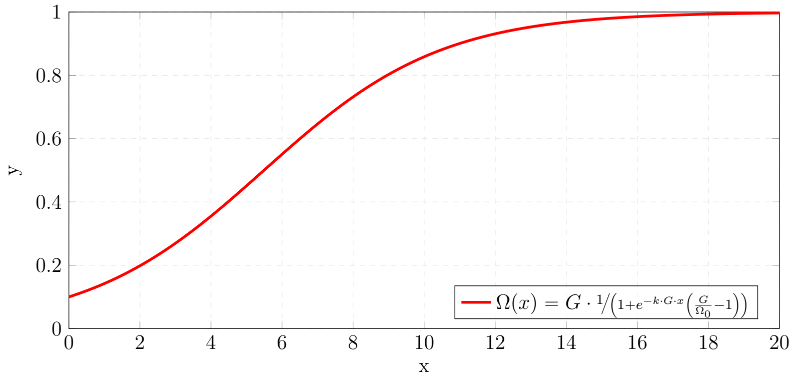


Figure 9.2: Sigmoidal limitation of the scale deviation for neighboring uncertainty ellipses (appendix A.2) by utilizing a logistic function. The logistic function is parametrized as follows: $G = 1.0$, $\Omega_0 = 0.1$, and $k = 0.4$.

- the scale deviations are allowed to differ in an exponential manner, proportional to the Euclidean distance of the corresponding measurements,
- an unlimited growth of the scale deviations is prevented, as the logistic function is limited by G.

Additionally, it is suggested to limit the range of adaptive scales h_i to an interval $[x, y]$, e. g. $[0.5, 3]$. This way, both the risk of potentially negative weights and of excessive inflation is circumvented. Commonly, negative weights have no real physical justification and should be, therefore, strictly avoided [4, 30, 62].

The formulation of the LP for the general case of n uncertainty ellipses is given by

Solution 9.2: Elliptical Kernels and Adaptive Adjustment of Scales

$$
\begin{array}{ll}
\textbf{maximize:} & \\
\qquad h_1 + \ldots + h_n & \Big|\ h_i \in [x, y] \\
\textbf{subject to:} & \\
\qquad \delta^m_{ij} h_i + \delta^m_{ji} h_j \leqslant \Delta_{ij} - \delta^b_{ij} - \delta^b_{ji} & \Big|\ \forall \text{ affected pairs } i, j \\
\qquad -\,\Omega_{ij} \leqslant h_i - h_j \leqslant \Omega_{ij} & \Big|\ \forall \text{ affected pairs } i, j
\end{array}
\qquad (9.14)
$$

The LP problem can be solved by means of the Simplex algorithm [24]. In practice, the Simplex algorithm is known to converge rapidly although it has an exponential worst-case runtime.

For improving the computational efficiency, an intersection test, e. g. by utilizing R-trees, can be utilized for pruning *not* reasonable pairings of measurements. Not reasonable pairings are, for example, ones that do not intersect, although, the

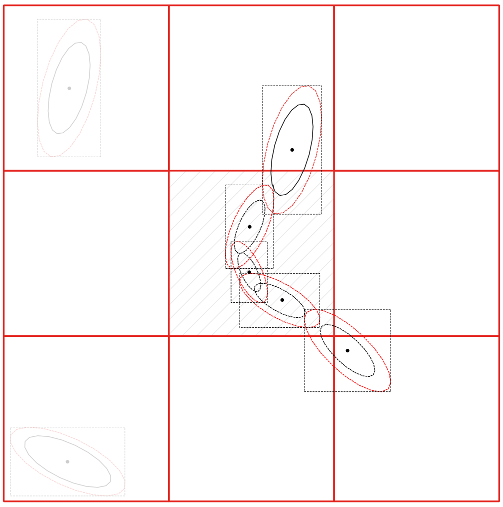


Figure 9.3: Pruning of uncertainty ellipses (appendix A.2) that do not intersect the central raster cell, although, extended to their maximal extent. Pruned uncertainty ellipses are faded out. For the pruning a fast R-tree-based intersection test is utilized.

corresponding uncertainties have been inflated to their maximal extent (figure 9.3). In practice, this way the number of equations can be significantly reduced.

9.5 Compensating for Temporal Decay

Areal sensor data is usually exposed to rapid decay. In order to take this into account, the temporal circular buffer-based weighting mechanism of section 4.3 is employed. The mechanism assigns a weight $\omega_i \in [0,1]$ to every measurement so that more recent measurements get a higher weight than measurements more in the past. Incorporating those weights leads to the following definitions:

Definition 9.5: Double-staged Areal Data Fusion with Temporal Decay

Let φ_i be the BF representing the covariance Σ_i of the measurement at location $\boldsymbol{s}_i$ with data value v_i, temporal weight $\omega_i \in [0,1]$, and $\mathcal{E}_{i,3}$ its truncation at a 3σ range, $i = 1, \ldots, n$:

Uncertainty function with temporal decay

$$U^{\text{tw}}(\boldsymbol{t}) = \sum_{i=1}^{n} \omega_i \cdot \varphi_i(\boldsymbol{t})|_{\mathcal{E}_{i,3}}. \tag{9.15}$$

Data function with temporal decay

$$\begin{aligned} D^{\text{tw}}(\boldsymbol{t}) &= \sum_{i=1}^{n} w_i^{\text{tw}}(\boldsymbol{t}) \cdot v_i, \\ w_i^{\text{tw}}(\boldsymbol{t}) &= \frac{\omega_i \cdot \varphi_i(\boldsymbol{t})|_{\mathcal{E}_{i,3}}}{U^{\text{tw}}(\boldsymbol{t})}. \end{aligned} \tag{9.16}$$

Intuitively, if more recent measurements occur in the environment of a location $\boldsymbol{t}$, possibly already existing measurements close to $\boldsymbol{t}$ contribute less to the value of $D^{\text{tw}}(\boldsymbol{t})$ than the new ones.

9.6 Submapping

As before, scalability is achieved by adaptive global submapping, as introduced in section 4.2. For evaluation of the data function $D(\boldsymbol{t})$ and the uncertainty function $U(\boldsymbol{t})$ at a location $\boldsymbol{t}$, the measurements located in the submap cell containing $\boldsymbol{t}$ and cells of its 8-neighborhood are considered. This means that measurements whose uncertainty exceeds the 8-neighborhood of their locations might be excluded. If the cell size is large compared to the typical uncertainties this has no severe implications on the quality of approximation, e. g. its continuity. The temporal decay mechanism is applied for every grid cell, under consideration of its 8-neighborhood (section 4.2).

9.7 Efficient Evaluation of the Uncertainty Function

Batch and cyclic updating according to the current temporal weights requires the calculation of the uncertainty function for multiple measurements at many locations. Since such updates happen frequently, the quest is for an efficient algorithmic implementation. The problem to be solved efficiently is specified as follows:

Problem 9.2: Efficient Evaluation of the Uncertainty Function

Input: An uncertainty function U according to equations 9.7 or 9.15, respectively, and an uncertainty raster $\boldsymbol{U}$

Output: The values of U at the pixels of $\boldsymbol{U}$.

The evaluation of the uncertainty function requires the accumulation of multiple Gaussians. The accumulation of multiple Gaussians is a general problem, and efficient algorithms have been suggested by multiple authors, such as the FGT [44, 45, 121] or derivatives of it [79, 94, 95, 100]. In particular, the FGT has been incorporated by Elgammal et al. [34] for improving the computational efficiency of KDE.

The basic idea of the FGT is to introduce two series expansions to the source points, i. e. the locations of the given measurements for the case considered here, and to the target points, i. e. the sampling points of the uncertainty raster cells. The far-field Hermite-expansion combines multiple source points under consideration of a common development point. It is utilized for determining the function value of the accumulated sources more time efficiently in the case that there are many nearby source points. Furthermore, the local Taylor-expansion (appendix A.3) combines multiple target points under consideration of a common development point. It is utilized for determining the function values of nearby targets more efficiently. Both expansions can be combined in the case of simultaneously many nearby source and target points.

The IFGT[11] [134, 135] aims to resolve the weaknesses of the FGT, mainly concerning the space partitioning and its performance at higher dimensions. To achieve this, it focuses on a single expansion type, the truncated Taylor-expansion (appendix A.3), and substitutes the basic box-based space partitioning of the FGT by a more advanced approach to clustering.

Another FGT-extension, the FIGTree[12]-algorithm [79], is based on the IFGT and aims to improve it even further. Weaknesses of the IFGT are a bad performance at low bandwidths and a complicated parameter adjustment. If parameters are chosen inappropriately, they can even lead to a worse computational performance than by direct evaluation. The first challenge is tackled by a tree data structure, the second one by introducing an automatic parameter selection scheme, which is based on the input data, desired accuracy, and chosen bandwidth.

9.7.1 Extension to Non-Radial Gaussian Basis Functions

Unfortunately, the FIGTree algorithm is, as well as its algorithmic predecessors, such as the FGT, IFGT, principally limited to radial Gaussian BFs, whose covariance matrices $\boldsymbol{C}$ feature a constant diagonal value σ^2, the so-called bandwidth:

$$\boldsymbol{C} = \begin{pmatrix} \sigma^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma^2 \end{pmatrix}. \tag{9.17}$$

Thus, either the assumed elliptic, therefore, non-radial CVD uncertainties of the vehicle localization need to be reduced to radial ones, e. g. via PCA[13] by averaging the determined axes or selecting the major one, or the FIGTree algorithm needs to be extended for non-radial Gaussian BFs.

[11] Improved Fast Gauss Transform (IFGT)
[12] Fast Improved Gauss Transform with Tree Data Structure (FIGTree)
[13] Principal Component Analysis (PCA)

For the extension of the FIGTree algorithm to non-radial Gaussian BFs, it can be exploited that a multivariate Gaussian distribution can be rewritten as a double summation of exponential terms

$$\begin{aligned}\varphi_{\boldsymbol{s},\boldsymbol{C}}(\boldsymbol{t}) &= \frac{1}{\sqrt{(2\pi)^{\dim(\boldsymbol{s})}\det(\boldsymbol{C})}} \cdot \exp\left(-\frac{1}{2}(\boldsymbol{t}-\boldsymbol{s})^T\boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{s})\right) \\ &= \frac{1}{\sqrt{(2\pi)^{\dim(\boldsymbol{s})}\det(\boldsymbol{C})}} \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{\dim(\boldsymbol{s})}\sum_{j=1}^{\dim(\boldsymbol{s})}(t_i-s_i)(t_j-s_j)\boldsymbol{C}^{-1}(i,j)\right),\end{aligned} \tag{9.18}$$

where $\boldsymbol{s}$ denotes the source point, $\boldsymbol{t}$ the target point, $\boldsymbol{C}$ the covariance matrix, and $\dim(\boldsymbol{s})$ the dimension of the vector $\boldsymbol{s}$.

The equation leads, in the case of a two-dimensional Gaussian distribution, to the product of four exponential terms,

$$\begin{aligned}\varphi_{\boldsymbol{s},\boldsymbol{C}}(\boldsymbol{t}) =& \frac{1}{\sqrt{(2\pi)^{\dim(\boldsymbol{s})}\det(\boldsymbol{C})}} \cdot \exp\left(-\frac{1}{2}\sum_{i=1}^{\dim(\boldsymbol{s})}\sum_{j=1}^{\dim(\boldsymbol{s})}(t_i-s_i)(t_j-s_j)\boldsymbol{C}^{-1}(i,j)\right) \\ \overset{2D}{=}& \frac{1}{\sqrt{(2\pi)^2\det(\boldsymbol{C})}} \\ &\cdot \exp\left(-\frac{1}{2}(t_1-s_1)(t_1-s_1)\boldsymbol{C}^{-1}(1,1)\right) \cdot \exp\left(-\frac{1}{2}(t_1-s_1)(t_2-s_2)\boldsymbol{C}^{-1}(1,2)\right) \\ &\cdot \exp\left(-\frac{1}{2}(t_2-s_2)(t_1-s_1)\boldsymbol{C}^{-1}(2,1)\right) \cdot \exp\left(-\frac{1}{2}(t_2-s_2)(t_2-s_2)\boldsymbol{C}^{-1}(2,2)\right),\end{aligned} \tag{9.19}$$

that can be rewritten to one-dimensional Gaussians.

As the covariance matrix $\boldsymbol{C}$ is symmetric, *only* two distinct cases, diagonal entries ①, and non-diagonal entries ②, need to be considered:

$$\begin{aligned}\varphi_{\boldsymbol{s},\boldsymbol{C}}(\boldsymbol{t}) \overset{\text{2D}}{=} & \frac{1}{\sqrt{(2\pi)^2 \det(\boldsymbol{C})}} \\ & \underbrace{\cdot \exp\left(-\frac{1}{2}(t_1 - s_1)(t_1 - s_1)\boldsymbol{C}^{-1}(1,1)\right)}_{①} \\ & \underbrace{\cdot \left(\exp\left(-\frac{1}{2}(t_1 - s_1)(t_2 - s_2)\boldsymbol{C}^{-1}(1,2)\right)\right)^2}_{②} \\ & \underbrace{\cdot \exp\left(-\frac{1}{2}(t_2 - s_2)(t_2 - s_2)\boldsymbol{C}^{-1}(2,2)\right)}_{①} .\end{aligned} \tag{9.20}$$

The first case ①, which is covering the diagonal elements $(i = j)$ of the covariance matrix $\boldsymbol{C}$, is denoted by

$$\exp\left(-\frac{1}{2}(t_i - s_i)(t_i - s_i)\boldsymbol{C}^{-1}(i,i)\right) = \exp\left(\frac{-(t_i - s_i)^2}{(\sqrt{2/\boldsymbol{C}^{-1}(i,i)})^2}\right). \tag{9.21}$$

The second case ②, which is covering the non-diagonal elements $(i \neq j)$ of the covariance matrix $\boldsymbol{C}$, is denoted by

$$\begin{aligned} & \left(\exp\left(-\frac{1}{2}\left(t_i - s_i\right)\left(t_j - s_j\right)\boldsymbol{C}^{-1}(i,j)\right)\right)^2 \\ &= \exp\left(-\left(t_i - s_i\right)\left(t_j - s_j\right)\boldsymbol{C}^{-1}(i,j)\right) \\ &= \exp\left(-\left(t_i - s_i\right)\frac{(t_i - s_i)}{(t_i - s_i)}\left(t_j - s_j\right)\boldsymbol{C}^{-1}(i,j)\right) \\ &= \exp\left(\frac{-\left(t_i - s_i\right)^2}{\left(\sqrt{1/\boldsymbol{C}^{-1}(i,j)}\right)^2}\right)^{\frac{\left(t_j - s_j\right)}{\left(t_i - s_i\right)}} . \end{aligned} \tag{9.22}$$

Afterwards, the three resulting radial Gaussians can be individually evaluated by means of the FIGTree algorithm. This increases the computational effort by the factor $(d\,(d+1))\,/2$ according to the dimension d, as the symmetry of the covariance matrix $\boldsymbol{C}$ can be exploited. As a consequence, the computational effort increases by approximately a factor of three for two-dimensional sources $\boldsymbol{s}$ and targets $\boldsymbol{t}$.

9.7.2 FIGTree-based Evaluation

The approach to the efficient evaluation of the uncertainty function (equations 9.7 and 9.15) presented in the following utilizes the FIGTree-algorithm. The challenge of multiple bandwidths, i. e. uncertainties, has to be tackled, as the FIGTree-algorithm does not support them out-of-the-box. To achieve this, it is exploited that CVD measurements often expose similar uncertainties. Therefore, clustering is applied to determine accumulations of source points of similar uncertainties. For this purpose, the well-known, density-based clustering algorithm DBSCAN[14] [37] is utilized.

For every identified cluster, the average uncertainty is calculated and the corresponding sources and targets are determined. The sources are determined in a straightforward manner by identifying each measurement that belongs to the current cluster. The identification of the relevant targets of a cluster considers the extent of the Gaussians. This is efficiently achieved by first computing the covariance ellipse (appendix A.2) of each Gaussian according to the desired range. Afterwards, a tight outer hull [81, 133] that encompasses all elliptic contours of a given cluster is computed. Then the target points are determined from the corresponding tight outer hull by a scanline-based approach. Finally, the accumulation of Gaussians is computed efficiently for every cluster, using the FIGTree-algorithm.

9.8 Algorithmic Summary of the Double-staged Approach

Figure 9.4 visualizes the approach to areal data fusion. At the first stage, the uncertainty function is calculated. At the second stage, the data function is determined based on the impact of the measurements gained by the uncertainty function. Both calculations can be performed in an incremental manner.

The incremental update of the uncertainty raster cells of the first stage is provided in algorithm 9.1. First, all grids cell $\mathcal{G}'$ which are affected by the new measurements $\mathcal{M}_{\mathrm{new}}$ are determined. Then the grids cells are processed separately. This way, the computation can be executed in parallel. For every grid cell, the interaction of the new points $\mathcal{M}_{\mathrm{new}}$, corresponding to the area of the currently inspected grid cell, and the old measurements $\mathcal{M}_{\mathrm{i}}$ that contributed to this grid cell, is analyzed. For the resulting subset of measurements $\mathcal{M}_i' \cup \mathcal{M}_{\mathrm{new}}$ an LP according to equation 9.14 is set up and solved. Afterwards, the temporal weights Ω_i are efficiently computed by means of pre-aggregation (section 4.3). Finally, affected historic measurements $\mathcal{M}_{\mathrm{i}}'$ are removed from the current uncertainty raster cell $\boldsymbol{U}_i$ and readded to it together with the new measurements $\mathcal{M}_{\mathrm{new}}$ corresponding to this grid cell.

[14] Density-based Spatial Clustering of Applications with Noise (DBSCAN)

Algorithm 9.2 outlines the alternative of FIGTree-based batch-update of the uncertainty raster. The FIGTree-based algorithm is pretty similar to the direct approach of algorithm 9.1. It actually differs by the computation of bandwidth clusters and the subsequent iteration over the determined ones. Within each iteration, the sources contributing to the current cluster and the associated targets are determined. Afterwards, the accumulation of Gaussians is computed efficiently by incorporating the FIGTree algorithm.

At the second stage, the sensor data is approximated by means of the not-normalized KDE computed in the previous stage. The incremental update of the data raster is shown in algorithm 9.3. First, all affected grid cells $\mathcal{G}'$ are determined. Then the uncertainty raster cell $\boldsymbol{U}_i$ is utilized for determining the overall impact of each new measurement. This information is subsequently used to add a new BF to the data raster cell $\boldsymbol{D}_i$. Prior, of course, all affected BFs have to be removed from the data raster cell.

Figure 9.5 summarizes the complete areal CVD fusion algorithm. It starts with the adaptive submapping. Then the two steps of the actual approximation are performed on a per-grid-cell basis: the uncertainty step ② and the subsequent data step ③.

9.9 Derivation of Lane Centerline Geometries

The uncertainty function (equation 9.7, respectively 9.15) can be utilized for determining up-to-date centerline geometries of lanes from CVD. A lane centerline is characterized by measurement locations of high density which are approximately arranged along a curve. This induces some sort of ridge of the uncertainty function, if the function values are considered as height values.

