UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Vehicle Self-Localization Using High-Precision Digital Maps

Dissertation

zur Erlangung des akademischen Grades des Doktors der Naturwissenschaften an der Fakultät für Informatik und Mathematik der Universität Passau

Dipl.-Inf. Andreas Schindler

Im Februar 2013

Erstgutachter:	PD Dr. Tobias Hanning
Zweitgutachter:	Prof. Dr. Tomas Sauer

Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung
Fakultät für Informatik und Mathematik
Universität Passau

To my parents

Acknowledgments

First of all, I would like to thank my academic advisors PD Dr. T. Hanning and Prof. Dr. T. Sauer for supervising this work. Their valuable comments and our professional discussions were essential for the achievement of this PhD thesis.

Furthermore, my appreciation goes to Dr. E. Fuchs (CEO FORWISS, Institute for Software Systems in Technical Applications). Due to his exceptional commitment I had the opportunity to take part in some ambitious and very interesting research projects.

Special thanks go to my colleague Dr. G. Maier, who contributed decisively to this work. His thesis on *Smooth Minimum Arc Paths* took me to the idea for the digital map model presented in the work at hand. Beyond that, he enriched discussions in countless hours with mathematical inputs and his friendly personality.

It was a great pleasure to work with all my colleagues during the time at FORWISS. I would explicitly like to thank F. Janda, S. Pangerl and S. Brummer for their support and contributions not only regarding the software implementation. We had a wonderful time, intensive and enriching in many senses. Special thanks go to Dr. A. Zimmermann, P. Barth, M. Berberich and M. Kuhnkies for the numerous helpful discussions.

Regarding the work in research projects, I would like to acknowledge BMW Group Research and Technology (by name F. Klanner, A. Rauch, K. Vogel, M. Gargano, T. Bleher and R. Feit) for the excellent cooperation and for providing the software framework and data sequences at the basis of this work. Furthermore, special thanks go to R. Krzikalla of the SICK AG for the fruitful and pleasant collaboration. I am indebted to R. Wertheimer for many valuable technical and methodical advice.

I wish to thank the Bayerische Vermessungsverwaltung for kindly providing aerial images for illustration purposes.

With regard to proof-reading, I would like to express my gratitude to M. Böhm.

However, I owe my most important acknowledgements to my family and friends. Their support and encouragement brought me happiness and were a refreshing source of life outside the challenges at work. I specifically thank Laura, whose love and dedication carried me through this PhD project.

Passau, February 2013 Andreas Schindler

Kurzfassung:
In den letzten Jahren haben Fahrerassistenzsysteme im Bereich der aktiven Sicherheit entscheidend zur Verringerung und zur Schadensbegrenzung von Verkehrsunfällen beigetragen. Drahtlose Übertragungstechnologien sowie Methoden zur fahrzeugübergreifenden Sensordatenfusion erlauben neuerdings den Übergang zu kooperativen Assistenzfunktionen. Zur konsistenten Integration von Umgebungsmodellen und der darauf aufbauenden Interpretation der Verkehrssituation stellen sich allerdings hohe Anforderungen an die Lokalisierungsgenauigkeit von Fahrzeugen. Derzeit verfügbare Technologien sind hierfür meist nicht leistungsfähig genug oder zu kostenintensiv für einen Serieneinsatz.
Die Dissertation stellt Methoden und Modelle für einen landmarkenbasierten Ansatz zur Fahrzeugeigenlokalisierung vor. Dabei besteht die Grundidee darin, Informationen aus der fahrzeuglokalen Umgebungsperzeption mit Daten einer hochgenauen digitalen Karte zu assoziieren, um schließlich auf die Position des Fahrzeugs zu schließen. Da derzeit keine digitalen Karten mit der geforderten Präzision und dem notwendigen Detaillierungsgrad im Fahrerassistenzbereich zur Verfügung stehen, wird ein neues Konzept zur Generierung hochgenauer Karten vorgestellt. Die darauf aufbauende probabilistische Eigenlokalisierungsstrategie fusioniert Daten einer Videokamera, eines Laserscanners, GPS und intrinsische Fahrzeugmesswerte in einem Partikelfilteransatz. Es wird aufgezeigt, dass mit der vorgeschlagenen Methodik globale Lokalisierungsgenauigkeiten deutlich unter einem Meter und Orienierungsgenauigkeiten unter einem Grad auch bei Geschwindigkeiten bis 100 km/h in Echtzeit erreicht werden können, was die von den Assistenzfunktionen gestellten Anforderungen erfüllt.
Die Kartenmodellierung basiert auf glatten Kreisbogensplines, also auf Kurven, die stückweise aus Kreisbögen und Strecken aufgebaut sind. Für die Kurvenpassung kommt ein Approximationsverfahren zum Einsatz, das für jede vorgegebene maximale Fehlertoleranz einen Kreisbogenspline mit minimaler Segmentzahl liefert. Diese Eigenschaften sind wertvoll für digitale Karten, denn sie bedeuten die Kontrollierbarkeit der Genauigkeit von Kartenelementen und die Minimierung des zur Speicherung benötigten Datenvolumens. Von diesen Vorteilen profitieren nicht nur die in dieser Arbeit vorgestellten Beobachtungsmodelle zur Fahrzeugeigenlokalisierung, sondern es ergibt sich auch ein Mehrwert für weitere Anwendungen im Automotivebereich.
Im Rahmen einer umfangreichen Auswertung basierend auf simulierten und realen Daten wird aufgezeigt, dass das vorgestellte Kartenkonzept der weit verbreiteten Modellierung mit Polygonen und anderen Kreisbogensplineapproximationen hinsichtlich der Effizienz von Berechnungen auf der Karte, dem Datenvolumen und dem Informationsgehalt überlegen ist. Darüber hinaus demonstriert eine Reihe von Experimenten die Robustheit des Lokalisierungsansatzes hinsichtlich unterschiedlicher Kartendetaillierung, Sensorkonfigurationen und Umgebungsbedingungen.
Damit stellt die Kartenmodellierung zusammen mit der Eigenlokalisierung ein viel versprechendes Konzept für zukünftige Systeme der aktiven Sicherheit im Straßenverkehr dar.

Schlagworte: Eigenlokalisierung, hochgenaue digitale Karte, Kreisbogensplines, Landmarken

Abstract:

In recent years, driver assistance systems have contributed significantly to the reduction of traffic accidents and the mitigation of crash consequences. Wireless communication technologies as well as sensor data fusion methods across vehicles enable cooperative assistance functions. However, the consistent integration of environment models and the subsequent interpretation of traffic situations impose high requirements on the self-localization accuracy of vehicles. State of the art technologies are often not effective enough for these purposes or they are too expensive for a series application.

This thesis presents methods and models for a landmark-based vehicle self-localization approach. The basic idea is to associate information from the vehicular environment perception with data of a high-precision digital map in order to deduce the vehicle's position. Since no digital map with the required precision and level of detail is available at present, a new concept for the generation of high-precision maps is proposed. The probabilistic self-localization strategy, which fuses data from a video camera, laser scanner, GPS and intrinsic vehicular measurements in a particle filter framework, satisfies the accuracy requirements defined by the applications. It is shown that a global localization accuracy significantly below one meter and an orientation accuracy below one degree can be reached even at a speed up to 100 km/h in real-time using the methods presented.

The map model is based on smooth arc splines, which are curves composed by smoothly joint circular arcs and line segments. For any given maximal tolerance, the applied curve approximation method generates a smooth arc spline with a minimum number of segments. These properties are most valuable for digital maps since they imply the checkability of accuracy of map elements as well as the minimization of data volume required for storing the map. Also, the advantages are profitable not only for the self-localization observation models defined in this work, but they represent an additional value for further automotive applications.

By means of an extensive evaluation of the map concepts using both simulated and real data, it is shown that the approach developed in this thesis outperforms the widely-used map modeling with polygons or other arc spline approximations when judged by criteria like efficiency of map calculations, data volume and information content. Moreover, a series of experiments demonstrates the robustness of the approach regarding different levels of details in the map, altering sensor configurations and environmental conditions.

It will be demonstrated that the mapping approach and the self-localization strategy presented in this work represent a promising concept for future systems within the field of active traffic safety.

Keywords: self-localization, high-precision digital map, arc splines, landmarks

Contents

Notations

Notation	Description
$\mathbb{N}$	Set of natural numbers starting with 0
$\mathbb{Z}$	Set of Integers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers
$\mathbb{S}_r(c)$	Sphere around $c \in \mathbb{R}^2$ with radius $r \in \mathbb{R}^+$
$[a, b]$	Closed interval of real numbers
M^T, resp. v^T	Transposition of a matrix M resp. of a vector v
M^{-1}	Inverse of an invertible matrix M
$\pi_k(v)$	Projection on the k-th component of the vector $v \in \mathbb{R}^n, 1 \leq k \leq n$
$SO(n)$	Special orthogonal group in $\mathbb{R}^n$
$\exists, \forall$	Existential quantifier and universal quantifier
$\mathfrak{S}^n$	Set of smooth arc splines with segment number $n \in \mathbb{N}$
$\mathfrak{P}^n$	Set of polygons with segment number $n \in \mathbb{N}$
$\dot{f}$	Derivative of a univariate function f
Df	Total derivative of a function f
$D_i f$	Partial derivative of a function f with respect to its i-th variable

Notation	Common usage
x, y, z	coordinates
λ, ϕ	geographic longitude λ and latitude ϕ
p, q	points
ψ	yaw angle
$\boldsymbol{x}$	(state) vector
i, j, k	indices
w, ω	parametrization w of a curve ω
s, γ	segment s of an arc spline γ
t, t_k	time parameter t and specific point in time t_k
Π	prototype as compact subset of $\mathbb{R}^2$

Chapter 1

Introduction

'You can't know where you are going
until you know where you have been.'

(Old saw)

Contents

1.1 Motivation

In recent years, many research activities in the field of advanced driver assistance technologies (cf. [Ko-FAS 13, interactIVe 13, HAVEit 11, INTERSAFE-2 11, PReVENT 08]) have brought active safety approaches from experimental to maturity phase. The accident analyses and statistics (cf. [GIDAS 08, DESTATIS 11]) show that these systems have the potential for significantly reducing the number of crashes or mitigating the injuries and damages caused by accidents. While modern driver assistance systems like ACC, ESP and electronic brake assist (EBA) help to decrease the frequency of accidents occurring in comparatively simple traffic scenarios, advanced active and preventive safety concepts are required to handle more complex situations. In the latter case, classical vehicular environment perception systems using onboard sensors to update a local environment model are often not sufficient to capture and interpret the surroundings, which is indeed substantial for any safety-related driver assistance function. To cope with this problem, cooperative perception concepts are considered within the joint project Ko-PER, which is part of the project initiative Ko-FAS [Ko-FAS 13], with the aim to let road users share their specific perception results using communication technology. In doing so, the integration of perceived objects from different vehicles increases the robustness and completeness of the individual knowledge on the surroundings. An example is depicted in Figure 1.1.1, where an occlusion is overcome by communicating and integrating the environment models of different road users.
In order to interpret the communicated data and to integrate different environment models consistently, precise and reliable information on the position and orientation (called *pose*) including the time of validity of the involved objects is crucial (cf. [Wertheimer 13]). This, in turn, necessitates that the individual observers are able to identify their own position (*self-localization* in

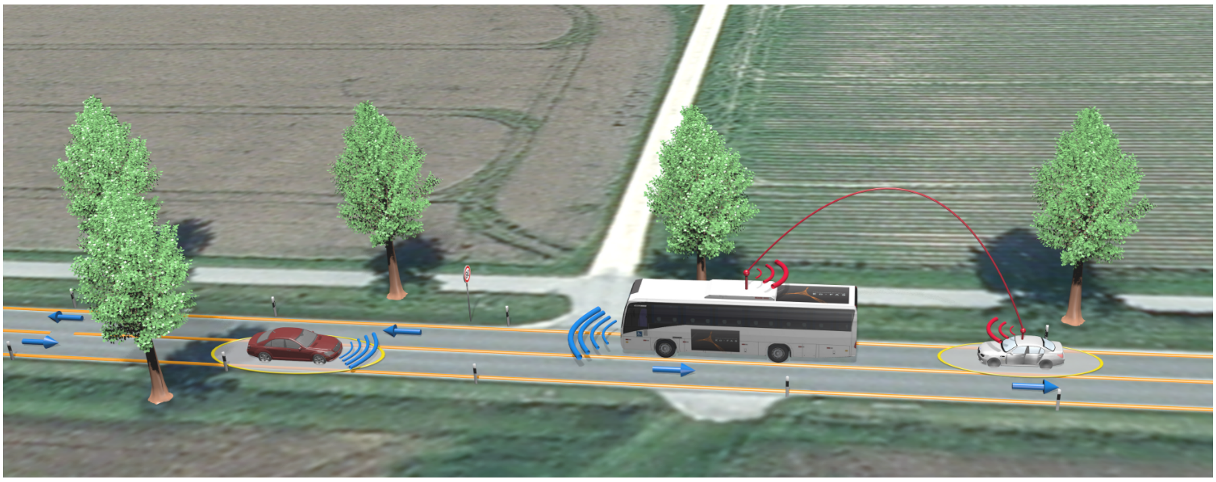


Figure 1.1.1: Cooperative perception: The passenger car on the right is not able to perceive the passenger car on the left using its onboard sensors due to an occlusion caused by the bus. However, this problem can be solved by sharing the environment models between the right car and the bus, which is actually able to observe the left vehicle.

terms of position, orientation and time). Within this context, a longitudinal and lateral position accuracy below 1 m and an orientation accuracy in the magnitude of 1° are required to distinguish between two adjacent vehicles and to associate road users and individual lanes. Hence, the vehicle self-localization represents an essential and challenging component for any cooperative system in the field of active safety.

Since standard *Global Navigation Satellite Systems (GNSS)* often cannot provide positioning results with the required accuracy and reliability defined above due to multi-path scattering, shading effects caused by the environment and atmospheric disturbances (cf. [GPS 08]), alternative and complementary global localization techniques must be considered.

In this thesis, a map-based vehicle self-localization approach is presented. The basic idea is to associate distinctive static objects, like road markings or traffic signs, which are detectable using onboard sensors, with corresponding objects in a high-precision digital map in order to deduce the vehicle's position and orientation. For that purpose, different components are essential: Suitable environment perception methods need to be developed in order to identify the relevant objects based on the vehicular sensor setup. Furthermore, a digital map holding the reference objects with a high global accuracy is required, where the specific requirements are met by the standard digital maps available in today's driver assistance systems. Finally, in order to cope with uncertainties regarding the sensor accuracy, the data processing and the map association, probabilistic methods and models are essential for the self-localization approach. Figure 1.1.2 gives a rough overview of the localization strategy presented in this work.

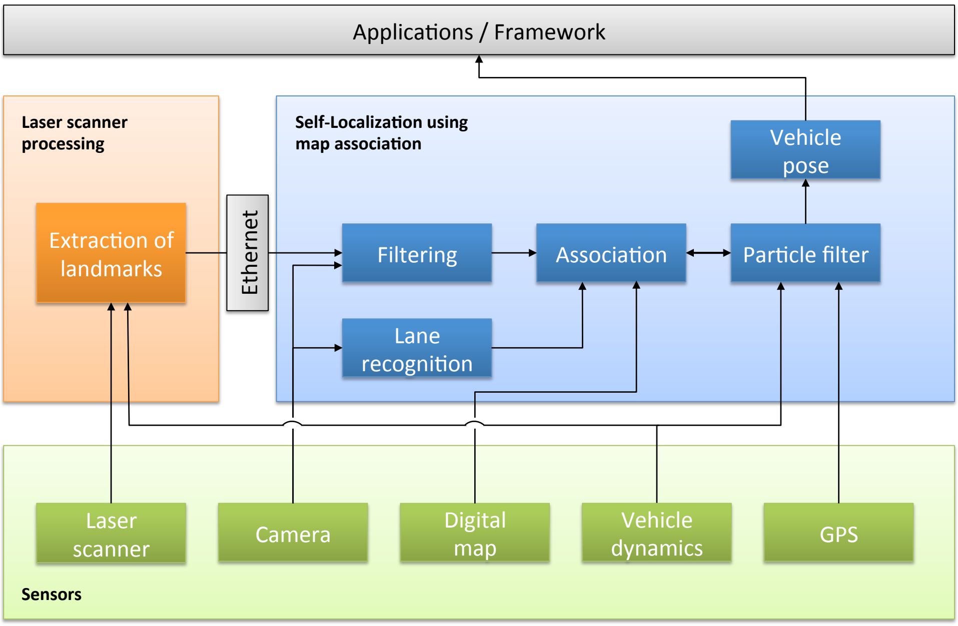


Figure 1.1.2: Overview of the self-localization approach: Environment perception methods based on a monocular gray value camera and a laser scanner are used to identify distinctive objects in the vehicle's surroundings. The sensor data processing allows detecting road markings using a video-based lane recognition system and extracting landmarks, like traffic signs, trees or reflection posts, realized in a specific laser scanner processing unit. These objects are associated with elements in the digital map in order to correct the position estimation. Intrinsic measurements of the vehicle dynamics, like the velocity and the turn rate, determine the motion model of the vehicle. Known in the field of stochastic state estimation, a particle filter is used to realize a probabilistic approach of the pose estimation, which is roughly initialized with GPS.

1.2 Contribution

The main contributions of this thesis can be summarized as follows:

- A model for highly accurate digital maps is presented which makes intensive use of smooth circular arc splines as a curve model for continuous structures like individual lanes and road markings. Compared to the widely-used polygon map model, the proposed model shows many advantageous properties regarding numerical computations, data volume for storage and the information content. The usage of the map model is not restricted to the vehicle self-localization approach in this thesis but it represents an additional benefit for many map-based driver assistance systems.

- In order to create a data basis for the generation of digital map elements, a strategy for the reconstruction of raw measurement points based on a vehicular data acquisition is proposed. The measurement points, extracted by environment perception methods, are reconstructed in a global coordinate system using a high-precision reference positioning system, time stamps for sensor data and different filtering and postprocessing techniques.

- For the generation of curved map elements based on reconstructed measurement points, a new method originated in the field of *reverse engineering* has been adapted and extended for the present mapping purposes. For any given tolerance, modeling the maximal deviation between the resulting curve and the input point sequence, the algorithm generates a smooth arc spline with the minimal possible number of curve segments. These properties are valuable for digital maps since they allow controlling the accuracy of the map elements and minimize the data volume for their storage regarding the chosen map model.

- A self-localization method is proposed which allows estimating the vehicle's pose based on the association of vehicular-perceived objects with elements in the digital map. Within this context, a sensor data fusion concept for a video camera, a laser scanner, GPS and inertial measurements of the vehicle is realized in a probabilistic particle filter approach. The involved observation models and resampling strategies extensively benefit from the particular properties of the digital map model.

- By means of an extensive evaluation of the presented map concepts using both synthetic and real data, it is shown that the mapping approach developed in this thesis generally outperforms comparable methods with polygons or other arc spline approximations regarding criteria like efficiency of map calculations, data volume and information content. Furthermore, a series of experiments shows that global localization errors significantly below 1 m and orientation errors below 1° can be achieved by using the presented self-localization methods on rural roads even at a speed of 100 km/h.

1.3 Outline

Chapter 2 introduces some fundamental models and methods required for the mapping and localization approach. Regarding the map modeling, arc splines as a special curve type play a leading role as geometric representation for map elements. Furthermore, stochastic models and methods for state estimation purposes are presented within this section.

In Chapter 3 the basic environment models are declared, including all relevant coordinate systems as well as the geometric and dynamic vehicle model. Then, all required environment perception methods for the vehicular data acquisition are explained in Chapter 4. In particular, the video-based lane recognition is detailed since it is required within the mapping and localization process.

After a requirements analysis and the discussion of common curve representations, the model of the digital map is presented in Chapter 5. This includes the definitions of individual lanes, road markings and landmarks that are necessary for the localization. It is shown how the modeled map elements can be generated by processing sensor data based on a vehicular acquisition.

The map-based vehicle self-localization approach is presented in Chapter 6 by defining observation models and strategies for the association of perceived objects with map elements.

In order to evaluate the performance of the presented mapping and localization techniques, a series of experiments and results is summarized in Chapter 7 with an outlook to possible future work.

CHAPTER 2

Fundamentals

'Most of the fundamental ideas of science are essentially simple, and may, as a rule, be expressed in a language comprehensible to everyone.'

(Albert Einstein)

Contents

In this chapter some fundamental definitions and methods are declared which are used in later sections of this work.

2.1 Arc splines

This section defines some relevant terms and notations concerning arc splines. The definitions mainly refer to [Maier 10], where the proofs of the stated properties can be found as well. Profound treatments of geometry are available in standard references (cf. [Cao 03, Hartshorne 00, Carmo 92, Dieudonné 60]).

2.1.1 Definitions

In the following, all of the terms related to orientation or relative position of structures refer to a Cartesian system with right-handed orientation in $\mathbb{R}^2$.

Definition 2.1.1.1 (Curve)
A continuous and piecewise continuously differentiable mapping $w : [a, b] \to \mathbb{R}^2$ is called *parametrization* (of a curve). In addition, the function $\Phi : [a, b] \to [c, d]$ expresses an order-preserving change of parameters if Φ is piecewise continuously differentiable and surjective and if $\dot{\Phi}(s) > 0$ holds except a finite number of points.

Any two parametrizations $w_1 : [a,b] \to \mathbb{R}^2$ and $w_2 : [c,d] \to \mathbb{R}^2$ are called equivalent if there exists a change of parameters $\Phi : [a,b] \to [c,d]$ with $w_1 = w_2 \circ \Phi$. The corresponding equivalence classes are called *curves*, denoted by ω.
A curve is called regular if there exists a parametrization $w : [a,b] \to \mathbb{R}^2$ whose derivative $\dot{w}$ does not vanish at any point.
For any parametrization w of ω, the *trace of* ω is given by $\operatorname{tr}(\omega) := w([a,b])$. In this context, $S(\omega) := w(a)$ is the *starting point* and $E(\omega) := w(b)$ is the *endpoint* of the curve. If there exists an injective parametrization of ω, the curve is said to be *simple*. A curve ω is said to be *closed* if $S(\omega) = E(\omega)$. A closed curve ω is a *Jordan curve* if there exists a parametrization $w : [a,b] \to \mathbb{R}^2$ whose restriction $w|_{[a,b[}$ is injective.
In the following, we only focus on

- simple or Jordan curves
- which are regular.

The term $\tau_\omega(x) := \frac{\dot{w}(t)}{\|\dot{w}(t)\|}$ defines the *tangent unit vector of* ω *at* $x = w(t)$. The corresponding orthogonal *normal unit vector* is denoted by $n_\omega(x) := \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right) \cdot \tau_\omega(x)$.
A parametrization of ω is called *arc length parametrization* if $\|\dot{w}(t)\| = 1$ holds for all $t \in [a,b]$.
If ω is simple, then each injective parametrization w induces a unique order:

$$x_1 \preceq x_2 \Leftrightarrow t_1 \leq t_2 \tag{2.1.1}$$

whenever $x_1 = w(t_1)$ and $x_2 = w(t_2)$.
It is obvious that any two injective parametrizations of ω induce the same order.
The length of a curve is given by

$$\operatorname{len}(\omega) := \int_a^b \|\dot{w}(t)\| \, dt. \tag{2.1.2}$$

The *Euclidean distance* of a point $x \in \mathbb{R}^2$ to a curve ω is determined by

$$d(x,\omega) := \operatorname{dist}(x, \operatorname{tr}(\omega)) \quad \text{with} \quad \operatorname{dist}(x, M) := \min_{x' \in M} \left\|x - x'\right\|_2 \tag{2.1.3}$$

for a non-empty, compact set $M \subset \mathbb{R}^2$ and Euclidean norm $\|\cdot\|_2$.

Definition 2.1.1.2 (Best approximating point)
For $x \in \mathbb{R}^2$, a point $x_0 \in \operatorname{tr}(\omega)$ is called *best approximating point* of x with respect to ω if $\|x - x_0\|_2 = d(x,\omega)$ holds.

Definition 2.1.1.3 ((Oriented) Arc / Segment)
Any connected and compact set $A \subseteq \mathbb{R}^2$ with $\operatorname{card}(A) > 1$ is called *arc* if it is a subset of a circle. A curve is called *(oriented) segment* if its trace is an arc or a line segment. The term "oriented" indicates that a curve is meant instead of its trace. This adjective will be omitted if the context is clear.
If the trace of a segment s is an arc, then $C(s)$ represents the *center* and $r(s)$ refers to the radius of the corresponding circle[1].

[1] This circle is unique, as the arc $A \subseteq \mathbb{R}^2$ contains an infinite number of points and a circle is already uniquely determined by three different non-collinear points.

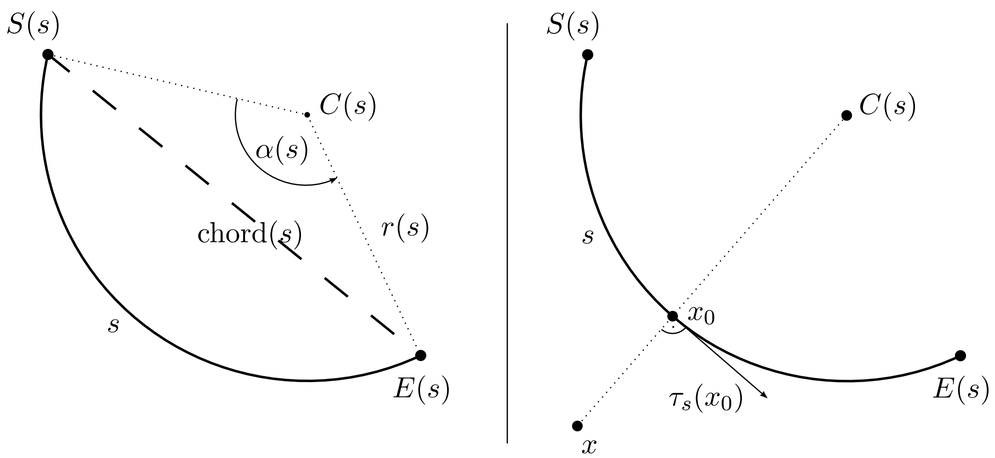


Figure 2.1.1: *Left:* An arc segment s with starting point $S(s)$, endpoint $S(s)$, center $C(s)$, radius $r(s)$, chord chord(s) and opening angle $\alpha(s)$. The orientation of the segment is counterclockwise. *Right:* Projection of the point x onto s and corresponding best approximating point x_0 with tangent unit vector $\tau_s(x_0)$.

The opening angle of s is described by

$$\alpha(s) := \sphericalangle((S(s) - C(s)), (E(s) - C(s))). \tag{2.1.4}$$

The line segment

$$\operatorname{chord}(s) := \{\lambda \cdot S(s) + (1 - \lambda) \cdot E(s) \mid \lambda \in [0, 1]\} \subseteq \mathbb{R}^2 \tag{2.1.5}$$

is called *chord* of s. If the trace of s is a line segment, then $\operatorname{chord}(s) = \operatorname{tr}(s)$ holds.
For $S(s) \neq E(s)$, $d(s) := E(s) - S(s)$ and $n(s) := \frac{1}{\|d(s)\|} \cdot \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right) \cdot d(s)$, the *orientation* $o(s)$ can be identified as

$$o(s) := \begin{cases} -1 & \text{if } \langle n(s) | \tau_s(S(s)) \rangle \geq 0 \quad (\text{“clockwise”}) \\ 1 & \text{else} \quad\quad\quad\quad\quad\quad\quad\quad\; (\text{“counterclockwise”}) \end{cases} \tag{2.1.6}$$

Regardless of the chosen parametrization, the *curvature* $\kappa(s)$ of an arc segment is constant: $\kappa(s) = \frac{o(s)}{r(s)}$.
In accordance with the previous definitions in Section 2.1.1.1, the length of an oriented line segment s can be expressed by $\|d(s)\| = \|E(s) - S(s)\|$.
In contrast, the length of an oriented arc segment is given by $\alpha(s) \cdot r(s)$.
For any arc segment s, the parametrization

$$w : [0, \operatorname{len}(s)] \to \mathbb{R}^2, \quad t \mapsto C(s) + r(s) \cdot \begin{pmatrix} \cos(\frac{t}{r(s)} + t_0) \\ \sin(\frac{t}{r(s)} + t_0) \end{pmatrix} \tag{2.1.7}$$

constitutes an arc length parametrization where $t_0 \in \mathbb{R}$, $w(0) = S(s)$ and $w(\operatorname{len}(s)) = E(s)$. Obviously, it is true that

$$\begin{aligned} \|\dot{w}(t)\| &= \left\| r(s) \cdot \begin{pmatrix} -\sin(\frac{t}{r(s)} + t_0) \cdot \frac{1}{r(s)} \\ \cos(\frac{t}{r(s)} + t_0) \cdot \frac{1}{r(s)} \end{pmatrix} \right\| \\ &= \sqrt{\left(-\sin\left(\frac{t}{r(s)} + t_0\right)\right)^2 + \left(\cos\left(\frac{t}{r(s)} + t_0\right)\right)^2} = 1. \end{aligned} \tag{2.1.8}$$

Likewise, an arc length parametrization of a line segment s is given by

$$w : [0, \operatorname{len}(s)] \to \mathbb{R}^2, \quad t \mapsto S(s) + t \cdot \frac{d(s)}{\|d(s)\|}. \tag{2.1.9}$$

Having introduced some basic segments, their composition can now be considered in order to provide a more flexible way of modeling. Originally, the term *spline* is used for curves composed of polynomials. Analogously, an arc spline is a curve that is piecewise defined by segments from 2.1.1.3.

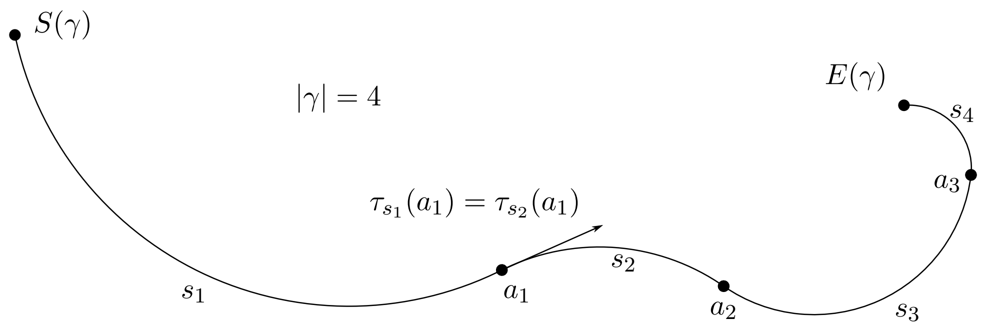


Figure 2.1.2: Smooth arc spline γ with segment number $|\gamma| = 4$ and breakpoints $a_1, \ldots, a_3$.