Problem 9.3: Derivation of Lane Centerline Geometries

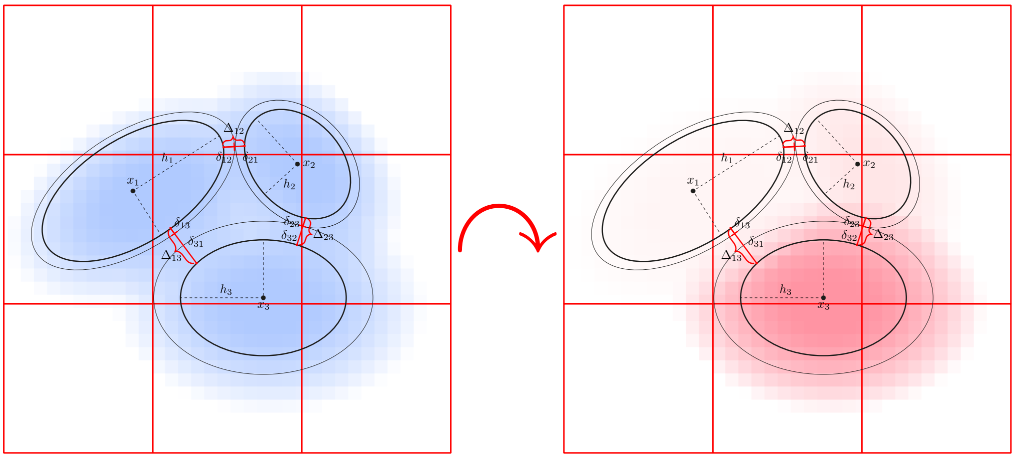
Input: The uncertainty function U of a set of areal measurements and its derivative.

Output: A set of ridge curves L which fulfill the constraints of lanes.

Constraints on lanes may concern the feasibility of their geometric shape, for example curvature restrictions, as commonly encountered in road engineering.

The first step of the algorithm determines peak points of the uncertainty function. For that purpose, additionally the first derivative of the uncertainty function U is calculated. It is obtained by utilizing the first derivative of the Gaussian [90] as BF of U:

$$\frac{\partial \varphi(\boldsymbol{t})}{\partial \boldsymbol{t}} = -\varphi(\boldsymbol{t}) \cdot \boldsymbol{C}^{-1} \cdot (\boldsymbol{t} - \boldsymbol{s}). \tag{9.23}$$



Uncertainty Raster

Data Raster

Figure 9.4: Visualization of the simplified working principle of double-staged areal data fusion. At first, the uncertainty raster and, subsequently, the data raster is updated.

```
input  : $\mathcal{G} = \{R_1 = (\boldsymbol{U}_1, \boldsymbol{D}_1, \mathcal{M}_1), \ldots, R_m = (\boldsymbol{U}_m, \boldsymbol{D}_m, \mathcal{M}_m)\}$,
         $\mathcal{M}_{\text{new}} = \{(\boldsymbol{x}_1, \boldsymbol{\Sigma}_1, v_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{\Sigma}_n, v_n)\}$
output : $\mathcal{G} = \{R_1 = (\boldsymbol{U}_1, \boldsymbol{D}_1, \mathcal{M}_1), \ldots, R_m = (\boldsymbol{U}_m, \boldsymbol{D}_m, \mathcal{M}_m)\}$
① Determine affected grid cells ($\mathcal{G}' \subseteq \mathcal{G}$)
$\mathcal{G}' = \text{affected\_grid\_cells}(\mathcal{G},\ \mathcal{M}_{\text{new}})$ ;
for $(\boldsymbol{U}_i, \mathcal{M}_i) \in \mathcal{G}'$ do
    ② Determine by new measurements affected old measurements
    $\mathcal{M}_{\text{i}}' = \text{affected\_meas}(\mathcal{M}_i,\ \mathcal{M}_{\text{new}})$ ;
    ③ Adjust scales of affected and new measurements
    $\mathcal{A} = \{a_j\} = \text{compute\_scales}(\mathcal{M}_i' \cup \mathcal{M}_{\text{new}})$ ;
    ④ Compute temporal weights
    $\Omega = \{\omega_j\} = \text{compute\_weights}(\mathcal{M}_i' \cup \mathcal{M}_{\text{new}})$ ;
    ⑤ Update uncertainty raster cells
    $\boldsymbol{U}_i = \text{remove\_measurements}(\boldsymbol{U}_i,\ \mathcal{M}_i')$ ;
    $\boldsymbol{U}_i = \text{add\_measurements}(\boldsymbol{U}_i,\ \mathcal{M}_i' \cup \mathcal{M}_{\text{new}},\ \mathcal{A},\ \Omega)$ ;
    $\mathcal{M}_i = \mathcal{M}_i \cup \mathcal{M}_{\text{new}}$ ;
end
return $(\mathcal{G} \setminus \text{affected\_grid\_cells}(\mathcal{G},\ \mathcal{M}_{\text{new}})) \cup \mathcal{G}'$ ;
```

Algorithm 9.1: Algorithmic outline of the uncertainty step of the double-staged areal data fusion.

```
input  : $\mathcal{G} = \{R_1 = (\boldsymbol{U}_1, \boldsymbol{D}_1, \mathcal{M}_1), \dots, R_m = (\boldsymbol{U}_m, \boldsymbol{D}_m, \mathcal{M}_m)\}$,
         $\mathcal{M}_{\text{new}} = \{(\boldsymbol{x}_1, \boldsymbol{\Sigma}_1, v_1), \dots, (\boldsymbol{x}_n, \boldsymbol{\Sigma}_n, v_n)\}$
output : $\mathcal{G} = \{R_1 = (\boldsymbol{U}_1, \boldsymbol{D}_1, \mathcal{M}_1), \dots, R_m = (\boldsymbol{U}_m, \boldsymbol{D}_m, \mathcal{M}_m)\}$
① Determine affected grid cells ($\mathcal{G}' \subseteq \mathcal{G}$)
$\mathcal{G}' = \text{affected\_grid\_cells}(\mathcal{G}, \mathcal{M}_{\text{new}})$ ;
for $(\boldsymbol{U}_i, \mathcal{M}_i) \in \mathcal{G}'$ do
    ② Determine by new measurements affected old measurements
    $\mathcal{M}'_{\text{i}} = \text{affected\_meas}(\mathcal{M}_i, \mathcal{M}_{\text{new}})$ ;
    ③ Adjust scales of affected and new measurements
    $\mathcal{A} = \{a_j\} = \text{compute\_scales}(\mathcal{M}'_i \cup \mathcal{M}_{\text{new}})$ ;
    ④ Compute temporal weights
    $\Omega = \{\omega_j\} = \text{compute\_weights}(\mathcal{M}'_i \cup \mathcal{M}_{\text{new}})$ ;
    ⑤ Cluster bandwidths
    $\mathcal{B} = \{\boldsymbol{\Sigma}_b\} = \text{cluster\_bandwidths}(\mathcal{A}, \mathcal{M}'_i \cup \mathcal{M}_{\text{new}})$ ;
    ⑥ Remove affected measurements
    $\boldsymbol{U}_i = \text{remove\_measurements}(\boldsymbol{U}_i, \mathcal{M}'_i)$ ;
    for $\boldsymbol{\Sigma}_b \in \mathcal{B}$ do
        ⑦ Filter sources with current bandwidth
        $\mathcal{M}_{\text{filt}} = \text{filter\_sources}(\mathcal{M}'_i \cup \mathcal{M}_{\text{new}}, \boldsymbol{\Sigma}_b)$ ;
        $\mathcal{T} = \{\boldsymbol{y}_k\} = \text{determine\_targets}(\mathcal{M}_{\text{filt}})$ ;
        ⑧ Update affected measurements
        $\boldsymbol{U}_i = \text{add\_measurements\_via\_FIGTree}(\boldsymbol{U}_i, \mathcal{M}_{\text{filt}}, \mathcal{T}, \boldsymbol{\Sigma}_b)$ ;
        $\mathcal{M}_i = \mathcal{M}_i \cup \mathcal{M}_{\text{filt}}$ ;
    end
end
return $(\mathcal{G} \backslash \text{affected\_grid\_cells}(\mathcal{G}, \mathcal{M}_{\text{new}})) \cup \mathcal{G}'$ ;
```

Algorithm 9.2: Algorithmic outline of the FIGTree-based batch-update of the uncertainty step of double-staged areal data fusion.

```
input  : $\mathcal{G} = \{R_1 = (\boldsymbol{U}_1, \boldsymbol{D}_1, \mathcal{M}_1), \dots, R_m = (\boldsymbol{U}_m, \boldsymbol{D}_m, \mathcal{M}_m)\}$,
         $\mathcal{M}_{\text{aff}} = \mathcal{M}'_i \cup \mathcal{M}_{\text{new}}$
output : $\mathcal{G} = \{R_1 = (\boldsymbol{U}_1, \boldsymbol{D}_1, \mathcal{M}_1), \dots, R_m = (\boldsymbol{U}_m, \boldsymbol{D}_m, \mathcal{M}_m)\}$
① Determine affected grid cells ($\mathcal{G}' \subseteq \mathcal{G}$)
$\mathcal{G}' = \text{affected\_grid\_cells}(\mathcal{G}, \mathcal{M}_{\text{aff}})$ ;
for $\boldsymbol{D}_i \in \mathcal{G}'$ do
    ② Update affected BFs
    $\boldsymbol{D}_i = \text{remove\_BFs}(\boldsymbol{U}_i, \boldsymbol{D}_i, \mathcal{M}_{\text{aff}})$ ;
    $\boldsymbol{D}_i = \text{add\_BFs}(\boldsymbol{U}_i, \boldsymbol{D}_i, \mathcal{M}_{\text{aff}})$ ;
end
return: $(\mathcal{G} \backslash \text{affected\_grid\_cells}(\mathcal{G}, \mathcal{M}_{\text{aff}})) \cup \mathcal{G}'$ ;
```

Algorithm 9.3: Algorithmic outline of the data step of the double-staged areal data fusion.

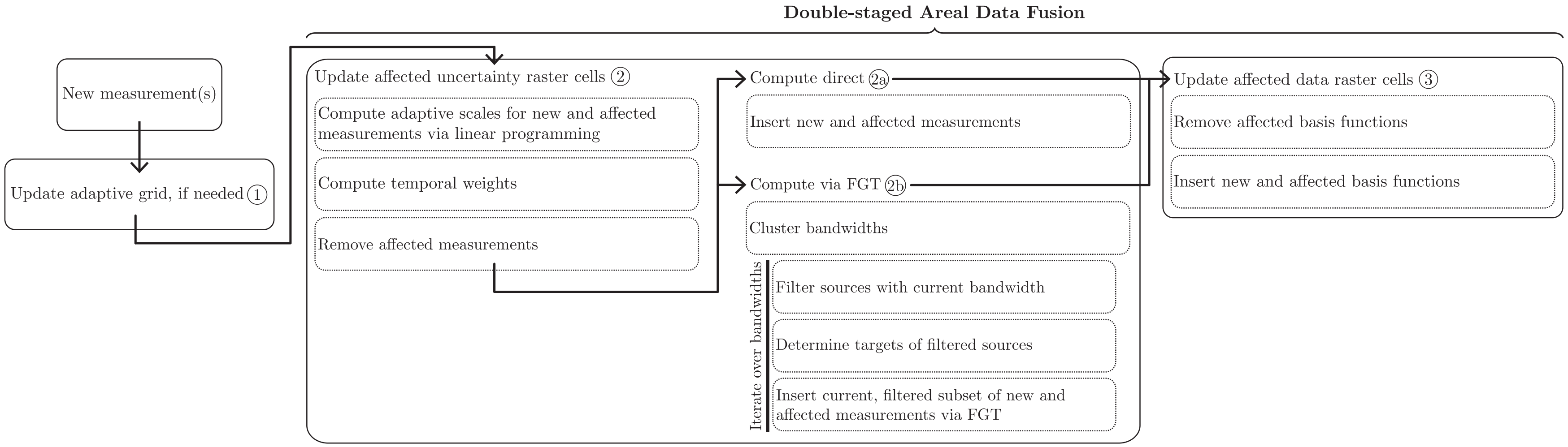

Double-staged Areal Data Fusion
New measurement(s)
Update adaptive grid, if needed ①
Update affected uncertainty raster cells ②
Compute adaptive scales for new and affected measurements via linear programming
Compute temporal weights
Remove affected measurements
Compute direct ②a
Insert new and affected measurements
Compute via FGT ②b
Cluster bandwidths
Iterate over bandwidths
Filter sources with current bandwidth
Determine targets of filtered sources
Insert current, filtered subset of new and affected measurements via FGT
Update affected data raster cells ③
Remove affected basis functions
Insert new and affected basis functions

Figure 9.5: Outline of the double-staged approach to the fusion of areal CVD via approximation. ① First, adaptive partitioning (section 4.2) is utilized for dividing the optimization problem into smaller subproblems. ② Second, uncertainty raster cells are updated either by the direct or the FGT-based approach. ③ Third, based upon the updated uncertainty raster cells the data raster cells are updated.

As before, the computation of the first derivative can be performed incrementally. Subsequently, for each trajectory point a corresponding peak point is determined:

Definition 9.6: Derivation of Lane Centerline Geometries: Identification of Peak Points

A peak point is identified when the slope of the first derivative of the uncertainty function U respectively U^{tw} at a pixel of the uncertainty raster is approximately 0,

$$\left\| \frac{\partial U(\boldsymbol{t})}{\partial \boldsymbol{t}} \right\|_2 \approx 0, \tag{9.24}$$

and the value of the uncertainty function at the pixel is among the highest ones within the 3σ covariance ellipse (appendix A.2) of the current trajectory point.

The second step connects the locations of peak points to seams. For this purpose, a histogram of the vehicle trajectory directions that contributed to this peak point is evaluated, and the peak point is connected with a preceding and succeeding peak point that is indicated as most probable by the histogram. Additional constraints are considered, such as the preference of seams with a constant curvature. The consideration of constraints has been identified to affect the overall robustness, even under tough conditions, in a positive manner. The approach features a linear runtime complexity in the number of identified peak points, as it comprehensively exploits the domain of the problem. Therefore, it can be regarded concerning its runtime and for this specific use-case, as superior to more generic ones, such as e. g. Graph Cut, that usually exhibit a non-polynomial runtime complexity [88].

The third step smooths the determined seams in order to compensate for small deviations induced by the discretization. For this purpose, parametric cubic spline curves are incorporated. Furthermore, main seams are identified by their average amplitude so that minor seams are clung to main seams. This is, for example, relevant for the use-case of highway entries or exits, where the main seam is the highway. Technically, clinging is achieved by setting the first and second derivative of the interconnection point of the minor cubic spline seam to the ones of the main cubic spline seam.

Figure 9.6 outlines the complete algorithm including adaptive submapping according to section 4.2. After new measurements have been acquired, and the corresponding uncertainty raster cells have been updated, the lane centerline geometries are determined.

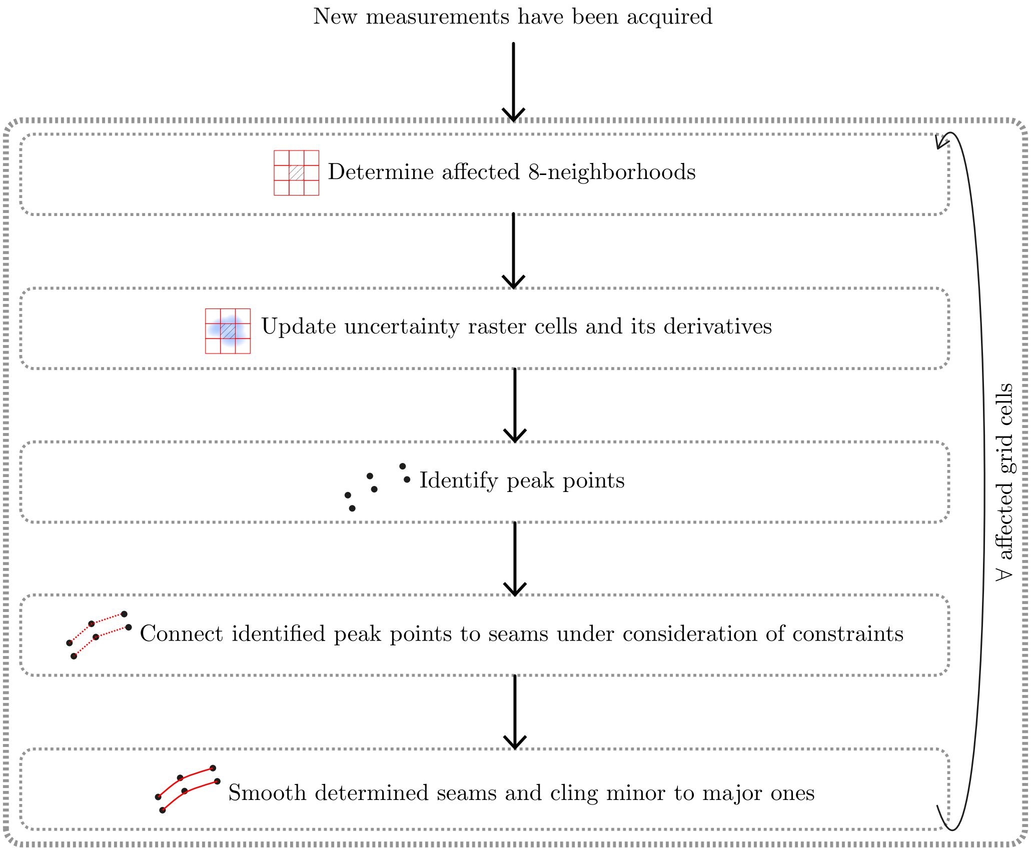

New measurements have been acquired
Determine affected 8-neighborhoods
Update uncertainty raster cells and its derivatives
Identify peak points
Connect identified peak points to seams under consideration of constraints
Smooth determined seams and cling minor to major ones
∀ affected grid cells

Figure 9.6: Outline of the algorithm for determining lane centerline geometries by means of the uncertainty function computed at the first stage of the double-stage areal data fusion approach.

9.10 Evaluation

In this section, first the experimental set-up is introduced (section 9.10.1). Subsequently, the double-staged approach to the incremental and scalable fusion of areal CVD is evaluated in detail (section 9.10.2). Afterwards, the improvement, in the case of batch inserts, deletions, and updates, by means of the FGT is analyzed (section 9.10.3). Furthermore, the approach proposed for the derivation of precise centerline geometries of lanes is evaluated (section 9.10.4). Finally, the findings are discussed and further enhancements are suggested.

9.10.1 Experimental Set-Up

As part of the evaluation, multiple trajectories were acquired by a testing vehicle (section 10.2). They resulted in approximately 5 000 distinct measurements per evaluated 100 m × 100 m grid cell on average. The utilized GNSS[15] receiver was parametrized to operate with a sampling rate of 10 Hz (section 3.2). The acquired GNSS pseudoranges were propagated together with the acquired areal CVD to

[15] Global Navigation Satellite System (GNSS)

the back-end-side. Afterwards, precise GNSS positions were derived by means of postprocessing (section 3.2.2). After postprocessing the vehicle trajectories, the measurements were geographically referenced by utilizing linear interpolation based on the corresponding GNSS and measurement timestamps. Finally, the acquired trajectories were multiply duplicated by adding Gaussian noise, with the intention to reduce the ecological impact of the acquisition of a big number of trajectories.

9.10.2 Direct Incremental Computation

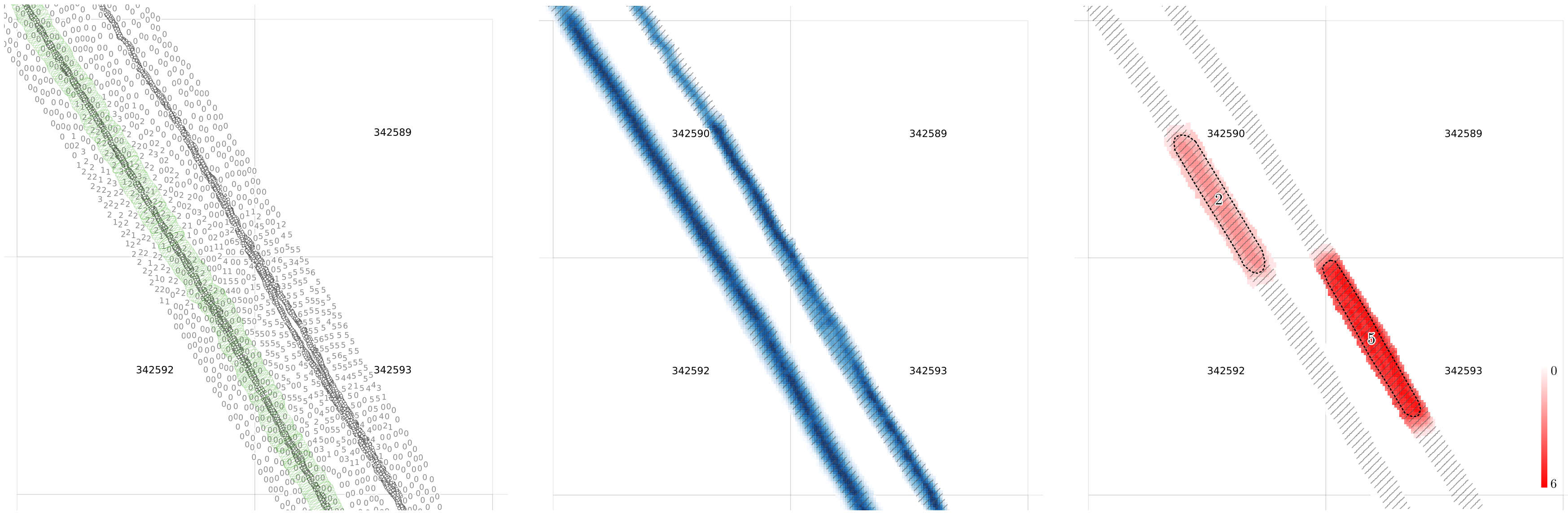
Figure 9.7 visualizes the results for exemplary friction test data with a known ground truth and by utilizing the direct approach to double-staged areal data fusion. The exemplary friction measurements are ranging from zero to six. Zero represents a good (usual) friction and six a bad one. It can be noticed that areas with a low friction are identified in a robust manner and with a fine granularity. Adaptive partitioning (section 4.2) has been parametrized to provide 100 m × 100 m raster cells with 1 m × 1 m pixels.

Furthermore, a detailed runtime analysis is provided in figure 9.8a. It can be noticed that the computational effort grows linearly with the number of measurements, with some minor deviations from the linear regression. However, they are not caused by the algorithm itself, but by the internal working principles of the database (caching, data persistence, memory management) and the operating system (scheduler, memory management). Furthermore, it may be noticed that the actual stamp computation consumes the major share of the overall computational effort. Pre- and postponed operations are nearly negligible. The averaged computational requirement for one single grid cell for both the uncertainty and the data function under consideration of its 8-neighborhood is approximately 4.53 s for 50 000 data points and 27.35 s for 250 000 ones. For determining the joint runtime of both stages the runtimes need to be doubled. The analysis reveals that the major computational effort corresponds to the stamp computation and optimizations of the algorithm should primarily focus it, for example by means of FGT.

9.10.3 FIGTree-based Batch Computation

In this section, the single-bandwidth and the multi-bandwidth FGT-based approach to the calculation of the uncertainty function are experimentally analyzed and compared to each other. The FGT could be theoretically also applied to the data function. However, many distinct data values would lead to lots of possibly single-element clusters. For that reason, the FGT is not very suitable for speeding up the second stage.

As the FIGTree algorithm for computing the FGT is out-of-the-box limited to radial Gaussian BFs, the non-radial Gaussian BFs are transformed by means of PCA

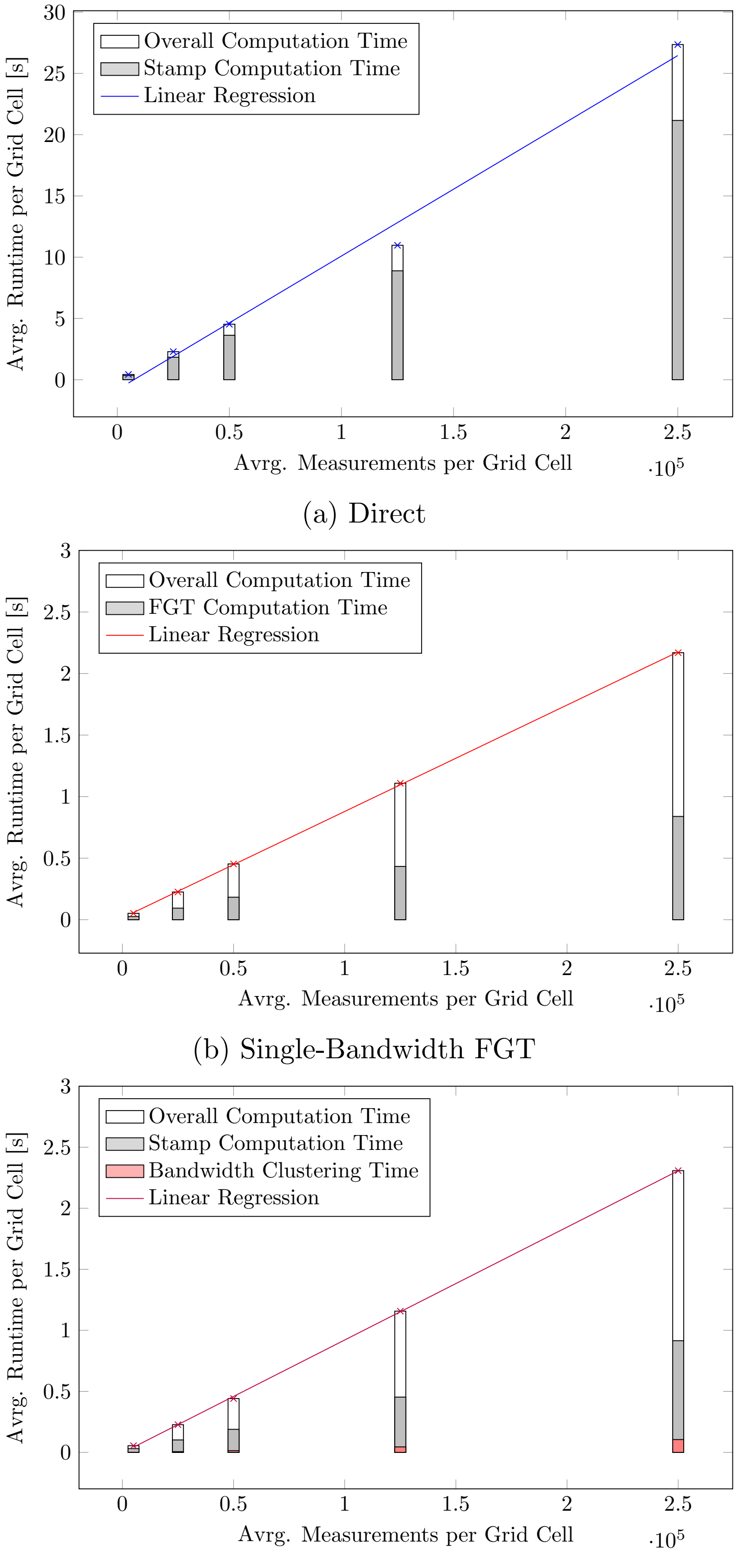


(a) Input Data

(b) Uncertainty Step

(c) Data Step

Figure 9.7: Exemplary friction data input with the corresponding localization uncertainties (for visual clarity the continuous data values have been rounded to integers and the localization uncertainties, highlighted in green ●, are provided for only one trajectory, cf. subfigure a), the subsequent uncertainty stage (subfigure b) and the data fusion with the corresponding ground truth (subfigure c) for a short road excerpt. The determined confidence area is hachured in gray ●.



(a) Direct

(b) Single-Bandwidth FGT

(c) Multi-Bandwidth FGT

Figure 9.8: Detailed average runtimes for the direct (subfigure a), the FGT-based (subfigure b), and the multi-bandwidth FGT-based approach (subfigure c) for approximately 5 000, 25 000, 50 000, 125 000, and 250 000 measurements *per* grid cell.

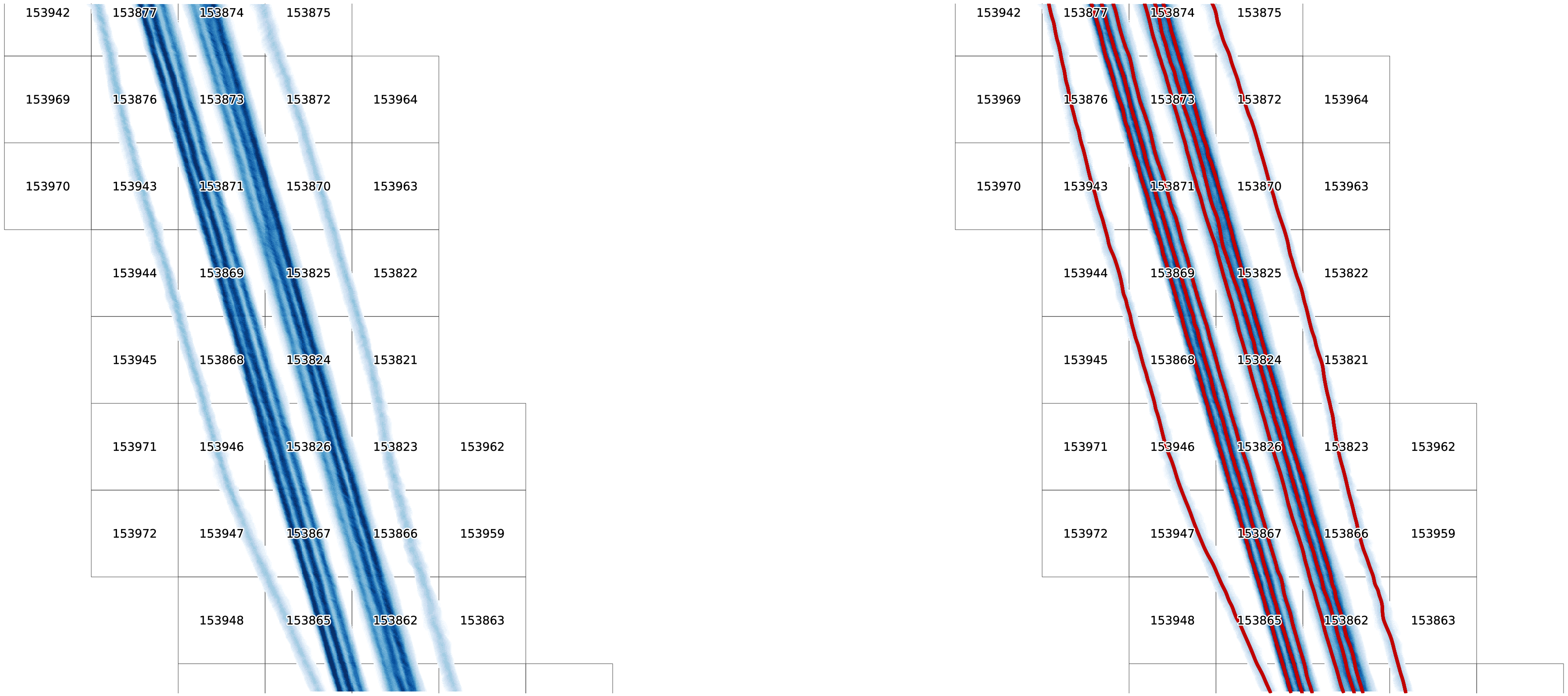
and computing the average value of the minor and major axes to radial ones. The subsequently provided results reflect radial Gaussian BFs and include the cost of the PCA and averaging.

However, by rewriting the multivariate Gaussian BFs by means of a double summation, as shown previously, the FIGTree algorithm can be as well utilized for non-radial Gaussian BFs. However, this way the computational effort is increased by a factor of three concerning two-dimensional Gaussian-BFs (section 9.7).

Figures 9.8b and c provide the average runtimes for the single- and the multi-bandwidth approach to double-staged approximation on a per grid-cell basis. While the single-bandwidth approach first computes an average bandwidth for every grid cell, the multi-bandwidth approach starts by computing clusters of bandwidths for every grid. The allowed bandwidth deviation is limited to $\epsilon_{\text{band}} = 0.1\,\text{m}$, what can be considered as minor in practice. It may be noticed that, by incorporating the FIGTree algorithm, the computational requirements can be reduced by more than a factor of ten, in the case of radial Gaussian BFs, and by more than a factor of three in the case of non-radial Gaussian BFs. The resulting computation time for the single-bandwidth approach is approximately 2.169 s for 250 000 measurements per grid cell. Furthermore, the multi-bandwidth extension is able to process 250 000 measurements in 2.308 s per grid cell. Thus the computational effort for the suggested multi-bandwidth extension is in practice nearly negligible, and hence it should be strongly preferred to the single-bandwidth approach.

9.10.4 Derivation of Lane Centerline Geometries

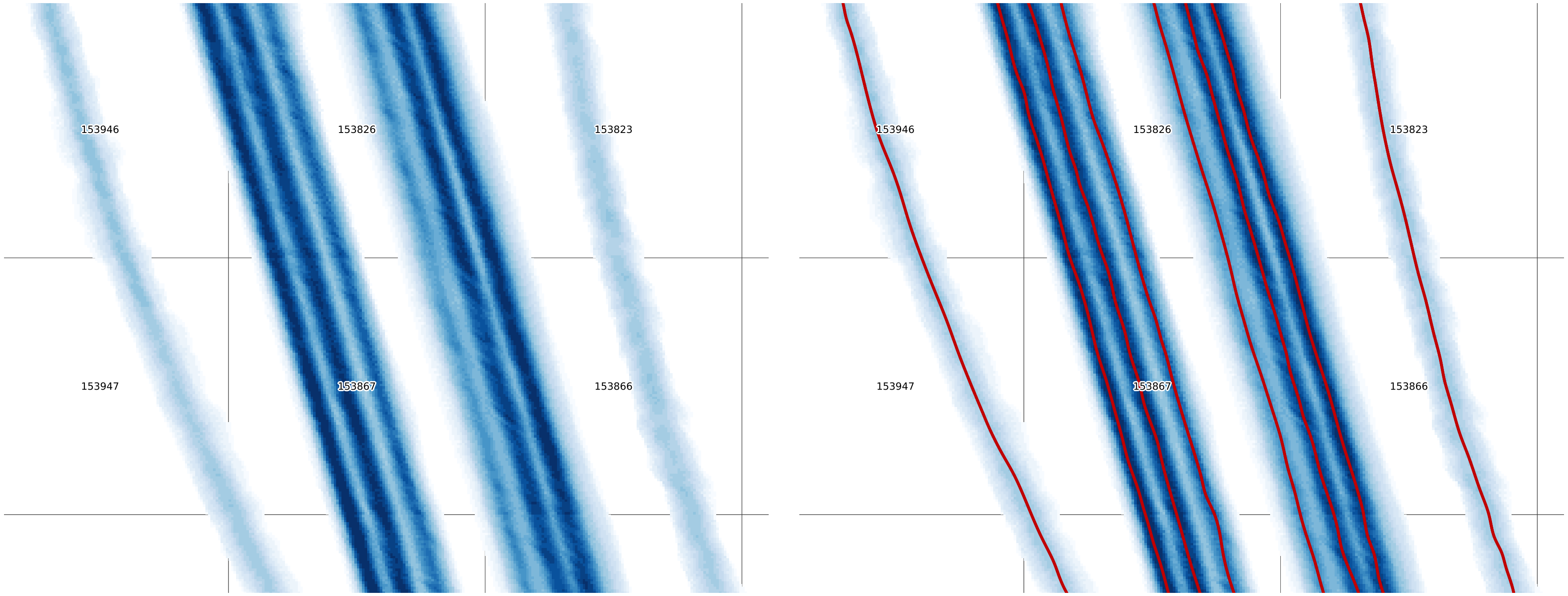
The approach to deriving lane centerline geometries is evaluated on a short excerpt of the German autobahn A9. Approximately ten trajectories per lane are used. Figure 9.9 provides a distant view and figure 9.10 a close-up view of both the uncertainty stage and the determined lane centerline geometries. It may be noticed that long hilltops are featured by the uncertainty raster cells, which are induced by the trajectories and the corresponding localization uncertainties of the vehicles. These hilltops are then algorithmically identified under the consideration of constraints, and subsequently combined to seams (section 9.9). The seams yield, even at low trajectory counts, highly precise and robust estimates of the actual lane centerline geometries. The fast convergence is, of course, positively affected by the precise vehicle localization due to GNSS postprocessing (section 3.2.2). This way, fused CVD can be, subsequently, employed to obtain highly up-to-date and precise lane centerline geometries.



(a) Uncertainty Stage

(b) Derived Lane Centerline Geometries

Figure 9.9: Derivation of lane centerline geometries (subfigure a) from intermediate results provided by the uncertainty step of double-staged areal data fusion (subfigure b). The results are based on approximately ten trajectories per lane for a short excerpt of the German Autobahn A9.



(a) Uncertainty Stage

(b) Derived Lane Centerline Geometries

Figure 9.10: Derivation of lane centerline geometries (subfigure a) from intermediate results provided by the uncertainty step of double-staged areal data fusion (subfigure b). The results are based on approximately ten trajectories per lane for a short excerpt of the German Autobahn A9.

9.11 Discussion

In this chapter, a novel, double-staged approach to the fusion of large-scale, areal CVD, such as temperatures, friction values, ambient light levels, traffic densities, signal strengths, or air qualities, has been presented in detail.

The positive characteristics of the proposed approach are mainly an incremental and scalable computation, the incorporation of measurement uncertainties and temporal weights, and the direct provision of confidence information, in contrast to existing state-of-the-art approaches. Furthermore, the evaluation of the suggested approach revealed an average runtime for a single grid cell and stage, under consideration of its 8-neighborhood, of 4.53 s for 50 000 data points and of 27.35 s for 250 000 data points. This results in an average processing time per measurement of approximately 100 μs, what can be considered as extremely feasible for large scale applications.

Additionally, a further enhancement of the suggested approach, in the case of batch inserts, deletions, and updates, which can be caused for example by a full temporal reweighting, by incorporating the FGT, has been investigated. A speed-up by more than a factor of ten for radial Gaussian BFs has been determined by utilizing a novel multi-bandwidth variant of the FIGTree approach. A further proposed extension of the FIGTree-algorithm for tackling non-radial Gaussian BFs, by rewriting the multivariate Gaussian BF to univariate ones, still leads to a speed-up by more than a factor of three. Further research concerning the batch update could focus the incorporation of non-radial Gaussian BFs in an even more efficient manner, e. g. by exploiting the correlations between the singular univariate terms.

10

Instantiation at the Vehicle- and Backend-side

This chapter presents the instantiation of the extraction, localization, and fusion pipeline for CVD[1] developed in the previous chapters, at both the vehicle- and the back-end-side, on the system level. A particular emphasis among the requirements stated in section 1.3 is put on universality. Universality means that the processing pipeline can be quickly adapted to new use cases and types of CVD. For this purpose, a universal CVD storage is proposed for the back-end-side as the foundation stone for the straightforward definition and application of universal processing functions.