Definition 2.1.1.4 (Arc spline)
Any simple curve or Jordan curve γ is called *arc spline* if there exists a finite family $(A_i)_{1 \le i \le n}$ of circular arcs or line segments such that the following holds:

$$\operatorname{tr}(\gamma) = \bigcup_{i=1}^{n} A_i \tag{2.1.10}$$

The minimum n for which a *defining sequence* $(A_i)_{1 \le i \le n}$ of γ exists is called *segment number* of γ, abbreviated by $|\gamma| := n$.
For any defining sequence $(A_i)_{1 \le i \le n}$ and $i \neq j$, it is true that $\operatorname{card}(A_i \cap A_j) \le 1$ due to the simplicity of the curve. According to [Maier 10], for any arc spline γ with $|\gamma| = n$, there exist a unique defining sequence $(A_i)_{1 \le i \le n}$ and a decomposition of segments $s_1, \ldots, s_n$ of γ such that $\operatorname{tr}(s_i) = A_i$. Therefore, we use the abbreviation $\gamma = s_1 \ldots s_n$.
The breakpoints $a_1, \ldots, a_{n-1} \in \mathbb{R}^2$ of an arc spline $\gamma = s_1 \ldots s_n$ are given by the non-empty intersections between A_i and A_{i+1} for $1 \le i \le n-1$.
With $l_i := \operatorname{len}(s_i)$, the length of an arc spline γ results in

$$\operatorname{len}(\gamma) = \sum_{i=1}^{n} l_i. \tag{2.1.11}$$

With the aid of the arc length parametrization w_i of the individual segments and the accumulated segment lengths $L_j := \sum_{i=0}^{j} l_i$ for $j \in \{1, \ldots n\}$, an arc length parametrization $w : [0, L] \to \mathbb{R}^2$ of γ can be expressed in the following way:

$$w(t) = \sum_{i=0}^{n} \underset{[L_{i-1}, L_i]}{\chi}(t) \cdot w_i(\underset{[0, l_i]}{\chi}(t - L_{i-1}) \cdot (t - L_{i-1})), \quad L_0 := 0, \tag{2.1.12}$$

where χ denotes the indicator function. This arc length parametrization of γ results from concatenating the arc length parametrizations of the individual segments.

Definition 2.1.1.5 (Smooth arc spline)
An arc spline $\gamma = s_1 \dots s_n$ with breakpoints $a_1 \prec \dots \prec a_{n-1}$ is said to be *smooth* if

$$\underset{1\leq i\leq n-1}{\forall} \tau_{s_i}(a_i) = \tau_{s_{i+1}}(a_i). \tag{2.1.13}$$

The set of all smooth arc splines with segment number $n \in \mathbb{N}$ is denoted by $\mathfrak{S}^n$. Furthermore, let $\mathfrak{S} = \mathfrak{S}^1$ and $\mathfrak{S}^\infty = \bigcup_{n\in\mathbb{N}} \mathfrak{S}^n$.

Definition 2.1.1.6 (Polygon)
An arc spline whose defining sequence consists exclusively of line segments is called *polygon*. A polygon can also be seen as a linear polynomial spline.
The set of all polygons with segment number $n \in \mathbb{N}$ is abbreviated by $\mathfrak{P}^n$. Analogous to the previous definition, let $\mathfrak{P} = \mathfrak{P}^1$ and $\mathfrak{P}^\infty = \bigcup_{n\in\mathbb{N}} \mathfrak{P}^n$.

2.1.2 Properties of arc splines

In the following, some of the fundamental properties of arc splines are shown.

2.1.2.1 Invariance

Arc splines are invariant with respect to rotation, isotropic scaling and translation $\mathbb{R}^2$. That means for any arc spline $\gamma \in \mathfrak{S}^\infty, \lambda \in \mathbb{R}, t \in \mathbb{R}^2$ and rotation matrix $R \in \mathbb{R}^{2\times 2}$, there exists an arc spline $\gamma' \in \mathfrak{S}^\infty$ such that the following holds:

$$\lambda \cdot R \cdot \mathrm{tr}(\gamma) + t = \mathrm{tr}(\gamma'). \tag{2.1.14}$$

For any set $M \subset \mathbb{R}^2$, the ε-*offset* is given by

$$M_\varepsilon := \left\{ x \in \mathbb{R}^2 \,\middle|\, \mathrm{dist}\,(x, M) = \varepsilon \right\}. \tag{2.1.15}$$

Using the normal unit vector $n_\omega(x)$ (cf. 2.1.1.1) and some parametrization $w : [a, b] \to \mathbb{R}^2$ of a curve ω, the *left* ε*-offset curve* $\omega_{\varepsilon,l}$ and the *right* ε*-offset curve* $\omega_{\varepsilon,r}$ can be expressed:

$$w_{\varepsilon,l}(t) := w(t) + \varepsilon \cdot n_\omega(w(t)), \tag{2.1.16}$$
$$w_{\varepsilon,r}(t) := w(t) - \varepsilon \cdot n_\omega(w(t)) \tag{2.1.17}$$

For a sufficiently[2] small ε, the ε-parallel curve of any (smooth) arc spline $\gamma = s_1 \dots s_n$ is again a (smooth) arc spline and the following holds for the resulting left and right offset arc spline $\gamma_l = s_{l,1} \dots s_{l,n}$ and $\gamma_r = s_{r,1} \dots s_{r,n}$, respectively:

$$\begin{aligned} C(s_{l,i}) &= C(s_{r,i}) = C(s_i) \\ r(s_{l,i}) &= r(s_i) - o(s_i) \cdot \varepsilon \\ r(s_{r,i}) &= r(s_i) + o(s_i) \cdot \varepsilon \end{aligned} \tag{2.1.18}$$

[2] In this case, an upper bound for the offset ε is given by $\inf\limits_{x\in\mathrm{tr}(\gamma)} \{\varepsilon' > 0 \,|\, B_{2\varepsilon'}(x) \cap \mathrm{tr}(\gamma) \text{ connected}\}$, where $B_r(x) := \left\{a \in \mathbb{R}^2 \,\middle|\, \|a - x\| \leq r\right\}$.

The calculation of offset curves plays a role in the generation of digital maps as it allows determining parallel structures like lanes or lane markings in a simple way.

2.1.3 Distance calculation to arc splines

This section explains appropriate methods for the calculation of the distance between any point $x \in \mathbb{R}^2$ and an arc spline $\gamma = s_1 \dots s_n$. That task is related to the computation of the corresponding best approximating point $x_0 \in \mathrm{tr}(\gamma)$ or with the calculation of the arc length parameter $t_0 \in [0, \mathrm{len}(\gamma)]$, respectively, for which $w(t_0) = x_0$ holds regarding an arc length parametrization w of γ.
As the distance between x and γ as well as the arc length parameter t_0 are directly resulting from the best approximating point x_0, the computation of x_0 is treated first. This can be separated into two parts:

1. The calculation of the best approximating point x_0 of x with respect to a segment s
2. The calculation of a segment $s^\star$ having the shortest distance to x regarding (2.1.3) or, in other words, $d(x, s^\star) = \min_i d(x, s_i)$

Calculation of the best approximating point of x with respect to a segment

Line segment: For any line segment s, let $l := \|E(s) - S(s)\|$ be the length of s and let $d(s) := \frac{E(s)-S(s)}{l}$ denote the normalized direction of the chord of s. Using $\lambda := \langle d(s) | x - S(s) \rangle$, the required best approximating point is given by

$$x_0 = \begin{cases} S(s) & \lambda \leq 0 \quad (x_0 \text{ is the starting point of } s) \\ S(s) + \lambda \cdot d(s) & 0 < \lambda < l \\ E(s) & l \leq \lambda \quad (x_0 \text{ is the endpoint of } s) \end{cases} \tag{2.1.19}$$

The arc length parameter t_0 can be expressed directly in terms of λ: $t_0 = \max\{0, \min\{\lambda, l\}\}$.

Arc segment: If s is an arc segment, the best approximating point x_0 corresponding to x can be determined in the following way:
First the projection x_0' of x onto the corresponding circle of s with radius $r(s)$ and center $C(s)$ is considered[3]: $x_0' := C(s) + r(s) \cdot \frac{x - C(s)}{\|x - C(s)\|}$. If x_0' is in $\mathrm{tr}(s)$, it is obviously true that $x_0 = x_0'$, otherwise x_0 is the starting point or the endpoint of s. The latter case can be treated by investigating the relative position of x_0' with respect to the chord of s:
Let $d(s) := E(s) - S(s)$ and $n \in \mathbb{R}^2$ with $n := o(s) \cdot \left(\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right) \cdot d(s)$ being the normal vector of the chord pointing in the direction of the arc. Then the best approximating point x_0 is determined in the following way:

$$x_0 = \begin{cases} x_0' & \langle n | x_0' - S(s) \rangle \geq 0 & (x_0' \text{ is in } \mathrm{tr}(s)) \\ S(s) & \langle n | x_0' - S(s) \rangle < 0 \wedge \|S(s) - x_0'\| \leq \|E(s) - x_0'\| & (x_0 \text{ is the starting point of } s) \\ E(s) & \langle n | x_0' - S(s) \rangle < 0 \wedge \|S(s) - x_0'\| > \|E(s) - x_0'\| & (x_0 \text{ is the endpoint of } s) \end{cases} \tag{2.1.20}$$

[3]Within this context the special case of $x = C(s)$ can be treated in practical applications. In this situation an arbitrary point in $\mathrm{tr}(s)$ is chosen as best approximating point and $d(x, s) = r(s)$ holds.

By means of the angle $\beta := \sphericalangle((S(s)-C(s)),(x_0-C(s)))$ the arc length parameter corresponding to x_0 can be expressed by $t_0 = \beta \cdot r(s)$.
In [Irle 09], a formulation for the computation of best approximating points is proposed which avoids the use of the segment radius by some geometrical reasoning. Within this approach, the slightly more complex calculations can improve the numerical stability when dealing with large segment radii.

Calculation of $s^\star$: The simplest way of calculating $s^\star$ is achieved by computing the distance between x and s_i for all $i \in \{1,\ldots,n\}$. Finally, the segment of least distance is chosen. This naive approach might be sufficient for small n depending on the application. However, it requires $O(n)$ calculations of distances as each segment is taken into account.
For arc splines of higher segment number, the above-mentioned approach might be computationally too expensive. In this case, some appropriate index structures for arc splines can be considered. Within this context, Quadtrees ([Samet 90]) or Voronoi diagrams ([De Berg 08, Aurenhammer 91]) represent advantageous data structures, since an appropriate decomposition of γ enables the calculation of $s^\star$ in $O(\log n)$ complexity.

Distance and arc length regarding γ: Using these calculations, the distance of x to an arc spline γ is finally given by the distance to the computed best approximating point x_0.
The arc length parameter of x_0 with respect to γ results in the sum of the lengths of all predecessor segments of $s^\star$ and the arc length parameter of x_0 with respect to $s^\star$.
To summarize, one can state that distances, best approximating points and corresponding arc length parameters can be calculated exactly and in a closed form for each segment. By means of appropriate data structures, these calculations can be realized in an efficient way for complete arc splines.

2.2 State estimation

This section deals with common methods for state estimation in the context of dynamic systems. Especially the Kalman filter and particle filter are discussed.

2.2.1 Dynamic systems

Some of the central components of this work, like the self-localization and the lane recognition, are model based systems. This means that there exists a formal description of the components that represents their relevant properties at a certain time. The set of all parameters used for this description is called by the state of the system. A dynamic system is characterized by a temporal variance of the state parameter. Typically, the state parameter at a certain time can not be determined directly. However, observations and measurements allow deducing the state of the system. Under realistic conditions, the correctness and completeness of the situation described by the system model cannot be achieved due to physical noise effects.

Therefore, a whole range of stochastic methods and models, which are suitable for state estimation, have been developed in the last decades. The techniques used within the context of this thesis are already treated extensively in the literature. That is why a detailed introduction to this topic is not presented here in favor of referring to appropriate standard works of the technical literature (cf. [Jazwinski 70, Meintrup 05]). In the following, the terms and concepts are mainly oriented on the treatment in [Tatschke 11].

In the framework of the Bayesian estimation theory, the situation at hand can be formulated such that the system state corresponds to a Markov process whose respective probability density function is estimated recursively over time. Within this context, dynamic models describe the temporal variation of the state whereas observation models allow integrating measurements for the correction of the state. The measurements can be interpreted as the visible variables of a *Hidden Markov model*.

In the following, a dynamic, time-discrete system is considered whose internal state is modeled by a $\mathbb{R}^n$-valued, stochastic process $(X_k)_{k\in\mathbb{N}}$. The index of the state refers to a strictly isotone sequence of points in time $t_k \in \mathbb{R}, k \in \mathbb{N}$. Likewise, the $\mathbb{R}^m$-valued, stochastic process $(Y_k)_{k\in\mathbb{N}}$ models the observations and the measurements of the system, respectively.

The state is supposed to have the Markov property, which means that the corresponding conditional density function and any realizations $x, x_0, \dots, x_n \in \mathbb{R}^n$ satisfy

$$p(X_{k+1} = x | X_0 = x_0, \dots, X_k = x_k) = p(X_{k+1} = x | X_k = x_k). \tag{2.2.21}$$

for any $k \in \mathbb{N}$. Concerning the observation process, the conditional independence of the internal state is assumed, which means that for any $y \in \mathbb{R}^m$

$$p(Y_k = y | X_0 = x_0, \dots, X_k = x_k) = p(Y_k = y | X_k = x_k) \tag{2.2.22}$$

holds as well as the pairwise independence over time for any $n \in \mathbb{N}$ and $k \in \{0, \dots, n\}$

$$\begin{aligned} p(Y_k = y_k, \dots, Y_n = y_n | X_k = x_k, \dots, X_n = x_n) \\ = \prod_{i=k}^{n} p(Y_i = y_i | X_k = x_k, \dots, X_n = x_n) \\ = \prod_{i=k}^{n} p(Y_i = y_i | X_i = x_i) \end{aligned}$$

for realizations $y_k, \dots, y_n \in \mathbb{R}^m$.

In the following, the abbreviation $\mathbb{Y}_k := \{Y_i = y_i \mid i \in \{0, \dots, k\}\}$ denotes a sequence of measurements for any $k \in \mathbb{N}$ and corresponding realizations $y_i \in \mathbb{R}^m$.

Using the state transition probability density $p(X_k = x | X_{k-1} = x_{k-1})$ from time t_{k-1} to t_k and the state probability $p(X_{k-1} = x_{k-1} | \mathbb{Y}_{k-1})$ for any realization $x_{k-1} \in \mathbb{R}^n$ by accounting for all previous observations, the conditional prediction density can be expressed using the *Chapman-Kolmogorov* equation:

$$p(X_k = x | \mathbb{Y}_{k-1}) = \int_{\mathbb{R}^n} p(X_k = x | X_{k-1} = x_{k-1}) p(X_{k-1} = x_{k-1} | \mathbb{Y}_{k-1}) dx_{k-1}. \tag{2.2.23}$$

Furthermore, the conditional state density taking into account the observation at time t_k can be expressed using Bayes' theorem

$$p(X_k = x | \mathbb{Y}_k) = \frac{p(Y_k = y_k | X_k = x) p(X_k = x | \mathbb{Y}_{k-1})}{\int_{\mathbb{R}^n} p(Y_k = y_k | X_k = x_k) p(X_k = x_k | \mathbb{Y}_{k-1}) dx_k}. \tag{2.2.24}$$

The equations (2.2.23) and (2.2.24) already indicate a recursive scheme, which is decisive for further specializations and algorithmic approaches for state estimation purposes. More details on this topic are to be found in [Bergman 99, Schön 06].

With regard to applications, an important special case of the above-mentioned equations arises if the dynamic model and the observation model are representable explicitly by functions and by assuming additive noise. In that case, the dynamic model can be expressed as a function $f : \mathbb{R}^n \times \mathbb{R}^2 \to \mathbb{R}^n$. Likewise, the observation model corresponds to a function $h : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$. Using the notation

$$f_k : \mathbb{R}^n \to \mathbb{R}^n, \quad x \mapsto f(x, t_k, \Delta t_k) \tag{2.2.25}$$

$$h_k : \mathbb{R}^n \to \mathbb{R}^m, \quad x \mapsto h(x, t_k) \tag{2.2.26}$$

for $k \in \mathbb{N}$ the equations

$$X_{k+1} = f_k(X_k) + W_k, \tag{2.2.27}$$

$$Y_k = h_k(X_k) + E_k \tag{2.2.28}$$

hold where the independent $\mathbb{R}^n$- and $\mathbb{R}^m$-valued random processes $(W_k)_{k \in \mathbb{N}}$ and $(E_k)_{k \in \mathbb{N}}$ are called *process noise* and *measurement noise*, respectively. The notation $\Delta t_k := t_{k+1} - t_k$ describes the difference between two points in time.

2.2.2 State estimation algorithms

Based on the terms introduced in Section 2.2.1, some specializations are considered below, which are relevant for this work. The aim of these methods is to estimate the probability density of the system process $(X_k)_{k\in\mathbb{N}}$ over time and to finally enable a *point estimate*. The latter implies calculating a vector $\hat{x}_{l|k} \in \mathbb{R}^n, l \geq k$ together with a quality criterion that represents the best estimation at time t_l, taking into account all measurements up to time t_k. The information extracted in that way is then provided to the contextual applications.

2.2.2.1 (Extended) Kalman filter

An important special case of the context discussed above is given by the assumption that both the dynamic model f and the observation model h are linear and that all involved noise processes are normally distributed. Under these assumptions, the so-called *Kalman filter* (cf. [Kalman 60, Kailath 00]) can be formulated as a recursive estimation algorithm, which has found application in numerous technological fields. One can show that, in this case, all related density functions from (2.2.23) and (2.2.24) are Gaussian. Hence, the best point estimation arises from the first two moments of the distribution, also referred to as mean and covariance matrix. This estimation is optimal among all point estimations.
The formulation of the filter equations of the Kalman filter is omitted at that point as it is a special case of the *Extended Kalman filter*, which is introduced in detail in [Schmidt 66] or [Schön 06]. In that case, f and h are generally nonlinear functions. Using the Taylor approximation up to degree one at the estimation $\hat{x}_{k|k}$ and $\hat{x}_{k|k-1}$, respectively, they can be used in a linearized way for any $x \in \mathbb{R}^n$:

$$f(x, t_k, \Delta_k t) \approx T^{(1)}_{f_k,\hat{x}_{k|k}}(x) = f_k(\hat{x}_{k|k}) + Df_k(\hat{x}_{k|k})(x - \hat{x}_{k|k}), \tag{2.2.29}$$

$$h(x, t_k) \approx T^{(1)}_{h_k,\hat{x}_{k|k-1}}(x) = h_k(\hat{x}_{k|k-1}) + Dh_k(\hat{x}_{k|k-1})(x - \hat{x}_{k|k-1}). \tag{2.2.30}$$

Concerning the process equation (2.2.27) and (2.2.28) the linearization results in

$$X_{k+1} = T^{(1)}_{f_k,\hat{x}_{k|k}}(X_k) + W_k \quad \text{and} \tag{2.2.31}$$

$$Y_k = T^{(1)}_{h_k,\hat{x}_{k|k-1}}(X_k) + E_k. \tag{2.2.32}$$

Finally, the equations of the Extended Kalman filter are given as follows:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - h(\hat{x}_{k|k-1}, t_k)), \tag{2.2.33}$$

$$P_{k|k} = P_{k|k-1} - K_k H_k P_{k|k-1}, \tag{2.2.34}$$

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}, t_k, t_{k+1} - t_k), \tag{2.2.35}$$

$$P_{k+1|k} = F_k(t_{k+1} - t_k) P_{k|k} F_k(t_{k+1} - t_k)^\top + Q_k(t_{k+1} - t_k), \tag{2.2.36}$$

$$K_k = P_{k|k-1} H_k^\top (H_k P_{k|k-1} H_k^\top + R_k)^{-1} \tag{2.2.37}$$

Here, $F_k(\Delta_k t)$ and H_k are the Jacobi matrices of the corresponding models f_k and h_k,

$$F_k(\Delta_k t) := Df_k(\hat{x}_{k|k}) \in \mathbb{R}^{n\times n}, \quad H_k := Dh_k(\hat{x}_{k|k-1}) \in \mathbb{R}^{n\times m}. \tag{2.2.38}$$

Furthermore, $\hat{x}_{1|0} \in \mathbb{R}^n$ is the starting value and $P_{1|0} \in \mathbb{R}^{n\times n}$ is a symmetric, positive-definite matrix. The symmetric, positive-definite matrices $Q_k \in \mathbb{R}^{n\times n}$ and $R_k \in \mathbb{R}^{m\times m}$ are called *process noise covariance* and *measurement noise covariance.*

Within the Kalman Filter equations, the estimated conditional density $\hat{p}(X_k = x|\mathbb{Y}_k)$ is Gaussian with mean $\hat{x}_{k|k}$ and covariance matrix $P_{k|k}$, called *estimate covariance matrix.* Likewise, the estimated conditional density $\hat{p}(X_{k+1} = x|\mathbb{Y}_k)$ is Gaussian with mean $\hat{x}_{k+1|k}$ and covariance matrix $P_{k+1|k}$.

The quantity $\hat{x}_{k+1|k}$ is named *predicted state* while $h(\hat{x}_{k|k-1}, t_k)$ represents the *predicted measurement* in contrast to the *real measurement* y_k. The expression $y_k - h(\hat{x}_{k|k-1}, t_k)$ is called *residual*, which is required for the *measurement update*, also called the *filtering step* (2.2.33), weighted by the *Kalman gain matrix* matrix K_k.

One can see that the point estimates for the states $\hat{x}_{k|k}$ and $\hat{x}_{k+1|k}$ as well as the corresponding estimate covariance matrices $P_{k|k}$ and $P_{k+1|k}$ are updated recursively within the equations (2.2.33) to (2.2.37).

As the Taylor approximation is considered only up to the polynomial degree one, the linearization may be error-prone concerning the point estimation of $\hat{x}_{k|k}$ and $P_{k|k}$. If the remainder term of the Taylor approximation of the chosen dynamic or observation model is significant, more sophisticated filtering techniques may be more appropriate.

The association of real and predicted measurements represents a decisive step within the filtering process, particularly in the presence of noise within the measurements. Wrong associations lead to incorrect measurement updates, which in turn may result in divergence of the estimation error. Therefore, several extensions of the Kalman filter have been presented in the literature dealing with uncertainties concerning the association of measurements [Bar-Shalom 95, Bar-Shalom 88].

The *Probabilistic Data Association (PDA)* filter [Bar-Shalom 09] allows the modeling of concurrent association constellations between cluttered measurements and a considered dynamic system in a statistical sense. This is realized by extending the conditional density function in (2.2.24) in order to model the probability of the different measurement associations, including the case when any measurement effectively corresponding to the considered dynamic system is missing and only noise is available within the measurements. A further extension is realized within the *Joint Probabilistic Data Association (PDA)* filter [Bar-Shalom 09, Chang 83], which allows modeling several dynamic systems at a time sharing their measurement spaces. In that case, the association of interfering measurements is even more challenging and necessitates modeling the probabilities for more combinatorial constellations. Further examples for filter extensions are the *Integrated Probabilistic Data Association (IPDA)* filter [Musicki 94] and the *Joint Integrated Probabilistic Data Association (JIPDA)* filter [Musicki 02], which introduce a target existence propagation model, the Unscented Kalman filter (cf. [Julier 95, Julier 97]) or the particle filter. Compared to the standard Kalman filter, these algorithms generally improve the filtering performance, whereas the computational efforts are increased.

The particle filter is described in the next section, since it is used for the proposed self-localization approach.

2.2.2.2 Particle filter

The fundamental advantage of a particle filter compared to a Kalman filter consists in making no assumptions on the probability distributions of the considered state. Furthermore, the dynamic model and the observation model can be nonlinear. This generalization allows the application of this estimation technique in much wider contexts. However, in general, these capabilities cause higher computational costs.
Basically, the terminology from [Tatschke 11] is used, where the corresponding introductions are worked out in detail. Some more in-depth treatment and extensions of the context summarized here can, for example, be found in [Doucet 98, Liu 98, Arulampalam 02] or [Schön 06].
The particle filter is based on a sequential Monte Carlo method to recursively estimate the probability densities in (2.2.23) and (2.2.24). Instead of solving the involved integrals numerically, the probability density function $p(x|\mathbb{Y})$ is approximated by a finite set of samples, also called *particles*,

$$p(X_k = x|\mathbb{Y}_s) \approx \sum_{i=1}^{M} w_k^{(i)} \delta(x - x_{k|s}^{(i)}) \tag{2.2.39}$$

for any $k, s \in \mathbb{N}$ and $x \in \mathbb{R}^n$ together with corresponding weights

$$\sum_{i=1}^{M} w_k^{(i)} = 1, \qquad \underset{i \in \{1,\ldots,M\}}{\forall} w_k^{(i)} \geq 0 \tag{2.2.40}$$

Here, δ denotes the Dirac delta function, $M \in \mathbb{N}^+$ is the number of samples $x_{k|s}^{(i)} \in \mathbb{R}^n$ of the $\mathbb{R}^n$-valued random variable $X_{k|s}^{(i)}$, and $w_k^{(i)} \in \mathbb{R}_0^+$ are the weights of the samples for $1 \leq i \leq M$.
Regarding the approximation (2.2.39), the samples $x_{k|s}^{(i)}$ must be determined as well as their corresponding weights $w_k^{(i)}$. The relation between this approach and the filter equation (2.2.24)

$$\begin{aligned} p(X_k = x \,|\, \mathbb{Y}_k) &= p(X_k = x \,|\, Y_k = y_k, \mathbb{Y}_{k-1}) && (2.2.41) \\ &= \frac{p(Y_k = y_k \,|\, X_k = x, \mathbb{Y}_{k-1})\, p(X_k = x \,|\, \mathbb{Y}_{k-1})}{\int_{\mathbb{R}^n} p(Y_k = y_k \,|\, X_k = x, \mathbb{Y}_{k-1})\, p(X_k = x \,|\, \mathbb{Y}_{k-1}) dx_k} && (2.2.42) \\ &= \frac{p(Y_k = y_k \,|\, X_k = x)\, p(X_k = x \,|\, \mathbb{Y}_{k-1})}{\int_{\mathbb{R}^n} p(Y_k = y_k \,|\, X_k = x, \mathbb{Y}_{k-1})\, p(X_k = x \,|\, \mathbb{Y}_{k-1}) dx_k} && (2.2.43) \end{aligned}$$

for any $k \in \mathbb{N}$ and $x \in \mathbb{R}^n$ can be expressed as follows:
The predicted conditional density $p(X_k = x \,|\, \mathbb{Y}_{k-1})$ corresponds to applying the dynamic model f_k to each particle, that is $x_{k+1|k}^{(i)} = f_k(x_{k|k}^{(i)})$ for all $i \in \{1, \ldots, M\}$.
In addition, the probability density $p(Y_k = y_k \,|\, X_k = x)$ can be formulated using the observation model with $k \in \mathbb{N}$ and $x \in \mathbb{R}^n$:

$$p(Y_k = y_k|X_k = x) = p(E_k = (y_k - h(x, t_k))) \tag{2.2.44}$$

where E_k, by analogy with (2.2.28), models a measurement noise process which is commonly assumed to be normally distributed. By means of approximating with a finite set of particles, the weights are given as

$$w_k^{(i)} = \frac{p(Y_k = y_k \mid X_k^{(i)} = x_{k|k-1}^{(i)})}{\sum_{j=1}^{M} p(Y_k = y_k \mid X_k^{(i)} = x_{k|k-1}^{(j)})} \tag{2.2.45}$$

for $k \in \mathbb{N}$ and $i \in \{1, \dots, M\}$.

The particle filter recursively applies a prediction step to the set of particles at each point in time, followed by the determination of the particle weights using (2.2.45). Then the particles are redistributed according to an appropriate resampling strategy. More details on resampling strategies and some extensions of the particle filter are described in [Tatschke 11].

The scheme in Algorithm 1 summarizes the individual steps of the *Sampling Importance Resampling (SIR)* particle filter algorithm.

Algorithm 1 SIR Particle filtering scheme

1. **Initialization** of the particles $\left(x_{1|0}^{(i)}\right)_{1 \le i \le M}$ of the $\mathbb{R}^n$-valued random variable $X_0^{(i)}$ for $1 \le i \le M$ at time t_0.
2. **Filtering step**: Calculation of the weights $\left(w_k^{(i)}\right)_{1 \le i \le M}$ of the predicted particles $x_{k|k-1}^{(i)}$ according to the density $p(Y_k = y_k | X_k^{(i)} = x_{k|k-1}^{(i)})$ using the measurements $y_k \in \mathbb{R}^m$.
3. **Resampling**: Generation of new samples $\left(x_{k|k}^{(i)}\right)_{1 \le i \le M}$ by means of sampling according to the weights $w_k^{(i)}$ based on the previous particles $\left(x_{k|k-1}^{(i)}\right)_{1 \le i \le M}$ such that the probability of choosing the sample $x_{k|k-1}^{(i)}$ corresponds to the weight $w_k^{(i)}$ for all $1 \le i \le M$.
4. **Prediction step**: Application of the dynamic model to all particles $\left(x_{k|k}^{(i)}\right)_{1 \le i \le M}$ in order to predict a new set of samples $\left(x_{k+1|k}^{(i)}\right)_{1 \le i \le M}$.
5. Set $t_k \leftarrow t_{k+1}$ and iterate from step 2.

Different approaches exist in order to enable a point estimation based on the resulting probability densities of the particle filter. For example, the empiric expectation

$$\hat{x}_{k|s} = \sum_{i=1}^{M} w_k^{(i)} x_{k|s}^{(i)} \tag{2.2.46}$$

and the covariance matrix

$$\hat{P}_{k|s} = \sum_{i=1}^{M} w_k^{(i)} \left(x_{k|s}^{(i)} - \hat{x}_{k|s}\right) \left(x_{k|s}^{(i)} - \hat{x}_{k|s}\right)^{\top} \tag{2.2.47}$$

can be extracted from the particle set. However, in general, this approach is only suitable if X_k is approximately normally distributed. If this is not the case, for example due to multi-modality, more sophisticated point estimations should be applied. Within this context, the wide field of pattern recognition opens up, since basically it is necessary to identify significant maxima from the approximated state density. In order to cope with this problem, clustering techniques like Mean-shift, Maximum-Likelyhood- or Variational-Inference methods can be applied, which are capable of dealing with multi-modality.

2.2.2.3 Application of the state estimation

Having introduced some basic methods of state estimation, the remaining open issues concerning the application of the estimation methods can be summarized.

For any concrete dynamic system, the state space needs to be specified as well as the dynamic model and the observation model in order to apply the presented filtering techniques. Furthermore, the involved parameters like the initialization values as well as the process and measurement noise need to be declared.

Regarding the integration of measurements, it should be noted that there might be several different observation models for one specific dynamic system. This is motivated by multi sensor systems that provide measurements from different sensors at a certain time in order to estimate the system state. In this case, the observation models are processed sequentially in the filtering step, such that it is sufficient to specify individual observation models.

Chapter 3

Environment modeling

'Complexity is one of the great problems in environmental design.'

(Christopher Alexander)

Contents

This chapter deals with modeling the vehicular environment. Therefore, the model of the ego-vehicle is introduced. Besides the description of the vehicle as a dynamic system whose parameters are considered over time, an overview of the used sensors is given.

For reasons of simplicity, the individual model components are characterized in their specific reference frame. This, however, requires the definition of relevant coordinate systems that are employed for purposes of perception, cartography and localization. In the following, the most important reference frames are characterized.

3.1 Coordinate systems

Within the context of this work, different coordinate systems are used to describe the individual subsystems with respect to their specific reference frame. In particular, global coordinate systems characterizing the vehicle pose for cartography and localization need to be distinguished from vehicular reference frames used for perception purposes.