Contents

10.1 Introduction

As motivated in chapter 1, the goal of the thesis is a universal, incremental, and scalable pipeline for the extraction, localization, and fusion of CVD. Those objectives have to be considered at the instantiation of the CVD-processing pipeline at the vehicle- and at the back-end-side. While the requirements "incrementality" and "scalability" mainly affect the fusion part of CVD-processing (chapters 7, 8, and 9), "universality" affects the fusion at the back-end-side as well as the extraction and localization at the vehicle-side.

This chapter presents a modular processing pipeline for both the vehicle- and back-end-side on the system level, which allows a straightforward adaptation to new use cases and types of data. The sensor data is preprocessed at the vehicle-side and,

[1] Collective Vehicle Data (CVD)

afterwards, the extracted areal data measurements and landmark observations are transmitted in a concise, whenever possible parametric manner to the back-end-side (chapter 8). This approach is motivated by the aim to limit the cellular traffic that is required for the transmission of CVD to the back-end-side. Additionally, this way the computational effort is alleviated at the back-end-side as it is partially shifted to the vehicle-side.

Figure 10.1 shows the CVD-processing pipeline from a macro-perspective. Processing at the vehicle-side, referred to as the "Acquisition-Pipeline", mainly concerns feature extraction and localization, and processing at the back-end-side, referred to as the "Fusion-Pipeline", concerns mainly the fusion of the acquired CVD. Both the vehicle- and back-end-side are designed to be fully parameterizable and reconfigurable so that they can be rapidly adapted to new use cases and types of CVD.

Sections 10.2 and 10.3 present details of the vehicle-side and back-end-side of the processing pipeline, respectively. Section 10.4 is devoted to the hierarchical template-based data storage. Section 10.5 gives information about the software frameworks used for implementation.

10.2 Processing at the Vehicle-side

The vehicle-side entity of the hybrid CVD-processing pipeline is instantiated by utilizing ADTF[2], a framework for the fast, filter-based prototyping of automotive applications. ADTF provides a runtime environment and basic filters, such as for accessing the CAN[3] and FlexRay buses, which are commonly used within modern vehicles. Furthermore, ADTF can be extended by own filters written in `C++` or `Python`. ADTF is not intended for use within series production. However, the concepts described in the following are intentionally kept general so that they can be easily transferred to series-grade ECU[4]-implementations.

For evaluation purposes, an Audi A6 has been equipped with an industry-grade CarPC running i. a. ADTF and featuring multiple CAN and FlexRay interfaces. Furthermore, the on-board localization has been replaced by a commercially viable, single-frequency `u-blox 6` GNSS[5]-receiver, which additionally allows to acquire GNSS-raw-data at a 10 Hz rate. For visual clarity, a slightly simplified test-setup is presented in figure 10.2.

Figure 10.3 visualizes the first entity of the hybrid CVD-processing pipeline from the micro-perspective. It consists of four major modules:

[2] Automotive Data and Time-triggered Framework (ADTF)

[3] Controller Area Network (CAN)

[4] Electronic Control Unit (ECU)

[5] Global Navigation Satellite System (GNSS)

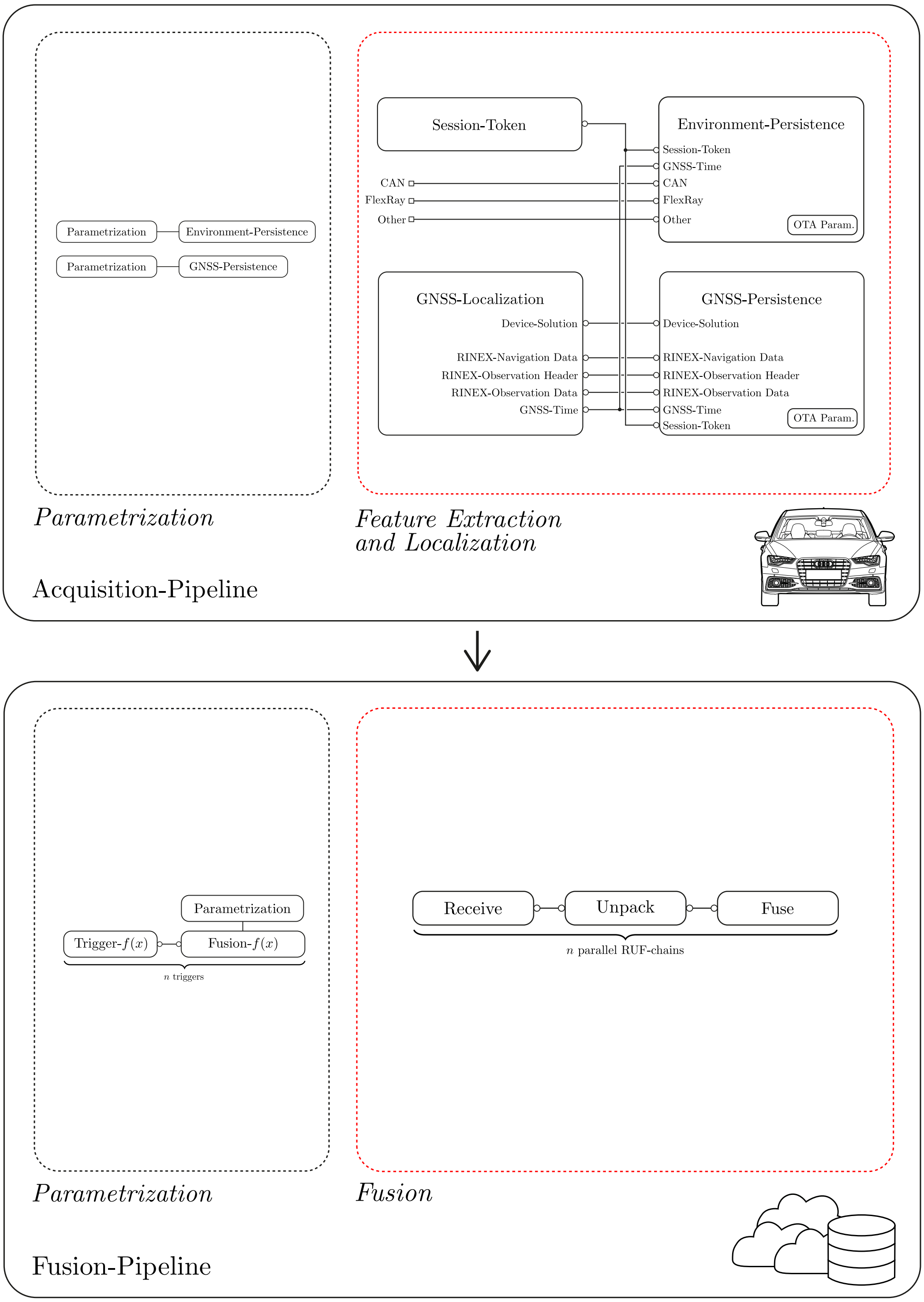


Figure 10.1: The hybrid CVD processing pipeline visualized from the macro-perspective for both the vehicle-side, referred to as the "Acquisition-Pipeline", and back-end-side, referred to as the "Fusion-Pipeline". Both the extraction and localization of CVD at the vehicle-side and the fusion of CVD at the back-end-side (right-hand side of the figure) can be rapidly adapted to new use cases and types of CVD by means of reconfiguration and parametrization (left-hand side of the figure).

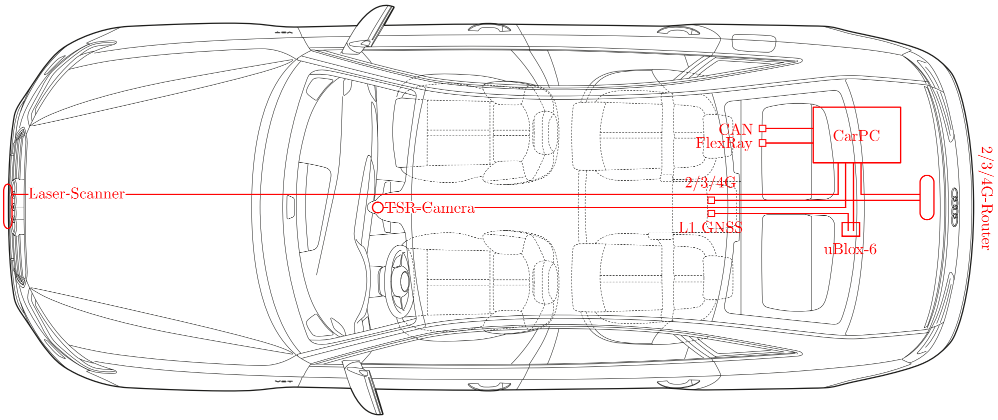


Figure 10.2: Simplified test-setup at the vehicle-side.

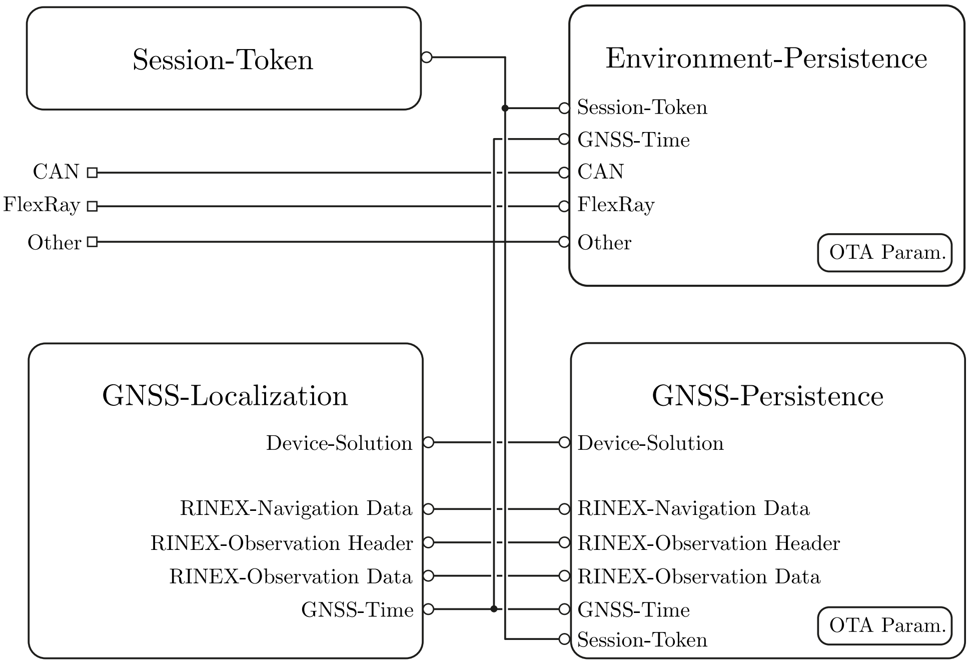


Figure 10.3: The hybrid CVD-processing pipeline at the vehicle-side, shown from the macro-perspective. It consists of four main modules: a module for determining a unique session-token, a GNSS-localization module, a GNSS-persistence module, and an environment-persistence module.

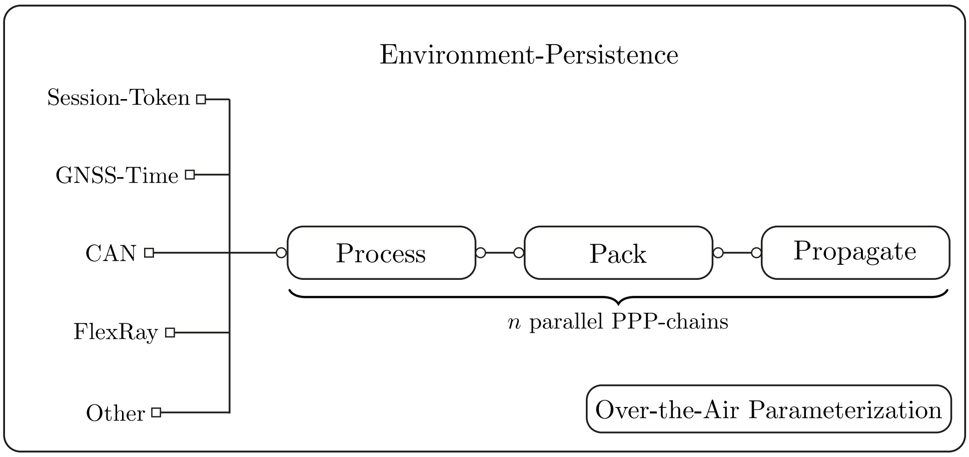


Figure 10.4: The "Environment-Persistence" module shown from the micro-perspective. It consists of multiple PPP-chains for processing, packing, and propagating environmental sensor data from the vehicle- to the back-end-side. This module can be parametrized over-the-air so that it can be easily adapted to new use cases and types of CVD.

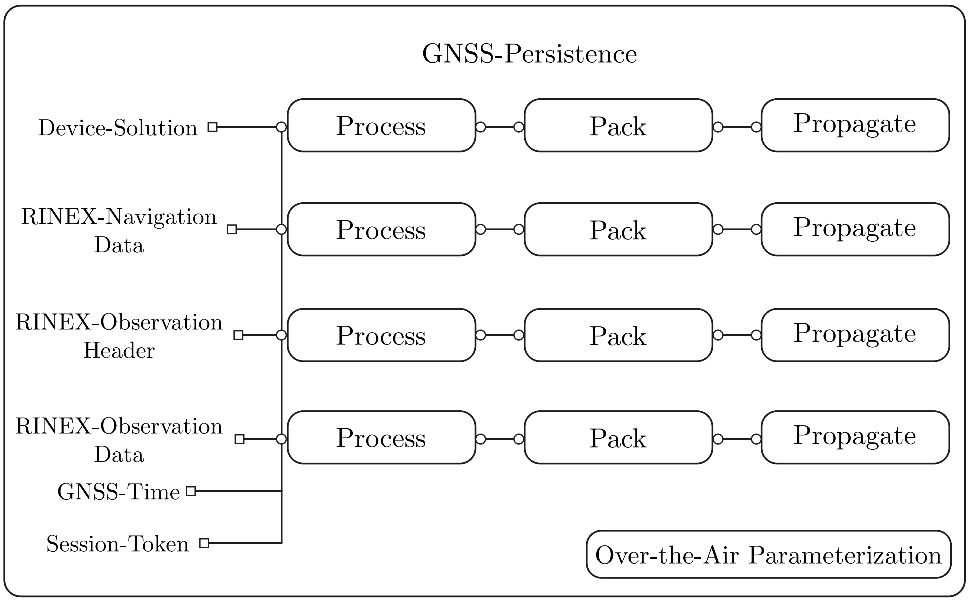


Figure 10.5: The "GNSS-Persistence" module shown from the micro-perspective. The module consists of four PPP-chains for processing, packing, and propagating the online solution determined by the GNSS-receiver and the corresponding raw data from the vehicle- to the back-end-side. This module is parametrized over-the-air so that it can be easily adapted to new use cases.

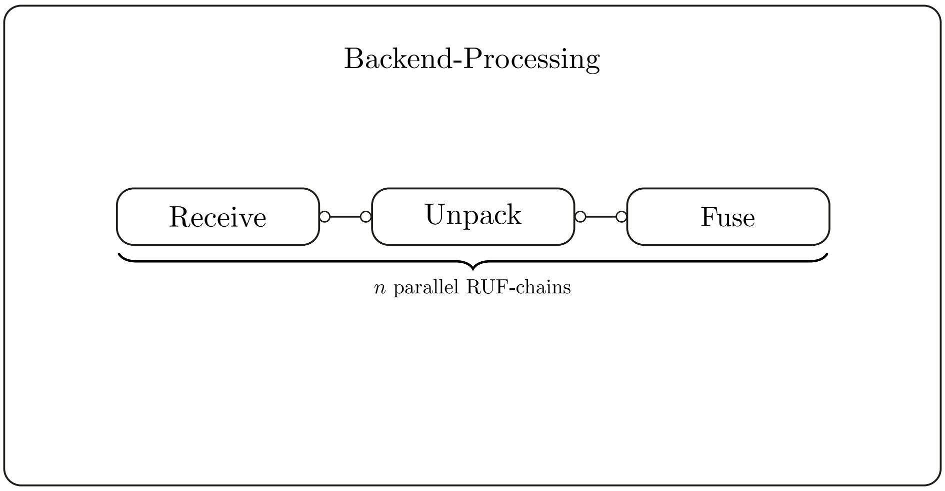


Figure 10.6: The RUF-chains are intended for receiving, unpacking, and fusing diverse types of CVD in a parallel manner at the back-end-side.

- a “Session-Token” module for requesting a unique session id from the back-end,
- a “GNSS-Localization” module providing a solution determined by the GNSS-receiver and the corresponding GNSS-raw-data intended for its postprocessing at the back-end-side (section 3.2.2),
- and two modules, the “GNSS-Persistence“ module and the “Environment-Persistence” module, which consist of multiple sub-modules responsible for localization and environment persistence.

The “Environment-Persistence” module (figure 10.4) consists of n parallel PPP[6]-chains, of which each is responsible for preprocessing, packing, and propagating one sensor datum or a group thereof simultaneously. This way the suggested hybrid CVD-processing pipeline can be rapidly adapted for new use cases and types of data, as it is based on standardized modules.

The “GNSS-Persistence” module (figure 10.5) consists of four parallel PPP-chains which are responsible for preprocessing, packing, and propagating the acquired localization data. One PPP-chain is persisting the online device solution. The three other PPP-chains are handling the GNSS-raw-data, which is subsequently utilized for determining a more precise solution to GNSS-based localization via postprocessing (section 3.2.2) at the back-end-side.

[6] Process-Pack-Propagate (PPP)

10.3 Processing at the Backend-side

The back-end-side of the CVD-processing pipeline is instantiated using PostgreSQL and PostGIS. Figure 10.6 shows the processing at the back-end-side. The first step of the RUF[7]-chains is receiving a certain type or a group of data and is followed by an unpacking step and a final fusion step.

For the actual data processing a custom database extension for PostgreSQL, named libAFD2, has been developed. The libAFD2 consists of the modules `mod_areal`, `mod_attenuation`, `mod_clustering`, `mod_core`, `mod_gnss`, `mod_landmarks`, and `mod_seam`.

`mod_core` provides basic functionality to all other modules, `mod_attenuation` incorporates temporal weighting, `mod_clustering` data association, `mod_gnss` the postprocessing of raw GNSS-data, `mod_landmarks` the fusion of arbitrary point-shaped and complex landmark observations, `mod_areal` the fusion of areal data, and `mod_seam` the determination of lane centerline geometries.

10.4 Hierarchical Template-based Data Storage

A common approach to storing CVD at the back-end-side is using self-contained tables for each distinct data type. Unfortunately, this approach does not exploit the similarities between different types of CVD and queries targeting certain classes of CVD are, therefore, unnecessarily difficult and, hence, prone to errors. Because of its benefits, a novel, hierarchical template-based storage scheme for CVD is used:

Definition 10.1: Hierarchical Template-based Storage Scheme

Template: A blueprint for a table scheme. Multiple instantiations of a template can be utilized to store data of a similar kind.

Hierarchical Template: Templates can be combined by inheritance to more specialized / complex ones. Inheritance can be applied in a cascaded manner. This means that combined templates can be again and again extended to suit the requirements of more specialized / complex data. The hierarchy of templates is induced by the order of inheritance.

Hierarchical Template-based Storage Scheme: A storage scheme that makes use of templates that are combined in a cascaded manner.

The hierarchical template-based storage scheme is achieved by exploiting the inheritance feature of modern databases, such as PostgreSQL. The storage scheme uses the possibility of formulating database queries in a way that they either target a singular

[7] Receive-Unpack-Fuse (RUF)

table within the inheritance hierarchy, or that they also include the inheriting child tables. This way queries based on classes of CVD, instead of their sole types, become possible.

For example, by exploiting the inheritance hierarchy, different types of complex landmarks can be utilized in a straightforward manner as additional point-shaped landmarks (point-shaped landmarks are defined to constitute a subclass of complex landmarks, cf. chapter 8). This is helpful since additional point-shaped landmarks can be of interest for improving the vehicle localization.

Furthermore, universal processing functions for different types of CVD are specified by requiring the target table of a function to inherit from certain table templates.

Consistency is assured by requiring that changes applied to template tables have also to be applied to all tables that inherit from them. This is a crucial feature for large-scale systems.

Scalability is achieved at a macro-level by utilizing PL/Proxy, a PostgreSQL extension developed by Skype, which allows to scale SQL[8]-queries to multiple database instances.

An introductory instance of the hierarchical template-based CVD storage scheme for the example of friction data is provided in figure 10.7. It is conceptually based on three storage layers, “root”, “refinement”, and “instantiation”. The suggested storage scheme has the property that all types of areal and point-shaped/complex landmark data are defined to inherit from the root reports template of the root layer. Furthermore, all types of vehicle traces, such as GNSS-position and -covariances, are defined to inherit from the root traces template of the root layer. Both root templates are specialized to match the particular needs imposed by vehicle friction reports of the refinement layer. Finally, three explicit instantiations are derived within the special example to store the (referenced) friction reports and the corresponding vehicle position traces on the layer of instantiations. This scheme can be applied throughout every sensor data item that needs to be persisted and/or processed at the fusion center.

A more advanced instance of the hierarchical template-based CVD storage scheme, although still simplified due to visual clarity, is shown in figure 10.8. It visualizes the representation of the two introduced categories of CVD, areal and point-shaped/complex landmark data (chapters 7, 8, and 9). It can be noticed that the definition of point-shaped landmark data as a subclass of complex landmark data is also reflected by this storage scheme. Furthermore, the templates are refined to fulfill the storage requirements of diverse types of CVD. The template for clustered reports

[8] Structured Query Language (SQL)

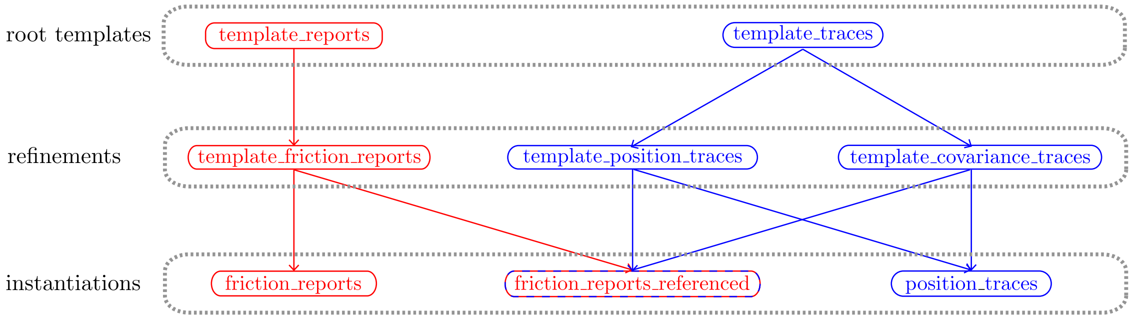


Figure 10.7: An introductory instance of the hierarchical template-based CVD-storage using the example of friction data. Additionally, the three layers "root", "refinement", and "instantiation" of the storage scheme are highlighted.

demonstrates one of the many capabilities to refine a given template for specific use cases.

10.5 Implementation

All modules of the back-end-side have been written in `C/C++` because of performance reasons, while glue code is usually written in PL/pgSQL, a procedural SQL-language, because of adaptability reasons. Furthermore PL/Proxy, a PostgreSQL extension developed by Skype which allows to scale SQL-queries to multiple database instances is used.

The vehicle-side mainly incorporates ADTF, Qt and RTKLIB as frameworks. The back-end-side additionally involves (except of ADTF) Eigen3, Boost, GDAL, FIGTree[9] and Qhull.

The software of the back-end processing-side consists of approximately 70.000 LOC[10], while the software of the vehicle-side consists of 5.000 LOC, both without including incorporated frameworks. Figure 10.9 shows the size of the individual libAFD2-modules in LOC.

10.6 Discussion

In this chapter, a hybrid processing pipeline for CVD has been presented which is instantiated at both the vehicle- and back-end-side. It has been designed in a way that it can be rapidly adapted to new use cases and types of CVD. The underlying concepts have not only been theoretically elaborated, but also implemented and evaluated on different real-world scenarios.

[9] Fast Improved Gauss Transform with Tree Data Structure (FIGTree)
[10] Lines of Code (LOC)

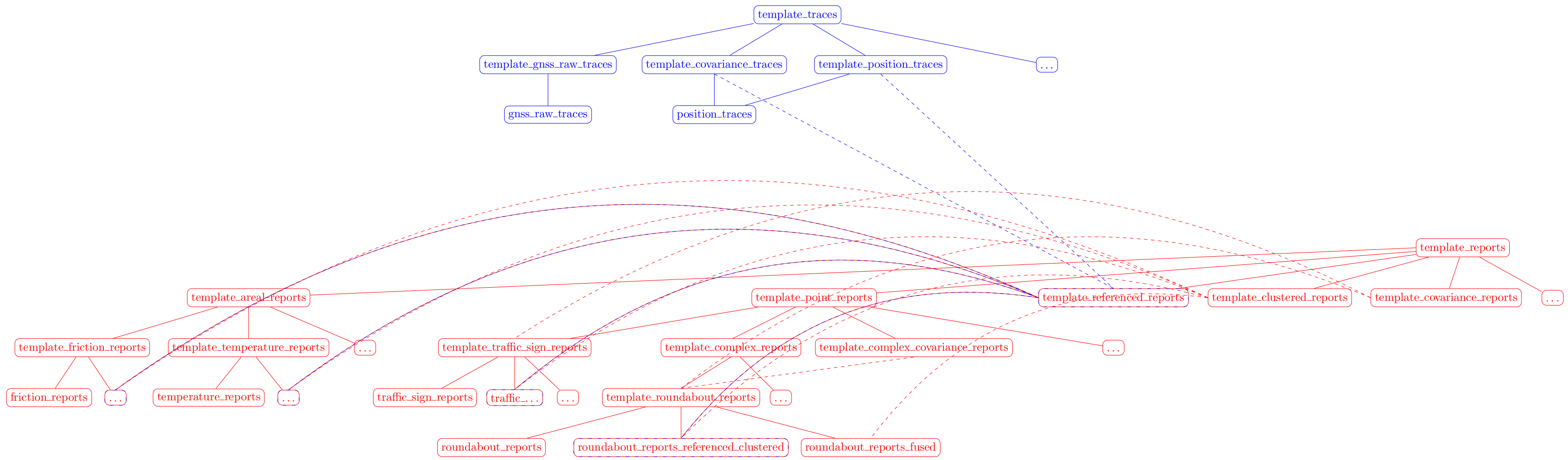


Figure 10.8: A more advanced instance of a hierarchical template-based storage for multiple types of CVD. It incorporates the distinction between areal and point-shaped/complex landmark data and their diverse refinements.

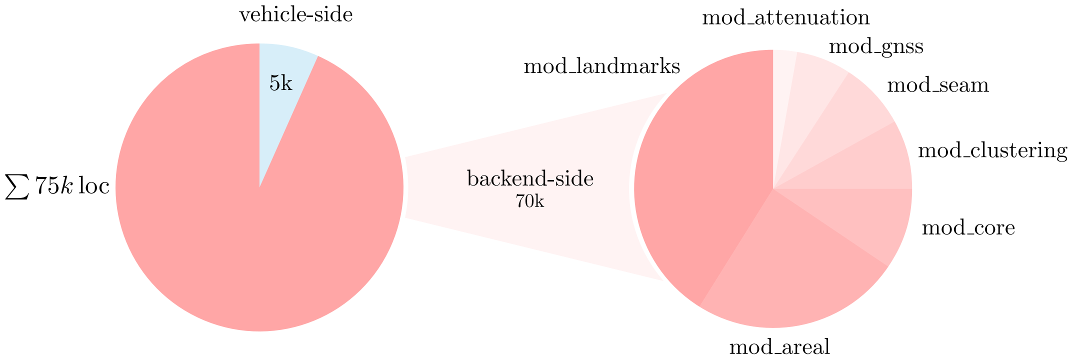


Figure 10.9: LOC count of the extraction, localization, and fusion pipeline for CVD for the vehicle- and back-end-side. Additionally, the diagram shows the LOC for the modules of the libAFD2 at the back-end-side.

The modularity of the design positively influences the simplicity of adaptation and the robustness, as common well-tested modules can be reutilized and reparametrized on-demand. Preprocessing and transformation of CVD into a concise parametric representation already at the vehicle-side has been shown to alleviate both the computational effort at the back-end-side and the required amount of cellular traffic. This can be assumed as crucial for large-scale applications.

Furthermore, the data storage is crucial for a generalized data processing. The requirement of supporting multiple data types is met by representing the similarities between different types of CVD by hierarchical templates instead of self-contained tables. This needs, of course, some additional initial effort. However, it may rapidly pay off by reduced subsequent maintenance and adaptation costs.

As an outlook, the suggested pipeline for extraction, localization, and fusion of CVD might be extended by an intelligent and demand-oriented acquisition of CVD which is triggered by the back-end-side. For this purpose, the ideas suggested by Ayala et al. [2] and Kerner et al. [66] could be incorporated and further refined. Additionally, the CVD storage could be further extended, for instance by unsupervised spatiotemporal partitioning based on approaches to multi-objective optimization.

11

Conclusions and Future Work

This chapter summarizes the main achievements of the thesis with respect to its main topics, extraction, localization, and fusion of CVD[1], and presents the main insights gained. Furthermore, possibilities of further improvements and extensions of the approaches are outlined.

Contents

11.1 Conclusions

This thesis has considered the topics "extraction", "localization", and "fusion" of CVD. The focus has been on universal, scalable, and incremental approaches which are additionally able to deal with temporal dynamics.

Since the precise vehicle localization is a foundation stone of an accurate fusion of CVD, it has been initially treated in detail. A series-tractable solution based upon the postprocessing of GNSS[2]-raw-data via PPP[3] at the back-end-side has been proposed for this purpose (section 3.2.2). An experimental analysis of the solution has shown a standard deviation of approximately 0.98 m instead of 2.28 m at open-sky conditions, what can be considered as a remarkable improvement.

Furthermore, general and adaptive approaches to scalability (section 4.2) by global submapping and to efficient deduction of temporal weights (section 4.3) for employment prior to, or within, the actual CVD fusion algorithms have been developed in the course of the thesis.

[1] Collective Vehicle Data (CVD)
[2] Global Navigation Satellite System (GNSS)
[3] Precise Point Positioning (PPP)

Subsequently, state-of-the-art approaches to non-Bayesian data association, ICNN[4], SCNN[5], and JCBB[6], have been examined in detail and extended to clustering ones and enhanced concerning their efficiency and robustness (chapter 6). Incorrect data associations are usually the reason for diverged fusion algorithms and spurious landmark estimates. Therefore, the precision of the clustering is strictly favored over the recall for the use-case of fusing CVD. It is expected that the amount of data provided by the vehicle fleet is more than sufficient for a precise fusion, and the clustering can be selective with respect to the input data. For this purpose, an impactful modification of JCBB which incorporates a generalization of BB[7], the so-called GBB[8], in combination with BFS[9], has been developed. This modification transforms JCBB into a robust multi-hypothesis clustering algorithm, as it is this way able to dynamically update the landmark map. Additionally, in the case of computational resource limitations or real-time constraints, a multi-objective cost-function for the selection of the η most promising hypotheses at each layer of the interpretation tree has been introduced. The weights of the cost function used have been determined by applying an optimization algorithm based on labeled ground truth data. Furthermore, since the JC[10] criterion is very restrictive in comparison to IC[11], SCNN/JCBB and their extensions ESCNN[12]/EJCBB[13] provide a lesser recall than ICNN and its extension EICNN[14], respectively. However, both new variants usually achieve more robust results.

The fusion of point-shaped landmark data has been performed within this thesis by well-known algorithms from the research fields of sensor data fusion/robotics, in particular (F)CI[15] and SLAM[16]. From the automotive point-of-view, aspects such as the incorporation of absolute instead of relative vehicle positions, scalability, incrementality, the provision of estimation uncertainties, and the compensation of temporal dynamics have been identified as essential. For this purposes, extended versions of EKF[17]- and BA[18]-based approaches to SLAM have been derived. The approaches have been experimentally evaluated and opposed to each other, with

[4] Individual Compatibility Nearest Neighbor (ICNN)
[5] Sequential Compatibility Nearest Neighbor (SCNN)
[6] Joint Compatibility Branch and Bound (JCBB)
[7] Branch and Bound (BB)
[8] General Branch and Bound (GBB)
[9] Breadth-First Search (BFS)
[10] Joint Compatibility (JC)
[11] Individual Compatibility (IC)
[12] Extended Sequential Compatibility Nearest Neighbor (ESCNN)
[13] Extended Joint Compatibility Branch and Bound (EJCBB)
[14] Extended Individual Compatibility Nearest Neighbor (EICNN)
[15] Covariance Intersection (CI)
[16] Simultaneous Localization and Mapping (SLAM)
[17] Extended Kalman Filter (EKF)
[18] Bundle Adjustment (BA)

respect to attributes essential from the automotive point-of-view, such as accuracy, robustness, and computational efficiency (section 7.5). The evaluation shows that overall good results were achieved by BA-based Full-SLAM which has been solved after linearization in a computationally efficient manner by means of QR[19]. Because approaches to Full-SLAM principally do not involve marginalization they are known to be more robust to outliers, what has also be confirmed by the experimental evaluation. Furthermore, it has turned out that the computational load of Full-SLAM is considerably reduced by linearization and by utilizing efficient methods for sparse matrix factorizations.

The fusion of point-shaped landmarks has been subsequently extended to an approach to fusion of complex landmarks (chapter 8). This has been motivated by the fact that complex landmarks, such as crosswalks, crossroads, roundabouts, traffic islands, sign gantries, or lane markings, are widespread in the automotive context. Representing complex landmarks by point-shaped descriptions, such as just by their center point, implies a considerable loss of useful information, and the incorporation of secondary attributes is, therefore, highly desirable. Challenges of complex landmarks are the often just partial observability due to sensor limitations or occlusions, and the varying number of degrees of freedom of certain complex landmarks, such as crossroads or roundabouts which can have an arbitrary number of junctions. For universal modeling of complex landmarks, a DAG[20]-based description has been introduced within this work. This parametric description by ADG[21]s allows, on the one hand, to reduce the cellular traffic for the transmission of complex landmark observations, compared to raw sensor data, and, on the other hand, to fuse complex landmark observations in a generalized, common manner. The ADGs are highly flexible and able to represent the interdependencies between corresponding attributes. This property can be considered as essential for the fusion of complex landmarks. The ADGs can be principally used in distinct approaches to SLAM. In this thesis, EKF-based Online-SLAM and BA-based Full-SLAM have been considered. Applications to complex roundabout observations and a hybrid scenario with additional traffic sign observations show promising results. An important strength of the approach is its ability to fuse distinct kinds of complex and non-complex landmark observations simultaneously, and therefore to exploit distinct cross-landmark correlations. It has turned out that this influences the overall convergence and robustness positively. However, there are still some interesting open research questions, such as the incorporation of explicit constraints, what will be discussed in the next section.

[19] QR Factorization (QR)
[20] Directed Acyclic Graph (DAG)
[21] Attribute Dependency Graph (ADG)

The fusion of areal data can be performed by interpolation or approximation. From literature two well-studied approaches are known, the interpolation via Kriging and RBFN[22]s. According to Fazio and Roisenberg [38], both approaches can be parametrized to provide comparable results, but RBFNs are much more time-efficient. Therefore, the fusion of areal data has been tackled in this thesis by an approach inspired by RBFN interpolation. To fulfill the requirements of fusion of CVD (section 1.3) a double-staged approach to areal data fusion has been proposed which has several important advantages against RBFN interpolation. At first, the new algorithm is able to incorporate elliptic instead of radial BF[23]s. This is useful since elliptic BFs can be canonically related to GNSS-based localization. By means of an LP[24]-based approach to determining the actual BF-weights, negative or unbalanced weights can be excluded, what is a well-known problem of interpolation with Kriging/RBFNs. Furthermore, the double-staged areal data fusion allows an incremental update due to its split computation. Additionally, information about the confidence of areal data fusion can be directly derived from the first stage, the uncertainty stage. This is essential if only a-priori data of a high confidence shall be provided to the vehicle fleet. Additionally, the intermediate results gained from the uncertainty stage can be reutilized, e. g. for the derivation of lane centerline geometries which can be later on used to assign fused areal data to corresponding lanes.

Because large amounts of data need to be processed and the accumulation of Gaussians to be performed is computationally expensive, the approach to areal data fusion has been further improved in the case of a batch insertion or updating, as e. g. for temporal reweighting. For this purpose, the FGT[25] has been used which allows a more efficient accumulation of Gaussians by means of a series expansion (section 9.7). A speed-up factor of more than ten has been achieved in experiments. Unfortunately, the FGT is commonly restricted to radial BFs. However, by rewriting the multidimensional Gaussian by means of a double-sum this problem can be solved, but at expense of computational efficiency. However, still a speed-up by more than a factor of three can be observed.

The approaches theoretically derived and elaborated in this thesis have been instantiated at the vehicle- and back-end-side. For this purpose, an Audi A6 has been equipped i. a. with an automotive-grade computer which allows to receive and send data via the CAN[26] and FlexRay buses. Furthermore, the vehicle has been equipped with several additional measurement units (chapter 10). A key aspect of the instantiation was the fulfillment of the requirement of universality as stated in section 1.3.

[22] Radial Basis Function Network (RBFN)
[23] Basis Function (BF)
[24] Linear Program (LP)
[25] Fast Gauss Transform (FGT)
[26] Controller Area Network (CAN)

For this purpose, a generic, modular acquisition, preprocessing, and propagation pipeline has been incorporated at the vehicle-side, which has been implemented on the basis of the automotive, rapid-prototyping framework ADTF[27]. The modular approach incorporated is intentionally kept very general so that it can be transferred in a straightforward manner to an ECU[28] implementation. The modularity is also given at the back-end-side, for which a PostgreSQL-based processing pipeline has been utilized. Furthermore, a custom database extension, the so-called libAFD2, which consists of multiple modules for the universal processing of distinct types of CVD, has been developed as part of this work. It uses the native PostgreSQL C-interface, the so-called SPI[29], for the efficient processing of huge amounts of data. At the back-end-side, the scalability of the presented approaches has been achieved at the micro level (vertical scalability) by adaptive partitioning (section 4.2) and at the macro level (horizontal scalability) via PL/Proxy. Furthermore, a generalized data storage scheme based on hierarchical templates has been incorporated, which exploits the similarities of distinct types of vehicular sensor data. It can be considered as the foundation stone of the universal processing.

11.2 Future Work

This section presents ideas and recommendations for possible improvements of the approaches presented in this thesis.

The accuracy of the trajectories gained by postprocessing, cf. section 3.2, can be further improved by incorporating more accurate ionospheric and tropospheric correction data. Alternatively, by incorporating a dual- or multi-frequency GNSS-receiver, ionospheric/tropospheric and multipath errors might be algorithmically reduced. Furthermore, the actually incorporated GNSS-antenna could be replaced with a more multipath resistant one. Both possibilities of improvement have not been pursued here because of cost and design constraints of a series-tractable solution, which was the goal of the thesis. An extension that should be strived in any case is the incorporation of vehicle egodata in the postprocessing at the back-end-side, to compensate for GNSS shadowing and outages.