3.1.1 World frame

The world frame is given by the World Geodetic System 1984 (WGS84). As a standardized geodetic world coordinate system it allows to describe the position of objects on the earth. The WGS84 reference ellipsoid is employed for approximating the terrestrial surface. Its center is

located in the geocenter, the semi-minor axis is defined by the connection between the geocenter and a pole and the semi-major axis is located in the equatorial plane.
The coordinates of objects are represented by two angular values, namely the geographic *longitude* $\lambda \in [-\pi, \pi]$ and the *latitude* $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ as well as the elevation over the WGS84 ellipsoid. The longitude describes the angle in east-west direction with origin at Greenwich meridian. The geographic latitude represents the angle between the equatorial plane and the normal vector of the ellipsoid at the considered point (cf. Figure 3.1.1a).
The world frame plays a role within the following contexts:

- Positioning measurements based on a *Global Navigation Satellite System (GNNS)* like the *Global Positioning System (GPS)* or the *Real-Time Kinematic Global Positioning System (RTK-GPS)* are given in the WGS84 coordinate system.
- In order to ensure exchangeability and compatibility with tools for the map generation, elements of the digital map are represented in this reference frame. On behalf of the self-localization and the associated map access, an intermediate step is applied: The positions of the relevant map elements are transformed into the navigation frame (see below) in order to enable appropriate caching strategies.
- Finally, the vehicle position calculated by the self-localization is provided to the frame work in the world frame to ensure comparability in a standardized way.

3.1.2 Navigation frame

The navigation frame is a Cartesian metric coordinate system. It results from a local approximation of the reference ellipsoid using a projection onto cylindric, conic or plane surfaces at a specific deployment point. The X-axis points eastwards, the Y-axis points in north direction and the Z-axis completes the right-handed trihedron in the direction of the ellipsoid normal. Within the geodetic context, coordinate systems of this shape exist in many different realizations. For instance, the Universal Transversal Mercator (UTM) system or the Gauss-Krüger system (cf. [Großmann 76, Torge 03, Seidel 06]) represent important examples at which the deployment point is located in the equatorial plane. In both cases, the earth's surface is divided into segments with respect to the longitude. The projection onto the tangential surface is then realized locally for each segment. The Gauss-Krüger transformation maps coordinates from the ellipsoid segment to coordinates in the X, Y-plane of the metric coordinate system with the following properties:

- The transformation is isogonal.
- The equator is mapped onto the horizontal axis of the metric coordinate system.
- The central meridian of a segment is mapped onto the vertical axis of the metric coordinate system.
- The transformation is isometric at least for the central meridian of a segment.

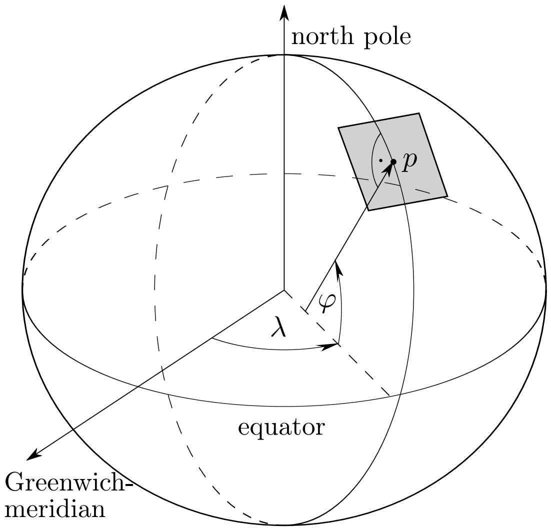
In general, positions in the world frame and the navigation frame can be converted when knowing the modeling parameter of the coordinate systems (c.f. [Großmann 76]). Depending on the chosen deployment point and the point selected for the transformation, positional errors occur during the conversion caused by the model approximation. For the UTM- or the Gauss-Krüger projection, these effects are reinforced at the transitional sections between neighboring segments. For that reason, local topocentric coordinate systems play an important role within the field of geodesy, where the deployment point, the topocenter, is not located at the equatorial plane but lies within the local region of interest. Transformation methods between local topocentric coordinates and geographic coordinates are treated in [Heck 03]. Using this concept, the projection error diminishes when transforming positions close to the deployment point[1]. Therefore, in this thesis, the deployment point is chosen in a local environment of the vehicle. It is determined at the initialization phase of the system using a GPS measurement. Any subsequently occurring transformation between the world frame and the navigation frame is then realized with respect to this deployment point.
In particular, the navigation frame is used for the following purposes:

- For the localization, the vehicle position is modeled in the navigation frame.
- All elements of the digital map provided by the caching strategy are given in this reference frame.
- The association of map elements with vehicular perception results is done in the navigation frame.
- For the generation of continuous map elements like lane representations or road markings, Cartesian metric coordinates are needed. For this purpose, the navigation frame is used as well.

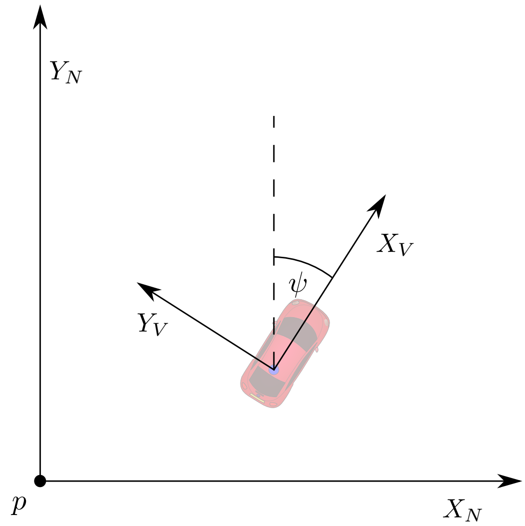
3.1.3 Vehicle frame

The vehicle frame is a local vehicular coordinate system, which means its origin moves with the vehicle. More precisely, it is located at the ground projection of the center of gravity of the vehicle. The X-axis is defined by the longitudinal axis lateral vehicle axis. To complete the Cartesian right-handed system, the Z-axis points upwards.
In order to describe a vehicle pose relative to the navigation frame, rotations with respect to the axis of the vehicle frame are considered as well.
According to the DIN 70000 guidelines ([DIN 01]), the rotation around the X-axis is called *rolling*, around the Y-axis as *pitching* and around the Z-axis as *yawing*. The aim of the ego-motion compensation is the calculation of the roll, pitch and yaw angles in order to account for them in the modeling of the vehicle dynamics. The corresponding roll, pitch and yaw rates describe the angular change per time unit.

[1]According to [Heck 03], the transformation accuracy between local topocentric and geographic coordinates is at the level of a few millimeter if the geographic longitudinal difference between the considered point and the topocenter is below 3°



(a) World frame with angular representation of a point using the longitude λ and the geographic latitude φ. The navigation frame at deployment point p is depicted by the gray tangential surface.



(b) Cartesian navigation frame at deployment point p with Y_N-axis point towards north and X_N-axis pointing eastwards. Furthermore, a local vehicle frame with X_V- and Y_V-axes as well as the yaw angle ψ. Illustration from the bird's eye perspective.

Figure 3.1.1: Relation between individual coordinate systems

In particular, the yaw angle $\psi \in [0, 2\pi]$ plays a decisive role within the scope of navigation, as it describes the orientation of the vehicle regarding the navigation frame. Historically, for that purpose, the angle between north (Y-axis of the navigation frame) and the vehicular longitudinal axis (X-axis of the vehicle frame) is considered in clockwise direction.
If the pose of a vehicle is known regarding the navigation frame, then the coordinates of a point can be easily transformed between the vehicle frame and the navigation frame.
Within this work, the vehicle frame is mainly of interest for the following purposes:

- The geometric model of the lane recognition system is defined in the vehicle frame.
- The coordinate system is an intermediate frame for the transformation of points between the navigation frame and some sensor specific reference frames. In particular, this is the case for cartography and localization.
- For the association of vehicular perception results and components of the digital map, the relevant sensor measurements are given in the vehicle frame.

3.1.4 Sensor specific frames

Besides the above mentioned reference frames, each of the sensors mounted on the vehicle possesses some specific coordinate systems, which are not presented in detail at that point. Knowledge on the mapping properties and the pose of a sensor allows the interpretation of its measurements regarding the vehicle frame if necessary and thus enables the

3.2 Ego-vehicle model

Having defined the relevant reference frames, the ego-vehicle model can be introduced in this section.
Modeling the vehicle motion within the context of dynamic systems has been widely discussed in literature [Mitschke 04, Gillespie 92], as it is the base for vehicle control and the interpretation of sensor data.
Essentially, the parameters modeling the ego vehicle refer to its pose and its dynamics. Within the context of this thesis, the commonly known *bicycle model* has been chosen to describe the motion of the ego vehicle [Riekert 40]. This model assumes a steerable front axis and a fixed rear axis at a constant distance. Furthermore, a constant steering angle is supposed within any observation interval. On these conditions, for any non-zero steering angle, the vehicle moves on a circular course, which motivates observing the curvature of the circle within the ego vehicle model. Figure 3.2.2 illustrates the bicycle model and its relevant parameters. The vectors v_f and v_b describe the velocity at the front and at the back wheel, respectively. The point S denotes the center of mass where the resulting velocity vector v of the vehicle is marked. It is easy to see that considering the equation

$$\tan(\delta) = \frac{l}{\sqrt{r^2 - l_b^2}} \tag{3.2.1}$$

with a constant steering angle δ leads to circular driving around C with radius r. Typically, one distinguishes between the orientation angle of the vehicular velocity vector, the so-called *course angle*, and the orientation of the vehicular longitudinal axis, called the *yaw angle*. According to [Riekert 40], the *slip angle* β describing this angular difference depends on the steering angle and the slip angle at the front and back wheel.
The considered vector describing the vehicle state at time $t_k \in \mathbb{R}$ can be summarized in the following way:

$$\boldsymbol{x}_k = \begin{pmatrix} x_k \\ y_k \\ \psi_k \\ v_k \\ c_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} \text{position x-coordinate} \\ \text{position y-coordinate} \\ \text{yaw angle} \\ \text{absolute value of the velocity} \\ \text{curvature of the circular track} \\ \text{slip angle} \end{pmatrix} \in \mathbb{R}^6 \tag{3.2.2}$$

All of these parameters refer to the navigation frame described in Section 3.1.2. As mentioned above, the curvature is trivially related to the radius r_k of the circular track by $|c_k| = \frac{1}{r_k}$ for any non-zero r_k (which is given under real conditions).

3.2.1 Dynamic model of the ego-vehicle

The dynamic model is dedicated to the pose prediction of a vehicle. A comparison of common dynamic models used for vehicle tracking is given in [Schubert 08]. For our purposes, the constant yaw rate and velocity model has been chosen, also called constant turn rate and velocity model

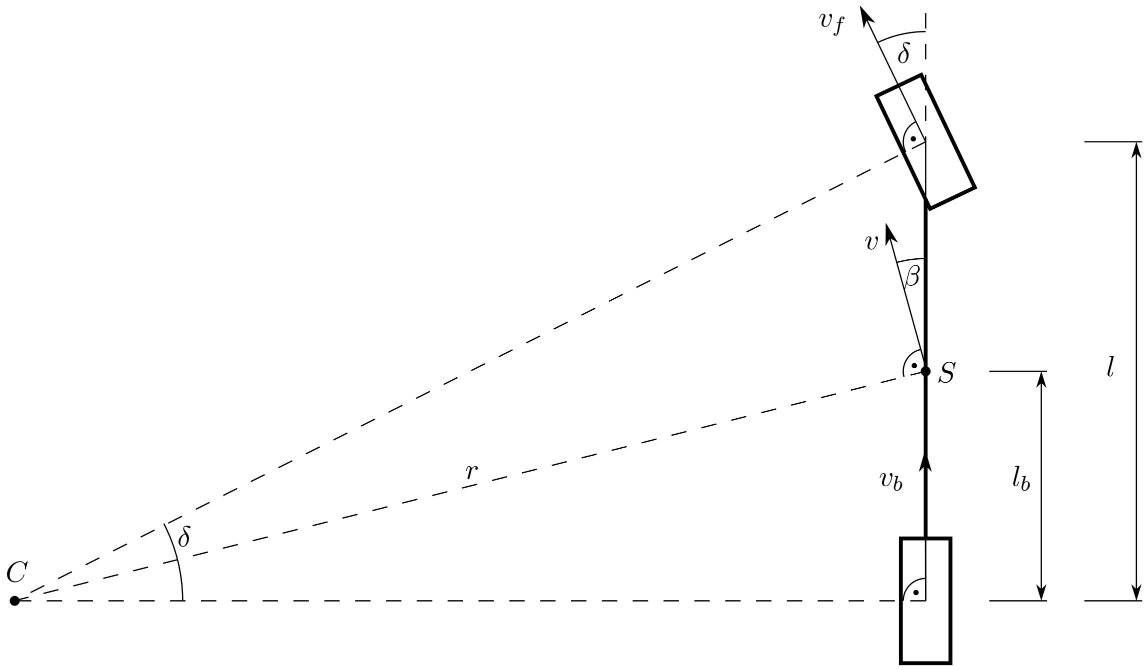


Figure 3.2.2: Bicycle model of the ego-vehicle. For any non-zero steering angle, the vehicle is moving on a circular track with radius r around the center C. In this case, the course angle of the vehicle does not equal the vehicle orientation (yaw angle) but it deviates by the slip angle β.

(CTRV). As mentioned above, these assumptions lead to a circular track of the vehicle. The following relations mainly refer to the notation in [Tatschke 11].
Let $\Delta t_k := t_{k+1} - t_k$ be the difference between any two points in time and v_k be the velocity at time t_k. Then the length of the circular arc the vehicle is driving on is given by $l_k = v_k \Delta t_k$. Using the curvature c_k of the circular track, the change of orientation can be expressed by

$$\Delta\psi_k = l_k c_k. \tag{3.2.3}$$

Thus, the predicted yaw angle for time t_{k+1} is

$$\psi_{k+1} = \psi_k + \Delta\psi_k = \psi_k + v_k c_k \Delta t_k. \tag{3.2.4}$$

Based on the change of orientation, the relative positional translation can be formulated (regarding the vehicle frame 3.1.3):

$$\begin{aligned} \Delta x_k &= \frac{\sin(\Delta\psi_k)}{c_k} \\ \Delta y_k &= \frac{1 - \cos(\Delta\psi_k)}{c_k} \end{aligned} \tag{3.2.5}$$

for any non-zero curvature c_k. In order to calculate the predicted vehicle position regarding the navigation frame, the relative positional changes need to be added to the position at time t_k, taking into account the course of the vehicle:

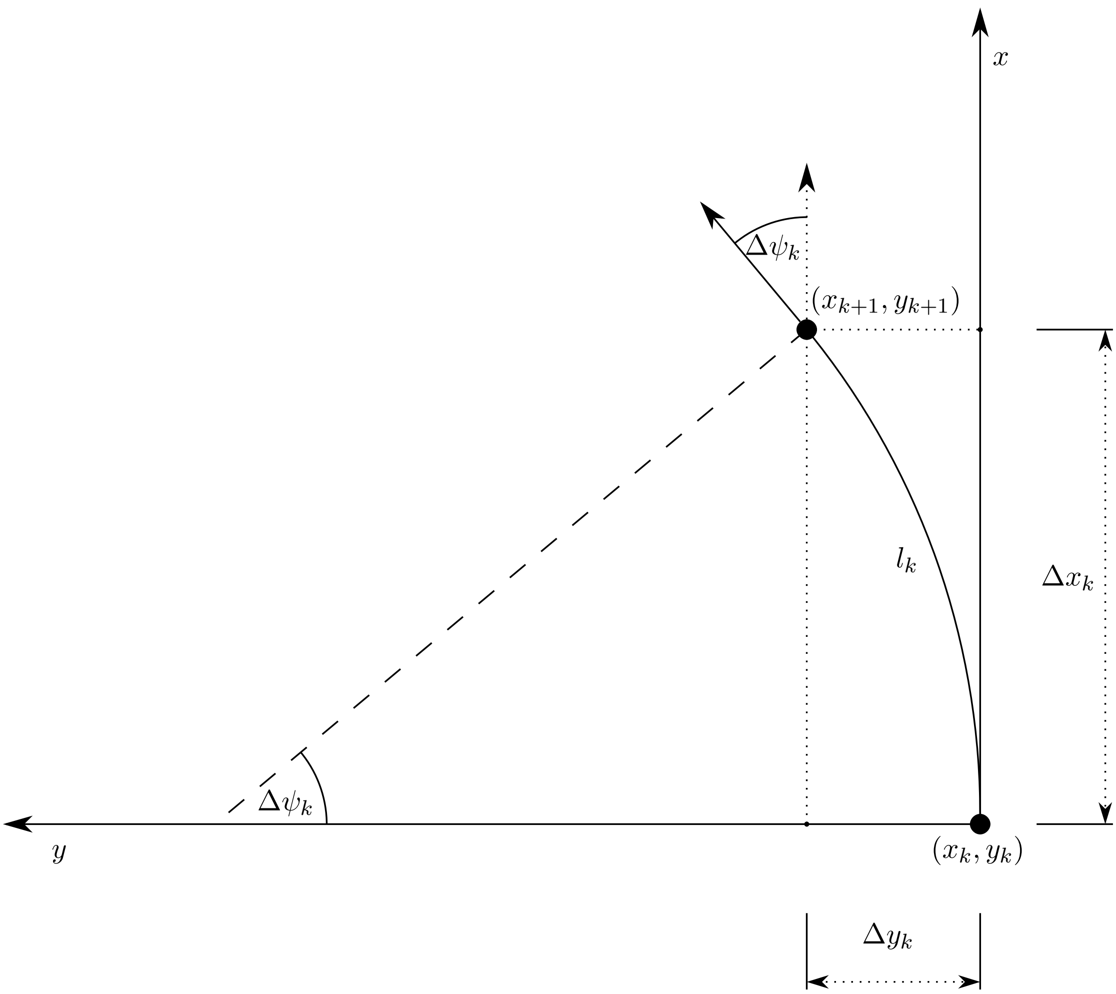


Figure 3.2.3: Prediction of the position (x_{k+1}, y_{k+1}) based on the bicycle model. The axes labeling refers to the vehicle frame at time t_k. The relative position changes $\Delta x_k, \Delta y_k$ and the relative orientation change $\Delta\psi_k$ are used for the pose prediction regarding the navigation frame.

For any $\alpha \in [0, 2\pi]$, let

$$R(\alpha) := \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{3.2.6}$$

be the corresponding rotation matrix. According to the definitions of the vehicle and the navigation frame as well as the yaw angle ψ_k (cf. Sections 3.1.2 and 3.1.3), the predicted vehicle position at time t_{k+1} is given by

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \end{pmatrix} = \begin{pmatrix} x_k \\ y_k \end{pmatrix} + R(\frac{\pi}{2} - \vartheta_k) \begin{pmatrix} \Delta x_k \\ \Delta y_k \end{pmatrix} \tag{3.2.7}$$

where $\vartheta_k := \psi_k + \beta_k$ is the course of the vehicle.

To summarize, the dynamic model of the ego-vehicle can be expressed (in accordance to the definitions in 2.2.1) as a function $f : \mathbb{R}^6 \times \mathbb{R}^2 \to \mathbb{R}$ with

$$f\left(\begin{pmatrix} x_k \\ y_k \\ \psi_k \\ v_k \\ c_k \\ \beta_k \end{pmatrix}, t_k, \Delta t_k\right) = \begin{cases} \begin{pmatrix} x_k + \Delta x \sin(\vartheta_k) + \Delta y \cos(\vartheta_k) \\ y_k - \Delta x \cos(\vartheta_k) + \Delta y \sin(\vartheta_k) \\ \psi_k + v_k c_k \Delta t_k \\ v_k \\ c_k \\ \beta_k \end{pmatrix} & \text{for } c_k \neq 0 \\ \begin{pmatrix} x_k + \cos(\vartheta_k) v_k \Delta t_k \\ y_k + \sin(\vartheta_k) v_k \Delta t_k \\ \psi_k \\ v_k \\ c_k \\ \beta_k \end{pmatrix} & \text{for } c_k = 0 \end{cases} \tag{3.2.8}$$

In addition to the discussed vehicle motion model, there exist several approaches for a partial or full ego-motion compensation concerning the translation and the rotation (pitch and roll) of the vehicle. Besides measuring the vehicle acceleration and rotation using an inertial measurement unit, video-based techniques, like optical flow and motion-from-structure, have been developed in order to estimate the motion parameter [Raudies 12, Golban 09]. As this topic is not at the main focus of this thesis, the corresponding compensation techniques are not discussed here.

Chapter 4

Perception

'What you see and what you hear depends a great deal on where you are standing.'

(C. S. Lewis, The Magician's Nephew)

Contents

In this chapter, the basic components of the environment perception system are presented, including the video-based lane recognition, the detection of landmarks using laser scanners, some intrinsic dynamics measurements of the vehicle as well as the localization methods used for cartographic purposes and the rough positioning of the self-localization.
Figure 4.0.1 shows the experimental vehicle, including some sensors relevant to this work:

- Monocular gray value camera behind the windshield, type IDS uEYE UI-6220SE-M-GL
- Laser scanners in the front of the vehicle, type SICK LD-MRS-400001
- GPS localization unit, type Novatel OEMV
- Real-time kinematic inertial measurement unit (RTK-GPS) as reference localization system, type OXTS RT3003

The technical specification of the experimental vehicle and the used sensors can be found in Appendix A.

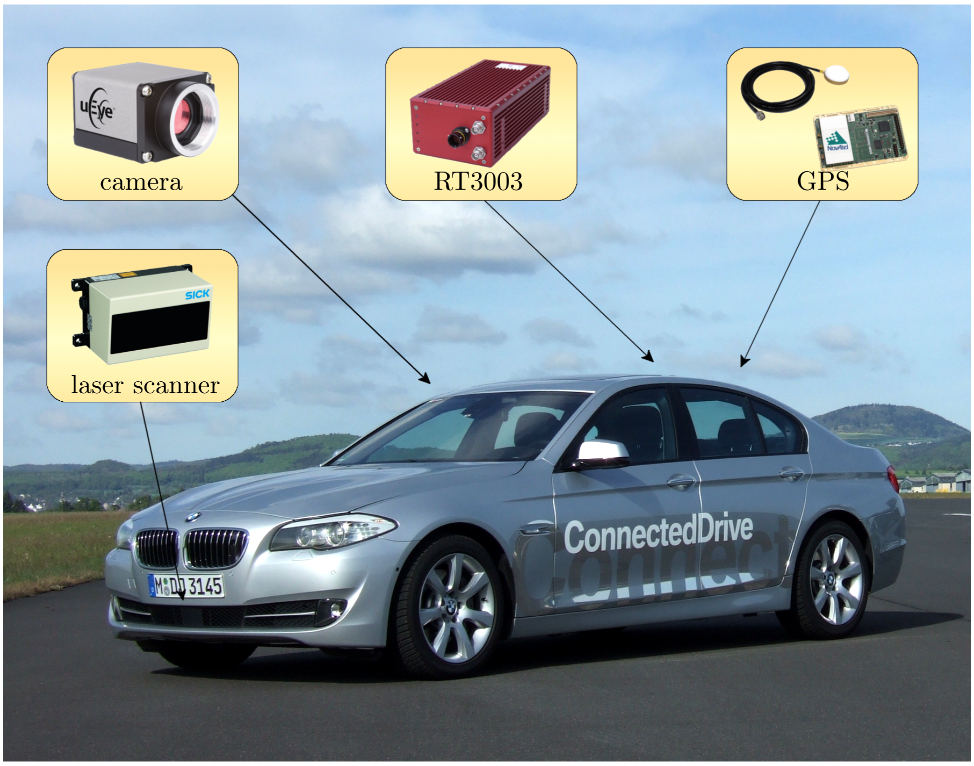


Figure 4.0.1: Experimental vehicle and the sensor setup relevant to this work.

4.1 Temporal and spatial interpretation of sensor data

In order to interpret the available sensor data appropriately, a temporal and spatial alignment within the overall system is crucial.

The spatial alignment of sensors with respect to the experimental vehicle is either known by the installation position or determined using appropriate calibration methods. Within this context, a distinction is made between the intrinsic calibration, where the basic mapping parameters of a sensor are calculated, and the extrinsic calibration, where the aim is to determine the pose of a sensor. In the present case it is assumed that all considered sensors are well-calibrated and, knowing their pose parameter, their measurements can be transformed from a sensor specific coordinate system into the vehicle frame if necessary.

Regarding the interpretation and association of measurements within the overall system, the temporal alignment of sensor data is essential. Due to a sophisticated method described in [Huck 11] and [Westenberger 11], the association of exact timestamps (temporal reference points) to sensor measurements is realized so that for each measurement considered in the following sections a corresponding timestamp $t_k \in \mathbb{R}^+$ is available that represents the beginning of the sensorial acquisition. If any quantity is indexed by some $k \in \mathbb{N}$, then it is related to its value at time t_k.

Furthermore, the camera and all laser scanners are synchronized. That means the sensors start their acquisition simultaneously at a defined point in time controlled by an external trigger signal.

4.2 Lane recognition

The lane recognition is a video-based system that allows the parameter estimation of a geometric road model in the local vehicular environment. For that purpose, the gray-value video camera integrated in the vehicle is used. Some appropriate image processing techniques allow the online extraction of road markings in digital images from the camera. The resulting measurements play several roles within this work: First, they represent the measurements of an observation model in terms of the state estimation (cf. Section 2.2.2) in order to estimate the parameter of the local road model. Second, the road marking measuring points form the data basis for the generation of digital map elements like road marking or lane representations. Finally, the lane recognition plays an essential role for the association with map elements regarding the self-localization approach. In order to apply the image data of the camera for the lane recognition, it is necessary to depict the camera modeling and, beyond that, the used image model. Afterwards, the described contour extraction allows the identification of distinctive gray-value changes in the camera image. Finally, modeling the lane recognition as a dynamic system explains the relation between the parameter estimation and the preceding contour extraction. The basic version of the lane recognition goes back to the system presented in [Tatschke 11]. This approach has been extended regarding the road model and adapted to the present framework requirements.

4.2.1 Camera modeling

A camera is a sensor that measures the light energy emitted by objects within its vision cone in direction of the camera. The light passes through the aperture controlling the exposure as well as through a focusing lens system and finally reaches a photosensitive area. The absorbed radiation creates some charge on the sensor surface, whose quantity can be measured and processed finally. Assuming an extended pinhole camera model (cf. [Hanning 10]), the mapping process can be described as follows:
The camera frame is a three-dimensional Cartesian coordinate system that has its origin within the camera. The X-axis points to the right, the Y-axis downwards and the Z-axis completes the right-handed system by pointing in the direction of the optical axis. Let $S \subset \mathbb{R}^3$ be the vision cone of the camera. Then, $T : S \to (\mathbb{R}^3 \setminus \{0\})$ describes the transformation from the vehicle frame (cf. Section 3.1.3) to the camera frame by rotation and translation. Furthermore, let

$$Pr : \mathbb{R}^2 \times (\mathbb{R} \setminus \{0\}) \to \mathbb{R}^2, \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} \frac{x}{z} \\ \frac{y}{z} \end{pmatrix} \tag{4.2.1}$$

be the projection from the camera frame onto the image plane of the camera. The modeling of optical distortions (for example radial, tangential or thin-prism distortions) can be incorporated by a mapping $\delta : \mathbb{R}^2 \to \mathbb{R}^2$. The transformation $\kappa : \mathbb{R}^2 \to \mathbb{R}^2$ with $a, b, c, u_0, v_0 \in \mathbb{R}$ converts camera coordinates (in metric units) to image coordinates (in pixel units):

$$\kappa : \begin{pmatrix} u \\ v \end{pmatrix} \mapsto \begin{pmatrix} a & c \\ 0 & b \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \tag{4.2.2}$$

The resulting camera mapping $K : S \to B$, which transforms any point $p \in S$ of the vision cone of the camera to a point in the image rectangle $B \subset \mathbb{R}^2$ within the image plane, can be summarized by

$$K : p \mapsto (\kappa \circ \delta \circ Pr \circ T)(p). \tag{4.2.3}$$

The sensor surface of the camera is usually organized in a matrix-like arrangement of individual sensors, the so-called hardware pixels. The light energy inciding on an individual sensor is integrated spatially and temporally (according to the exposure). After the quantization of these individual measurings to a finite number of discrete values $\{0, \ldots, 2^d - 1\}, d \in \mathbb{N}^+$, the digital camera image can be defined: Each hardware pixel of the $(w \times h)$-matrix arrangement with $w, h \in \mathbb{N}^+$ is associated with a pixel coordinate pair $(p_1, p_2) \in Pix := \{0, \ldots, w-1\} \times \{0, \ldots, h-1\}$. Finally, any function $g : Pix \to \{0, \ldots, 2^d - 1\}$ describes a *digital image*, which, in case of a gray-value camera, assigns a gray-value intensity to each pixel coordinate.
Let the mapping $\nu : B \to Pix$ denote the identification of hardware pixels with pixel coordinates mentioned before. Then each point $p \in S$ within the vision cone of the camera can be associated with the pixel coordinate $(\nu \circ K)(p)$.
Besides this projection, the corresponding reprojection is often of interest for image processing purposes as well. The latter maps pixel coordinates to view rays.
Let $T^{-1} : T[S] \to S, \delta^{-1} : \mathbb{R}^2 \to \mathbb{R}^2$ and $\kappa^{-1} : B \to \mathbb{R}^2$ denote the inverse mapping of T, δ and κ, and let

$$Pr^\star : \mathbb{R}^2 \to \mathbb{R}^2 \times (\mathbb{R} \setminus \{0\}), \quad \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{4.2.4}$$

be the reprojective mapping onto the $z = 1$ plane of the camera frame. Using the camera reprojection

$$K^\star : B \to S, \quad b \mapsto (T^{-1} \circ Pr^\star \circ \delta^{-1} \circ \kappa^{-1})(b) \tag{4.2.5}$$

and the focal point $f \in \mathbb{R}^3$, each image point $b \in B$ can be mapped to the view ray

$$R(b) := \left\{ f + \lambda \cdot (K^\star(b) - f) \in \mathbb{R}^3 \,\middle|\, \lambda \geq 0 \right\}. \tag{4.2.6}$$

The parameter calculation of the camera projection and reprojection is called camera calibration. An extensive treatment of this topic can be found in [Hanning 10] or [Hanning 05].

4.2.2 Contour extraction

The bright appearance of road markings usually forms a sharp contrast to the underlying road surface in order to facilitate lane keeping for the driver. This circumstance can be used for detecting road markings using image processing techniques. Within this context, the extraction of distinctive gray-value changes in the camera image plays a decisive role. Since the extraction of contours is used in a wide range of image processing applications, there exist numerous definitions and extraction approaches for contours (cf. [Smith 07, Kellner 06, Haralick 92]).
In the present case, the methods described in [Haas 00] and [Pisinger 03] are applied. The core algorithm can be summarized as follows:

1. **Extraction of contour points**
   A contour point is a pixel coordinate located on a distinctive gray-value change in the image. For any gray-value image, the extraction step uses some discrete gradient filters to result in a set of contour points.

2. **Thinning and decomposition into unbranched contour point sequences**
   Using a context sensitive thinning method, the contour points determined this way are processed and decomposed into connected contour point sequences. In order to avoid outliers and isolated points as well as for focusing on prominent components, some further filtering steps of the point sequences are applied.

3. **Polygon fitting**
   The individual contour point sequences are approximated by polygons enabling a compact encoding of the resulting contours.

The result of the contour extraction is shown in Figure 4.2.2b in an exemplary manner. Using the terms of Section 2.1.1.5 and 2.1.1.6, the resulting contour $C \subset \mathfrak{P}^\infty$ approximated by polygons is picked up again within the scope of the observation model of the lane recognition.

4.2.3 Local road model

During the last decades, a multitude of approaches to video-based lane recognition systems has been proposed in the literature. As a base for many lane keeping applications, some of these methods have reached their maturity phase in the automobile industry. A survey of different modeling approaches is given in [McCall 06], some examples are [Tatschke 08, Polychronopoulos 07, Dickmanns 92]. Most of these approaches are based on the visibility of road markings, which form a strong contrast to the road surface. Besides, there exist some methods dedicated to the detection of the drivable area on unmarked roads (cf. [Schindler 09, Franke 07]). The geometric road model used in this work is widely used in the literature and proved of value in many lane recognition systems. The basic version of the model goes back to work in [Dickmanns 92], which is also shown in detail in [Tatschke 11]. In this approach, the geometry of the locally considered road markings is described by an approximated clothoid model (cf. Section 5.2.2). Using a small-angle approximation and the Taylor approximation of the clothoid model, the ego-lane is represented by a parametrization of the left and the right road marking[1], respectively:

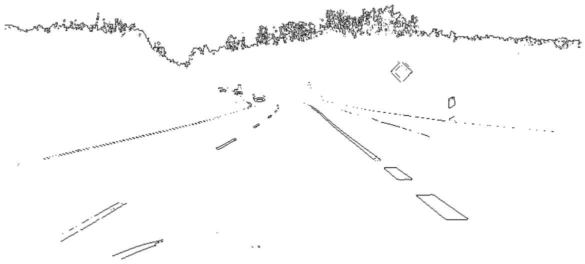
$$m_{left} : [0, L] \to \mathbb{R}^2, \quad l \mapsto \begin{pmatrix} l \\ o_y + \frac{w}{2} + \varphi l + \frac{c_0 l^2}{2} + \frac{c_1 l^3}{6} \end{pmatrix} \tag{4.2.7}$$

with $L \in \mathbb{R}^+$ and

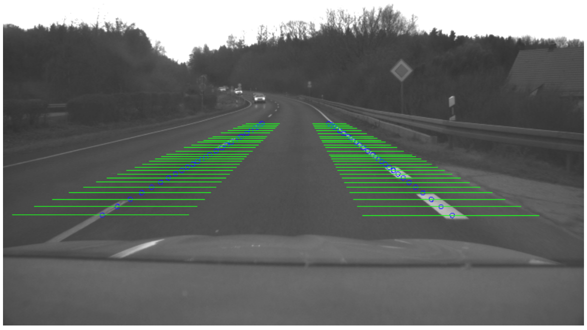
$$m_{right} : [0, L] \to \mathbb{R}^2, \quad l \mapsto \begin{pmatrix} l \\ o_y - \frac{w}{2} + \varphi l + \frac{c_0 l^2}{2} + \frac{c_1 l^3}{6} \end{pmatrix}. \tag{4.2.8}$$

[1]Since, in reality, any road marking has a certain width, its geometric description refers to the middle of the road marking.

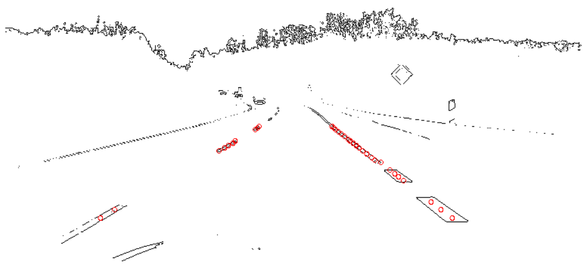
(a) Input image



(b) Contour extraction



(c) Predicted measurements $Y_{FS}^{(P)}$ (blue) and corresponding search paths (green)



(d) Extracted real measurements Y_{FS} (red) using Algorithm 2

(e) Residual (yellow) of the correspondences between predicted and real measurements

(f) Detail of Figure 4.2.2e

Figure 4.2.2: Processing steps of the lane recognition

Here, $o_y \in \mathbb{R}$ represents the lateral offset of the vehicle with respect to the middle of the lane and $w \in \mathbb{R}$ is the lane width. The quantities $c_0 \in \mathbb{R}$ and $c_1 \in \mathbb{R}$ are called starting curvature and curvature change rate, respectively, since the curvature $c : \mathbb{R} \rightarrow \mathbb{R}$ of a clothoid changes linearly with its argument: $c(l) = c_0 + l \cdot c_1$. The angle $\varphi \in [0, 2\pi]$ describes the difference between the orientation of the vehicle and the tangential direction at the starting point of m_{left}.

The individual components of the equations above refer to the X, Y-plane of the vehicle frame (cf. 3.1.3). Figure 4.2.3 illustrates the local road model.

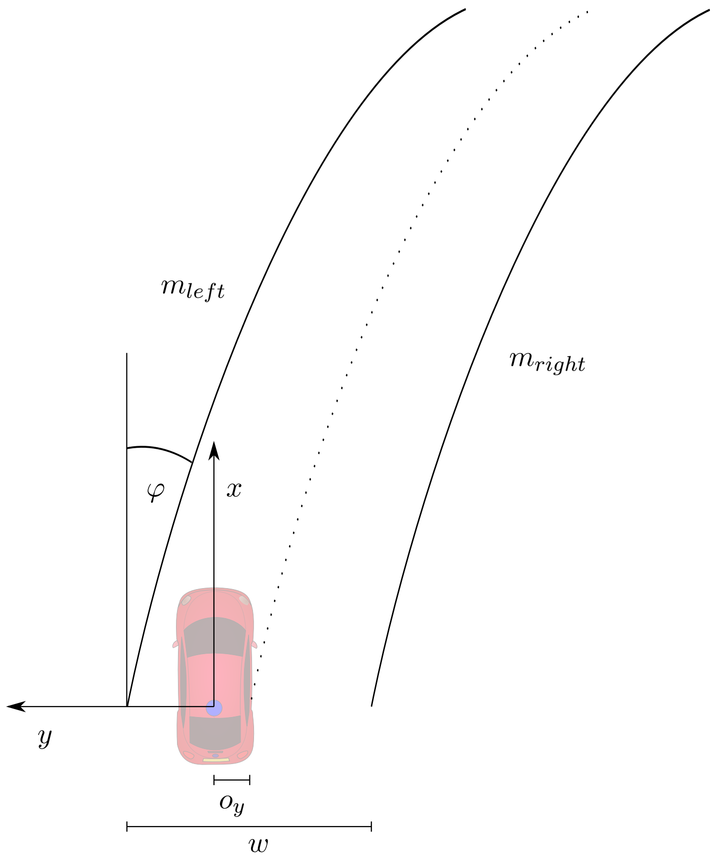


Figure 4.2.3: Geometric road model with lateral offset o_y, lane width w, curvature parameters c_0 and c_1 as well as the relative orientation angle φ. The axes refer to the vehicle frame.

To summarize, the state vector

$$\boldsymbol{x} = (o_y, w, c_0, c_1, \varphi)^T \tag{4.2.9}$$

describes the parameters that are estimated using the techniques presented in chapter 2.2.2.

4.2.3.1 Dynamic model of the lane recognition

Assuming that all considered parameters remain constant within any observation interval Δt, a corresponding linear dynamic model can be formulated as follows:

$$f : \mathbb{R}^5 \times \mathbb{R}^2 \to \mathbb{R}^5, \quad (\boldsymbol{x}, t_k, \Delta t_k) \mapsto \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & v_k \Delta t_k & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \boldsymbol{x}, \tag{4.2.10}$$

where the velocity of the vehicle at time t_k is denoted by v_k. In the strict sense, the dynamic model of the ego-vehicle (cf. Section 3.2) would have to be considered in order to predict the relative pose parameter o_y and φ. However, this aspect is omitted at that point since the local road model is defined with respect to the vehicle instead of a definition in the navigation frame. Based on empirical investigations, no significant losses due to this simplification have been observed.

4.2.3.2 Observation model of the lane recognition

The integration of measurements into the filtering process requires the definition of an observation model. In this case, determining a predicted measurement based on a predicted state is realized by sampling the geometric model at predefined points followed by their projection into the camera image according to Section 4.2.1. For that purpose, a finite sampling sequence $I \subset [0, L]$ of the road model is considered. This allows defining the observation model as a family of functions $(h_{left,i})_{i \in I}, (h_{right,i})_{i \in I} : \mathbb{R}^5 \times \mathbb{R} \to B$ for any state $\boldsymbol{x}_k \in \mathbb{R}^5$ at time t_k and $i \in I$ with

$$h_{left,i}(\boldsymbol{x}_k, t_k) = K \begin{pmatrix} m_{left}(i) \\ 0 \end{pmatrix} \quad \text{and} \quad h_{right,i}(\boldsymbol{x}_k, t_k) = K \begin{pmatrix} m_{right}(i) \\ 0 \end{pmatrix}. \tag{4.2.11}$$

Consistent with the definitions in Section 2.2.2, the corresponding predicted measurements result from applying the observation models to a given predicted state $\boldsymbol{x}_k \in \mathbb{R}^5$. With

$$Y_L^{(P)} := \{h_{left,i}(\boldsymbol{x}_k, t_k) \mid i \in I\} \quad \text{and} \quad Y_R^{(P)} := \{h_{right,i}(\boldsymbol{x}_k, t_k) \mid i \in I\}, \tag{4.2.12}$$

the image points

$$Y_{LR}^{(P)} := Y_L^{(P)} \cup Y_R^{(P)} \tag{4.2.13}$$

form a set of predicted measurements of the lane recognition at time t_k. In order to calculate the residual needed for the filtering step, each predicted measurement must be associated with a real measurement. As indicated in Section 4.2.2, for that purpose, the extracted polygonal representation of image contours is picked up again. For each predicted measurement, a search path in the form of a line segment in the image coordinate system is defined first. Its starting point and endpoint are related to the corresponding sampling position of the road model in the following way: For a parameter $d \in \mathbb{R}^+$ and for all $i \in I$, the search path on the left side of the road model with

$$a_{left,i} = K \left(\begin{pmatrix} m_{left}(i) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ d \\ 0 \end{pmatrix} \right) \in \mathbb{R}^2, \qquad b_{left,i} = K \left(\begin{pmatrix} m_{left}(i) \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ d \\ 0 \end{pmatrix} \right) \in \mathbb{R}^2 \tag{4.2.14}$$

is given by

$$s_{left,i} := \{(1-\lambda) \cdot a_{left,i} + \lambda \cdot b_{left,i} \mid \lambda \in [0,1]\} \tag{4.2.15}$$

and analogously for the right side of the road model. The predicted measurements including their corresponding search paths are depicted in Figure 4.2.2c. For each predicted measurement $p \in Y_{LR}^{(P)}$, Algorithm 2 extracts a real measurement, if applicable. Therefore, the intersection points of the corresponding search path and the contour $C \subset \mathfrak{P}^{\infty}$ within the image rectangle $B \subset \mathbb{R}^2$ of the image g are considered first. With the aid of the image gradient $grad_u(g)(x)$ in u-direction of the image coordinate system at the contour intersection point x, it is decided whether the intersection point potentially belongs to the left or to the right edge of a road marking. The basic idea is that a positive u-component of the image gradient direction indicates a gray value change from dark to bright, which rather corresponds to the left edge of a road marking. If there exist suitable contour intersections at both sides of a road marking, their middle point is determined and selected as the corresponding real measurement. However, if a contour intersection point is located only on one of the two sides, then this point is chosen. Finally, if no intersection could be identified at all or if the order of potential edge points is not plausible, then no real measurement can be associated with the predicted measurement. In this case, the filtering update does not account for that predicted measurement at this particular time. In [Tatschke 11], some more heuristic filtering steps, which also validate the local direction of the contour at the intersection, are proposed for the extraction of contour intersections.

Applying Algorithm 2 to all predicted measurements $p \in Y_{LR}^{(P)}$ results in a finite set of corresponding real measurements $Y_{LR} \subset B$, which is depicted exemplarily in Figure 4.2.2d. Finally, this association allows the calculation of the filter residual and the measurement update (cf. Figure 4.2.2e and 4.2.2f). Furthermore, the real measurements are used for the generation of digital map elements and they serve for the association with road markings of the map in the context of the self-localization.

The process and measurement noise covariance matrices modeling the noise processes within the state estimation need to be determined by system identification techniques. Some appropriate methods on this topic are found in [Soderstrom 89, Goodwin 77, Graupe 72, Eykhoff 74, Walter 97].

Remark 4.2.3.1

A video-based method for detecting and classifying painted arrow markings on the road is proposed in [Maier 11a]. After an identification step of arrow marking candidates in the input image, the contour is extracted and reprojected onto the X, Y-plane of the vehicle frame. Then, a curve-based classification approach is applied in order to compare the reprojected contour with some standardized arrow prototypes represented as arc splines.

As arrow markings on the road can be seen as landmarks, this method can be used for both enriching digital maps and supporting the self-localization by comparing online detected arrow markings with corresponding elements of the digital map.

Algorithm 2 Measurement association

Input: A predicted measurement $p \in Y_{LR}^{(P)}$ and a contour $C \subset \mathfrak{P}^\infty$ of the image g

Output: A real measurement $q \in B$ (image rectangle B) corresponding to p if applicable

Calculate the search path s corresponding to p according to (4.2.15)
// Determine all contour intersections $S \subset B$ of s within g:
$S = B \ \cap s \ \cap \left(\bigcup_{k \in K} \operatorname{tr}(k)\right)$
// Subdivide S into potential edge measurements of a road marking
$S_{left} = \{x \in S \mid grad_u(g)(x) > 0\}$ resp. $S_{right} = S \setminus S_{left}$
$Q = \emptyset$
if $S_{left} \neq \emptyset$ **then**
  // Calculate the closest point to p within S_{left}:
  $q_{left} = \underset{r \in S_{left}}{\operatorname{argmin}} \|p - r\|$
  $Q = Q \ \cup \{q_{left}\}$
end if
if $S_{right} \neq \emptyset$ **then**
  // Calculate the closest point to p within S_{right}:
  $q_{right} = \underset{r \in S_{right}}{\operatorname{argmin}} \|p - r\|$
  $Q = Q \ \cup \{q_{right}\}$
  // Check the order of the point selections q_{left} and q_{right}:
  if $S_{left} \neq \emptyset \ \wedge \pi_1(q_{right}) < \pi_1(q_{left})$ **then**
    $Q = \emptyset$
  end if
end if
if $Q \neq \emptyset$ **then**
  // Calculate the mean of Q:
  $q = \frac{1}{\operatorname{card} Q} \sum_{r \in Q} r$
  return q
end if
return $\emptyset$

4.3 Intrinsic vehicle measurements

The experimental vehicle used within this scope provides a range of intrinsic measurements giving valuable clues to relevant parameter of the vehicle dynamics. This includes

- the absolute value of the vehicular velocity $v \in \mathbb{R}_0^+$, which is determined using a wheel speed sensor.
- the yaw rate $\dot{\psi} \in \mathbb{R}$ of the vehicle. This quantity describes the rotation rate of the vehicle regarding the Z-axis of the vehicle frame. The yaw rate can be measured by a rate sensor. For any velocity $v \in \mathbb{R}_0^+$, the relation between the yaw rate and the curvature $c \in \mathbb{R}$ of the circular track described in the ego-vehicle model (cf. Section 3.2) is simply

$$c := \begin{cases} \frac{\dot{\psi}}{v} & \text{for } v > 0 \\ 0 & \text{for } v = 0 \end{cases} \tag{4.3.16}$$

- the slip angle $\beta \in [0, 2\pi]$, defined by the angular difference between the vehicle orientation (yaw angle) and its driving direction (course angle). The relation between these terms is explained in Section 3.2.

The CAN-Bus of the vehicle transmits all of these measurements to the relevant processing units.

4.4 Detection of landmarks using laser scanners

In the front area of the vehicle, a total of three laser scanners are available for environment perception purposes. Such a sensor is based on time-of-flight measurements of pulsed laser emissions. Using a rotating mirror, it scans the environment in a fan-shaped way with several planes. The interpretation of reflected echo pulses allows reconstructing 3D measuring points, which in turn yield conclusions on objects in the current scene. Based on that, an adequate processing of this laser data in terms of a grid-based fusion enables the extraction of landmark hypotheses (cf. [Weiss 07, Heenan 05]). Within this thesis, a landmark represents on object of small lateral dimension, like traffic signs, reflection posts or trees. For each acquisition time of the sensor, the processing system applied yields a list of landmark hypotheses $\{p_i \in \mathbb{R}^2 \mid i = 1, \ldots, n \in \mathbb{N}\}$, represented by their coordinates in the X, Y-plane of the vehicle frame.
Due to the measuring principle of the laser scanner, the corresponding data processing does not allow a direct type classification of a landmark. However, at least its diameter can be determined approximately. This quantity, in turn, can ease the association of locally detected candidates and landmarks of the digital map. The automated classification of landmark hypotheses, for example regarding the type of a traffic sign, can be achieved using image processing techniques within the scope of sensor data fusion [Janda 11, Pangerl 10]. These approaches do not only simplify the association of landmarks but also enable the automated enrichment of digital maps with classified landmarks.

4.5 Satellite-based localization

To determine the global position and orientation of a vehicle, basically two satellite-based localization units are available. Both systems provide the vehicle position in the world frame (WGS84-coordinates, described in Section 3.1.1) as well as the yaw angle, which describes the vehicle orientation regarding north (cf. Section 3.1.3).

4.5.1 GPS

The *Global Positioning System (GPS)* enables the calculation of positions based on satellites orbiting the earth. Using radio-communication, these satellites regularly emit their own positions together with a corresponding timestamp. Basically, any GPS device receiving at least four different satellite signals is capable of determining its own position based on the time-of-flight interpretation of the transmitted information (cf. [Hofmann-Wellenhof 03, Strang 08, Grewal 01]). The orientation of a vehicle can be determined using integrated navigation techniques based on the temporal filtering of positions or by considering measurements of an electronic compass.

Thus, the orientation of the vehicle is not measured directly by the GPS system, but it is added to the resulting pose measurements for the sake of simplicity:

$$\begin{pmatrix} \lambda \in [-\pi, \pi] \\ \varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \psi \in [0, 2\pi] \end{pmatrix}_{(GPS)} = \begin{pmatrix} \text{geographic longitude} \\ \text{geographic latitude} \\ \text{yaw angle} \end{pmatrix} \tag{4.5.17}$$

The accuracy of the satellite-based pose estimation is mainly dependent on atmospherical noise effects as well as multi-path propagation errors due to reflecting objects (such as buildings and trees). Subject to these environmental circumstances, the positional accuracy of the standard GPS is up to single- or low two-digit range of meters.
Regarding the self-localization approach in this thesis, the standard GPS is used for the system initialization and a rough positioning. This positioning technique can be considered as a low-cost solution and is thus expected to be integrated in future vehicles on a large scale.

4.5.2 RTK-GPS

The satellite-based positioning results can be enhanced considerably by integrating inertial measurements of the vehicle dynamics and evaluating the carrier phase of the GPS signals (cf. [Wendel 07, Groves 07]). Using an *Inertial Measurement Unit (IMU)*, the rotation rates and accelerations of the vehicle can be measured precisely. By integrating these measurements in the localization algorithm, the so-called *Real-Time Kinematic (RTK)-GPS* system reaches a global accuracy of 2 cm on good terms when using particular differential correction services like [ASCOS 12].
Analogous to the standard GPS, the measurements of the RTK-GPS unit can be summarized to

$$\begin{pmatrix} \lambda \in [-\pi, \pi] \\ \varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ \psi \in [0, 2\pi] \end{pmatrix}_{(RTK-GPS)} = \begin{pmatrix} \text{geographic longitude} \\ \text{geographic latitude} \\ \text{yaw angle} \end{pmatrix} \tag{4.5.18}$$

This high-cost positioning solution is used for the generation of the digital map as well as the assessment of the self-localization approach presented in this work. In the latter case, the RTK-GPS system is used as a high-precision reference system as detailed in [Vogel 07].

Chapter 5

Digital Map

'All the roads we have to walk are winding.'
(Oasis, English rock band)

Contents

Digital maps represent abstractions of the reality. Within this context, a multitude of different peculiarities exist regarding the level of detail and abstraction. This diversity allows focusing on the specifics of the considered environment. Depending on the aim of the map usage, different requirements arise concerning the modeling and the representation of the map data.

In the introductory Chapter 1 the digital map was already presented as an important component of the self-localization. Indeed, the map represents the base for a whole range of driver assistance systems. That way, it does not only contain the map elements relevant for the self-localization presented in this work, but it also provides information that is useful for many further applications.

In order to enable this additional value, a closer look at the requirements arising from different applications is necessary before making decisions on the map modeling. Besides the content-related requirements, some more general aspects have to be taken in account: Mathematical, numerical and algorithmic particularities of the representation and the processing of map data must be considered to ensure an efficient usage of the map.

In Section 5.1, the requirements for the digital maps are collected. The rather informal conditions from the applications' point of view are then translated into technical aspects. After a discussion of common approaches known in the literature in Section 5.2, the map modeling of this work is presented in Section 5.3.

Furthermore, Section 5.4 depicts appropriate methods for the generation of the digital map. It is shown how a curve representation can be calculated whose accuracy is controlled while its efficient processing and a compact data storage is achieved.

The last part is dedicated to concepts of practical usage and fast access to some map information.

5.1 Requirements analysis

In the following, some requirement aspects are shown that arise from map-based driver assistance systems and from which the technical conditions on the map modeling are derived.

5.1.1 Requirements from the applications' side of view

5.1.1.1 Map-based Self-localization

The basic idea of the map-based self-localization is the association of vehicular perception results with map elements in order to infer the pose of the vehicle. For that purpose, the map necessitates an adequate representation of the comparable *landmarks*.
For the matter in hand, this includes storing some information on

- landmarks of low lateral dimension
  - traffic signs
  - reflection posts
  - trees
- road markings, like
  - continuous markings
  - dashed markings
  - arrow markings
  - stop lines

Not only need these elements to be represented geometrically by their shapes and poses. Some additional semantic attributes are also required specifying their real meaning, like the type of a traffic sign.

5.1.1.2 Lane-level accurate information

For many applications in the field of track guiding and navigation, detailed information is required on the lane a vehicle is driving on. Therefore, *individual lanes* need to be stored in the map.
Lane keeping systems use the *curvature information* of lanes in order to generate curve speed warnings. In particular, for automated / autonomous driving, highly accurate reference tracks are necessary. For these purposes, the curvature of a reference lane is commonly used for steering control. In order to avoid a non-smooth steering maneuver, both the precision of curvature and its continuity are desirable.
Furthermore, the interpretation of situations requires information on the *drivable direction* of individual lanes, especially in overtaking scenarios. The allowed driving directions at lane-level need to be provided by the map.
The *elevation profile* of a lane is of multiple importance: If a vehicle is driving towards a camber hiding the upcoming course of the road, an assistance function can react on the resulting increase of safety risk if the digital map allows to suggest the occlusion (cf. figure 5.1.1).

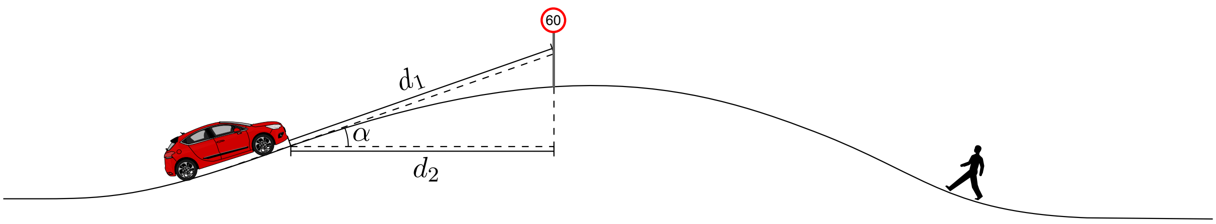


Figure 5.1.1: Occlusion caused by a camber. The lane elevation profile of a digital map enables safety-related assistance functions in this case. Furthermore, the local inclination angle α allows the correction of vehicular measured distances (d_1) to objects with respect to distances regarding the planar surface: $d_2 = \cos(\alpha) \cdot d_1$.

For the interpretation of situations, assumptions on the local road surface are often required. Within this context, an elevation profile stored in the digital map allows a more detailed modeling of the vehicular environment. For instance, some *inclination information* of a lane enables the correction of a vehicular measured landmark position by accounting for the local rise, as depicted in Figure 5.1.1. This correction plays a role in the vehicle self-localization (cf. Section 7.2.3.4).
Finally, the *width of a lane* represents an essential information for assistance functions dealing with the drivable space. Hence, for accuracy reasons, a continuous lane width is preferable to a static value. According to [BFV 93], winding road sections are typically constructed with an extended lane width to provide a safer transition for long vehicles like trucks.

5.1.1.3 Pose prediction

Safety applications like collision avoidance and turn assistance systems require the *temporal and spatial prediction of vehicles* in order to react at an early stage in case of critical situations. Therefore, an estimation of a vehicle's position at a certain point of time is needed. This can be realized by suggesting the anticipated driven path based on the velocity of the vehicle and the time difference considered for the prediction. Taking into account the course of lanes stored in a digital map, a pose prediction of the vehicle is achieved.

5.1.1.4 Map matching

The situation analysis represents an important component in safety related assistance systems. The aim is to assess a traffic situation based on the environment model of a vehicle in order to initiate reactions like warnings, evasion maneuvers or emergency braking. The environment model is typically enriched by the vehicular perception. Moreover, cooperative systems integrate information transmitted from other vehicles or the infrastructure using communication technologies.
In the context of risk assessment of a situation, an important task is to associate vehicles from the environment model to individual lanes of a digital map, which is commonly called map matching. This assignment allows deciding whether a different vehicle is driving on the same lane as the ego-vehicle. Thus, the association of objects to their most likely lanes represents another requirement on the map and on the algorithms processing the map data, respectively. Figure 5.1.2 shows a scenario including the ego-vehicle and three other vehicles driving on a

three-lane road. The association of individual lanes and vehicles is realized by their minimal distance.

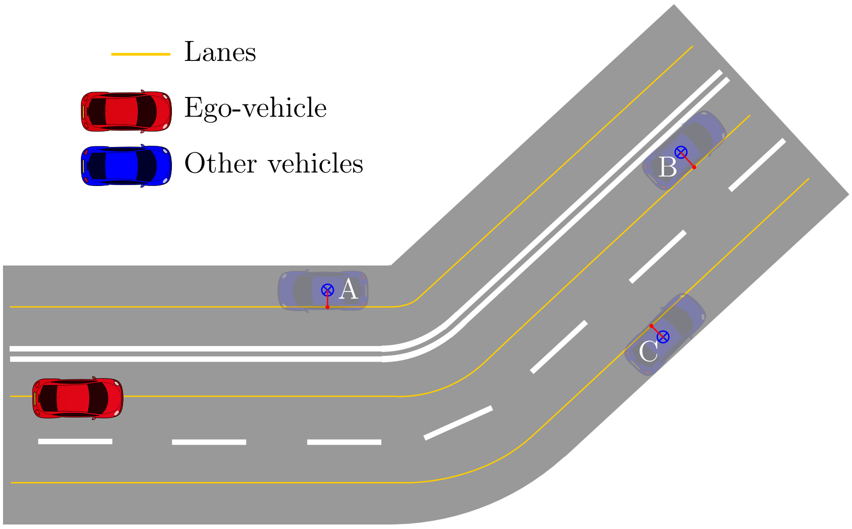


Figure 5.1.2: Map matching of vehicles. Vehicle A is located on the opposite lane of the ego-vehicle, which in turn shares the lane with vehicle B. C is driving on a parallel lane.

5.1.1.5 Intersection assistance

Intersection assistance systems are designed to support the driver for safe transitions at road intersections. In these complex scenarios, the situation analysis is of particular importance. For an automated emergency braking, the distance between a vehicle and elements of the infrastructure, like stop lines or lines of sight, need to be observed over time. In this context, a digital map is useful if it contains such structures in the data base. Furthermore, the distance between a vehicle and, for instance, a stop line on the associated lane must be computable based on the map information.

5.1.2 Requirements from a technical point of view

Based on the requirements from the applications' side of view, some technical aspects are derived in the following.

5.1.2.1 Accuracy of the digital map

As mentioned in the introduction, the required accuracy of the vehicle self-localization is in the range of 1 m regarding the position. As the digital map contains the structures needed by the self-localization, the accuracy of the digital map needs to be at least as high as required by the localization. In order to reduce the sources of error for the localization, it is reasonable to require a considerably higher accuracy of the offline generated map. Assuming that the accuracy of the referenced map elements exceeds the localization accuracy by one order of magnitude, the global precision requirements of the digital map lie in the range of a few centimeter.

However, the question arises which methods allow measuring the actual accuracy of a digital map. For that purpose, Section 7.1.4 lists appropriate approaches for the map evaluation.

5.1.2.2 Curve representation

For continuous elements of the digital map, like road markings or lanes, a curve representation as defined by Section 2.1.1.1 has significant advantages compared to a representation as a finite point sequence:

- Due to the continuity of curves, best approximating points for a given reference point can generally be calculated with an arbitrary accuracy.
- In principle, the data volume for storing a curve is lower than that of a point sequence, as only the defining parameters of the curve need to be represented, in contrast to all elements of a point sequence.

5.1.2.3 Efficient distance calculation

Almost every requirement from Section 5.1.1 implies the calculation of distances between points and continuous structures of the digital map:

- For the self-localization (cf. 5.1.1.1), distances between measuring points arising from the environment perception and continuous representations of road markings must be calculated.
- Likewise, providing lane-level accurate information (cf. 5.1.1.2) requires that the lane of smallest distance to a vehicle position is determined.
- Again, the map matching of vehicles (cf. 5.1.1.4) requires the identification of the lane next to the position of a vehicle. Hence, distances between points and curves must be calculated.
- Finally, the information on distances between vehicles and elements of the infrastructure (cf. 5.1.1.5) immanently requires their calculation.

As a consequence, an essential technical criterion on the digital map is given by the efficient calculation of point to curve distances. Formally, for any point $x \in \mathbb{R}^2$ and a curve ω, the distance $d(x, \omega) = \operatorname{dist}(x, \operatorname{tr}(\omega))$ needs to be determined, that is, a best approximating point $x_0 \in \operatorname{tr}(\omega)$ has to be found such that $\|x - x_0\|_2 = d(x, \omega)$. As it is expected that these calculations are required at high frequency during the runtime of the system, the efficiency of the involved computations is decisive.

5.1.2.4 Data volume

In principle, the data volume needed for storing the digital map should be as small as possible. On the one hand, this is justified by the restricted capacity of the memory medium of the target vehicle. On the other hand, a higher amount of data (regarding the same type of representation)

requires, in turn, a higher computational effort for its processing. However, the processing time is strongly limited due to the real-time conditions of the considered application.
Regarding the curve representation of the digital map, the above mentioned requirement implies an efficient encoding of the curve data. In other words, the curve should be completely represented by as few parameters as possible. Within this context, the *minimum description length* (cf. [Grünwald 05, Rissanen 85]) can be interpreted as the minimal data volume (in bits or the number of parameters) which is required for encoding the curve.

5.1.2.5 Offset curves

As many of the structures, like road markings and lanes on road sections, are widely parallel by means of ε-offset curves (cf. Section 2.1.2.1), the calculation of such curves is essential. That way, the map can be extended with parallel structures in a simple way without the need for an additional data base.
From a technical point of view, determining offset curves for a given representation is a further criterion.

5.1.2.6 Real-time access to map sections

During the runtime of a map-based application, the efficient access to a map section is decisive. This means that all relevant map information has to be provided for any (reasonable) chosen region of interest in real-time. Thus, some appropriate caching strategies are mandatory.