The method for determining adaptive temporal weights of section 4.3 can adjust in an unsupervised manner to the frequency of the CVD acquisition and provides this way realistic decay rates. However, there are a few parameters of the underlying data structure, such as the bin width and count, which are expected by now to be set manually according to the characteristics of the input data. As an alternative, meth-

[27] Automotive Data and Time-triggered Framework (ADTF)
[28] Electronic Control Unit (ECU)
[29] Server Programming Interface (SPI)

ods for the appropriate selection of these parameters by multi-objective optimization based on ground truth data could be explored.

The parametric representation of complex landmarks by ADGs in chapter 8 might be further extended by allowing cascaded subgraphs, so-called dynamic blocks, and the incorporation of explicit constraints. Especially the definition of explicit constraints might be useful in many cases, as restrictions, such as orthogonality, parallelism, symmetry, limited angles, and lengths, could improve the robustness in complicated scenarios, while simultaneously reducing the required observation count. The incorporation of constraints could be performed i. a. by penalization terms or by utilizing KKT[30]. The incorporation of KKT would be the more elegant and, therefore, the favored way. Possible resulting challenges could be the linearization and the subsequent derivation of the matrix notation required for efficiently solving the equation system by QR, what is expected to become even more complex than for complex landmarks without explicit constraints.

In section 9.10.4, the non-normalized KDE[31] has been analyzed for ridges which correspond to vehicle trajectories and thus to centerlines of lanes. The robustness of this approach could be further improved by incorporating possible lane markings detected by vehicles equipped with TSR[32]-cameras or lidars. Furthermore, coupled GNSS-localization can be incorporated to compensate for GNSS-shadowing and outages, as e. g. observed in urban scenarios, to improve the overall robustness.

The incorporation of the FGT has improved the computational efficiency of the double-staged approach to areal data fusion (section 9.7). A further improvement for the case of elliptic Gaussian distributions might possibly be achieved by a stronger exploitation of the structure of the double summation by a common consideration of its distinct terms.

The hierarchical template-based data storage scheme of section 10.4 could be further extended by concerning the spatiotemporal partitioning of the stored data. Furthermore, unsupervised partitioning based on a multi-objective optimization could be investigated.

An issue not treated in the thesis is the classification of road network scenarios based on the common consideration of individual, fused features. An important scenario from the automotive point of view are roadworks. Roadworks are usually not defined by a singular feature, however, they are constituted of multiple ones, such as roadwork and/or masked signs, special lane markings, delineators, or a dirty roadway. A possible roadwork scenario is presented in figure 11.1. It exhibits a detour

[30] Karush–Kuhn–Tucker Conditions (KKT)
[31] Kernel Density Estimation (KDE)
[32] Traffic Sign Recognition (TSR)

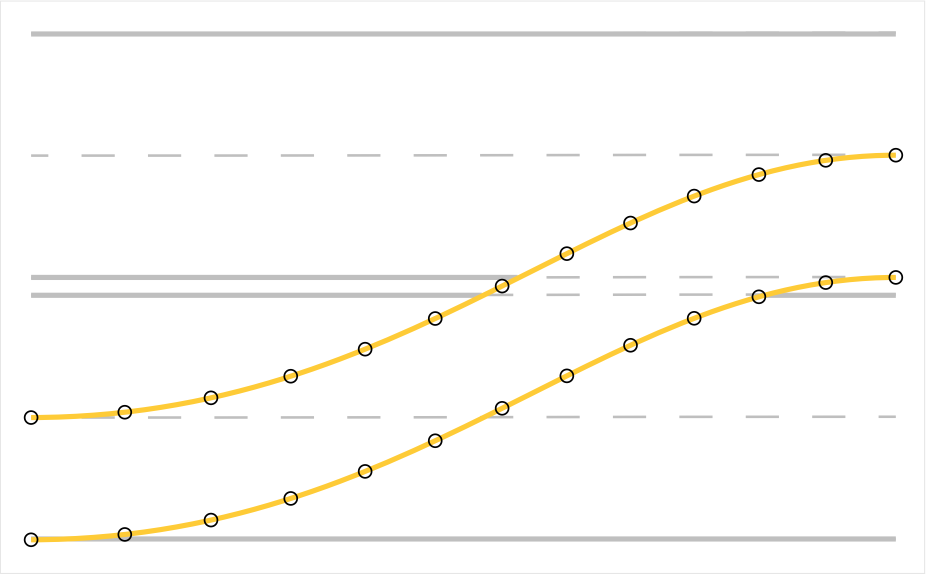


Figure 11.1: An exemplary roadwork scenario with a detour to the opposing traffic lane, yellow lane markings =, and delineators ○.

to the opposing traffic lane, yellow lane markings, and delineators. For a human it is easily recognizable as a roadwork – to detect it algorithmically can be considered as challenging, because of an extensive amount of actual manifestations. The previously mentioned features can be utilized to identify roadworks and to determine their extents so that succeeding vehicles can be informed in advance about an approaching roadwork. The precise models of roadworks can be subsequently utilized for next generation piloted, driver assistance, or comfort functions.

For the identification of possible roadworks, the following procedure is proposed. At first, potential roadwork areas should be identified, what can be done by a variation of fusion of areal data presented in chapter 9. Different types of observed roadwork features are incorporated in an areal voting mechanism. Possible roadwork areas, this means areas whose confidence is above a cut-off confidence level, can then be fused by the algorithms presented in chapters 7 and 8. The distinct fused features can be subsequently provided to classification approaches. A general classification approach would be again useful, as not only roadworks, but also entities defined by multiple features, such as play streets, or parking lots, are of interest to be mapped and identified collaboratively.

A
Appendix

A.1 Mahalanobis Distance Metric

The Mahalanobis distance [76] is a measure for the dissimilarity of two samples $\boldsymbol{a}$ and $\boldsymbol{b}$ of a multi-variate random variable with covariance $\boldsymbol{\Sigma}$:

$$\|\boldsymbol{a}, \boldsymbol{b}\|_{\boldsymbol{\Sigma}} \; = \sqrt{(\boldsymbol{a} - \boldsymbol{b})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{a} - \boldsymbol{b})}. \tag{A.1}$$

In the case that $\boldsymbol{\Sigma}$ does take the shape of an identity matrix, the Mahalanobis distance is equal to the Euclidean one.

A.2 Covariance Ellipse

The covariance ellipse (also called uncertainty/error ellipse respectively covariance/uncertainty/error ellipsoid for more than two dimensions) of a random variable $\boldsymbol{x}$ with mean $\overline{\boldsymbol{x}}$ and covariance $\boldsymbol{C}_{\boldsymbol{x},\boldsymbol{x}}$ with range c is defined as

$$\mathcal{E}_{\boldsymbol{x},c} = \left\{ \tilde{\boldsymbol{x}} \mid (\tilde{\boldsymbol{x}} - \overline{\boldsymbol{x}})^T \boldsymbol{C}_{\boldsymbol{x},\boldsymbol{x}}^{-1} (\tilde{\boldsymbol{x}} - \overline{\boldsymbol{x}}) \leqslant c^2 \right\}. \tag{A.2}$$

A.3 Linearization of Non-linear Functions

The Taylor series in one variable is defined by

$$T_\infty \; f_a(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n \tag{A.3}$$

with a being the linearization point and $f^{(n)}$ the n-th derivative of a given function $f(x)$. By only considering the 1st-order term, the Taylor series can be utilized for the linearization of non-linear functions:

$$T_1 \; f_a(x) \; \approx f(a) + f'(a) \cdot (x - a). \tag{A.4}$$

The 1st-order Taylor series in two variables is denoted by

$$T_1\ f_{a,b}(x,y) \ \approx f(a,b) + \left.\frac{\partial f(x,y)}{\partial x}\right|_{a,b}(x-a) + \left.\frac{\partial f(x,y)}{\partial y}\right|_{a,b}(y-b) \tag{A.5}$$

with (a,b) being the chosen linearization point.

Analogously, the approach can be extended to n variables.

A.4 Bounding Box of an Ellipse

An ellipse can be defined in parametric form by

$$\begin{pmatrix} x(t) \\ y(t) \end{pmatrix} = \begin{pmatrix} c_x + a \cdot \cos(t) \cdot \cos(\alpha) - b \cdot \sin(t) \cdot \sin(\alpha) \\ c_y + a \cdot \cos(t) \cdot \sin(\alpha) + b \cdot \sin(t) \cdot \cos(\alpha) \end{pmatrix} \tag{A.6}$$

with a being the length of its major axis, b the length of its minor axis, $(c_x, c_y)^T$ its center point, α its rotation around the center point, and $t \in [0, 2\pi]$ the parameter.

The horizontally and vertically extremal points of an ellipse are characterized by vertical and horizontal tangents, respectively. They can be determined by setting $x'(t)$ respectively $y'(t)$ equal to zero:

$$\begin{aligned} x'(t) \ &= c_x + a \cdot \cos(t) \cdot \cos(\alpha) - b \cdot \sin(t) \cdot \sin(\alpha) \\ &= \ - a \cdot \cos(\alpha) \cdot \sin(t) - b \cdot \sin(\alpha) \cdot \cos(t) \\ &= 0, \\ y'(t) \ &= c_y + a \cdot \cos(t) \cdot \sin(\alpha) + b \cdot \sin(t) \cdot \cos(\alpha) \\ &= \ - a \cdot \sin(\alpha) \cdot \sin(t) + b \cdot \cos(\alpha) \cdot \cos(t) \\ &= 0. \end{aligned} \tag{A.7}$$

By substituting $\cos(\alpha)$ respectively $\cos(t)$ by λ_α respectively λ_t, the derivative of the parametric ellipse equation (equation A.7) is rewritten to

$$\begin{aligned} x'(t) \ &= \ - a \cdot \lambda_\alpha \cdot \sqrt{1-\lambda_t^2} - b \cdot \sqrt{1-\lambda_\alpha^2} \cdot \lambda_t \ = \ 0, \\ y'(t) \ &= \ - a \cdot \sqrt{1-\lambda_\alpha^2} \cdot \sqrt{1-\lambda_t^2} + b \cdot \lambda_\alpha \cdot \lambda_t \ = 0 \end{aligned} \tag{A.8}$$

with

$$\begin{aligned} \cos(\alpha) &= \lambda_\alpha, \\ \cos(t) &= \lambda_t, \\ \sin(\alpha) &= \sqrt{1-\lambda_\alpha^2}, \\ \sin(t) &= \sqrt{1-\lambda_t^2}. \end{aligned}$$

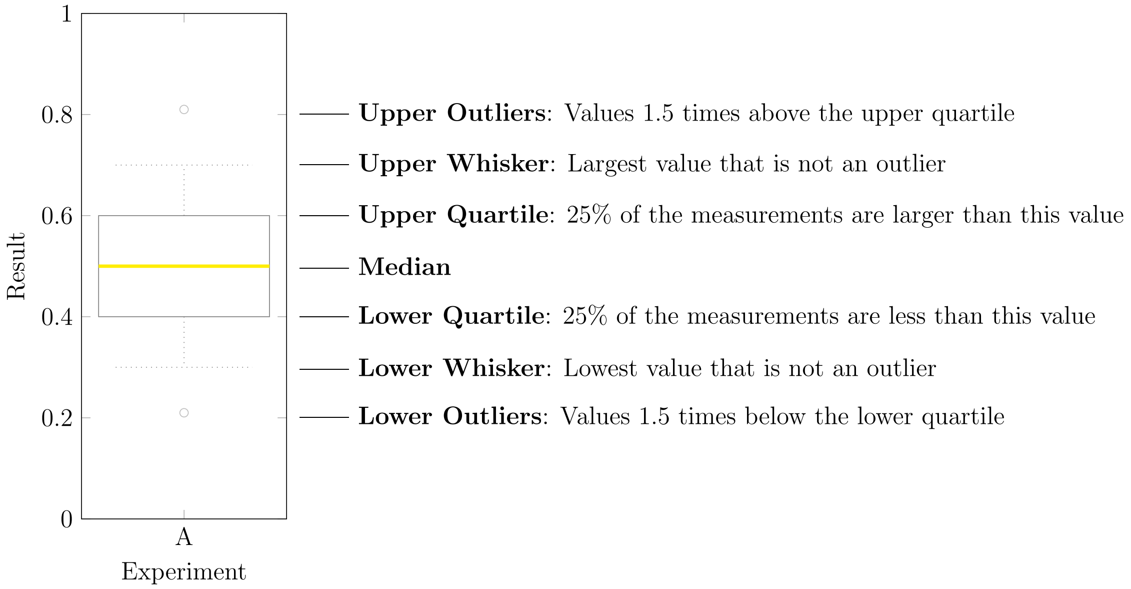


Figure A.1: An exemplary box-whisker-plot that does feature common indicators, such as the median, upper and lower quartiles, uppermost and lowest values, and upper and lower outliers.

The intermediate solutions are then obtained as

$$\begin{aligned}\lambda_t^{1,\,2} &= \pm\frac{a\cdot\lambda_\alpha}{\sqrt{a^2\cdot\lambda_\alpha^2-\lambda_\alpha^2\cdot b^2+b^2}},\\ \lambda_t^{3,\,4} &= \pm\frac{a\cdot\sqrt{\lambda_\alpha^2-1}}{\sqrt{a^2\cdot\lambda_\alpha^2-\lambda_\alpha^2\cdot b^2-a^2}}.\end{aligned} \tag{A.9}$$

After resubstituting λ_α respectively λ_t by $\cos(\alpha)$ respectively $\cos(t)$ (equation A.9), the final solutions are

$$\begin{aligned}t_{1,\,2} &= \cos^{-1}\left(\pm\lambda_t^{1,\,2}\right),\\ t_{3,\,4} &= \cos^{-1}\left(\pm\lambda_t^{3,\,4}\right).\end{aligned} \tag{A.10}$$

The bounding box is obtained from the minimum and maximum x- and y-coordinates of the ellipse points with those parameters.

A.5 Box-Whisker-Plot

A box-whisker-plot (figure A.1) is an efficient approach for simultaneously providing common indicators, such as the median, upper and lower quartiles, uppermost and lowest values, and upper and lower outliers, for a given set of measurements. While the box is unambiguously defined within literature, the whiskers, and outliers are not [77]. The whiskers are defined within this work to indicate the uppermost and the lowest non-outlier value. Furthermore, outliers are defined to be 1.5 times below the lower quartile, respectively 1.5 times above the upper quartile. In the case that multiple experiments are conducted to gain more robust results, the resulting indicators of the averaged box-whisker-plot are defined to be the averages of the distinct values. However, upper and lower outliers are an exception, as they are not averaged, and their sets are instead concatenated.

Glossary

Adaptive Cruise Control

ACC is a commonly radar- and/or camera-based driver assistance system that tracks surrounding vehicles and automatically adjusts the velocity of the own vehicle.

Ambiguity Resolution

In the context of GNSS-based localization, AR denotes a class of methods for resolving ambiguities that could lead to multiple feasible position estimates.

Artificial Neural Network

ANNs aim to mimic the reasoning of organisms by incorporating interconnected artificial neurons. These neurons are able to interact with each other by exchanging values. By means of supervised learning, ANNs can be trained to weight the received values appropriately, and to provide the sought-for results at the output layer. Commonly, ANNs are subdivided into three layers of neurons: the input layer, the processing (hidden layer), and the output layer.

Attribute Concatenation Graph

An ACG is induced by all those vertices for which a directed path exists to the vertex which calculates the AA. This vertex is the root of the tree. An ACG is denoted by 🔗 $(\boldsymbol{a})$ where $\boldsymbol{a}$ is the root.

Attribute Dependency Function

Every vertex of an ADG represents an ADF and is labeled with a corresponding elementary LA. It transforms the corresponding input LA under consideration of the input AA(s) and provides the result as an output AA.

Attribute Dependency Graph

An ADG denotes a concise, parametric description of complex landmarks by a DAG. This way, the interdependencies between attributes can be appropriately modeled and incorporated within subsequent data association or fusion algorithms.

Attribute Dependency Tree

A special case of an ADG whose vertices have at most one ingoing edge. For ADTs the ACGs become paths.

Automotive Data and Time-Triggered Framework

ADTF is a framework that is intended for the rapid prototyping of automotive applications by incorporating a modular, filter-based approach to data processing. In so doing, predefined filters, such as for the communication via CAN/FlexRay, or custom ones may be incorporated.

Auxiliary Attribute

An AA denotes one of the inputs or the output of an ADF.

Basis Function

BFs are well-known fundamental functions, such as Gaussian, Cauchy, or Epanechnikov, that are utilized within other functions or algorithms.

Branch and Bound

BB is an approach to a more efficient traversing of the exponential solution space of distinct optimization problems instead of enumerating all possible candidate solutions. For this purpose, BB splits the problem space recursively into sub-problem spaces, the so-called branches. All new branches are checked against the upper bound of the so far best-known solution and only followed up when they can principally lead to a better solution. Otherwise, branches are bounded, it means discarded. This way, BB can provide a feasible solution usually more efficiently than by exhaustive search. In rare cases BB can still degenerate to an exhaustive search.

Breadth-First Search

BFS is a method for the traversing of tree and graph data structures. BFS explores nodes by beginning at the root node and proceeding with the direct neighbor nodes. Subsequently, the method moves on to the next level within the node hierarchy. This procedure is repeated until all nodes have been visited.

Bundle Adjustment

BA is an approach for simultaneously refining the geometry of a scene acquired by a sensor and its relative motion. Additionally, the estimation of the exact sensor parameters is often incorporated within the refinement process. BA is usually expressed as a non-linear LS problem, whose solution is subsequently approached by a non-linear solver, such as LM, or after linearization by a linear one.

Collective Vehicle Data

CVD refers to all kinds of information that is continuously or on demand acquired by a vehicle fleet. For this purpose, the data of principally all vehicular sensors can be utilized. By transmitting this data to a central instance, extensive information about the road network and its current state can be derived. This information can be subsequently backpropagated to the vehicle fleet and incorporated into next generation driver assistance, comfort, or piloted functions.

Column Approximate Minimum Degree Ordering Algorithm

COLAMD is an approach for permuting a given matrix in a way that sparser factorizations, this means with less fill in, are achieved.

Compressed Receiver Independent Exchange Format

C-RINEX is a compressed version of the RINEX file format.

Constant Acceleration

CA denotes a linear vehicular motion model that does assume a constant acceleration between two given time intervals.

Constant Curvature and Acceleration

CCA denotes a curvilinear vehicular motion model that does assume a constant curvature and acceleration between two given time intervals.

Constant Steering Angle and Velocity

CSAV denotes a curvilinear vehicular motion model that does assume a constant steering angle and velocity between two given time intervals.

Constant Turn Rate and Acceleration

CTRA denotes a curvilinear vehicular motion model that does assume a constant turn rate and acceleration between two given time intervals.

Constant Turn Rate and Velocity

CTRV denotes a curvilinear vehicular motion model that does assume a constant turn rate and velocity between two given time intervals.

Constant Velocity

CV denotes a linear vehicular motion model that does assume a constant velocity between two given time intervals.

Controller Area Network

CAN is a message-based protocol that is commonly incorporated within vehicles for the communication between distinct ECUs. The protocol was officially released in 1986.

Covariance Intersection

CI is an algorithm for the fusion of consistent random variables with an unknown cross-correlation. In so doing, a user-defined criterion is incorporated, e. g. the minimization of the determinant or trace of the resulting fused covariance matrix.