5.1.3 Map modeling approaches in the literature

Depending on the assistance application that the maps are intended to supply, the level of abstraction and thus the representation of the digital map varies significantly in the literature. Besides the explicit modeling of lanes as curves, which is discussed in the following sections, some examples of map modeling approaches are given for both comfort and safety applications.
One of the first comfort applications based on digital maps was for navigation purposes [Groves 07, Grewal 01]. In order to find the shortest or fastest route from one point to another, mainly the distances of locations and connecting roads are of interest. Therefore, an abstract graph-based representation of the map is sufficient and an accurate geometric model is not needed. These kind of maps are enriched by addresses and points of interest to provide comfort functions to the driver. In addition, current traffic information on congestions can be included using communication services.
Going more into detail, maps can contain information on the individual road types and pavement state [Herold 05], which is useful for road maintenance and comfort applications like adaptive light control or entertainment functions. Based on map data, new concepts on increasing fuel efficiency and traffic optimization are developed.
In general, for safety applications, a more detailed representation of maps is needed. Originated from the field of robotic exploration, *Simultaneous Localization And Mapping (SLAM)* approaches (cf. [Wang 11, Nüchter 10, Aycard 10]) are employed for localization and mapping purposes of

road vehicles. SLAM maps are typically represented in a feature-based way, i.e. using attributed point clouds.

Furthermore, there exist map representations using closed polygons [Betaille 10] and extensions to round shapes [López-Pérez 12] in order to describe the drivable area of a road. This modeling allows characterizing complex road boundaries in urban environments.

The ADASIS protocol [Ress 08] defines a standardized access to map data for transportation systems. While the topological correctness and the relative accuracy of the provided information is high, the global accuracy and the level of detail of map elements required for the work at hand can not be provided up to now. Roads are commonly represented by a single curve while their attributes on the number and direction of lanes allow approximating parallel lane instances.

5.2 Discussion of curve models

In this section, several established curve models are discussed regarding their suitability for modeling continuous structures in digital maps. The set of criteria is based on the requirements defined in the previous sections.

5.2.1 Polygons

The most elementary but widely used (cf. [Mattern 10a, Gerlach 09, Noyer 08, Xu 96]) model for continuous structures in digital maps are polygons (cf. Section 2.1.1.6). A survey of methods for the generation of lane representations based on different curve models is given in [Chen 10]. The *OpenStreetMap (OSM)* project [OSM 12] generates and provides map data based on the creative commons license. Its basic geometric primitives comprise *nodes* (points) and *ways* (polygons).

The complexity of calculating distances between points and polygons or determining best approximating points is comparable to the complexity of circular arc splines (cf. Section 2.1.3): After the calculation of the polygon segment that is closest to the given point, the distance calculation can be achieved in a simple closed form.

A polygon does not contain any curvature information since for each segment the curvature disappears. Moreover, any tangent-continuous transition of segments is impossible.

In general, any offset curve of a polygon is an arc spline, as depicted in Section 2.1.2. Thus, polygons are not invariant regarding offsetting.

Since any polygon of segment number $n \in \mathbb{N}$ is already uniquely defined by its starting point, its breakpoints and the endpoint, storing a polygon, requires $2n + 2$ (floating point) numbers.

Concerning the approximation of a point sequence by a polygon, there exist several methods in the literature (e.g. [Douglas 73, Ghosh 91, Kolesnikov 03]). Respecting any predefined tolerance, the *Minimum Link Path (MLP)* ([Suri 86]) yields an approximation with the minimal segment number:

Given a simple closed polygon P (in this case P approximately represents the ε-offset of the input point sequence) with a source and destination vertex, the MLP-algorithm produces a sequence of line segments connecting the source and destination inside P with a minimal number of segments.

In [Pink 10], a method for the generation of polygonal map elements based on aerial images is

proposed. The image regions of road markings that are visible in an aerial image are extracted using image processing techniques, followed by a line fit of the connected components.
Since polygons do not provide any immediate curvature information, their suitability for modeling road sections is limited. In principle, the individual segments could be attributed additionally with separately determined curvature information. However, no continuous curvature can be realized using that model. Curved road sections can only be approximated using a large set of breakpoints, in order to guarantee a given positional accuracy. This, however, increases the data volume of the digital map and thus the computational effort of processing algorithms. Furthermore, the calculation of point to curve distances is hampered since more segments have to be considered for the distance calculation.

5.2.2 Clothoids

According to the road construction regulation in [BFV 93] and [Richter 08], a turn of a typical road in rural areas consists – at least constructionally – of a sequence of a straight line segment, a clothoid, a circular arc, a clothoid and again a straight line segment. Clothoids are usually defined using the Fresnel integrals. For instance, the clothoid starting at the origin with curvature zero is determined by

$$K : [0, L] \to \mathbb{R}^2, \quad K(t) := a \begin{pmatrix} C(t/a) \\ S(t/a) \end{pmatrix}, \tag{5.2.1}$$

$$C(t) := \int_0^t \cos\left(\frac{\pi}{2}s^2\right) ds, \quad S(t) := \int_0^t \sin\left(\frac{\pi}{2}s^2\right) ds, \tag{5.2.2}$$

$$a = \sqrt{\pi L R}, \tag{5.2.3}$$

where $L \in \mathbb{R}$ is the curve length, $R \in \mathbb{R}$ is the curvature radius at the endpoint of the clothoid and $a \in \mathbb{R}$ is a scaling factor. The curvature of a clothoid changes linearly with respect to the arc length. This property is used for the road construction since, that way, clothoids provide a smooth steering phase when passing the lane sections. Regarding the modeling of curvature, clothoids are the best curve model for this kind of road since they equal the constructional model. However, clothoids show certain disadvantages in digital maps: Curve approximations, point to curve distance calculation, the computation of some best approximating points and even the drawing of the curve are computationally expensive as they imply nonlinear optimizations.
In [Walton 05, Walton 90], clothoid splines are treated, which in analogy to Section 2.1.1 represent curves that are piecewise defined by clothoids and some approximation methods based on input polygons are proposed. In general, the offset curve of a clothoid is not a clothoid anymore. However, there exist approximate solutions ([Meek 90]). Regarding the approximation of clothoids and their offset curves by other curve types, a strategy is proposed in [Wang 01] that relies on Bézier curves and B-splines. Furthermore, an arc spline approximation of a clothoid is presented in [Meek 04]. The authors show that, using n-times the number of segments, the approximation error behaves like $O(n^{-2})$.
Regarding the data volume, at least the starting point and the tangential direction at the starting point must be stored together with two values out of either the curve length, the scaling factor or the curvature at the end point.

5.2.3 Polynomial splines

Within this context, planar polynomial splines are curves that are piecewise composed by polynomials of degree $n \in \mathbb{N}$ with values in $\mathbb{R}^2$.
Let $s : [0,1] \to \mathbb{R}^2$ be a polynomial spline that is differentiable at every point. If the best approximating point of an arbitrary point $x \in \mathbb{R}^2$ regarding s is not located on the starting point or on the endpoint of s, then the parameter t_0 of the best approximating point satisfies $\langle \dot{s}(t_0) | x - s(t_0) \rangle = 0$. In general, solving this equation implies finding the roots of a polynomial of degree $2n-1$. For $n = 2$, there exist some closed solving formulas for this problem, which are possibly numerically critical. For $n > 2$, the above mentioned equation is generally not solvable in a closed form at all (cf. [Bosch 06]). Instead, some iterative approximation methods can be applied, like bisection or Newton's method. These approaches are computationally much more expensive than the closed solutions available for line segments or circular arcs.
Basically, polynomial splines of degree n enable $n-1$-times differentiability, which makes them suitable for modeling the curvature as far as $n \geq 3$ is chosen. For $n = 3$, the second derivative is piecewise affine.
Polynomial splines are generally not invariant with respect to offset curves (cf. [Farin 02]).
Regarding the data volume, for any polynomial spline represented with respect to the compact B-Spline basis its breakpoints and the guiding polygon need to be stored. Alternatively, the curve can be reconstructed considering the breakpoints and the n-th derivatives of the spline at the breakpoints. For Bézier curves, the defining guiding points need to be stored.
Road modeling approaches using polynomial splines are found for instance in [Chen 10, Koutaki 06]. There are several methods for the interpolation or approximation with polynomial splines (cf. [De Boor 01, Nürnberger 89]). In this situation, the choice of the breakpoints (or the choice of the B-Spline knot sequence) is crucial for both the accuracy and the total number of segments of the spline. However, the approximation with polynomial splines respecting a given error tolerance and minimizing the number of segments has remained an unsolved problem so far. Since the accuracy and the computational processing effort, which is also affected by the segment number, impact the performance of any map-based application, the commonly known approximation techniques are rethought in favor of a novel approximation scheme based on circular arc splines.

5.2.4 Arc splines

The basic definitions and properties of arc splines have been introduced in Section 2.1.1 and 2.1.2, respectively. The particularly efficient calculation of point to curve distances and best approximating points for arc splines has been shown in Section 2.1.3.
The curvature of an arc spline is a step function. In other words, it is piecewise constant. Since continuity of curvature is of importance for applications like the autonomous driving, some methods that generate this continuity based on arc splines are proposed in Section 5.3.4.4.
As mentioned in Section 2.1.2, offset curves of arc splines are arc splines again and they can be determined directly using the equations in (2.1.18).
For any given point sequence and a chosen tolerance, the *Smooth Minimum Arc Path*-algorithm from [Maier 10] generates a smooth arc spline that minimizes the segment number while not

exceeding a given tolerance. Thus, applying the method described in Section 5.4, the accuracy of the resulting map can be controlled for generating continuous map elements. Moreover, the minimality of segments ensures that the computational effort for processing map data as well as storing the map is minimized regarding any other arc spline approximation technique.

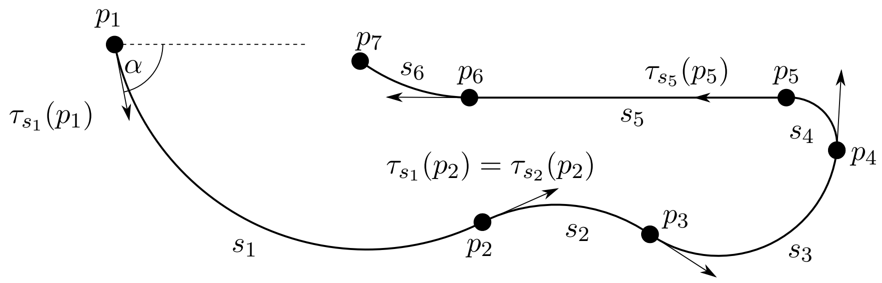


Figure 5.2.3: Smooth arc spline γ with six segments. All segments s_i except for s_5 are circular arcs. The angle α determines the orientation of $\tau_{s_1}(p_1)$. The equality of the tangent unit vectors $\tau_{s_i}(p_i)$ and $\tau_{s_{i+1}}(p_i)$ indicates the smoothness at p_i.

Using the tangential continuity, a smooth arc spline $\gamma = s_1 \dots s_n$ can be stored efficiently, as depicted in Figure 5.2.3 and detailed in [Schindler 12]: Based on the starting point $p_1 = S(s_1)$ and the first breakpoint $p_2 = E(s_1)$ together with the angle of the starting tangential direction $\tau_{s_1}(S(s_1))$, the first segment s_1 is already uniquely determined. Due to the tangential smoothness of γ, one has $\tau_{s_1}(E(s_1)) = \tau_{s_2}(S(s_2))$, which in turn enables the reconstruction of the second segment s_2 by also considering $S(s_2)$ and $E(s_2)$. Iteratively, the whole arc spline γ is encoded uniquely by $n+1$ points (starting point, breakpoints and endpoint) as well as the angle of the starting tangent, which requires $2n+3$ floating point numbers in total. In comparison, a polygon defined by the chords of the segments would require $2n+2$ floating point numbers. Therefore, the following should be noted:

- Due to the increased flexibility of arc splines, a significant reduction of the segment number is generally to be expected when approximating with arc splines in comparison to polygons, given a desired tolerance.

- Justified by the comparatively lower number of segments and the distance calculation in closed form, the computation of best approximating points is in general more efficient compared to polygons.

- Beside the more realistic modeling of road elements, arc splines directly provide curvature information.

In this sense, modeling with smooth arc splines outperforms modeling with polygons with respect to the information content, the data volume and the efficiency of processing.
Furthermore, arc splines comply with the standards of the *Geography Markup Language (GML)* [GML 12] developed by the *Open Geospatial Consortium (OGC)* for modeling geometry within *Geographic Information Systems (GIS)* [Heywood 06, Maguire 97]. Thus, using arc splines as a curve model ensures compatibility with standard GIS tools and map databases.

5.2.5 Selection of arc splines

Under consideration of the requirements analysis in Section 5.1, a final evaluation of the different curve types can be carried out regarding their suitability for modeling in digital maps. Therefore, the alternative curve types are summarized in table 5.2.1 together with their appropriateness regarding the set of criteria.

	Distance calculation	*Offset generation*	*Data volume*	*curvature information*
Polygon	+	+	--	--
Clothoid	--	--	++	++
Pol. spline	-	--	+	+
Arc Spline	++	++	+	+

Table 5.2.1: Summary of the evaluation of different curve types regarding the set of criteria. ++ means: "very suitable", +: "suitable", -: "unsuitable", --: "very unsuitable".

Due to the advantageous properties of arc splines, this curve type is chosen for modeling continuous structures of the digital map in this work.

5.3 Modeling

After the requirements analysis and the justification for the choice of arc splines as curve model, this section deals with the modeling of the digital map. All of the modeled elements consist of a geometric representation and some additional semantic information realized in a list of attributes.

5.3.1 Separation of the elevation

First, all geometric structures are defined two-dimensional in the X, Y-plane of the navigation frame (cf. Section 3.1.2). This modeling already satisfies the requirements of many applications. Additionally, each lane representation is connected to an elevation profile, which is described in more detail in Section 5.3.4.1.

5.3.2 Landmarks

Definition 5.3.2.1 (Landmark)
Any landmark $l = (p_l, A_l)$ consists of a pair of coordinates $p_l \in \mathbb{R}^2$ defining its position and a list of optional attributes that describe the geometry and the type of l in more detail.
Table 5.3.2 gives an overview of the considered attributes in A_l. In Figure 5.3.4, a map section is illustrated including some classified landmarks.
In principle, arrow markings on the road surface are representable by the geometry of their contour. However, since their appearance is standardized by [BFV 93], it is sufficient to describe an arrow marking by its position, its orientation and its type as a landmark.
The set of all landmarks is denoted by $\mathfrak{L}$.

Attribute	*Range*	*Description*
Diameter	$\mathbb{R}^+$	Diameter regarding the X, Y-plane
Height	$\mathbb{R}^+$	Height of the landmark in Z-direction
Orientation	$[0, 2\pi[$	type specific orientation of the landmark; Clockwise angle to the Y-axis
Type	{*tree, reflection_post, stop, give_way, traffic_light, speed_limit_s* ($s \in \mathbb{N}$)*, arrow_marking_left, arrow_marking_right, arrow_marking_forward, arrow_marking_left_right, arrow_marking_left_forward, arrow_marking_right_forward, arrow_marking_left_right_forward*}	Type of the landmark

Table 5.3.2: Attributes of a landmark

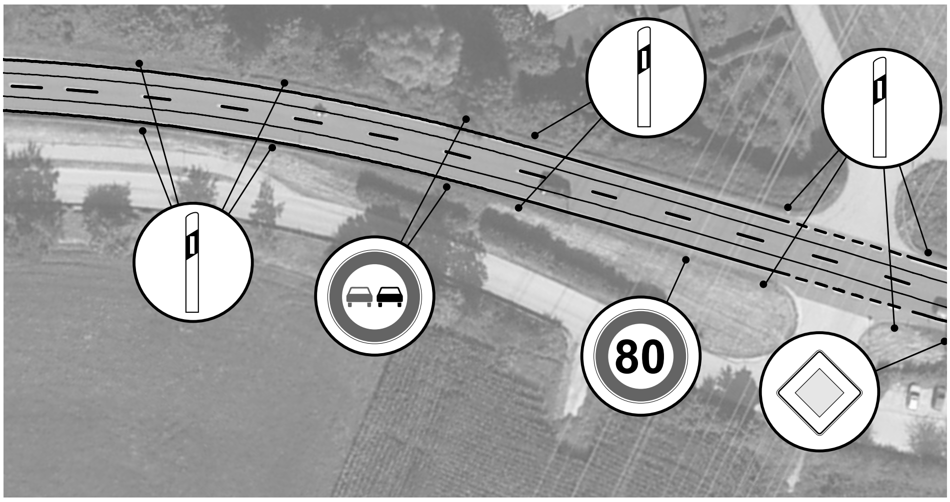


Figure 5.3.4: Landmarks including semantic classification. Aerial image by courtesy of Bayerische Vermessungsverwaltung

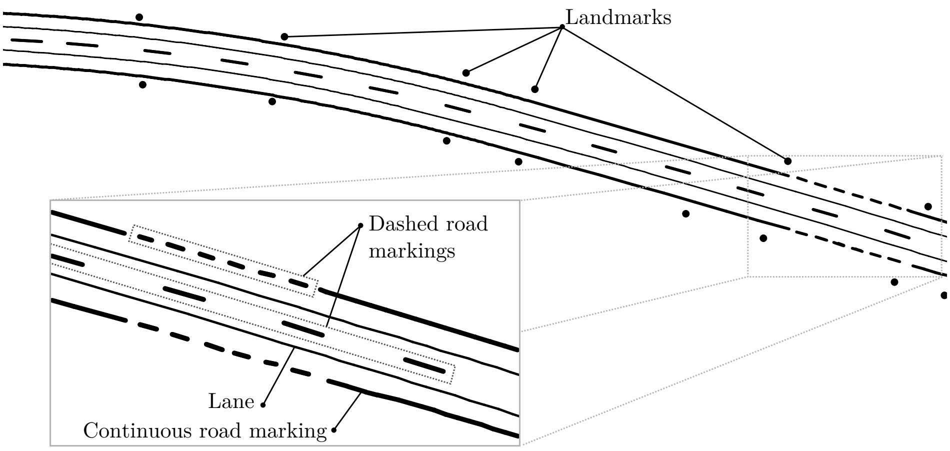


Figure 5.3.5: Map section including lanes, road markings and landmarks.

5.3.3 Road markings

Definition 5.3.3.1 (Road markings)
Any road marking (γ, A) is composed by a smooth arc spline $\gamma \in \mathfrak{S}^{\infty}$ and an optional list of attributes A.

Arc splines are used for modeling both continuous road markings and individual line segments of dashed road markings (cf. Figure 5.3.5). Note that, according to definition 2.1.1.4, line segments are smooth arc splines, as well.

Possible attributes of A are summarized in table 5.3.3. The attribute *Width* refers to the average width of the road marking in the direction of the normal vectors of γ. The road marking type *marking* models a marking in longitudinal direction of a lane, while *stop_ line* represents a stop line, for instance, at an intersection or a junction. In addition, the (virtual) line of sight type *line_ of_ sight* is considered at where a vehicle needs to stop in case that no other traffic rule applies.

The set of all road markings is denoted by $\mathfrak{M}$.

Attribute	*Range*	*Description*
Width	$\mathbb{R}^+$	Average width of the marking
Type	{*marking, stop_line, line_of_sight*}	Type of the marking
Color	{*white, yellow*}	Color of the marking

Table 5.3.3: Attributes of a road marking

5.3.4 Lanes

The model of a lane describes the course of the middle of a real lane.

Definition 5.3.4.1 (Lanes)
A lane (γ, ν, A) consists of two smooth arc splines $\gamma, \nu \in \mathfrak{S}^\infty$ and an optional list of attributes A. The spline γ represents the course of the lane in the X, Y-plane of the navigation frame, where the order (2.1.1) of γ determines the drivable direction[1]. Additionally, the arc spline ν describes the elevation profile of the lane which is depicted in more details in the following sections.
The set of all lanes is denoted by $\mathfrak{R}$.

5.3.4.1 Elevation profile and 3D-curve

According to the definitions in Section 2.1.1.1, let $l_\gamma := \operatorname{len}(\gamma)$ be the length of γ and let $l_\nu := \operatorname{len}(\nu)$ be the length of ν, respectively. The corresponding elevation profile that provides an elevation value for each arc length parameter of the planar lane course γ is a function $h : [0, l_\gamma] \to \mathbb{R}$, for which

$$\left\{(t, h(t)) \in \mathbb{R}^2 \,\middle|\, t \in [0, l_\gamma]\right\} = \operatorname{tr}(\nu) \tag{5.3.4}$$

holds. In general, for an arbitrary arc spline $\nu \in \mathfrak{S}^\infty$, h is only a relation. Since the inclination on real roads is significantly lower than $90°$ with respect to the local tangential plane on earth, it can be assumed that h is well-defined.
In the following, the properties of the chosen modeling are investigated. That way, the differentiability of h can be shown, which characterizes the realism of the proposed curve model: Let $g : [0, l_\gamma] \to \mathbb{R}^2$ be an arc length parametrization of γ and let $n : [0, l_\nu] \to \mathbb{R}^2$ be an arc length parametrization of $\nu = \nu_1 \dots \nu_k$ with breakpoints $b_i := \begin{pmatrix} t_i \\ h(t_i) \end{pmatrix}$ for $i \in \{1, \dots, k-1\}$ with $t_0 := 0$ and $t_k := l_\gamma$.
Let $t \in [0, l_\gamma]$ be arbitrarily chosen.

1. Case: $t \in]t_{i-1}, t_i[$ for $i \in \{1, \dots, k\}$
Let $r > 0, C \in \mathbb{R}^2$ be the radius and the center of ν_i. Then the following is true for

$$F : \mathbb{R}^2 \to \mathbb{R}, \quad (a, b) \mapsto (C_1 - a)^2 + (C_2 - b)^2 - r^2 : \tag{5.3.5}$$

[1]The case of different allowed drivable directions can be modeled by introducing specific attributes in A.

$$F(t, h(t)) = 0 \quad \text{and} \quad D_2F(t, h(t)) = -2(C_2 - h(t)) \neq 0, \tag{5.3.6}$$

according to the assumptions. Otherwise the tangential direction would be parallel to the Y-axis at $(t, h(t))$.

The implicit function theorem (cf. [Forster 11]) guarantees the existence of some neighborhoods $I, J \subset \mathbb{R}$ of t and $h(t)$ as well as a uniquely defined continuously differentiable function $\varphi : I \to J$ with $\varphi(t) = h(t)$. After a possible shrinking of I, the equality $\varphi|_I = h|_I$ results from the uniqueness and thus h is continuously differentiable in t.

2. Case: $t = t_i$ for a $i \in \{1, \dots, k-1\}$

Likewise, we have $D_2F_i(t, h(t)) \neq 0$ and $D_2F_{i+1}(t, h(t)) \neq 0$ for the segments ν_i and ν_{i+1}, respectively.

The implicit function theorem guarantees the existence of some neighborhoods $I_1, I_2 \in \mathbb{R}$ of t and some neighborhoods $J_1, J_2 \in \mathbb{R}$ of $h(t)$ as well as two continuously differentiable functions

$$\varphi_1 : I_1 \to J_1 \text{ with } \varphi_1(t) = h(t) \text{ and} \tag{5.3.7}$$

$$\varphi_2 : I_2 \to J_2 \text{ with } \varphi_2(t) = h(t). \tag{5.3.8}$$

Hence, h is differentiable from the left and from the right. Furthermore, the implicit function theorem assures that

$$\dot{\varphi}_i(a) = -\frac{D_1F_i(a, \varphi_i(a))}{D_2F_i(a, \varphi_i(a))} \text{ for } i \in \{1, 2\}. \tag{5.3.9}$$

Since ν is smooth, the tangential unit vectors of ν_i and ν_{i+1} equal at the breakpoint b_i and $\dot{\varphi}_1(t) = \dot{\varphi}_2(t) = \dot{h}(t)$ holds. Hence, h is differentiable.

The function

$$f : [0, l_\gamma] \to \mathbb{R}^3, \quad t \mapsto \begin{pmatrix} g(t) \\ h(t) \end{pmatrix} \tag{5.3.10}$$

is thus a regular parametrization of the 3D-curve of the lane because its components are smooth and $\dot{g}(t) \neq 0$ for all $t \in [0, l_\gamma]$ and hence $\dot{f}(t) \neq 0$.

That way, the course of elevation ν is modeled separately from γ, which is in accordance to the German guidelines for the road construction [BFV 93]. In particular, $|\gamma| = |\nu|$ does not hold imperatively. Figure 5.3.6 shows an example of the planar course of a lane. The corresponding elevation profile is depicted in Figure 5.3.7. Finally, Figure 5.3.8 shows the resulting 3D-curve, which combines the planar course and the elevation profile.

Calculating the length of a lane section According to the preceding proof, the function $\tilde{n} : [0, l_\gamma] \to \mathbb{R}^2$ with the components $\tilde{n}_1(t) = t$ and $\tilde{n}_2(t) = h(t)$ is a parametrization of ν.

Using the proposed modeling, the length of the elevation profile ν equals the length of the

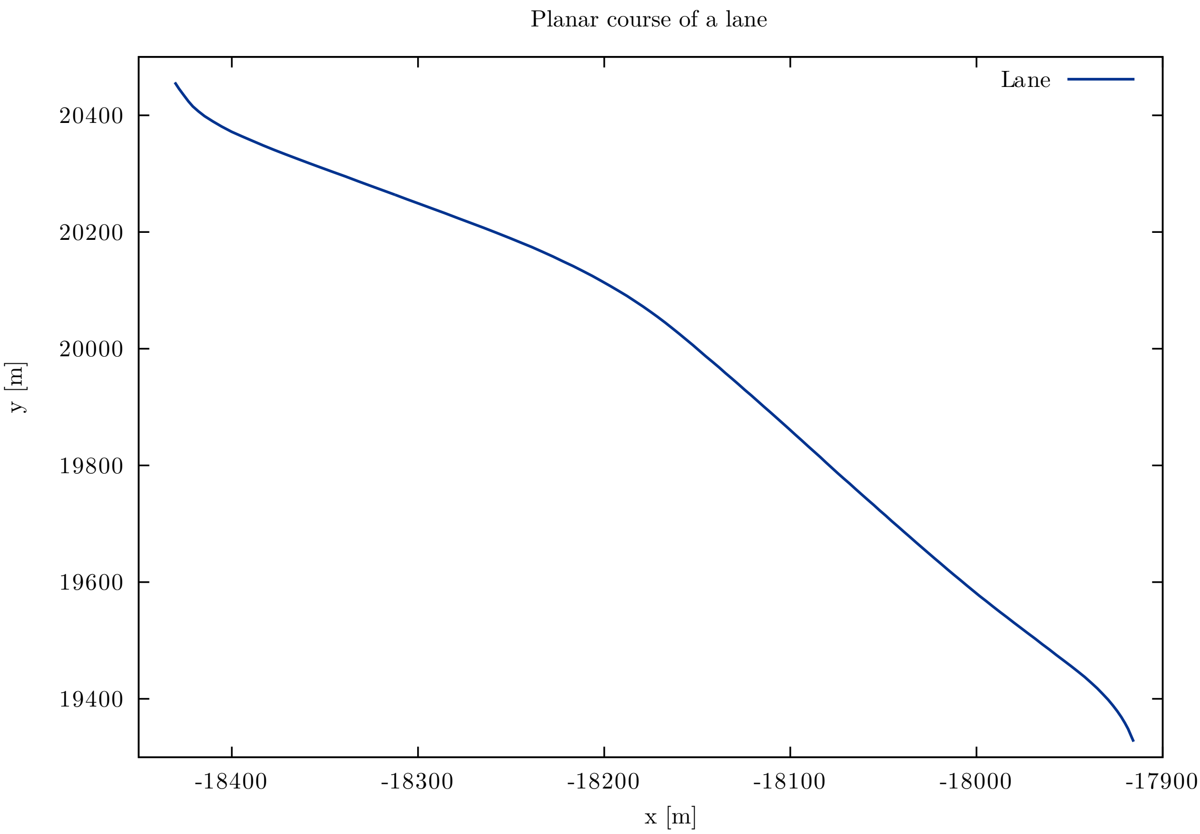


Figure 5.3.6: Planar course of a lane in the X, Y-plane of the navigation frame (arc spline γ).

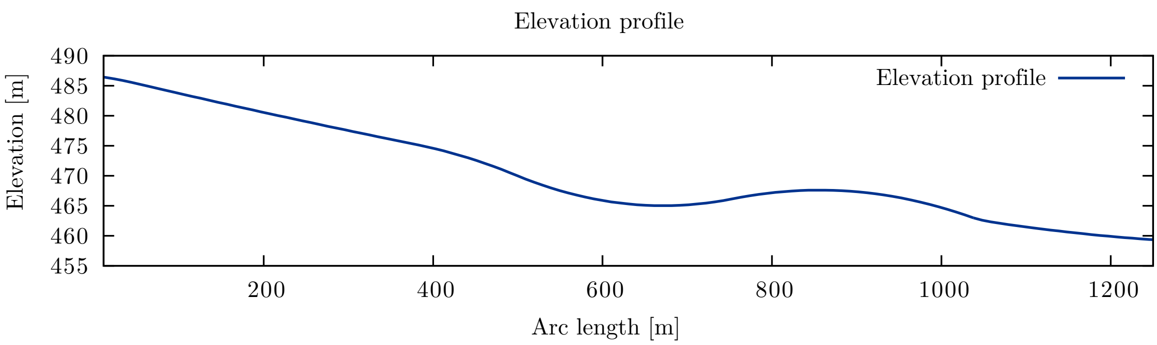


Figure 5.3.7: Elevation profile of a lane (arc spline ν).

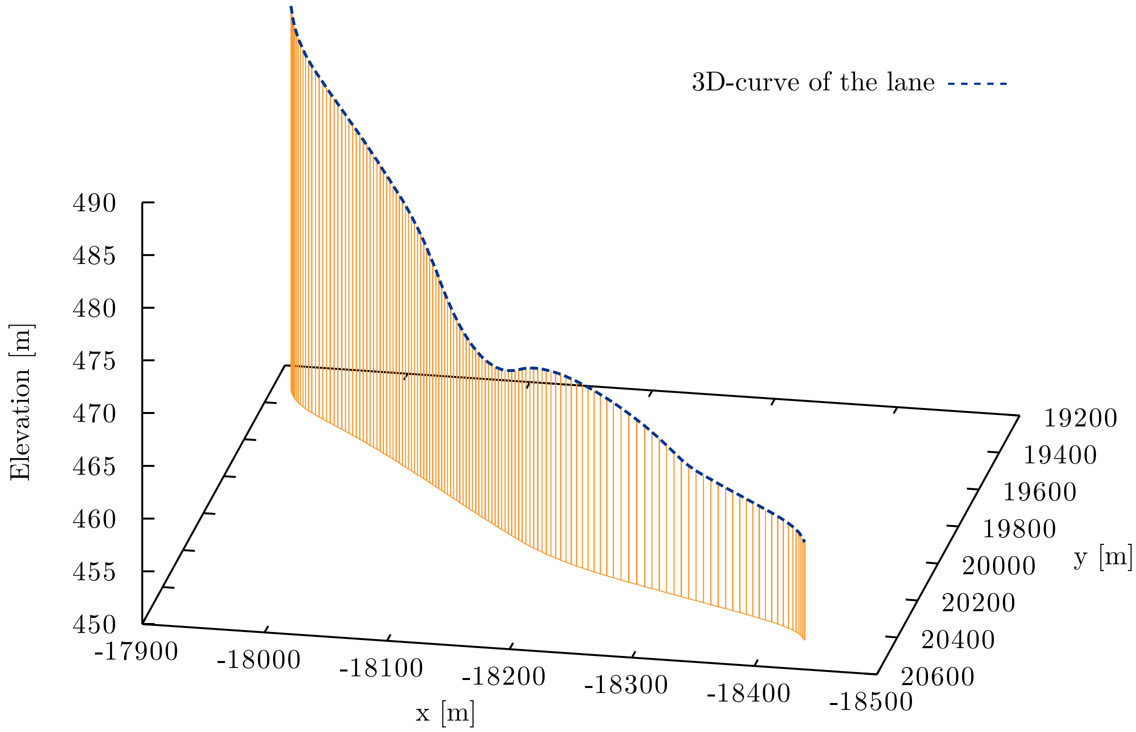


Figure 5.3.8: Resulting 3D-curve η of the lane. Some perpendicular lines on the planar course γ are shown as well.

3D-curve η, since γ is arc length parametrized, and it is true that:

$$\begin{aligned}
\operatorname{len}(\nu) &= \int_0^{l_\gamma} \left\|\dot{\tilde{n}}(t)\right\| dt \\
&= \int_0^{l_\gamma} \sqrt{(\dot{\tilde{n}}_1(t))^2 + (\dot{\tilde{n}}_2(t))^2} dt \\
&= \int_0^{l_\gamma} \sqrt{1 + (\dot{h}(t))^2} dt \\
&= \int_0^{l_\gamma} \sqrt{\|\dot{g}(t)\|^2 + (\dot{h}(t))^2} dt \\
&= \operatorname{len}(\eta)
\end{aligned} \tag{5.3.11}$$

This property is advantageous for applications that need the distance between two points p_1 and p_2 on the lane regarding the 3D-curve representation[2]: Instead of calculating the arc length difference between p_1 and p_2 regarding the 3D-curve, it is sufficient to simply focus on the arc length difference regarding the elevation profile ν:

[2]This kind of information is relevant to intersection assistance systems, where the distance between a vehicle on a lane and a stop line lying ahead at an intersection must be calculated.

Let $p_1, p_2 \in \mathrm{tr}(f)$ and $t_1, t_2 \in [0, l_\gamma]$ with $f(t_1) = p_1$ and $f(t_2) = p_2$. Then the distance between p_1 and p_2 regarding the 3D-curve is given by

$$d_f(p_1, p_2) := \int_{t_1}^{t_2} \left\| \dot{f}(t) \right\| dt = \int_{t_1}^{t_2} \left\| \dot{\tilde{n}}(t) \right\| dt.$$

According to Section 2.1.1.4, the length of an arc spline sums up from the lengths of its segments that in turn can be calculated in a closed form.
From an implementation's side of view, the calculation of $d_f(p_1, p_2)$ can be further simplified if the individual segment lengths are stored in an accumulated form in the data structure for an arc spline. The desired distance then results from the distance of the arc length parameter t_1 and t_2, which can be determined efficiently using the methods described in Section 2.1.3.
To summarize, using the modeling proposed above, point distances can be calculated very efficiently regarding both the planar representation and the 3D-curve of a lane.

5.3.4.2 Attributes of a lane

Table 5.3.4 gives an overview of the possible attributes that are considered additionally for lanes. Beside the continuous course of the lane width (cf. Section 5.3.4.3), a static value of the average lane width is available, which already satisfies the requirements of many applications.

Attribute	*Range*	*Description*
Width	$\mathbb{R}^+$	Static value of the average lane width
Type	{*high_way, motor_way, primary, residential, cycleway*}	Lane type

Table 5.3.4: Attributes of a lane.

5.3.4.3 Additional profiles

Analogous to the modeling of the elevation profile, the integration of additional profiles is possible. For assistance functions relying on exact information on the lane width, a continuous representation of the course of the lane width is preferable compared to some static information. According to the guidelines on road constructions [BFV 93], the lane width can be extended locally in turns, in order to facilitate the transit of long vehicles like trucks. Analogous to the elevation profile, the access to the local lane width is realized by the arc length parameter of the planar lane representation.
Furthermore, the cross-slope of a road can be modeled using an additional profile in the same way.

5.3.4.4 Extended curvature information

As discussed in Section 5.1.1.2, precise reconstruction of the original curvature is essential. In particular, the linear curvature characteristics of the clothoid parts should be approximated in a preferably exact manner. However, the curvature of an arc spline is a step function and therefore not continuous. Though there are approaches for G^2-smoothing of arc splines [Li 05], the following method results in a continuous and piecewise affine curvature characteristic: Let γ be a smooth arc spline with arc length parametrization $g : [0, l] \to \mathbb{R}^2$ and let $t_0 := 0 < t_1 < \cdots < t_m := l$ denote the arc length parameters of the corresponding breakpoints with respect to g. Hence the corresponding curvature function of γ is

$$\kappa : [0, l] \to \mathbb{R}, \quad \kappa = \sum_{j=1}^{m} \kappa_j \underset{[t_{j-1}, t_j[}{\chi}, \tag{5.3.12}$$

where $\chi_{[x,y[}$ is the characteristic function of the interval $[x, y[$ and κ_j are the curvature values of the particular segments. We then define an approximation $\tilde{\kappa} : [0, l] \to \mathbb{R}$ by

$$\tilde{\kappa}(t) := \frac{1}{b(t) - a(t)} \int_{a(t)}^{b(t)} \kappa(s) ds, \tag{5.3.13}$$

$$\text{with} \quad a(t) := \max(0, t - \Delta t) \quad \text{and} \quad b(t) := \min(l, t + \Delta t) \tag{5.3.14}$$

where $\Delta t > 0$ is a parameter controlling how strong the influence of the curvature values of the adjacent segments are. Using the mean value theorem, it is easy to show the continuity of $\tilde{\kappa}$ and the piecewise affinity follows from the integration of a step function. For arc splines, $\tilde{\kappa}$ is a sum which can be calculated very efficiently. More details and results on this extension are to be found in [Schindler 12].

5.3.5 Digital map / map section

Finally, based on the preceding model descriptions, the term *digital map* or *map section* can be defined:

Definition 5.3.5.1 (Digital map / map section)
Any triple (L, M, R) of landmarks $L \subset \mathfrak{L}$, road markings $M \subset \mathfrak{M}$ and lanes $R \subset \mathfrak{R}$ is called digital map or map section, respectively.

5.4 Map generation

This section is dedicated to methods for the generation of digital map elements in terms of the modeling proposed in part 5.3. Figure 5.4.9 gives an overview of the data processing chain for generating map elements.

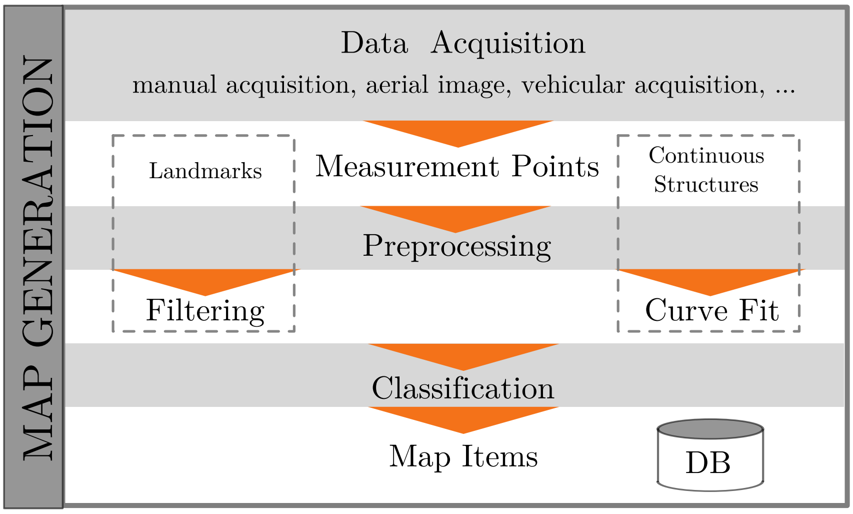


Figure 5.4.9: Processing chain for the map generation.

The basic data set for the determination of map elements consists of measurement points provided by the data acquisition step:

- For landmarks, these measurement points correspond to their position.
- In order to reconstruct road markings, measurement points on the corresponding line markings are necessary.
- The determination of a lane representation requires measurement points along the middle of a lane. The elevation profile can be created as long as some elevation data is available for each measurement point.

Before scoping the geometric computation of map elements, the data sources for the required measurement points should be clarified:

5.4.1 Data sources for the map generation

5.4.1.1 Use of existing map data

If some highly precise map data is available (e.g. from land surveying offices) then this information can be used for a change of curve representation. Therefore, the existing curve is sampled in order to extract some raw measurement points. One example is sampling a curve based on the construction plans of a road.

5.4.1.2 Aerial images

Furthermore, the evaluation of aerial images can be used for extracting raw measurement points. In [Pink 11] or [Kümmerle 09], some image processing techniques are proposed in order to automatically extract measurement points on road markings based on high-definition aerial images. That way, large map sections can be processed efficiently. This kind of method requires some ortho-rectified images of high-definition that are exactly georeferenced. Dealing with this high-quality aerial images involves some potential sources of errors. In particular, physically caused projection errors mostly occur in hilly areas. Furthermore, it should be noted that, even for highly precise aerial images with a resolution of 10 cm / pixel, one pixel corresponds to the magnitude of the desired map accuracy. Finally, measurement points on road markings can only be extracted if they are actually visible on the aerial images, which might not be true in case of occluding objects like trees, buildings, vehicles or shadows.

5.4.1.3 Manual acquisition

Using highly precise geodetic surveying systems (like the one specified in Appendix A.6), single measurements can be made with centimeter accuracy under appropriate conditions. This procedure allows high precision of measurements, but it is time consuming and thus expensive. However, the method is suitable for the generation of reference data for evaluation purposes. In Ko-PER, this has been done to confirm the accuracy of the digital map created with the methods presented in this work.

5.4.1.4 Vehicular acquisition

An alternative for the generation of measurement points is given by the vehicular data acquisition on roads using onboard perception systems. In that case, the vehicle needs to be equipped with a localization unit as well as a sensor system that allows the extraction of the desired raw measurement points. This approach has been chosen for this thesis and it is elucidated in the following sections.

5.4.2 Preprocessing of measurement points

Measurement points for generating landmarks can be created by the laser scanner processing described in Section 4.4. Since this extraction results in a set of landmark candidates with regard to the vehicle frame, they need to be transformed into the navigation frame in order to store their position in the digital map.

The transformation of measurement points from the vehicle frame to the navigation frame is done using the position and the yaw angle of the vehicle determined by the RTK GPS unit. Again, let $\psi \in \mathbb{R}$ be the vehicle yaw angle and let $\boldsymbol{x} \in \mathbb{R}^3$ its position in the navigation frame. Then, any measurement point $\boldsymbol{p} \in \mathbb{R}^3$ is transformed from the vehicle frame into the navigation frame by rotation and translation:

$$T : \mathbb{R}^3 \to \mathbb{R}^3, \quad \boldsymbol{p} \mapsto M\left(\frac{\pi}{2} - \psi\right) \cdot \boldsymbol{p} + \boldsymbol{x} \tag{5.4.15}$$

where, for any $\alpha \in \mathbb{R}$, M is the rotation matrix

$$M(\alpha) := \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

For the transformation of measurement points extracted by the sensor data processing at a certain point in time t_k, the transformation parameter $\boldsymbol{x}$ and ψ from (5.4.15) are required for t_k as well. However, the RTK-GPS unit might not provide this information at that specific time in general. In that case, the vehicle's position and yaw angle are predicted using the RTK-GPS measurements preceding t_k and the vehicle motion model described in Section 3.2.1.

Concerning the measurement points for road markings and lanes, the video-based lane recognition presented in Section 4.2 can be used for the data acquisition. The algorithms described there result in a finite set of real measurements $Y_{LR} \subset B$ in the image rectangle $B \subset \mathbb{R}^2$ within the image plane. Using the camera reprojection $K^\star$ from (4.2.5), the real measurement points in Y_{FS} are reprojected onto the road surface. This is realized by intersecting for each point $b \in Y_{LR}$ the view ray $R(b)$ defined by (4.2.6) with the $Z = 0$ plane of the vehicle frame[3]. This step yields the finite set

$$P_{LR} := \bigcup_{b \in Y_{LR}} (R(b) \cap \{Z = 0\}) \subset \mathbb{R}^3 \tag{5.4.16}$$

of reconstructed road marking measurement points based on the lane recognition. Like the landmark candidates, the road marking measurement points are transformed into the navigation frame using transformation (5.4.15).

In order to create a data basis of measurement points for a whole road section, the cartographic vehicle passes all relevant lanes capturing the environment with the onboard sensor system. For each sensor acquisition frame at time t_k, the current measurement points $P_{LR,k}$ are reconstructed in the navigation frame using T_k from (5.4.15) with the transformation parameter for time t_k, yielding to a finite set

$$P_{RM} := \bigcup_{k \in \mathbb{N}} (T_k(P_{LR,k})) \subset \mathbb{R}^3 \tag{5.4.17}$$

of reconstructed road marking measurement points in the navigation frame. Likewise, the reconstructed landmark measurement points are denoted by

$$P_{LM} \subset \mathbb{R}^3. \tag{5.4.18}$$

Due to physical measuring inaccuracies of the used sensors, the reconstructed measurement points may be inconsistent regarding their positions. In order to cope with this problem, post-processing techniques are available in the field of robotics. Simultaneous Localization and Mapping (SLAM) approaches can be used to model this task as a global optimization problem (cf. [Rauch 13, Nüchter 10]). Furthermore, standard filtering techniques based on RANSAC methods (cf. [Fischler 81]) are suitable to remove outliers within the input point sequences.

[3] In reasonable practical configurations, these intersections always exist and they are located next to the sampled points of the local road model described in Section 4.2.3.2.

5.4.3 Generation of map elements

Once the reconstructed landmark measurement points P_{LM} are available, they are used for generating landmarks for the digital map according to the specification in Section 5.3.2. Details on clustering and filtering techniques concerning the landmark generation are not treated further at that point but they are found in [Weiss 11, Heenan 05, Weiss 07, Fuerstenberg 05].
Instead, this section focuses on the generation of continuous map elements, like road markings and lanes, based on the reconstructed measurement points of road markings P_{RM} defined in (5.4.17). According to the map modeling presented in Section 5.3, continuous map elements are essentially represented by smooth arc splines. Therefore, the following part deals with the computation of smooth arc splines based on the reconstructed and filtered point data set P_{RM}.
To summarize, the following steps are performed:

1. **Segmentation** During the data acquisition process, measurement points for different road marking segments may have been collected. Since each segment is represented by an individual arc spline, the input data needs to be segmented into connected components. In particular, this is necessary for dashed road markings.

2. **Arc spline fit** After the segmentation, each component of the input points is approximated by a smooth arc spline.

3. **Elevation profile** As long as there is some information on the elevation of a component, the corresponding elevation profile can be computed.

4. **Storage** Finally, the map elements are stored in the map data base to provide persistent access to the digital map.

Since the elevation profile is separated (cf. Sections 5.3.1 and 5.3.4.1) from the planar curve representation in the X,Y-plane of the navigation frame, we focus on the planar projection of the input points at first. Therefore, let

$$P := \left\{ \begin{pmatrix} \pi_1(p) \\ \pi_2(p) \end{pmatrix} \in \mathbb{R}^2 \,\middle|\, p \in P_{RM} \right\} \tag{5.4.19}$$

denote the finite set of projected input points for the subsequent steps.

5.4.3.1 Segmentation

The aim of the segmentation is the separation of P into components such that, ideally, each component corresponds exactly to one real road marking component. Therefore, a graph-based approach can be applied:
Let $G = (V, E)$ be a Euclidean graph where the coordinates of the nodes $v \in V$ are identified with the points in P. With d denoting the Euclidean distance, the set of edges E in G is defined as $\{(u, v) \in V \times V | 0 < d(u, v) \leq \delta\}$. The distance δ is determined by the density of the measuring points and it is smaller than the minimum distance between two adjacent road marking segments.
Let $\mathfrak{C}$ be the set of connected components of G. The set of corresponding points of a connected component $C \in \mathfrak{C}$ is denoted by P_C.

5.4.3.2 Arc spline fit

In the next step, the input points P_C of each connected component $C \in \mathfrak{C}$ are approximated by a smooth arc spline. The individual segments of dashed road markings are represented as line segments, which are indeed smooth arc splines with only one segment. Depending on the dimensions[4] of C, a simple line segment is fit or the more general approximation with smooth arc splines is applied to P_C.

Line segment case The best approximating line $l := \{a + \lambda \cdot b \mid \lambda \in \mathbb{R}\}$ for some $a, b \in \mathbb{R}^2$, is used which minimizes $\sum_{p \in P_C} \operatorname{dist}(l, p)^2$, where $\operatorname{dist}(l, p) := \min_{x \in l} d(x, p)$ with the Euclidean distance d. This approximation problem is solved using standard least-squares methods. The corresponding line segment is the smallest connected subset of l including all projections of P_C. If the lengths of the segments are known a priori, the segments can be used as an initial solution for an optimization process with respect to the pose of the line segment. However, this postprocessing should be handled carefully, as the painted road markings in reality often differ from the construction plan.

General arc spline case It is desirable to compute a curve that approximates not only the extracted points up to unavoidable fitting errors but also describes them effectively, i.e. with minimal complexity. Such a characterization allows coping with requirements and applications motivated in Section 5.1.

In the general case, the arc spline approximation technique developed in [Maier 10] is applied to P_C. Since the methodology of that work is used extensively to generate map elements in the present case and thus plays a decisive role, it is summarized in the following.

A so called *tolerance channel* describes the feasible area for the resulting smooth arc spline. The method controls the approximation error by a geometric model: Only solutions staying inside the tolerance channel around P_C are taken into account. The width of this channel represents the user-specified maximum tolerance which can even vary locally. The canonical shape of a tolerance channel modeling a maximum error ε is given by the set of points which have Euclidean distance of at most ε to the open polygon running through P_C. The resulting boundary curve, which is in fact an arc spline, is approximated by a simple closed polygon. This way, a geometric constraint is obtained, which can be adjusted in a comfortable and intuitive manner. In addition, two disjoint edges of the tolerance polygon, s and d, are fixed and act as start and destination segment of the channel (see Figure 5.4.12). Details on the generation of tolerance channels are found in [Schindler 11, Maier 10, Drysdale 08] or in [Heimlich 08, Held 05], where both symmetric and asymmetric tolerance zones are introduced which are generated using Voronoi diagrams.

Any smooth arc spline staying inside the tolerance channel and connecting s and d with a minimum number of segments solves the problem. Such a curve is called *Smooth Minimum Arc Path (SMAP)*. Note that the breakpoints are not required to be part of P_C but they have to be determined automatically. This has considerably positive effects on the resulting number of

[4]Relevant decision criteria are the maximal distance between any two points in P_C and the statistical spread of P_C.

segments. In contrast, all other currently-known methods using arc splines have no theoretical bounds concerning the number of segments.

Definition 5.4.3.1
Any triple (P, s, d) is called *tolerance channel* if P is a simple closed polygon and s and d are two disjoint edges of P denoting the start and the destination.

Since P is a closed curve, it is reasonable to consider the *interior of* P. According to the Jordan curve theorem (cf. [Dieudonné 60]), P divides $\mathbb{R}^2 \setminus \mathrm{tr}(P)$ into two connected components. One of them is bounded and defines the interior $\mathring{P}$ of P. The closure of the interior is denoted by $\bar{P} := \mathrm{tr}(P) \cup \mathring{P}$. Furthermore, it can be distinguished between the *left channel side* and the *right channel side* by imaging to 'stand' on $\mathrm{tr}(s)$ and looking into the interior $\mathring{P}$ of P.
In the following, let (P, s, d) be a tolerance channel. To keep the notation as simple as possible, it is assumed that the two vertices of s are convex: A vertex v is convex if the interior angle at v is strictly smaller than $180°$. The definitions for the general case as well as the proofs of all subsequent theorems can be found in [Maier 10].

Definition 5.4.3.2
A point $a \in \bar{P}$ is said to be *circularly visible (from s with respect to $\bar{P}$)* if there exists a segment γ in $\bar{P}$ that has its starting point on s and ends in a. The set of all circularly visible points from s with respect to $\bar{P}$ is denoted by V. An oriented arc γ, as above, is called *visibility arc.*

The main instrument of the SMAP algorithm is the calculus of *alternating restrictions.* These are points visibility arcs and $\mathrm{tr}(P)$ have in common, as indicated in Figure 5.4.10. In the left image, the plotted visibility arc is touched by P from the left and from the right at the points $a_1, \ldots, a_6$. Thus, they are called *left* and *right restriction points.*
At a right restriction point, the visibility arc cannot be moved to the right without either exceeding the tolerance boundary or violating the starting condition.
The *alternating number* of a sequence of restriction points, ordered with respect to the order (2.1.1) of the visibility arc, is given by the number of changes of channel sides between the consecutive restriction points increased by one: For two side changes, at least three restriction points are required on alternating channel sides resulting in an alternating number of three. Thus, the alternating number is not defined directly by the number of restriction points but it is determined by the number of side changes, as depicted in Figure 5.4.10.
Let ∂V be the boundary of V. In [Maier 10], it is shown that the boundary points in $\partial V \setminus \mathrm{tr}(P)$ lie on arcs, and the corresponding visibility arcs are called *blocking arcs* if they are maximally extended with respect to inclusion in $\bar{P}$. These arcs distinguish themselves from the other visibility arcs as they have at least three *alternating* restriction points a_1, a_2, a_3 ordered as a_1, a_2 and a_3 according to the order (2.1.1) of the arc. They satisfy the following condition: Either a_1 and a_3 are left restriction points and a_2 is a right restriction point or a_1 and a_3 are right restriction points and a_2 is a left restriction point. An example of alternating restrictions can be found in Figure 5.4.11. Arcs passing through three alternating restriction points can be described in an efficient manner regarding an algorithmic approach as they uniquely determine the three degrees of freedom an arc has.

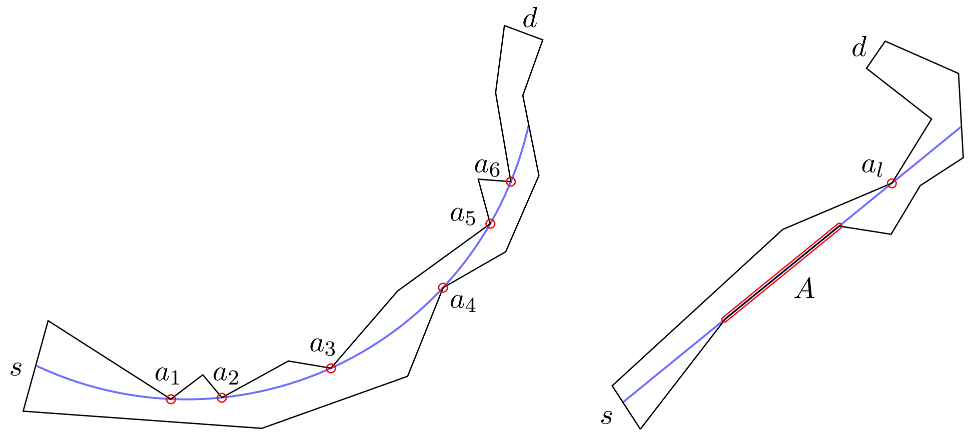


Figure 5.4.10: Tolerance channels and visibility arcs with respect to s.
Left: a_1, a_2, a_3, a_5 and a_6 are left restriction points; a_4 is a right restriction point. The alternating number is three.
Right: Visibility arc with an infinite number of right restriction points A and one left restriction point a_l. The alternating number is two.

Every connected component of $\bar{P} \setminus V$ is separated from V by exactly one blocking arc. The blocking arc corresponding to the connected component including d is called the *window (with respect to s)*, for which one can show the following characterization:

Remark 5.4.3.1 (Window characterization)
Let γ be a maximally extended visibility arc with endpoint on the left side of P. Then γ is the window if and only if there are three alternating restriction points a_1, a_2, a_3 where a_3 is a right restriction point. Similar conditions hold if γ has its endpoint on the right side of P.

All possible configurations of left and right restrictions are given as follows: The corresponding arc

- passes through three vertices
- passes through two vertices and touches an edge of P or
- passes through one vertex and touches two edges of P

Having analyzed the circular visibility set V, the next step is to characterize the sets V^i of all points which can be reached by $i = 2, \ldots, k$ segments till V^k intersects d. Therefore, the following gives a criterion for deciding if an oriented arc can be smoothly continued or not.
Oriented arcs in P satisfying the so called *continuation condition (CC)* can be smoothly joined to a visibility arc. An oriented arc γ satisfies the CC with respect to an oriented arc η if either γ smoothly joins η or there are two intersection points x_1, x_2 of γ and η s.t. the order induced by γ and the one induced by η are equal. An illustration can be found in Figure 5.4.12. The CC can be summarized as follows:

Remark 5.4.3.2 (Continuation condition)
Let $x \in \bar{P} \setminus V$ and let C be the connected component of $\bar{P} \setminus V$ containing x. Then, $x \in V^2$ if and

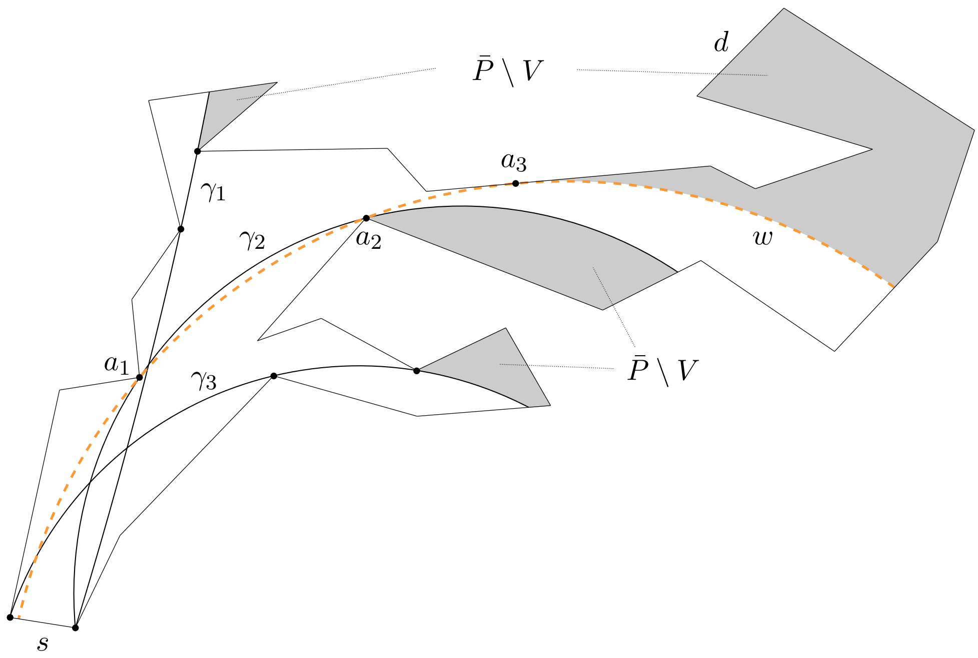


Figure 5.4.11: Visibility set with respect to the start segment s. The shaded area is not circularly visible and γ_1, γ_2 and γ_3 are blocking arcs. Restriction points are marked with dots. The window w is highlighted and dashed with its alternating restriction points a_1, a_2, a_3. In contrast to the blocking arcs $\gamma_1, \gamma_2, \gamma_3$, the end point of the window w lies on the opposite channel side of the last restriction point a_3.

only if there exists an oriented arc γ in $V \cup C$ ending in x and satisfying the CC with respect to the corresponding blocking arc.

In this case, even an arc γ that is extremal can be chosen, i.e. having at least two alternating restrictions, which is fundamental for a constructive approach and hence for the algorithmic design.

Therefore, a characterization of the set V^2 can be formulated by examining all oriented arcs γ satisfying the conditions above. However, not the whole set V^2 needs to be considered but only the component leading to d. This meets in elucidating a "modified" tolerance channel with the corresponding window as starting segment. The only differences to the kind of tolerance channels considered so far are the more complicated starting requirements given by Theorem 5.4.3.2. In [Maier 10], it is shown that the theorems presented above hold for this kind of tolerance channel as well. Especially, the window of V^2 is characterized in the same manner.

It should be recalled that touching the start segment, which is here the window of V, is a left or right restriction point. In fact, this channel can be interpreted as a shrinked subset of the original one, where the starting segment is given by the window instead of s.

Theorem 5.4.3.2 can now be used inductively, exploiting the properties of the sets V^k this way. In the following, it is assumed that the successively resulting windows have exactly three alternating restrictions. The general case requires some slight modifications, which cannot be treated within this scope. The outcome of this is a two step greedy algorithm traversing P from s to d in the forward step and back again from d to s in the backward step.

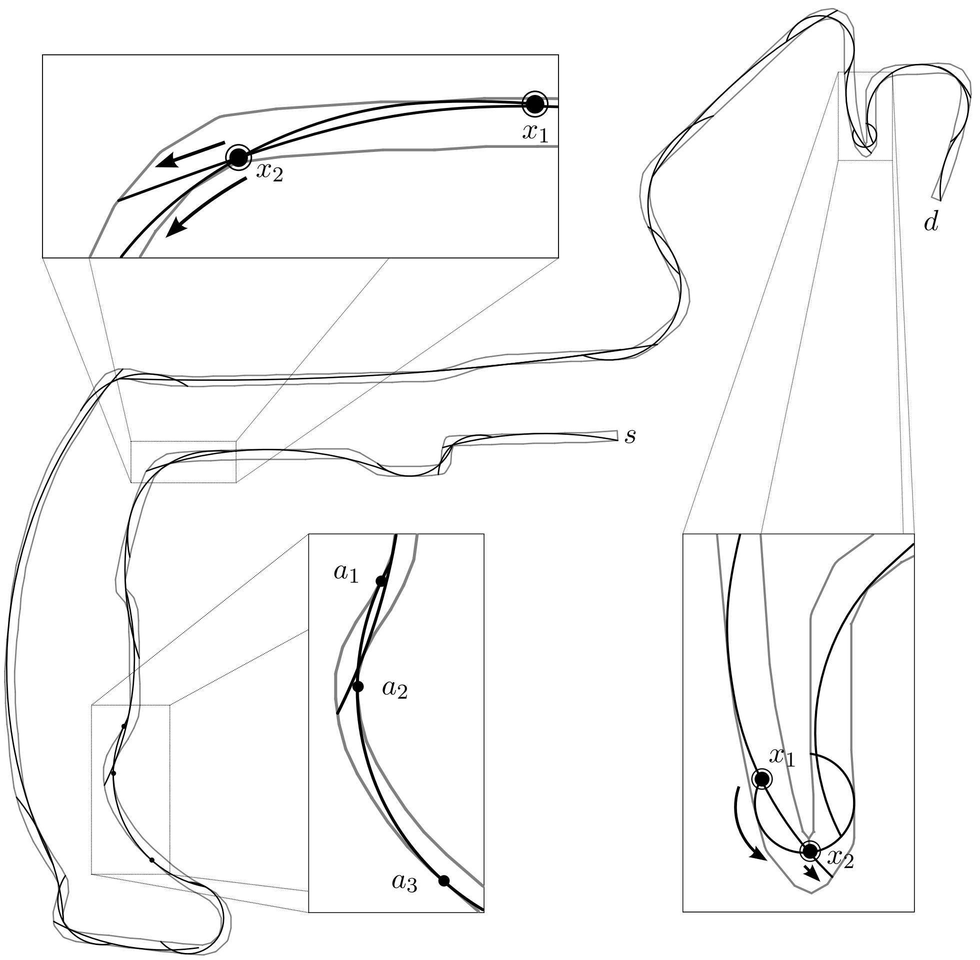


Figure 5.4.12: Visualization of the forward step of the SMAP algorithm. In the bottom left zoom, three alternating restriction points a_1, a_2, a_3 are marked with dots.