Covariance Union

CU is an extension of CI. It allows to additionally incorporate inconsistent measurements. In the case of consistent ones, the approach is equal to CI.

Cumulative Density Function

The CDF provides information about the probability with which the PDF of a random variable will take a value less than or equal to x.

Density-Based Spatial Clustering of Applications with Noise

DBSCAN is a density-based clustering algorithm. The output of the clustering algorithm is mainly defined by two parameters, the search radius ϵ and the number of required measurements κ within the search radius ϵ. DBSCAN is able of classifying measurements as noise. This is highly beneficial in the case of erroneous measurements. DBSCAN features an average runtime complexity of $O(n \log n)$ when tree-based index data structures, such as R-trees, are incorporated. DBSCAN is one of the most common clustering algorithms.

Depth-First Search

DFS is a method for the traversing of tree and graph data structures. DFS explores nodes by beginning with a root node and proceeding within the node hierarchy until a leaf node is encountered. Subsequently, the method steps one level back and continues the DFS with a neighboring node. This procedure is repeated until all nodes have been visited.

Differential Global Positioning System

DGPS utilizes a reference station to improve the accuracy of GPS. By knowing its exact position a reference station is able to provide delta information (difference between the exact and the currently measured position). This delta information can be utilized by GPS receivers to compensate for satellite ephemeris errors/clock drifts and ionospheric/tropospheric distortions.

Digital Audio Broadcasting

DAB denotes a digital radio transmission standard. It is utilized in several countries within Europe and Asia. Furthermore, DAB does allow to transmit arbitrary additional data, such as GNSS correction data, parallel to the broadcasted audio.

Directed Acyclic Graph

A DAG is a directed graph without cycles.

Electronic Control Unit

Within the automotive context, the term ECU does refer to embedded systems that control or regulate parts of a vehicle.

Extended Individual Compatibility Nearest Neighbor

An extension of ICNN that aims to increase the overall robustness/efficiency and allows the consideration of constraints.

Extended Joint Compatibility Branch and Bound

An extension of JCBB that aims to increase the overall robustness/efficiency and allows the consideration of constraints.

Extended Kalman Filter

The EKF is an extension of the KF that can handle non-linear models by means of linearization. Both the EKF and the KF are designed to fuse measurements with known cross-correlations.

Extended Sequential Compatibility Nearest Neighbor

An extension of SCNN that aims to increase the overall robustness/efficiency and allows the consideration of constraints.

Fast Covariance Intersection

FCI is a closed form solution to CI and usually provides a better computational efficiency.

Fast Gauss Transform

The FGT is a procedure intended for the fast batch summation of Gaussians by incorporating Taylor and Hermite expansions for the source and target points.

Fast Improved Gauss Transform with Tree Data Structure

The FIGTree denotes an improved FGT.

FlexRay

The bus can be considered as a faster alternative to the CAN bus. It does provide both static and dynamic segments by means of TDMA. Static segments are preallocated, therefore, stronger real-time guarantees are assured than by the CAN bus. Furthermore, dynamic segments allow event-triggered messages and can be, therefore, regarded as CAN-like messages.

General Branch and Bound

GBB aims to provide a more general formulation of the BB search paradigm. Differences between BB and GBB are i. a. that GBB allows a more generic pruning than the comparison of lower and upper bounds by ordinary BB and the utilization of domain-specific knowledge. The introduced generalizations allow that common search procedures, such as e. g. A*, can be formulated as a special case of GBB.

Global Navigation Satellite System

GNSS is a satellite-aided global localization service grounded upon multilateration. For this purpose, precise timestamps that are obtained from atomic clocks are emitted by each satellite. Subsequently, these timestamps are received by GNSS receivers, and the position thereof is derived from the measured signal runtimes, the so-called pseudoranges. Known manifestations of GNSS are: GPS, GLONASS, BeiDou, and Galileo.

Global Positioning System

GPS is a global navigation satellite system developed by the U. S. Department of Defense, which provides a localization deviation of less than 10 m for civil usage.

Improved Fast Gauss Transform

The IFGT denotes an improved FGT.

Individual Compatibility

IC is a χ^2-based test that is intended for determining if an individual sample is probable of originating from a given distribution.

Individual Compatibility Nearest Neighbor

A data association algorithm based on the IC-test.

Inertial Measurement Unit

An IMU is a device that is able to measure the relative motion of a vehicle by utilizing a combination of accelerometers, gyroscopes, and magnetometers. A combination of relative inertial and global GNSS-based localization is highly sensible in order to compensate for GNSS outages and deviations.

International GNSS Service

IGS is a service of the International Association of Geodesy that provides GNSS correction data, in particular, more accurate ephemerids and clocks when compared to the broadcast ones.

Joint Compatibility

JC is a χ^2-based test that is intended for determining if a set of sample points is probable of jointly originating from a given distribution.

Joint Compatibility Branch and Bound

A BB-based data association algorithm that incorporates both the IC- and JC-test.

Kalman Filter

The KF is a filter that allows to fuse multiple measurements based on linear models with known cross-correlations in an ideal manner.

Karush–Kuhn–Tucker Conditions

The KKT conditions can be considered as a generalization of the Lagrange multipliers. While the Lagrange multipliers allow solving optimization problems under the consideration of equality constraints, the KKT conditions allow to also consider inequality ones.

Kernel Density Estimation

KDE is a non-parametric approach for determining the PDF of a random variable. The PDF is approximated by a sum of BFs, such as Gaussian, Cauchy, or Epanechnikov.

Landmark Attribute

A LA denotes an elementary attribute of an ADG that is transformed by an ADF.

Least Squares

LS approaches minimize the sum of squared errors. This way, they are able to provide a solution to a given optimization problem.

Levenberg Marquardt

The LM algorithm is intended for iteratively determining a solution to a non-linear LS problem. It combines the Gauss–Newton method with the one of gradient descent for fast convergence.

libAFD2

The libAFD2 is a novel framework that has been developed within the scope of this thesis. It is intended for the generalized fusion of CVD and does intensively utilize the SPI for all computationally-intensive functions and for the glue code.

Linear Programming

LP is a technique that does allow to model and solve linear optimization problems under the consideration of equality and inequality constraints. The procedure exploits the fact that the solution space of every linear program is spanned by a convex polytope. The Simplex method by Dantzig is a well-known algorithmic approach for determining the optimal solution (if existent) of a linear program.

Lines of Code

LOC is a metric for describing the size of a computer program by providing the total number of source code lines.

Minimum Covariance Determinant

The MCD method is a robust estimator for the first and second moment of a set of measurements. The basic idea of this method is to sample multiple subsets, to compute for each subset the first and second moment, and to return the first and second moment for the subset with the smallest covariance determinant (determinant of the second moment). This way, a more robust estimate of the first and second moment can be achieved in the case of noisy measurements.

PL/Proxy

PL/Proxy is a PostgreSQL extension that allows to partition and scale queries to multiple database instances. PL/Proxy has been originally developed by Skype.

PostgreSQL

PostgreSQL is an object-relational database that is being steadily developed since 1996. It is commonly regarded as the most advanced open source database. Hence, it features an extensive amount of features and can be even further extended by means of own or third-party extensions. Furthermore, it is known to be feasible for large-scale applications, such as Skype (prior to the acquisition by Microsoft).

Precise Point Positioning

PPP is a method for determining more accurate position estimates via GNSS by additionally considering the carrier phase. Furthermore, precise satellite ephemerids and clocks are commonly incorporated for this purpose.

Principal Component Analysis

The PCA is a statistical procedure that is intended for determining an orthogonal coordinate system for a set of n-dimensional measurements. The orthogonality implies that the so-called eigenvectors, which denote the determined coordinate system, are linearly independent. Furthermore, the PCA provides so-called eigenvalues, which reflect the variances of the eigenvectors.

Probability Density Function

The PDF of a continuous random variable denotes its relative likelihood to take a certain value. A PDF is always bounded to the range $[0, 1]$ and its integral is always equal to one.

Process-Pack-Propagate-chain

PPP-chains are processing chains at the *vehicle-side* for preprocessing, packing, and propagating sensor data from the vehicle- to the back-end-side.

QR Factorization

The QR factorization of a matrix is denoted by: $A = QR$. Q is an orthogonal matrix and R an upper triangular one. The QR factorization can be, for instance, utilized for determining the solution of a linearized LS problem. There exist well-researched methods, such as the Gram–Schmidt process, Householder reflections, or Givens rotations, for computing the QR factorization in both a robust and efficient manner.

Radial Basis Function

RBFs are well-known BFs whose function value is only dependent on the distance between their origin and a given value.

Radial Basis Function Network

An RBFN is an ANN that incorporates RBFs as activation functions.

Realtime Positioning Service

The SAPOS EPS service does provide real-time DGPS correction data. The incorporation of the EPS service for online localization does usually lead to a precision between 1 m and 3 m.

Receiver Independent Exchange Format

RINEX is a receiver independent and widely supported file format for the exchange of GNSS raw data.

Receive-Unpack-Fuse-chain

RUF-chains are processing chains at the *back-end-side* for receiving, unpacking, and fusing sensor data that has been provided by the vehicles.

RTKLIB

The RTKLIB is a C++-framework and toolkit for GNSS-based localization. It supports manifold online and offline localization methods and features a wide range of parameterization options.

Satellite Based Augmentation System

SBAS is a system that enhances GNSS-based localization by providing additional information. In particular, these are more accurate satellite ephemerids, satellite clock drifts, and ionospheric/tropospheric correction data.

Satellite Positioning Service of the German National Survey

The SAPOS service of the German National Survey is a commercial, highly-reliable DGPS service. It is grounded upon a network of approximately 260 reference stations in Germany. The derived correction data is provided in the shape of two products: the EPS real-time positioning service and the HEPS high precision real-time positioning service. While the EPS service does usually lead to an online localization precision between 1 m and 3 m, the HEPS service does usually lead to an online localization precision between 1 cm and 2 cm.

Sequential Compatibility Nearest Neighbor

A data association algorithm based on both the IC- and JC-test.

Server Programming Interface

PostgreSQL's SPI is intended for the extension of its functionality by the development and utilization of user-defined functions and data types. For this purpose, the C/C++ programming language is utilized. The utilization of C/C++ can be regarded as especially beneficent in the case of demanding computations and in the case when third party frameworks, such as Boost or Eigen3, shall be involved.

Simultaneous Localization and Mapping

SLAM can be regarded as a class of methods that allow a vehicle to build a map of an unknown area, while simultaneously localizing itself within.

Structured Query Language

SQL is a standardized database query language for relational databases, which is intended for schema creation / modification / deletion, data inserts / updates/ deletions / queries. SQL is supported by (nearly) all common relational databases. However, it is usually supported within distinct scopes and some databases feature own extensions.

Traffic Sign Recognition

TSR is an automotive camera-based system that is intended for the recognition of traffic signs.

World Geodetic System 1984

WGS84 is a global coordinate system for the geographic referencing of landmarks on earth and the nearby outer space. The coordinate origin of the WGS84 coordinate system is located at the earth's center of mass. Coordinates within the WGS84 coordinate system are provided in degrees with regard to their latitude and longitude, and in meters with regard to their altitude. Furthermore, WGS84 is the reference coordinate system of GPS.

List of Abbreviations

AA	Auxiliary Attribute
ACC	Adaptive Cruise Control
ACG	Attribute Concatenation Graph
ADF	Attribute Dependency Function
ADG	Attribute Dependency Graph
ADT	Attribute Dependency Tree
ADTF	Automotive Data and Time-triggered Framework
ANN	Artificial Neural Network
AR	Ambiguity Resolution
BA	Bundle Adjustment
BB	Branch and Bound
BF	Basis Function
BFS	Breadth-First Search
CA	Constant Acceleration
CAN	Controller Area Network
CCA	Constant Curvature and Acceleration
CDF	Cumulative Density Function
CI	Covariance Intersection
COLAMD	Column Approximate Minimum Degree Ordering Algorithm
C-RINEX	Compressed Receiver Independent Exchange Format
CSAV	Constant Steering Angle and Velocity
CTRA	Constant Turn Rate and Acceleration
CTRV	Constant Turn Rate and Velocity
CU	Covariance Union
CV	Constant Velocity
CVD	Collective Vehicle Data

DAB	Digital Audio Broadcasting
DAG	Directed Acyclic Graph
DBSCAN	Density-based Spatial Clustering of Applications with Noise
DFS	Depth-First Search
DGPS	Differential Global Positioning System
ECU	Electronic Control Unit
EICNN	Extended Individual Compatibility Nearest Neighbor
EJCBB	Extended Joint Compatibility Branch and Bound
EKF	Extended Kalman Filter
EPS	Realtime Positioning Service
ESCNN	Extended Sequential Compatibility Nearest Neighbor
FCI	Fast Covariance Intersection
FGT	Fast Gauss Transform
FIGTree	Fast Improved Gauss Transform with Tree Data Structure
GBB	General Branch and Bound
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IC	Individual Compatibility
ICNN	Individual Compatibility Nearest Neighbor
IFGT	Improved Fast Gauss Transform
IGS	International GNSS Service
IMU	Inertial Measurement Unit
JC	Joint Compatibility
JCBB	Joint Compatibility Branch and Bound
KDE	Kernel Density Estimation
KF	Kalman Filter
KKT	Karush–Kuhn–Tucker Conditions

LA	Landmark Attribute
LM	Levenberg Marquardt
LOC	Lines of Code
LP	Linear Program
LS	Least Squares
MCD	Minimum Covariance Determinant
PCA	Principal Component Analysis
PDF	Probability Density Function
PPP	Precise Point Positioning
QR	QR Factorization
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
RINEX	Receiver Independent Exchange Format
RUF	Receive-Unpack-Fuse
SAPOS	Satellite Positioning Service of the German National Survey
SBAS	Satellite Based Augmentation System
SCNN	Sequential Compatibility Nearest Neighbor
SLAM	Simultaneous Localization and Mapping
SPI	Server Programming Interface
SQL	Structured Query Language
TSR	Traffic Sign Recognition
WGS84	World Geodetic System 1984

List of Algorithms

List of Definitions

List of Figures

List of Problems

List of Solutions

List of Tables

Bibliography

[1] Audi, BMW, and Daimler. The Future is HERE: Tomorrow's Mobility Begins with Real-time Digital Maps, 2015. (Cited on pages 21 and 24)

[2] D. Ayala, J. Lin, O. Wolfson, N. Rishe, and M. Tanizaki. Communication Reduction for Floating Car Data-based Traffic Information Systems. *International Conference on Advanced Geographic Information Systems, Applications, and Services*, pages 44–51, 2010. (Cited on pages 22, 23, and 219)

[3] T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping: Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006. (Cited on pages 49, 96, and 145)

[4] R. J. Barnes and K. You. Adding Bounds to Kriging. *Mathematical Geosciences*, 24(2):171–176, 1992. (Cited on page 188)

[5] R. Bishop. *Intelligent Vehicle Technology and Trends*. Artech House Publishers, 2005. (Cited on pages 19 and 20)

[6] A. Björck. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics, 1996. (Cited on pages 97 and 128)

[7] J. L. Blanco, J. González-Jiménez, and J. A. Fernandez-Madrigal. Sparser Relative Bundle Adjustment (SRBA): Constant-Time Maintenance and Local Optimization of Arbitrarily Large Maps. *IEEE International Conference on Robotics and Automation*, pages 70–77, 2013. (Cited on page 116)

[8] M. L. Boas. *Mathematical Methods in the Physical Sciences*, volume 2. Wiley, 1966. (Cited on page 153)

[9] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas Framework for Scalable Mapping. *IEEE International Conference on Robotics and Automation*, 2:1899–1906, 2003. (Cited on page 48)

[10] S. Breitenberger, B. Gruber, and M. Neuherz. Extended Floating Car Data – Potenziale für die Verkehrsinformation und notwendige Durchdringungsraten. *Straßenverkehrstechnik*, 48(10):522–531, 2004. (Cited on page 24)

[11] R. P. Brent. *Algorithms for Minimization without Derivatives*. Courier Corporation, 1973. (Cited on page 100)

[12] E. Brockfeld, S. Lorkowski, P. Mieth, and P. Wagner. Benefits and Limits of Recent Floating Car Data Technology – An Evaluation Study. *WCTR Conference*, 2007. (Cited on page 20)

[13] E. Brockfeld, B. Passfeld, and P. Wagner. Validating Travel Times Calculated on the Basis of Taxi Floating Car Data with Test Drives. *World Congress on Intelligent Transport Systems*, 2007. (Cited on pages 20 and 22)

[14] Bundeskartellamt. BMW, Daimler, and Audi Can Aquire Nokia's HERE Mapping Service, 2015. (Cited on page 21)

[15] J. Carlson. *Mapping Large, Urban Environments with GPS-Aided SLAM*. PhD thesis, Carnegie Mellon University, 2010. (Cited on page 28)

[16] J. A. Castellanos, J. Neira, and J. D. Tardós. Map Building and SLAM Algorithms. *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*, 2006. (Cited on pages 81 and 92)

[17] H. J. Chang, C. S. G. Lee, Y. C. Hu, and Y.-H. Lu. Multi-Robot SLAM with Topological/Metric Maps. *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1467–1472, 2007. (Cited on page 104)

[18] B. Chen, L. Liu, Z. Zou, and X. Xu. A Hybrid Data Association Approach for SLAM in Dynamic Environments. *International Journal of Adavanced Robotic Systems*, 10(118), 2013. (Cited on page 92)

[19] C. Chong and S. Mori. Convex Combination and Covariance Intersection Algorithms in Distributed Fusion. *International Conference on Information Fusion (FUSION)*, 2001. (Cited on page 99)

[20] J. Civera, O. G. Grasa, A. J. Davison, and J. M. M. Montiel. 1-point RANSAC for Extended Kalman Filtering: Application to Real-time Structure from Motion and Visual Odometry. *Journal of Field Robotics*, 27(5):609–631, 2010. (Cited on pages 91 and 92)

[21] A. J. Cooper. A Comparison of Data Association Techniques for Simultaneous Localization and Mapping. Master's thesis, Massachusetts Institute of Technology, 2005. (Cited on pages 64 and 92)

[22] J. P. Costa, L. Pronzato, and E. Thierry. A Comparison between Kriging and Radial Basis Function Networks for Nonlinear Prediction. *IEEE Nonlinear Signal and Image Processing (NSIP)*, 80(4):726–730, 1999. (Cited on page 182)

[23] N. Cressie. The Origins of Kriging. *Mathematical Geosciences*, 22(3):239–252, 1990. (Cited on page 182)

[24] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1998. (Cited on pages 185, 186, and 188)

[25] T. A. Davis. SuiteSparseQR: Multifrontal Multithreaded Rank-revealing Sparse QR-factorization. *ACM Transactions on Mathematical Software*, 38(1), 2011.