The Forward Step: After having found the first window ω_1 by identifying arcs with three alternating restrictions, the next windows ω_i can be computed such that the conditions of Theorem 5.4.3.2 and Theorem 5.4.3.1 are satisfied. In particular, ω_i has to satisfy the CC with respect to ω_{i-1}. The procedure is stopped when a point of d is circularly visible, and a visibility arc satisfying Theorem 5.4.3.2 and ending in d is computed. As it can be seen in Figure 5.4.12, the windows do not represent a smooth arc spline. However, the computed windows are used in the backward step to generate a SMAP.

The Backward Step: The lastly calculated arc ω_k represents the last segment of the resulting SMAP. In particular, the minimum segment number is k. The predecessor segments are then determined by touching their successor and by two alternating restrictions. The whole procedure is finished when s has been reached. In Figure 5.4.13, the backward step is visualized and the box shows a single backward step: γ_i joins its successor γ_{i+1} and satisfies the CC with respect to ω_{i-1}, as the two emphasized dots indicate.

Figure 5.4.14 shows the reconstruction result of a real lane section.

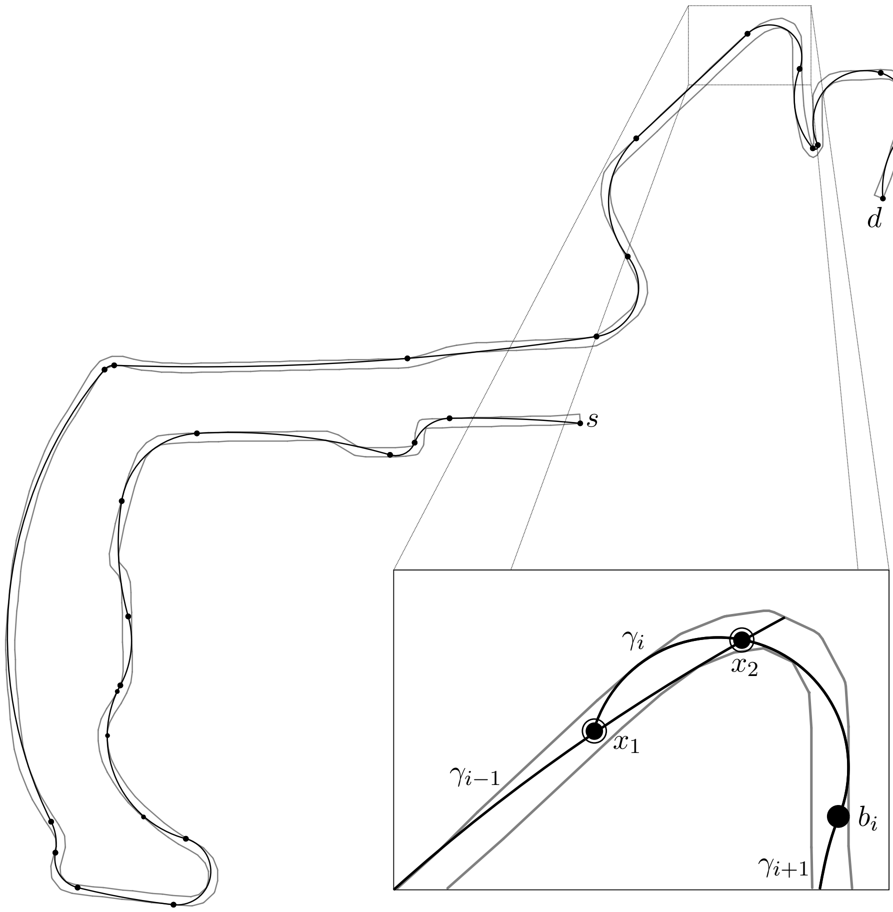


Figure 5.4.13: Visualization of the backward step of the SMAP algorithm. The overview shows the SMAP result. Simple dots indicate breakpoints of the spline. The zoom shows a single backward step, where γ_i smoothly joins its successor γ_{i+1} at the breakpoint b_i and satisfies the CC with respect to ω_{i-1} by intersecting at x_1 and x_2.

Incorporation of line segments Considering digital maps, it has to be expected that straight road sections appear on real roads, in particular in the case of highways. Due to numerical reasons, the standard version of the SMAP algorithm produces circular arc segments with large radii in this case. However, it is preferable that these straight sections are represented by line segments instead. This is not only motivated by the more realistic modeling. Also, numerical pitfalls can be evaded for further calculations on the digital map. For that reason, an extension of the SMAP algorithm is proposed in [Maier 13], respecting predefined line segments while the minimality of the segment number can still be guaranteed.

Computational complexity Though generating a SMAP guarantees a smooth arc spline with the least possible number of segments with respect to any given accuracy, the approximation algorithm does not satisfy real time requirements. The best known implementation of this algorithmic approach has a quadratic worst case complexity regarding the number of input points. However, the computational time does not play an important role here since the digital map can be generated offline.

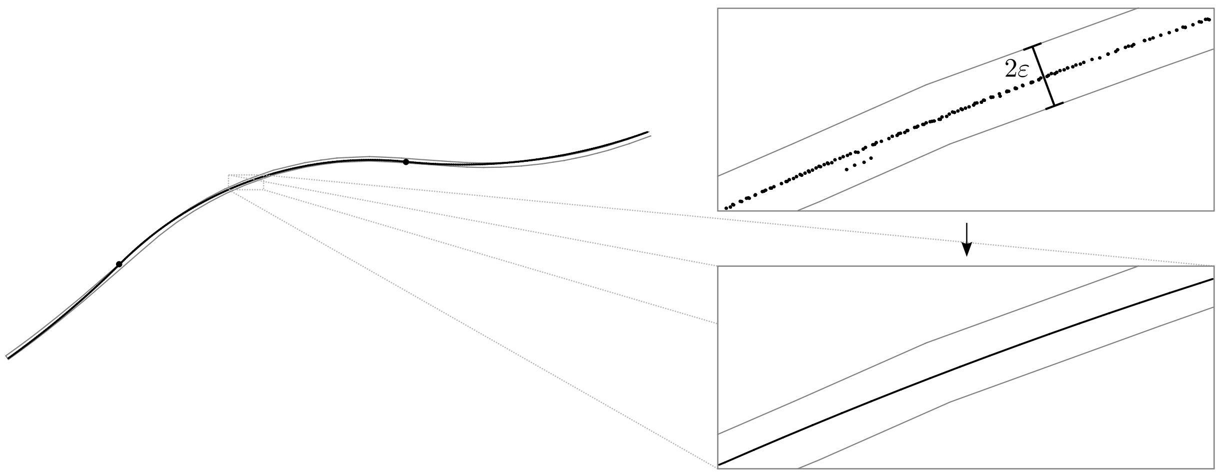


Figure 5.4.14: Illustration of the arc spline approximation for a lane section: Input data points, ε-tolerance channel and resulting smooth minimum arc path (SMAP).

Restriction of the start and end condition According to the theory in [Maier 10], the start and end segment of a tolerance channel can degenerate to points. This restriction allows modeling mandatory starting and end conditions of the resulting arc spline. In particular, this is useful when individual map elements should be connected at predefined points e.g. at intersections.

5.4.3.3 Elevation profile

Once an arc spline representation γ of a lane with arc length $l > 0$ is available, the corresponding elevation profile can be computed. For this purpose, let t_i be the run length parameter of the input data points p_i with respect to their projection on γ. The elevation information h_i of the measuring points, which is in fact their Z-coordinate, is used in order to create the input data (t_i, h_i) for the elevation profile. Next, the SMAP algorithm is applied to these points resulting in a smooth arc spline ν representing the elevation profile as defined in (5.3.4.1).

5.4.3.4 Storage

Finally, all computed map elements are stored in a database to provide persistent map access for the applications. To achieve greatest possible compatibility and interchangeability the approved OpenStreetMap ([OSM 12]) database and file scheme is used for data storage. Due to this open format, new elements and attributes can easily be added to the map. For instance, new tags have been introduced to enable the storage of arc splines in OSM ways.

5.5 Summary

In this chapter, the requirements for the digital map have been gathered from the applications' side of view. Based on these conditions, the technical requirements have been deduced. After a discussion of modeling approaches known in the literature, a map model has been chosen that represents continuous structures as smooth arc splines.

Finally, methods for the generation of the modeled map elements have been described. The core algorithm for generating smooth arc splines not only guarantees a predefined approximation accuracy but it also produces splines with the minimal possible number of segments. This, indeed, has considerable effects on the quality of the resulting digital map: The global accuracy of map elements can be controlled while the minimal possible description of the resulting curves minimizes the required data volume for storing the map as well. Furthermore, the low segment number of arc splines reduces the computational costs for calculating best-approximating points, which represents an essential criterion for the map modeling. To sum it up, the proposed models and methods for the digital map largely meet the requirements and thus are well-suited for a range of applications, thereunder the vehicle self-localization that is treated in the next chapter.

Chapter 6

Self-localization

'The first steps are worthless
unless the path be followed to the end.'

(Shankara, 780-820 AD)

Contents

In the preceding chapters, methods for modeling and perceiving the vehicle's environment were introduced. The relevant reference frames were defined as well as estimation techniques for the model parameters of dynamic systems. After the introduction of the digital map model, adequate methods for the generation of high-precision digital maps allow storing the reference data that is required for the landmark-based vehicle self-localization approach described in this chapter.

Before scoping on the methods and models used within this thesis, localization approaches known in the literature are summarized. Determining the pose of a vehicle is one of the key problems of robotics. Since this thesis accounts for the conditions of real road scenarios, it is focused on landmark-related vehicle localization approaches for advanced driver assistance systems.

6.1 Localization approaches in the literature

The so-called *Monte Carlo localization (MCL)* comprises probabilistic techniques based on particle filters in order to determine the position of a robot taking into account measurements from onboard sensors (cf. [Thrun 00]). A standard work on probabilistic methods for robot localization is [Thrun 05].

Originating from the field of robotic terrain exploration, the *Simultaneous Localization and Mapping (SLAM)* approaches pursue a strategy in which a digital map is built up during a driving maneuver and, simultaneously, the vehicle's pose is estimated relatively to this map (cf. [Wang 11, Nüchter 10, Aycard 10, Durrant-Whyte 06, Thrun 02, Montemerlo 02]). Due to the immanent mutual dependence of the map building process and the vehicle localization, errors concerning the resulting localization accuracy are summing up over time. However, if the vehicle passes through a section which is already represented in the map, then some optimization techniques can be applied to enhance the digital map as well as the localization estimation [Gutmann 99, Thrun 98].

Landmark-based localization approaches using laser scanners have been investigated in several works (cf. [Weiss 11, Schubert 09, Weiss 05b, Weiss 05a]). In order to increase traffic safety at urban intersections, some methods have been developed to provide vehicle positioning in these environments (e.g. [INTERSAFE-2 11, SAFESPOT 09, Ahlers 09, Rössler 06, Fuerstenberg 05]). Furthermore, the work in [Wimmer 11] deals with the special conditions on vehicle localization within road construction sites. A method for mapping and localizing in the context of autonomous driving using a 360° laser scanner is presented for instance in [Levinson 11].

In the past, some video-based methods for vehicle self-localization have been proposed. The SLAM approach in [Se 02] uses invariant image features [Lowe 99] for localization purposes based on image feature matching. The work in [Pink 11, Pink 10] uses stereo image reconstruction in order to match the extracted visual landmarks on the road surface on a feature level.

The approach in [Mattern 10b, Mattern 09, Mattern 08] uses Hough transformation to detect line markings. The comparison of expected and real image contents is realized based on texture interpretation. Within this approach, a texture feature based map is sufficient in contrast to some explicit geometric modeling of map elements. Video-based localization methods working directly on aerial images are proposed in [Napier 10, Dogruer 08]. These methods require some ortho-rectified aerial images of high-definition that are exactly georeferenced. A grid map based localization approach, which combines sensor data from video and laser scanners, is presented in [Konrad 12].

In contrast to the positioning methods mentioned above, the following localization approach makes intensive use of the particular properties of the map model introduced within this thesis. In particular, the efficient calculability of best approximating points to map elements plays a decisive role within the observation models and resampling strategies.

6.2 Overview

The goal of the self-localization is to determine the pose (position and orientation) of the ego-vehicle within the navigation frame[1]. The principle of the landmark-based localization is to associate the detected objects from the vehicle environment perception with elements of the digital map in order to deduce the vehicle's pose from the correspondences. Therefore, several modeling components need to be declared.

Due to physical measuring inaccuracies, the position and shape of perceived objects in the vehicle's surroundings are potentially erroneous. Furthermore, errors in the association step are possible, which affects the resulting pose estimation in a negative way. Hence, it is suitable to apply some probabilistic techniques for the pose estimation, like the ones introduced in Section 2.2. Within this work, a particle filter model has been chosen as it allows modeling multiple hypotheses (corresponding to multiple modes in the state probability density) as well as incorporating different observation models of sensor measurements in a flexible way. Modeling measurement inaccuracies within the observation models allows coping with the problems of erroneous data associations. In accordance to the recursive scheme of the particle filter described in Section 2.2.2.2, a dynamic model allows predicting the state of the dynamic system while an observation model enables incorporating measurements in order to correct the estimated state.

The vehicle model has already been introduced in Section 3.2. Again, the considered state at time t_k consists of the vector

$$\boldsymbol{x}_k = \begin{pmatrix} x_k \\ y_k \\ \psi_k \\ v_k \\ c_k \\ \beta_k \end{pmatrix} = \begin{pmatrix} \text{position x-coordinate} \\ \text{position y-coordinate} \\ \text{yaw angle} \\ \text{absolute value of the velocity} \\ \text{curvature of the circular track} \\ \text{slip angle} \end{pmatrix} \in \mathbb{R}^6, \qquad (6.2.1)$$

modeling the position and the orientation of the vehicle together with its dynamic parameters. The dynamic model of the vehicle, describing its motion in time and space, is summarized in equation (3.2.8).

In order to apply the filtering techniques introduced in Section 2.2, the observation models for the measurement update need to be defined. Therefore, the following components remain to be declared in order to explain the self-localization approach at hand:

- Initialization of the system
- Access to the digital map
- Observation models for the individual sensors: camera, laser scanner, vehicle dynamics and GPS
- Association of perceived objects in the vehicle's surroundings with map elements

These aspects are discussed in subsequent sections.

[1] As mentioned in Section 3.1.1, the resulting pose estimation can be transformed into the world frame using the transformations described in Section 3.1.2.

6.3 Initialization

In the initialization phase (cf. step 1 of the particle filtering scheme 1 in Section 2.2.2.2), GPS is used to define the initial pose parameters of the particles. Therefore, let $\left(x_{1|0}^{(i)}\right)_{1\leq i\leq M}$ be the set of $M \in \mathbb{N}^+$ particles at the initialization time t_0. Furthermore, let the geographic longitude $\lambda_0 \in [-\pi, \pi]$, the geographic latitude $\varphi_0 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and the yaw angle $\psi_0 \in [0, 2\pi]$ be the measurements of the GPS unit at time t_0. Since the pose parameters are modeled in the navigation frame within the particle filter, the position coordinates (λ_0, φ_0) are expressed in the navigation frame using the transformation described in Section 3.1.2. Let therefore $(l_0, b_0) \in \mathbb{R}^2$ be the corresponding coordinates of (λ_0, φ_0) with respect to the navigation frame. Let $(v_0, c_0, \beta_0) \in \mathbb{R}^3$ be some initial values for the absolute value of the velocity, the curvature of the driven track and the slip angle.

For the initialization, the particles $\left(x_{1|0}^{(i)}\right)_{1\leq i\leq M}$ are normally distributed with regard to the expectation vector $(l_0, b_0, \psi_0, v_0, c_0, \beta_0) \in \mathbb{R}^6$ and a covariance matrix $M_0 \in \mathbb{R}^{6\times 6}$. M_0 is determined by the accuracy of the GPS measurement as well as by the accuracy of the intrinsic dynamics measurements of the vehicle. All uncertainty values are available from the sensor data.

6.4 Map access

For the association of perceived objects with elements of the digital map, only map sections of a local environment around the vehicle are of interest. Therefore, a caching strategy is applied in order to provide an efficient access to map data. Within the set of currently known spatial data structures, *quadtrees* represent a suitable model for that purpose [De Berg 08, Samet 90].

Definition 6.4.0.3 (Quadtree)

A quadtree is a rooted tree in which each inner node has exactly four children. Each node v refers to a rectangle $r(v)$ that covers a subset of $\mathbb{R}^2$. If a node v has children $v_{NW}, v_{NE}, v_{SW}, v_{SE}$, then their corresponding rectangles are defined by the quadrants of $r(v)$:

Let $r(v) := [x_0, x_1] \times [y_0, y_1] \subset \mathbb{R}^2$ with $x_0, x_1, y_0, y_1 \in \mathbb{R}, x_0 < x_1, y_0 < y_1$ and $x_{mid} := \frac{x_0+x_1}{2}$ and $y_{mid} := \frac{y_0+y_1}{2}$. The rectangles of the children of v are given by

$$r(v_{NW}) := [x_0, x_{mid}] \times [y_{mid}, y_1] \tag{6.4.2}$$

$$r(v_{NE}) :=]x_{mid}, x_1] \times [y_{mid}, y_1] \tag{6.4.3}$$

$$r(v_{SW}) := [x_0, x_{mid}] \times [y_0, y_{mid}[\tag{6.4.4}$$

$$r(v_{SE}) :=]x_{mid}, x_1] \times [y_0, y_{mid}[. \tag{6.4.5}$$

Fig. 6.4.1 shows a quadtree with the corresponding recursive decomposition into quadrants.

Thus, a quadtree models a recursive decomposition of a two-dimensional space. Beside the rectangle $r(v)$, each node v contains some data corresponding to $r(v)$. For our purposes, each node v refers to a map section (L_v, M_v, R_v) of landmarks $L_v \subset \mathfrak{L}$, road markings $M_v \subset \mathfrak{M}$ and lanes $R_v \subset \mathfrak{R}$ (cf. definition (5.3.5.1)) which is defined in the following way: The root of the

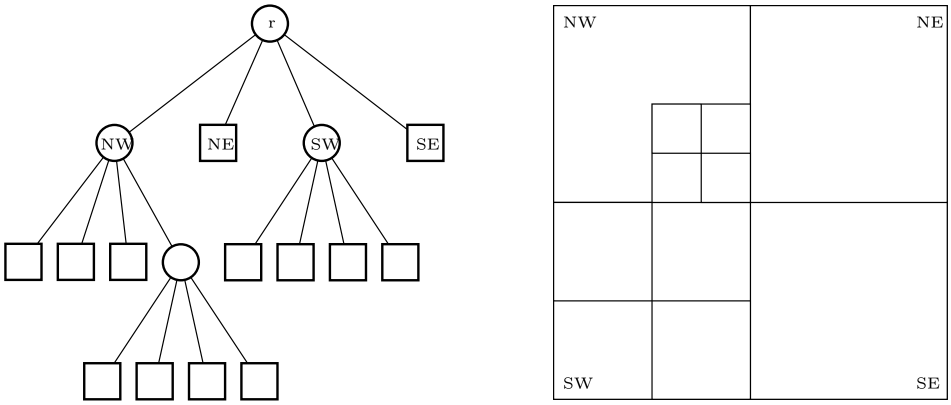


Figure 6.4.1: left: Quadtree of height 4 with root r, inner nodes (circles) and leafs (squares). right: Corresponding subdivision.

quadtree refers to the entire digital map $(L, M, R) \subset \mathfrak{L} \times \mathfrak{M} \times \mathfrak{R}$. The map section corresponding to any node v is given by a subset of (L, M, R) having a non-empty intersection with $r(v)$:

$$L_v := \{(p, A) \in L \mid p \in r(v)\} \tag{6.4.6}$$

$$M_v := \{(\gamma, A) \in M \mid \operatorname{tr}(\gamma) \cap r(v) \neq \emptyset\} \tag{6.4.7}$$

$$R_v := \{(\gamma, \nu, A) \in R \mid \operatorname{tr}(\gamma) \cap r(v) \neq \emptyset\} \tag{6.4.8}$$

According to its hierarchical definition, a quadtree is constructed by recursively decomposing the covered areas of nodes together with the associated map sections. The decomposition is stopped at a leaf v when either the number of elements in the map section of v is below a given threshold or a predefined maximum height of the tree is reached.

In principle, the use of spatial data structures like quadtrees is motivated by the reduction of storing space. In comparison to a static grid representation, quadtrees allow aggregating homogeneous data, which corresponds to empty map sections in the present case. Furthermore, many operations like search queries for points or regions, required for best-approximating point calculations, can be realized significantly faster compared to a grid representation. However, the hierarchical structure of quadtrees imposes an overhead for storing data. While point queries in spatial data can often be realized efficiently using standard index structures of databases, queries for curves like arc splines need to be handled with more sophisticated methods. The work in [Stone 11] treats these challenges and gives a complexity analysis for relevant queries as well.

At the initialization time of the self-localization system, a quadtree representing the digital map is constructed from the map database, where it is sufficient to focus on map sections of a few square kilometers around the vehicle position. Whenever the vehicle approaches the boundary of the map section represented in the quadtree, the tree is updated according to a new region of the map.

Once the quadtree cache is available, arbitrary map sections can be retrieved from it efficiently by specifying the desired region of interest. This kind of query is required for the map data association. For any given position $(x, y) \in \mathbb{R}^2$ and $\rho \in \mathbb{R}^+$, let $r := [x - \rho, x + \rho] \times [y - \rho, y + \rho] \subset \mathbb{R}^2$.

The map section map_r corresponding to the region of interest r is given by

$$map_r := (L_r, M_r, R_r) \subset \mathfrak{L} \times \mathfrak{M} \times \mathfrak{R} \tag{6.4.9}$$

with

$$L_r := \{(p, A) \in L \mid p \in r\} \tag{6.4.10}$$

$$M_r := \{(\gamma, A) \in M \mid \mathrm{tr}(\gamma) \cap r \neq \emptyset\} \tag{6.4.11}$$

$$R_r := \{(\gamma, \nu, A) \in R \mid \mathrm{tr}(\gamma) \cap r \neq \emptyset\} \tag{6.4.12}$$

Fig. 6.4.2 shows a quadtree decomposition of a digital map section in the north of Munich.

6.5 Observation models

Observation models allow the integration of sensor measurements in order to deduce the state of the considered dynamic system. According to the filtering methodology introduced in Section 2.2, any observation model is a function from the state space $\mathbb{R}^n$ and the time T to the measurement space $\mathbb{R}^m$:

$$h : \mathbb{R}^n \times T \to \mathbb{R}^m \tag{6.5.13}$$

The associated measurement uncertainty, modeled by the covariance matrix of the measurement noise process in (2.2.28) and (2.2.44), is either given by the sensor characteristics or it is determined empirically for each observation model by means of a system identification step (cf. [Soderstrom 89, Goodwin 77, Graupe 72, Eykhoff 74, Walter 97]), which is not discussed further within this work.
In the following, the essential observation models used for the vehicle self-localization are presented.

6.5.1 Observation model for road markings

The core of the observation model for road markings lies in the association of measurements coming from the lane recognition system with road marking elements from the digital map. The video-based lane recognition is described in Section 4.2. For each camera frame, the data processing provides a set of real measurement points. Using the reconstruction techniques described in Section 5.4.2, for each frame a finite set of measurement points $P_{LR} \subset \mathbb{R} \times \{0\}$ in the X, Y-plain of the vehicle frame is available (cf. equation(5.4.16)).
Furthermore, for any vehicle position $(x, y) \in \mathbb{R}^2$, represented within the state model (6.2.1), and $\rho \in \mathbb{R}^+$, the map section $map_r = (L_r, M_r, R_r) \subset \mathfrak{L} \times \mathfrak{M} \times \mathfrak{R}$ corresponding to the region of interest r is extracted using the equations (6.4.9) to (6.4.12). The value of ρ is chosen such that all perceived and reconstructed objects lie within the region of interest[2].
Before scoping on the observation model, the association of the reconstructed measurement points P_{LR} and the map section map_r should be declared. This is realized by formulating the task as a prototype fitting problem.

[2]In the present case, it sufficient to choose $\rho > L$ for the evaluation distance L in (4.2.7) within the local road model of the lane recognition system. In the reference implementation, a value of $\rho = 50\,\mathrm{m}$ proved to be appropriate.

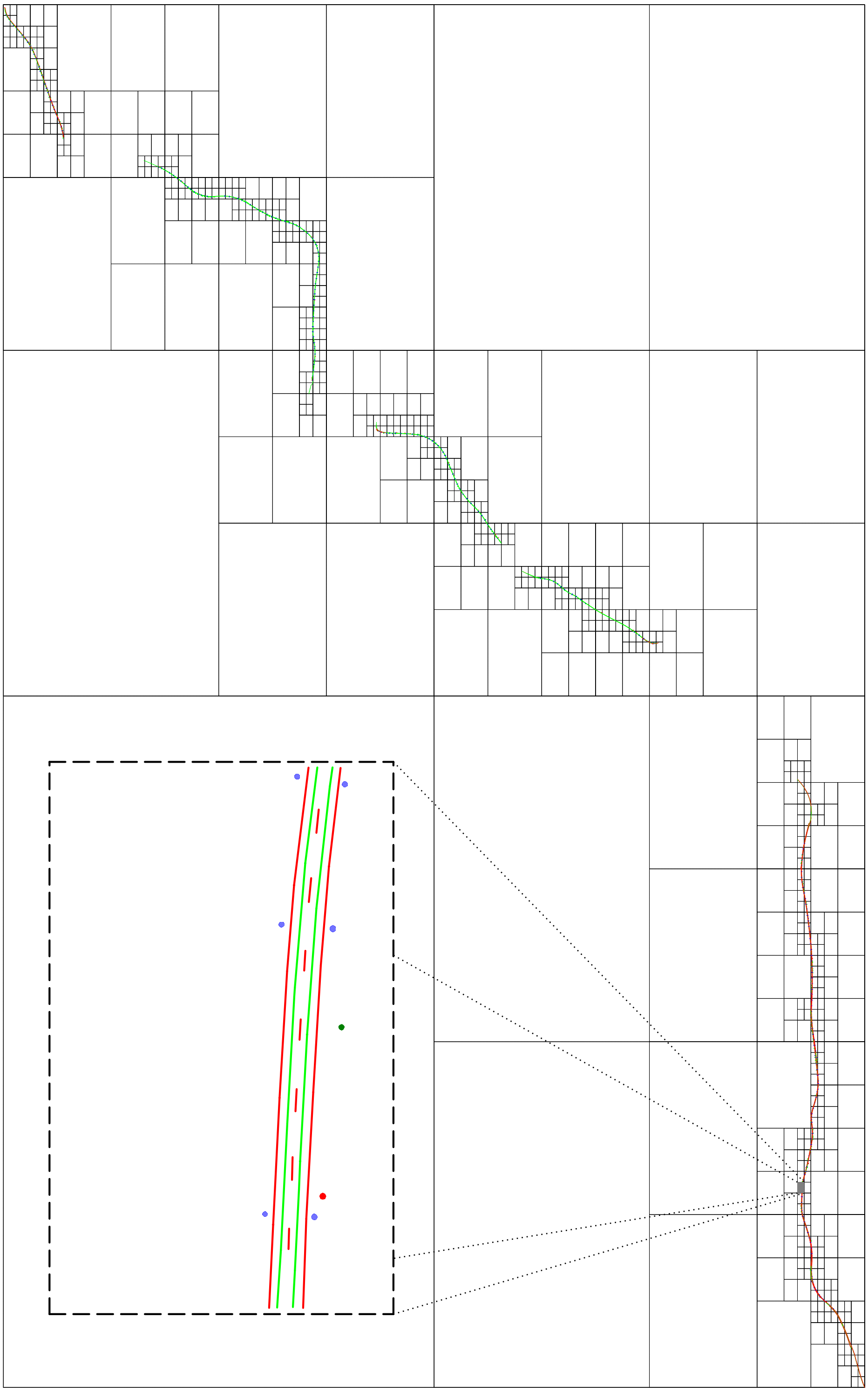


Figure 6.4.2: Quadtree decomposition of a digital map up to the tree height 8. The zoom shows the map section of a leaf, restricted to its corresponding rectangle. Road markings are drawn in red while lanes are marked in green. Blue dots mark reflection posts while the red and green dots represent a traffic sign and a tree.

6.5.1.1 Prototype fitting

Shape matching or shape registration is the basis for many computer vision techniques, such as image segmentation and pose estimation with applications in object recognition and quality assurance tasks. Therefore, several publications on shape matching can be found in the literature. A survey and a short summary is given in [Veltkamp 99] or [Rosenhahn 06]. Most of these approaches rely on classic explicit shape representations given by points, which are possibly connected by lines or other types of curve segments in order to form a shape.
The most common method working on explicit shape representations is the *iterated closest point (ICP)* algorithm (cf. [Besl 92]): Given two shapes and an error metric, the task is to find a mapping in an admissible class of transformations which leads to the minimum distance between the two shapes. The ICP algorithm is then searching for an optimal rigid motion $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ matching a finite point set $A \subset \mathbb{R}^2$ to another set $B \subset \mathbb{R}^2$ as follows: For each point $y \in A$, the best approximating point $x_y \in B$ regarding the Euclidean distance is calculated. Then, the optimal transformation $\tilde{\Phi}$ is determined that minimizes the sum of squared distances $\left\|\tilde{\Phi}(y) - x_y\right\|^2$ between pairs of closest points (y, x_y). Having this transformation applied to the point set A, the three steps explained above are repeated until the algorithm converges. The convergence of this algorithm is ensured to the next local minimum of the sum of squared distances between closest points. Hence, a good initial estimate is required to ensure reaching the sought solution. In order to improve the rate of convergence and to match partially overlapping point sets, several variants of the ICP algorithm have been developed in the last decades (cf. [Rusinkiewicz 01]).
The so-called *prototype fitting*, which was introduced in [Donner 97], is a generalization of the ICP algorithm. Within this work, broader classes of admissible transformations are treated, algorithmic improvements have been made and the corresponding theoretical bounds and convergence behaviors are analyzed. For our purposes, the reference geometry is encoded as a compact subset $\Pi \subset \mathbb{R}^2$, e.g. as a union of boundary curves, and it is called *prototype*. If points $y \in A$ have been extracted within a measurement process, the *prototype fitting problem* is the challenge to find a feasible transformation $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ minimizing the sum of squares

$$\sum_{y \in A} \operatorname{dist}\left(\Phi(\Pi), y\right)^2, \tag{6.5.14}$$

where dist denotes the Euclidean distance[3]. Within any prototype fitting problem, the determination of best approximating points $x_y \in \Pi$ for each $y \in A$ is the bottleneck of the computing time. Therefore, a suitable encoding of Π, which minimizes the computational requirements for the best approximating point calculation, is essential.
Naturally, the existence of optimal motions can only be assured if some restrictions and assumptions on the feasible transformations are made. In the present case, the admissible transformations are restricted to rotations and translations, which are mappings of the form

$$T_{\varphi,t} : \mathbb{R}^2 \to \mathbb{R}^2, \quad x \mapsto A_\varphi \cdot x + t, \tag{6.5.15}$$

[3]The question whether the prototype Π or the points A are transformed by Φ can be decided according to the computational effort. The prototype fitting procedure, described in the following, can be formulated for both decisions analogously.

with $\varphi \in [0, 2\pi[, t \in \mathbb{R}^2$ and

$$A_\varphi := \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \in SO(2) \tag{6.5.16}$$

where $SO(2)$ denotes the special orthogonal group[4] in $\mathbb{R}^2$. Hence, optimal parameters $\varphi \in [0, 2\pi[$ and $t \in \mathbb{R}^2$ are searched within the prototype fitting problem. The set $\mathcal{T}$ of all feasible mappings is defined by

$$\mathcal{T} := \left\{ T_{\varphi,t} : \mathbb{R}^2 \to \mathbb{R}^2 \;\middle|\; \varphi \in [0, 2\pi[, t \in \mathbb{R}^2 \right\}. \tag{6.5.17}$$

It should be noted that, even for a larger class of admissible transformations, the pose estimation regarding rotation and translation is important to have a first match. Afterwards, possibly more sophisticated transformations, like non-isotropic scalings, projections or spline deformations can be taken into account. Thus, focusing on the class $\mathcal{T}$ is sufficient within this scope. Then, the problem can be solved very fast by an iterative approach.