(Cited on page 126)

[26] T. A. Davis, J. R. Gilbert, S. I. Larimore, and E. G. Ng. Algorithm 836: COLAMD, a Column Approximate Minimum Degree Ordering Algorithm. *ACM Transactions on Mathematical Software*, 30(3):377–380, 2004. (Cited on page 127)

[27] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007. (Cited on pages 113 and 161)

[28] C. De Fabritiis, R. Ragona, and G. Valenti. Traffic Estimation and Prediction Based on Real Time Floating Car Data. *IEEE Conference on Intelligent Transportation Systems*, pages 197–203, 2008. (Cited on pages 20 and 21)

[29] F. Dellaert and M. Kaess. Square Root SAM. *International Journal of Robotics Research*, pages 177–184, 2005. (Cited on pages 61, 104, 116, 120, 126, and 127)

[30] C. V. Deutsch. Correcting for Negative Weights in Ordinary Kriging. *Computers & Geosciences*, 22(7):765–773, 1996. (Cited on page 188)

[31] G. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001. (Cited on page 132)

[32] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006. (Cited on page 96)

[33] S. Eichler. Anonymous and Authenticated Data Provisioning for Floating Car Data Systems. *IEEE International Conference on Communication Systems*, pages 1–5, 2006. (Cited on page 24)

[34] A. Elgammal, R. Duraiswami, and L. S. Davis. Efficient Kernel Density Estimation Using the Fast Gauss Transform with Applications to Color Modeling and Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1499–1504, 2003. (Cited on page 190)

[35] T. Emter. Probabilistic Localization and Mapping for Mobile Robots. *Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics*, 2009. (Cited on page 145)

[36] T. Emter and T. Ulrich. Visuelle Information zur robusten Zuordnung von Landmarken für die Navigation mobiler Roboter. *Forum Bildverarbeitung*, pages 95–106, 2010. (Cited on page 145)

[37] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *ACM SIG Knowledge Discovery and Data Mining (SIGKDD)*, 96(34):226–231, 1996. (Cited on page 194)

[38] V. S. Fazio and M. Roisenberg. Spatial Interpolation: An Analytical Comparison between Kriging and RBF Networks. *ACM Symposium on Applied Computing*, pages 2–7, 2013. (Cited on pages 182 and 224)

[39] P. Filzmoser. A Multivariate Outlier Detection Method. *TU Wien*, 2004. (Cited on page 70)

[40] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha. Visual Simultaneous Localization and Mapping: A Survey. *Artificial Intelligence Review*, 43(1):55–81, 2015. (Cited on page 28)

[41] H. Gavin. The Levenberg-Marquardt Method for Nonlinear Least Squares Curve-Fitting Problems. *Duke University*, pages 1–15, 2013. (Cited on pages 119 and 120)

[42] G. H. Golub and R. J. Plemmons. Large Scale Geodetic Least Squares Adjustment by Dissection and Orthogonal Decomposition. *Linear Algebra and Its Applications*, 34:3–28, 1979. (Cited on page 128)

[43] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, 2013. (Cited on pages 126 and 127)

[44] L. Greengard and J. Strain. The Fast Gauss Transform. *SIAM Journal on Scientific and Statistical Computing*, 12(1):79–94, 1991. (Cited on page 190)

[45] L. Greengard and X. Sun. A New Version of the Fast Gauss Transform. *Documenta Mathematica*, pages 575–584, 1998. (Cited on page 190)

[46] W. E. L. Grimson. The Combinatorics of Object Recognition in Cluttered Environments using Constrained Search. *Artificial Intelligence*, 44(1):121–165, 1988. (Cited on page 72)

[47] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A Tutorial on Graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010. (Cited on pages 115 and 119)

[48] J. E. Guivant and E. M. Nebot. Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, 2001. (Cited on page 49)

[49] A. Guttman. *R-trees: A Dynamic Index Structure for Spatial Searching*, volume 14. ACM, 1984. (Cited on pages 49 and 76)

[50] W. Haller. The New Instructions for the Construction of Roundabouts. *Straßenverkehrstechnik, Forschungsgesellschaft für Straßen- und Verkehrswesen*, 2007. (Cited on pages 146, 147, and 155)

[51] S. Hammarling and C. Lucas. Updating the QR-factorization and the Least Squares Problem. Technical report, University of Manchester, 2008. (Cited on pages 97 and 128)

[52] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. (Cited on page 119)

[53] Y. Hatanaka. A Compression Format and Tools for GNSS Observation Data. *Bulletin of the Geographical Survey Institute*, 2008. (Cited on page 35)

[54] P. Heroux and J. Kouba. GPS Precise Point Positioning using IGS Orbit Products. *Physics and Chemistry of the Earth*, 26(6):573–578, 2001. (Cited on page 32)

[55] P. Heroux, J. Kouba, P. Collins, and F. Lahaye. GPS Carrier-Phase Point Positioning with Precise Orbit Products. *KIS*, pages 5–8, 2001. (Cited on page 32)

[56] A. Hesselbarth. *GNSS-Auswertung mittels Precise Point Positioning*, volume 57 of *DVW-Schriftenreihe*. Wißner-Verlag, 2009. (Cited on page 32)

[57] W. Huber, M. Lädke, and R. Ogger. Extended Floating-car Data for the Acquisition of Traffic Information. *World Congress on Intelligent Transport Systems*, pages 1–9, 1999. (Cited on pages 19 and 20)

[58] IGS. IGS Products, 2016. (Cited on pages 31 and 32)

[59] iMAR. iTraceRT-F400, 2016. (Cited on page 34)

[60] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Courier Corporation, 1970. (Cited on page 98)

[61] T. Jeske. Floating Car Data from Smartphones: What Google and Waze Know About You and How Hackers Can Control Traffic. *BlackHat Europe*, 2013. (Cited on page 25)

[62] A. G. Journel. Constrained Interpolation and Qualitative Information - The Soft Kriging Approach. *Mathematical Geosciences*, 18(3):269–286, 1986. (Cited on page 188)

[63] S. J. Julier and J. K. Uhlmann. A Non-Divergent Estimation Algorithm in the Presence of Unknown Correlations. *American Control Conference*, 1997. (Cited on pages 61, 98, 99, and 100)

[64] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. (Cited on pages 104 and 105)

[65] E. Kaplan and C. Hegarty. *Understanding GPS: Principles and Applications*. Artech House Publishers, 2005. (Cited on pages 30, 31, and 32)

[66] B. S. Kerner, C. Demir, R. G. Herrtwich, S. L. Klenov, H. Rehborn, and A. Haug. Traffic State Detection with Floating Car Data in Road Networks. *IEEE Conference on Intelligent Transportation Systems*, pages 44–49, 2005. (Cited on pages 23, 24, and 219)

[67] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A General Framework for Graph Optimization. *IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011. (Cited on pages 119 and 127)

[68] A. H. Land and A. G. Doig. An Automatic Method of Solving Discrete Programming Problems. *Journal of the Econometric Society*, pages 497–520, 1960. (Cited on page 81)

[69] J. Levinson and S. Thrun. Robust Vehicle Localization in Urban Environments Using Probabilistic Maps. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4372–4378, 2010. (Cited on page 28)

[70] Q. Li, T. Zhang, H. Wang, and Z. Zeng. Dynamic Accessibility Mapping Using Floating Car Data: A Network-constrained Density Estimation Approach. *Journal of Transport Geography*, 19(3):379–393, 2011. (Cited on page 22)

[71] Q. Li, T. Zhang, and Y. Yu. Using Cloud Computing to Process Intensive Floating Car Data for Urban Traffic Surveillance. *Journal of Geographical Information Science*, 25(8):1303–1322, 2011. (Cited on page 24)

[72] Y. Li, S. Li, Q. Song, H. Liu, and M. Meng. Fast and Robust Data Association using Posterior Based Approximate Joint Compatibility Test. *IEEE Transactions on Industrial Informatics*, 10(1):331–339, 2014. (Cited on pages 64 and 76)

[73] C. Liu, X. Meng, and Y. Fan. Determination of Routing Velocity with GPS Floating Car Data and WebGIS-based Instantaneous Traffic Information Dissemination. *Journal of Navigation*, 61(02):337–353, 2008. (Cited on page 20)

[74] S. Lorkowski, E. Brockfeld, P. Mieth, B. Passfeld, K. U. Thiessenhusen, and R. P. Schäfer. Erste Mobilitätsdienste auf Basis von Floating Car Data. *AMUS*,

pages 93–100, 2003. (Cited on pages 20 and 21)

[75] F. Lu and E. Milios. Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4:333–349, 1997. (Cited on page 116)

[76] P. C. Mahalanobis. On the Generalized Distance in Statistics. 1936. (Cited on page 229)

[77] R. McGill, J. W. Tukey, and W. A. Larsen. Variations of Box Plots. *The American Statistician*, 32(1):12–16, 1978. (Cited on page 231)

[78] F. Mickler. Spurgenaue GPS Fahrzeuglokalisierung für moderne Fahrerassistenzsysteme durch die Übertragung von Korrekturdaten über Digitalradio (DAB), 2016. (Cited on pages 29 and 37)

[79] V. Morariu, B. V. Srinivasan, V. C. Raykar, R. Duraiswami, and L. S. Davis. Automatic Online Tuning for Fast Gaussian Summation. *Advances in Neural Information Processing Systems*, pages 1113–1120, 2009. (Cited on pages 190 and 191)

[80] J. J. Moré. The Levenberg-Marquardt Algorithm: Implementation and Theory. *Numerical Analysis*, pages 105–116, 1978. (Cited on page 119)

[81] A. Moreira and M. Y. Santos. Concave Hull: A k-Nearest Neighbours Approach for the Computation of the Region Occupied by a Set of Points. *INSTICC Press*, 2007. (Cited on page 194)

[82] W. J. Morphet. *Simulation, Kriging, and Visualization of Circular-Spatial Data*. PhD thesis, Utah State University, 2009. (Cited on page 182)

[83] D. E. Myers. Matrix Formulation of Co-Kriging. *International Association for Mathematical Geology*, 14(3):249–257, 1982. (Cited on page 182)

[84] D. S. Nau, V. Kumar, and L. Kanal. General Branch and Bound, and Its Relation to A* and AO*. *Artificial Intelligence*, 23, 1984. (Cited on pages 74, 80, 81, and 92)

[85] J. Neira and J. D. Tardós. Data Association in Stochastic Mapping using the Joint Compatibility Test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001. (Cited on pages 61, 62, 64, 65, 66, 70, 71, 72, 73, 74, 77, 78, 79, 81, 89, and 91)

[86] W. Niehsen. Information Fusion Based on Fast Covariance Intersection Filtering. *International Conference on Information Fusion (FUSION)*, 2:901–904, 2002. (Cited on page 100)

[87] J. Nieto, T. Bailey, and E. Nebot. Scan-SLAM: Combining EKF-SLAM and Scan Correlation. In *Field and Service Robotics*, pages 167–178. Springer, 2006.

(Cited on page 145)

[88] N. Paragios, Y. Chen, and O. D. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer, 2006. (Cited on page 199)

[89] K. Pearson. On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it Can be Reasonably Supposed to have Arisen from Random Sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1992. (Cited on page 67)

[90] K. B. Petersen and M. S. Pedersen. *The Matrix Cookbook*. Technical University of Denmark, 2005. (Cited on page 195)

[91] M. Raaijmakers and M. E. Bouzouraa. Circle Detection in Single-Layer Laser Scans for Roundabout Perception. *IEEE International Conference on Intelligent Transportation Systems*, pages 2636–2643, 2014. (Cited on page 146)

[92] M. Rahmani, E. Jenelius, and H. N. Koutsopoulos. Route Travel Time Estimation Using Low-frequency Floating Car Data. *IEEE Conference on Intelligent Transportation Systems*, pages 2292–2297, 2013. (Cited on page 22)

[93] S. Rass, S. Fuchs, M. Schaffer, and K. Kyamakya. How to Protect Privacy in Floating Car Data Systems. *ACM International Workshop on Vehicular Inter-Networking*, pages 17–22, 2008. (Cited on page 24)

[94] V. C. Raykar. The Fast Gauss Transform with all the Proofs. Technical report, University of Maryland, 2006. (Cited on page 190)

[95] V. C. Raykar, C. Yang, R. Duraiswami, and N. Gumerov. Fast Computation of Sums of Gaussians in High Dimensions. Technical report, University of Maryland, 2005. (Cited on page 190)

[96] R. Reid. Jointly Compatible Pair Linking for Visual Tracking with Probabilistic Priors. *Australasian Computer Science Conference*, pages 35–42, 2011. (Cited on page 91)

[97] M. Reinhardt, B. Noack, and U. D. Hanebeck. Closed-form Optimization of Covariance Intersection for Low-Dimensional Matrices. *International Conference on Information Fusion (FUSION)*, pages 1891–1896, 2012. (Cited on pages 100, 101, and 102)

[98] M. Reinthaler, B. Nowotny, F. Weichenmeier, and R. Hildebrandt. Evaluation of Speed Estimation by Floating Car Data within the Research Project DMotion. *World Congress on Intelligent Transport Systems*, 2007. (Cited on pages 20 and 22)

[99] S. T. Roweis and R. R. Salakhutdinov. Simultaneous Localization and Surveying with Multiple Agents. *Switching and Learning in Feedback Systems*, pages 313–332, 2005. (Cited on page 104)

[100] R. S. Sampath, H. Sundar, and S. K. Veerapaneni. Parallel Fast Gauss Transform. *IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–10, 2010. (Cited on page 190)

[101] R. R. Schaller. Moore's Law: Past, Present and Future. *IEEE Spectrum*, 34(6), 1997. (Cited on page 116)

[102] R. Schubert, E. Richter, and G. Wanielik. Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking. *International Conference on Information Fusion (FUSION)*, pages 1–6, 2008. (Cited on page 38)

[103] J. Schweitzer. Modular Positioning using Different Motion Models. *International Conference on Machine Control and Guidance*, 2012. (Cited on pages 38 and 40)

[104] S. Skibinski. Dynamische Georeferenzierung ortsbezogener Flottendaten für intelligente Kartendienste. Master's thesis, TU Dortmund, 2012. (Cited on pages 16 and 18)

[105] S. Skibinski. Fahrzeugflotten-basierte Messung von Umgebungsdaten. DE102015000394, 2015. (Cited on page 16)

[106] S. Skibinski. Kartographieren von Fahrspuren mittels Fahrzeugflottendaten. DE102015000399, 2015. (Cited on page 16)

[107] S. Skibinski. Verfahren zum Bereitstellen von Informationen eines Objekts in einer Verkehrssituation und System. DE102015001194, 2015. (Cited on page 14)

[108] S. Skibinski. Verfahren zur Bereitstellung von Informationen über zumindest ein Objekt in einem Umgebungsbereich eines Kraftfahrzeugs und Systems. DE102015001247, 2015. (Cited on page 16)

[109] S. Skibinski. Verfahren zur Erzeugung von Informationen über die Umgebung eines Kraftfahrzeugs und System. DE102015001193, 2015. (Cited on page 15)

[110] S. Skibinski. Method for Providing Information about at Least One Object in a Surrounding Region of a Motor Vehicle and System. WO2016120001, 2016. (Cited on page 16)

[111] S. Skibinski. Method for Providing Information about at Least One Object in a Surrounding Region of a Motor Vehicle and System. EP3158295, 2016. (Cited on page 16)

[112] S. Skibinski. Method for Providing Information of an Object in a Traffic Situation, and System. WO2016120004, 2016. (Cited on pages 14 and 16)

[113] S. Skibinski. Fusion von a priori Kartendaten und sensorbasiertem Umfeldmodell. *BMWi Kooperatives Hochautomatisiertes Fahren (Ko-HAF)*, 2017. (Cited on page 15)

[114] S. Skibinski, A. Lamprecht, and J. Mohring. Positionsbestimmung eines stationären Verkehrsobjekts mittels einer zentralen Serveranordnung. DE102013009856, 2013. (Cited on pages 16 and 18)

[115] S. Skibinski, H. Terhorst, J. F. Weichert, and H. Müller. Large-Scale Fusion of Collective, Areal Vehicle Data. *IEEE International Conference on Multisensor Fusion and Integration (MFI)*, 2015. (Cited on pages 14, 15, 16, and 17)

[116] S. Skibinski, F. Weichert, and H. Müller. Parametric Fusion of Complex Landmark Observations Present within the Road Network by Utilizing Bundle-Adjustment-based Full-SLAM. *International Conference on Information Fusion (FUSION)*, 2016. (Cited on pages 14, 15, 16, and 18)

[117] S. Skibinski, F. Weichert, and H. Müller. Selected Aspects Important from an Applied Point of View to the Fusion of Collective Vehicle Data. *IEEE International Conference on Multisensor Fusion and Integration (MFI)*, 2016. (Cited on pages 14, 15, and 18)

[118] S. Skiena. *The Algorithm Design Manual*. Springer, 1998. (Cited on page 81)

[119] P. Smith. Single-Camera SLAM using the SceneLib Library. *Robotics Research Group Talk*, 2005. (Cited on pages 113 and 161)

[120] R. Smith, M. Self, and P. Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. *Autonomous Robot Vehicles*, pages 167–193, 1990. (Cited on pages 41, 61, 103, and 115)

[121] J. Strain. The Fast Gauss Transform with Variable Scales. *SIAM Scientific and Statistical Computing*, 12(5):1131–1139, 1991. (Cited on page 190)

[122] T. Takasu. RTKLIB: Open Source Program Package for RTK-GPS. *FOSS4G*, 2009. (Cited on page 34)

[123] T. Takasu. RTKLIB: An Open Source Program Package for GNSS Positioning, 2011. (Cited on pages 32 and 34)

[124] T. Takasu, N. Kubo, and A. Yasuda. Development, Evaluation and Application of RTKLIB: A Program Library for RTK-GPS. *GPS/GNSS Symposium 2007*, 2007. (Cited on page 34)

[125] T. Takasu and A. Yasuda. Development of the Low-Cost RTK-GPS Receiver with an Open Source Program Package RTKLIB. *International Symposium on GPS/GNSS*, pages 4–6, 2009. (Cited on page 34)

[126] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005. (Cited on pages 61, 103, and 104)

[127] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle Adjustment – A Modern Synthesis. Technical report, INRIA, 2000. (Cited on page 116)

[128] B. Turan. Using Shape Information from Natural Tree Landmarks for Improving SLAM Performance. Technical report, Bilkent University, 2012. (Cited on page 145)

[129] J. K. Uhlmann. Covariance Consistency Methods for Fault-Tolerant Distributed Data Fusion. *International Conference on Information Fusion (FUSION)*, 4(3):201–215, 2003. (Cited on pages 102, 103, and 115)

[130] C. J. Van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, 2004. (Cited on page 85)

[131] W. Vandenberghe, E. Vanhauwaert, S. Verbrugge, I. Moerman, and P. Demeester. Feasibility of Expanding Traffic Monitoring Systems with Floating Car Data Technology. *IET Intelligent Transport Systems*, 6(4):347–354, 2012. (Cited on page 24)

[132] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina, 1997. (Cited on page 104)

[133] J. Xu, Y. Feng, Z. Zheng, and X. Qing. A Concave Hull Algorithm for Scattered Data and its Applications. *International Congress on Image and Signal Processing*, 5:2430–2433, 2010. (Cited on page 194)

[134] C. Yang, R. Duraiswami, and L. S. Davis. Efficient Kernel Machines Using the Improved Fast Gauss Transform. *Advances in Neural Information Processing Systems*, pages 1561–1568, 2004. (Cited on page 191)

[135] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved Fast Gauss Transform and Efficient Kernel Density Estimation. *IEEE International Conference on Computer Vision*, pages 664–671, 2003. (Cited on page 191)

[136] X. Zheng and P. Palffy-Muhoray. Distance of Closest Approach of Two Arbitrary Hard Ellipses in Two Dimensions. *Physical Review E*, 75(6), 2007. (Cited on page 186)

[137] X. S. Zhou and S. I. Roumeliotis. Multi-Robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case. *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1785–1792, 2006. (Cited on page 104)

[138] J. M. Zogg. *GPS, Essentials of Satellite Navigation.* μ-blox AG, 2003. (Cited on pages 28 and 32)

[139] G. Zunino. *Simultaneous Localization and Mapping for Navigation in Realistic Environments*. PhD thesis, KTH Royal Institute Of Technology, 2002. (Cited on pages 103 and 141)