To begin with, it is assumed that some initial transformation parameters φ_0, t_0 are known such that the transformed prototype approximately fits to the given points. In order to obtain such an initialization, some appropriate methods are presented in [Maier 11b].

In any case, after having found an initial transformation, the following iteration is performed: Starting with $\Pi^{(0)} := \Pi$, in the j-th step for $j \in \mathbb{N}$, the best approximating points $x_i^{(j)}$ of y_i with respect to the set

$$\Pi^{(j)} := T_{\varphi_{j-1}, t_{j-1}}(\Pi^{(j-1)}) \tag{6.5.18}$$

are computed. Using the abbreviation

$$\widetilde{x}_i^{(j)} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} x_i^{(j)} \text{ for all } i = 1, \ldots, n \tag{6.5.19}$$

and denoting the barycenters of $x_1^{(j)}, \ldots, x_n^{(j)}$ and $y_1, \ldots, y_n$ by

$$\mu_{x^{(j)}} := \frac{1}{n} \sum_{i=1}^{n} x_i^{(j)} \text{ and } \mu_y := \frac{1}{n} \sum_{i=1}^{n} y_i, \tag{6.5.20}$$

the optimal values $\varphi_j \in [0, 2\pi[$ and $t_j \in \mathbb{R}^2$, i.e.

$$\sum_{i=1}^{n} \left\| T_{\varphi_j, t_j}(x_i^{(j)}) - y_i \right\|^2 = \min_{\varphi \in [0,2\pi[, t \in \mathbb{R}^2} \sum_{i=1}^{n} \left\| T_{\varphi,t}(x_i^{(j)}) - y_i \right\|^2 \tag{6.5.21}$$

can be derived in a closed form (cf. [Maier 11b, Donner 97]): Using $c_j, s_j \in \mathbb{R}$ with

$$c_j = \frac{1}{\rho} \sum_{i=1}^{n} (x_i^{(j)} - \mu_{x^{(j)}})^T y_i, \;\; s_j = \frac{1}{\rho} \sum_{i=1}^{n} (\widetilde{x}_i^{(j)} - \mu_{x^{(j)}})^T y_i, \tag{6.5.22}$$

where $\rho = \sum_{i=1}^{n} \left\| x_i^{(j)} - \mu_x^{(j)} \right\|^2$, the optimal rotation angle[5] is given by

$$\varphi_j = \arccos \left(\frac{c_j}{\sqrt{c_j^2 + s_j^2}} \right). \tag{6.5.23}$$

[4] The special orthogonal group in $\mathbb{R}^n$ is defined by $SO(n) := \left\{ M \in \mathbb{R}^{n \times n} \;\middle|\; \det(M) = 1,\; M^T = M^{-1} \right\}$ for any $n \in \mathbb{N}$.

[5] As explained in Section 7.2.1, it is not necessary to calculate the angle φ_j explicitly. Instead, accounting for its sine and cosine values is sufficient.

The optimal translation is given by $t_j = \mu_y - A_{\varphi_j}\mu_{x^{(j)}}$. Using the abbreviation $T_j := T_{\varphi_j,t_j}$, the value

$$E_j := \frac{1}{n}\sum_{i=1}^{n}\left\|T_j(x_i^{(j)}) - y_i\right\|^2 \tag{6.5.24}$$

indicates the prototype fitting error at step j. The sequence $(E_j)_{j\in\mathbb{N}}$ is monotonic decreasing since for all $j > 0$ it is true that

$$E_j = \frac{1}{n}\sum_{i=1}^{n}\left\|T_j(x_i^{(j)}) - y_i\right\|^2 \tag{6.5.25}$$

$$\overset{(a)}{\leq} \frac{1}{n}\sum_{i=1}^{n}\left\|x_i^{(j)} - y_i\right\|^2 \tag{6.5.26}$$

$$\overset{(2.1.3)}{=} \frac{1}{n}\sum_{i=1}^{n}\operatorname{dist}\left(\Pi^{(j)}, y_i\right)^2 \tag{6.5.27}$$

$$\overset{(6.5.18)}{=} \frac{1}{n}\sum_{i=1}^{n}\operatorname{dist}\left(T_{j-1}(\Pi^{(j-1)}), y_i\right)^2 \tag{6.5.28}$$

$$\overset{(b)}{\leq} \frac{1}{n}\sum_{i=1}^{n}\| \underbrace{T_{j-1}(\underbrace{x_i^{(j-1)}}_{\in\Pi^{(j-1)}})}_{\in\Pi^{(j)}} - y_i\|^2 = E_{j-1} \tag{6.5.29}$$

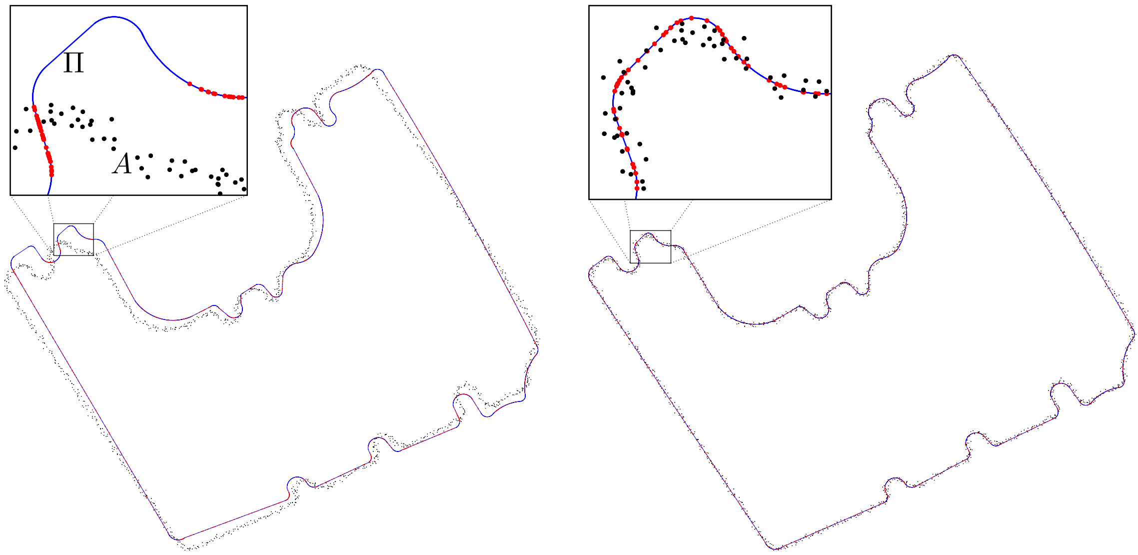
Inequality (a) holds since T_j minimizes the optimization problem formulated in (6.5.21). Furthermore, inequality (b) is true since for any $i \in \{1,\ldots,n\}, j \in \mathbb{N}$ and for any $p \in \Pi^{(j)}$

$$\operatorname{dist}\left(\Pi^{(j)}, y_i\right) = \left\|x_i^{(j)} - y_i\right\|_2 \leq \|p - y_i\|_2 \tag{6.5.30}$$

holds by definition of the best approximating points (2.1.1.2).
By increasing j, the best approximating points $x_i^{(j+1)}$ of y_i with respect to $\Pi^{(j+1)}$ can be computed and the least squares problem is solved iteratively. This alternating procedure is continued while E_j is greater than some given threshold or the difference between the predecessor error and the current error is not smaller than some given threshold. Figure 6.5.3 illustrates the prototype fitting of a curve prototype and some noisy fitting points.

6.5.1.2 Prototype Encoding

In order to achieve an efficient computation of this iterative method, it is decisive that the calculation of the best approximating points with respect to Π is very fast. As already discovered in [Besl 92] and [Donner 97], the necessity of a fast determination of best approximating points does not depend on a special choice of the optimization method but it is also crucial when using any other nonlinear optimization algorithm, like Gauß-Newton or Levenberg-Marquardt (cf. [Nocedal 99]).
Indeed, the efficiency of the calculation of closest points depends on the encoding of the prototype Π. Therefore, a description of Π as a union of curves having a preferably low number of segments and providing a fast calculation of best approximating points is preferable. Furthermore, high flexibility for modeling the desired geometric pattern is needed. Since almost all ICP methods are



(a) Initial situation

(b) Final situation after convergence

Figure 6.5.3: Illustration of the prototype fitting. The prototype Π is encoded as an arc spline in blue. The noisy fitting points A are drawn in black and their best approximating points on Π are depicted in red.

based on point encodings of the prototype, sophisticated point selection approaches are needed to achieve efficiency improvements. Obviously, a curve representation, as described above, has considerable advantages over these techniques regarding accuracy, time and storage space.

One possible choice of such a curve fulfilling these criteria are arc splines, as discussed in Section 2.1. Methods for efficiently calculating best-approximating points with respect to arc splines are shown in Section 2.1.3.

6.5.1.3 Observation model

In the present case, the prototype fitting principle is used for the observation model in the following sense: Based on the map section $map_r = (L_r, M_r, R_r)$ from (6.4.9) around the current position estimation of the vehicle, the prototype consists of the set of arc splines contained in the road markings of map_r.

$$\Pi := \bigcup_{(\gamma,A)\in M_r} \operatorname{tr}(\gamma) \tag{6.5.31}$$

Furthermore, the fitting points required by the prototype fitting correspond to the measurement points P_{LR} (cf. 6.5.1) extracted by the lane recognition. The initial transformation parameters are based on the pose parameters $(x_k, y_k, \psi_k) \in \mathbb{R}^3$ of the vehicle state $\boldsymbol{x}_k$ (6.2.1) at time t_k. Related to the inverse mapping of equation (5.4.15), the initial transformation T_0 is given by (6.5.15) with the parameters

$$\varphi_0 := \psi_k - \frac{\pi}{2} \in [0, 2\pi[\tag{6.5.32}$$

$$t_0 := -A_{\varphi_0} \cdot \begin{pmatrix} x_k \\ y_k \end{pmatrix} \in \mathbb{R}^2. \tag{6.5.33}$$

Regarding the filtering methodology described in Section 2.2.2.2, the fitting points $P_{LR} = y_1, \ldots, y_n$ represent some real measurements of a measurement update. Their corresponding predicted measurements are given by the best approximating points with respect to the prototype Π transformed by the initial transformation. Therefore, the road marking observation model can be expressed as a family of functions $(h_i)_{i=1,\ldots,n} : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}^2$ for any state $\boldsymbol{x}_k \in \mathbb{R}^6$ at time t_k:

$$h_i(\boldsymbol{x}_k, t_k) = \underset{p \in T_0(\Pi)}{\operatorname{argmin}} \operatorname{dist}(y_i, p) \tag{6.5.34}$$

where dist denotes the Euclidean distance[6]. Thus, the residual vector (cf. (2.2.44))

$$v_k := \begin{pmatrix} y_1 - h_1(\boldsymbol{x}_k, t_k) \\ \vdots \\ y_n - h_n(\boldsymbol{x}_k, t_k) \end{pmatrix} \tag{6.5.35}$$

is related to the prototype fitting error (6.5.24) since

$$\frac{1}{n} \sum_{i=1}^{n} \left\| x_i^{(1)} - y_i \right\|^2 = \frac{1}{n} \left\| v_k \right\|^2 \tag{6.5.36}$$

for the best approximating points $x_i^{(1)}$ of y_i for $i = 1, \ldots, n$ with respect to the transformed prototype $T_0(\Pi)$.

Figure 6.5.4 illustrates the prototype fitting principle for road markings. The measurement points of the lane recognition (red pyramids) are associated with the best approximating points on the road markings. The correspondences allow computing the transformation for the best fit.

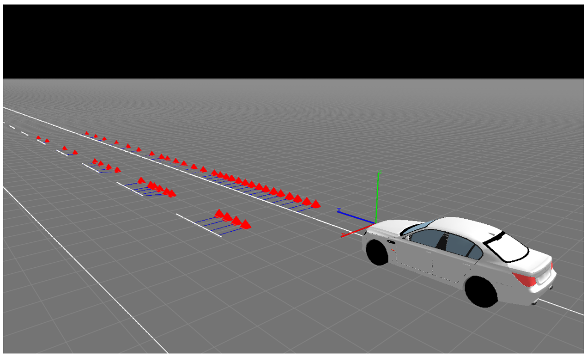
6.5.2 Observation model for landmarks

The real measurements of the landmark observation model are given by detections of the laser scanner processing described in Section 4.4. The result is a list of landmark hypotheses $H_k := \{p_1, \ldots, p_n \in \mathbb{R}^2\}, n \in \mathbb{N}$ at time t_k, represented by their coordinates in the X, Y-plain of the vehicle frame.

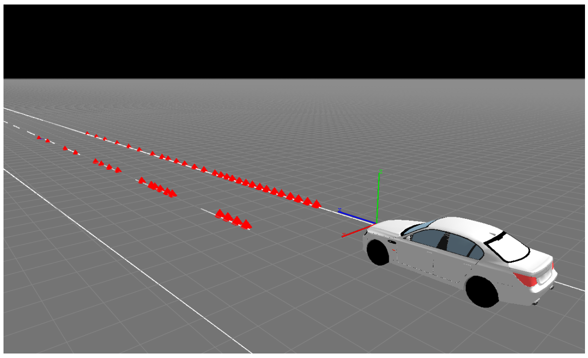
Analogous to the road marking observation model (6.5.1.3), let $map_r = (L_r, M_r, R_r)$ denote a map section around the current position estimation. Referring to (5.3.2.1), let

$$L_k := \bigcup_{(p,A) \in L_r} \{p\} \tag{6.5.37}$$

[6] In general, the term $\underset{x \in D}{\operatorname{argmin}} f(x)$ is set-valued. In this case, an adequate selection strategy is applied, such that we write $x_{min} = \underset{x \in D}{\operatorname{argmin}} f(x)$ instead of $x_{min} \in \underset{x \in D}{\operatorname{argmin}} f(x)$



(a) Association of measurement points with road markings



(b) Constellation after the prototype fitting

Figure 6.5.4: Illustration of the prototype fitting on road markings

be the set of landmark positions represented within map_r. Applying transformation T_0 as defined in Section 6.5.1.3 with respect to the current vehicle state estimation $\boldsymbol{x}_k$ to L_k leads to the predicted measurements of the landmark observation model. Thus, the observation model can be seen as a function $h : \mathbb{R}^6 \times \mathbb{R} \to \mathcal{P}(\mathbb{R}^2)$:

$$h(\boldsymbol{x}_k, t_k) = T_0(L_k) \tag{6.5.38}$$

for any $\boldsymbol{x}_k \in \mathbb{R}^6, k \in \mathbb{N}$ and where $\mathcal{P}(\mathbb{R}^2)$ denotes the power set of $\mathbb{R}^2$.
The association of the real measurements H_k and the predicted measurements $T_0(L_k)$ is a problem that has been treated widely in the literature. As already discussed in Section 2.2.2.1, the more general data association problem between real and predicted measurements can be handled with probabilistic techniques presented in [Bar-Shalom 09, Bar-Shalom 95, Musicki 94]. Especially in the field of vehicle localization on urban or rural road some approaches are discussed in [Weiss 11, Pink 11, Weiss 05b, Lu 97].

6.5.3 Observation model for vehicle dynamics

In the present case, the dynamics parameters of the vehicle state $\boldsymbol{x}_k$ (6.2.1) are directly observable from the intrinsic measurements described in Section 4.3. These real measurements are represented by a vector $(v_k, c_k, \beta_k) \in \mathbb{R}^3$, denoting the absolute value of the velocity v_k, the curvature of the driven circular track c_k and the slip angle β_k. Hence, the observation model for the vehicle dynamics parameters can be expressed as a linear function $h : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}^3$ for any $\boldsymbol{x}_k \in \mathbb{R}^6, k \in \mathbb{N}$ with

$$h(\boldsymbol{x}_k, t_k) = A \cdot \boldsymbol{x}_k \tag{6.5.39}$$

and

$$A := \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.5.40}$$

6.5.4 Observation model for GPS

The GPS observation model describes the way, GPS measurements are integrated in the filtering process. Again, let $\left(x_{k|k-1}^{(i)}\right)_{1\leq i\leq M}$ be the set of $M \in \mathbb{N}^+$ particles at time t_k according to the notations in Section 2.2.2.2. Furthermore, let the geographic longitude $\lambda_k \in [-\pi, \pi]$, the geographic latitude $\varphi_k \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and the yaw angle $\psi_k \in [0, 2\pi]$ be the measurements of the GPS unit at time t_k. Since the pose parameters of the vehicle are modeled in the navigation frame, the position coordinates (λ_k, φ_k) need to be transformed from the world frame to the navigation frame using the methods described in Section 3.1.2. Let therefore $(l_k, b_k) \in \mathbb{R}^2$ denote the corresponding coordinates of (λ_k, φ_k) with respect to the navigation frame. Thus, the real measurements are given by the vector $(l_k, b_k, \psi_k) \in \mathbb{R}^3$ and the GPS observation model is a function $h : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R}^3$ for any $\boldsymbol{x}_k \in \mathbb{R}^6, k \in \mathbb{N}$ with

$$h(\boldsymbol{x}_k, t_k) = A \cdot \boldsymbol{x}_k \tag{6.5.41}$$

and

$$A := \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{6.5.42}$$

However, GPS is only used for a rough reinitialization due to its expected positional inaccuracy. Regarding the particle filter principle, this aspect can be modeled explicitly by applying the GPS observation model not to all particles, but to only a (small) subset $\left(x_{k|k-1}^{(i)}\right)_{1\leq i\leq j}$ of the particles $\left(x_{k|k-1}^{(i)}\right)_{1\leq i\leq M}$ for $j \in \mathbb{N}, j < M$.

6.6 Resampling strategies

Within the particle filtering scheme discussed in Section 2.2.2.2, in each iteration, a resampling strategy is applied in order to redistribute the particles $\left(x_{k|k}^{(i)}\right)_{1\leq i\leq M}$ according to their weighting $w_k^{(i)}$ for all $1 \leq i \leq M$. In addition to this procedure, some alternative resampling strategies can be considered that allow modeling explicitly some preliminary knowledge on the particle distribution.

6.6.1 Resampling by prototype fitting

The goal of the prototype fitting described in Section 6.5.1.1 is to determine a feasible transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$ minimizing the sum of squares

$$\sum_{y\in A} \operatorname{dist}\left(T(\Pi), y\right)^2, \tag{6.6.43}$$

for a given finite set of fitting points $A \subset \mathbb{R}^2$, some compact prototype $\Pi \subset \mathbb{R}^2$ and the Euclidean distance dist.

In the present case, this method can be used for a resampling strategy of particles in the following sense: Analogous to the road marking observation model in Section 6.5.1.3, let the prototype Π

be defined by the road markings given in equation (6.5.31) and let the fitting points be given by P_{LR} (cf. 6.5.1), extracted by the lane recognition. Furthermore, the initial transformation T_0 is based on the pose parameters $(x_k, y_k, \psi_k) \in \mathbb{R}^3$ of the vehicle state $\boldsymbol{x}_k$ (6.2.1) at time t_k as in Section 6.5.1.3.

Since the prototype fitting determines a transformation that best fits Π onto P_{LR} in the least-squares sense above, it can be used to correct the pose estimation represented in each particle $\left(x_{k|k}^{(i)}\right)_{1\leq i\leq M}$ at time t_k in a direct way. Therefore, for any $j \in \mathbb{N}$ let $T_j := T_{\varphi_j,t_j}$ denote the optimal feasible transformation at step j of the prototype fitting according to (6.5.21) with $\varphi_j \in [0, 2\pi[$ and $t_j \in \mathbb{R}^2$. Let $n \in \mathbb{N}$ denote the number of applied fitting steps leading to the final transformation

$$T^\star := T_n \circ \cdots \circ T_0. \tag{6.6.44}$$

It is easy to show that $\mathcal{T}$ is closed with respect to composition, hence, $T^\star = T_{\varphi^\star,t^\star}$ is an element of $\mathcal{T}$. The inverse[7] mapping $(T^\star)^{-1} : \mathbb{R}^2 \to \mathbb{R}^2$ of $T^\star$ can now be used to characterize the parameters of the corrected pose estimation.

$$(T^\star)^{-1} := A_{-\varphi^\star} \cdot id - A_{-\varphi^\star} \cdot t^\star \tag{6.6.45}$$

with $A_{-\varphi^\star}$ from (6.5.16). Finally, the corrected state pose parameters $x_k^\star, y_k^\star, \psi_k^\star \in \mathbb{R}$ can be expressed in terms of the components in (6.6.45):

$$\begin{pmatrix} x_k^\star \\ y_k^\star \end{pmatrix} := -A_{-\varphi^\star} \cdot t^\star \tag{6.6.46}$$

$$\psi_k^\star := \varphi^\star + \frac{\pi}{2} \tag{6.6.47}$$

Based on these corrected pose parameters and using (6.5.15) and (6.5.32), the parameters of the initial transformation $T_0^\star$ result in

$$\varphi_0 = \varphi^\star \tag{6.6.48}$$

$$t_0 = -A_{\varphi_0} \cdot \begin{pmatrix} x_k^\star \\ y_k^\star \end{pmatrix} = t^\star \tag{6.6.49}$$

which in turn leads to $T_0^\star = T_{\varphi^\star,t^\star} = T^\star$. As already introduced in (6.5.24), let the prototype fitting error at step j be defined by

$$E_j := \frac{1}{m} \sum_{i=1}^{m} \left\| T_j(x_i^{(j)}) - y_i \right\|^2, \tag{6.6.50}$$

where $x_i^{(j)}$ denote the best approximating points of y_i with respect to $\Pi^{(j)}$. Based on the argumentation in (6.5.25), it is true that

[7]It is easy to show that $(T^\star)^{-1} \circ T^\star = id$

$$
\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m}\operatorname{dist}\left(T_0^{\star}(\Pi), y_i\right)^2 &= \frac{1}{m}\sum_{i=1}^{m}\operatorname{dist}\left(T^{\star}(\Pi), y_i\right)^2 && (6.6.51)\\
&= \frac{1}{m}\sum_{i=1}^{m}\operatorname{dist}\left(T_n \circ \cdots \circ T_0(\Pi), y_i\right)^2 && (6.6.52)\\
&= \frac{1}{m}\sum_{i=1}^{m}\operatorname{dist}\left(\Pi^{(n+1)}, y_i\right)^2 && (6.6.53)\\
&\leq \frac{1}{m}\sum_{i=1}^{m}\operatorname{dist}\left(T_n(x_i^{(n)}), y_i\right)^2 && (6.6.54)\\
&= E_n \leq E_0 && (6.6.55)\\
&= \frac{1}{m}\sum_{i=1}^{m}\operatorname{dist}\left(T_0(x_i^{(0)}), y_i\right)^2. && (6.6.56)
\end{aligned}
$$

Hence, the parameter correction in (6.6.46) and (6.6.47) improves the pose estimation regarding the prototype fitting error. As mentioned in the introduction of Section 6.6, the presented resampling strategy is applied to the particles $\left(x_{k|k}^{(i)}\right)_{1\leq i\leq M}$ or to a subset of it.

6.6.2 Resampling by map matching

Based on the assumption that vehicles are likely to drive on a specific lane within the drivable area of a road, another resampling strategy is related to the map matching principle: For any given vehicle position, the best approximating point on the nearest lane within a local map section is computed.

Analogous to definition (6.4.9), let $map_r = (L_r, M_r, R_r)$ be a local map section around the position $p = (x_k, y_k) \in \mathbb{R}^2$ and orientation $\psi_k \in \mathbb{R}$ of the vehicle state $\boldsymbol{x}_k$ (6.2.1) at time t_k. Furthermore, let $\Gamma \subset \mathbb{R}^2$ be defined by

$$
\Gamma := \bigcup_{(\gamma,\nu,A)\in R_r} \operatorname{tr}(\gamma). \qquad (6.6.57)
$$

Using the methods in Section 2.1.3 and (2.1.3), a point $p_0 \in \Gamma$ is computed satisfying

$$
\|p - p_0\|_2 = \operatorname{dist}(p, \Gamma) \qquad (6.6.58)
$$

and it represents the new vehicle position regarding this resampling strategy.

Let $\gamma_0 \in \mathfrak{S}$ denote the arc spline of a lane for which $p_0 \in \operatorname{tr}(\gamma_0)$ holds. Since the drivable direction on a lane can be determined according to Section 5.3.4, the tangent vector $\tau_{\gamma_0}(p_0) \in \mathbb{R}^2$ (cf. (2.1.1.1)) can be used to deduce a likely corresponding orientation of the vehicle.

However, since the assumption that the vehicle is always driving on lanes modeled within the digital map represents a strong assumption, this resampling strategy should be handled carefully. In the present case, it is rather used for initialization purposes taking into account a priori knowledge from the map by applying the resampling method to a small subset of the particles.

Analogous to this resampling strategy, more hypotheses on vehicle poses can be modeled explicitly by spreading a subset of particles on each individual neighboring lane in a local environment of the vehicle.

Chapter 7

Results

'There are no such things as applied sciences,
only applications of science.'
(Louis Pasteur)

Contents

7.1 Results from the map generation

In the following, some map generation results are shown concerning the application of the presented methods on both synthetic and real data. In the first section some results for the approximation of a clothoid are presented. Then, in order to evaluate the map building methods on different kinds of roads, some results of the arc spline approximation are shown based on data of a typical highway and a winding mountain road. Finally, the mapping results for the digital rural map in Ko-PER are presented.

7.1.1 Approximation of clothoids

As discussed in Section 5.2.2, turns on a road in rural areas are – at least constructionally – commonly composed by a sequence of a straight line segment, a clothoid, a circular arc, a clothoid and again a straight line segment. Defined by the equations (5.2.1) to (5.2.3), the considered clothoid is determined by the length $L \in \mathbb{R}$ of the curve, the curvature radius $R \in \mathbb{R}$ at the endpoint and the scaling factor $a \in \mathbb{R}$, respectively.

Since clothoids are not part of the lane model within this work, the question arises how clothoids can be approximated by smooth arc splines. Such an approximation scheme is presented in [Meek 04]. The so-called *discrete clothoid*, which is a smooth arc spline, results by approximating a clothoid with a sequence of arcs such that the length of the individual arcs are constant, as depicted in Figure 7.1.1. The curvature of the discrete clothoid, which is a step function, increases in each step by $\frac{1}{R \cdot m}$ where m is the number of arc segments. While the method proposed in [Meek 04] allows controlling the approximation error, it does not guarantee the minimal possible number of arc segments. However, the authors show that using n-times the number of segments the approximation error behaves like $O(n^{-2})$.

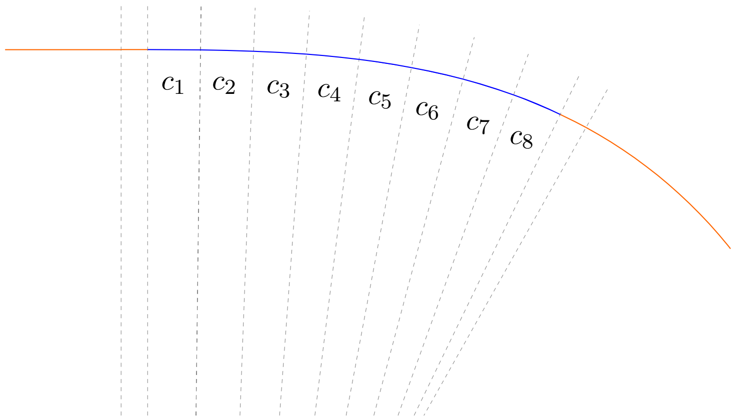


Figure 7.1.1: Approximation of a clothoid by a smooth arc spline (discrete clothoid, in blue) with 8 segments $c_1, \dots, c_8$ smoothly joining a given line and arc segment.

In the following experiment, clothoids of different lengths L and radii R at the endpoint are considered. Given an approximation tolerance $\varepsilon \in \mathbb{R}$, the approximation result described in [Meek 04] is compared with the arc spline approximation used in this thesis. Furthermore, the number of segments of the corresponding *Minimum Link Path (MLP)*[1] (cf. Section 5.2.1, [Suri 86]) is indicated.

According to the road construction regulation in [BFV 93] and [Richter 08], the clothoid parameter a is restricted with respect to R:

$$\pi \frac{R}{3} \leq a \leq \pi R \tag{7.1.1}$$

Using (5.2.3), the minimal and maximal length L_{min} and L_{max} is given by

$$L_{min} = \frac{R}{9}, \quad L_{max} = R \tag{7.1.2}$$

These lower and upper bounds correspond to the minimal and maximal deviation angle between the tangents at the start point and at the endpoint of the clothoid under all feasible clothoid parameters a respecting (7.1.1).

[1] Again, given a simple closed polygon P – in this case P approximately represents the ε-offset of the clothoid – with a source and destination vertex, the MLP-algorithm produces a sequence of line segments connecting the source and destination inside P with a minimal number of segments.

	15	25	50	100	250	500	1000	2500	5000
MLP	3	3	4	6	9	12	17	27	38
DC	1	2	2	3	4	6	8	13	18
SMAP	1	2	2	2	3	4	5	7	10

Table 7.1.1: Resulting segment numbers for a clothoid approximations with the Minimum Link Path (MLP), the discrete clothoid (DC) and the arc spline approximation used within this thesis (SMAP) with respect to the tolerance $\varepsilon = 0.1\,\mathrm{m}$ and different arc lengths L (in meters). The values correspond to the extremal clothoid parameter $a = \pi R$ or $L = R$, respectively.

The resulting number of segments for the approximation of clothoids of different lengths using the Minimum Link Path, the discrete clothoid and a SMAP are listed in table 7.1.1 and illustrated in Figure 7.1.2. One can see that the SMAP approximation outperforms both the MLP and the discrete clothoid regarding the number of segments.

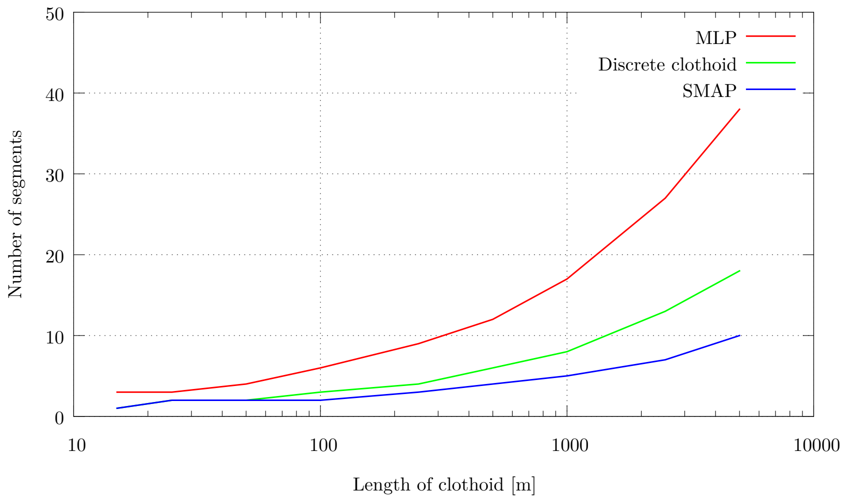
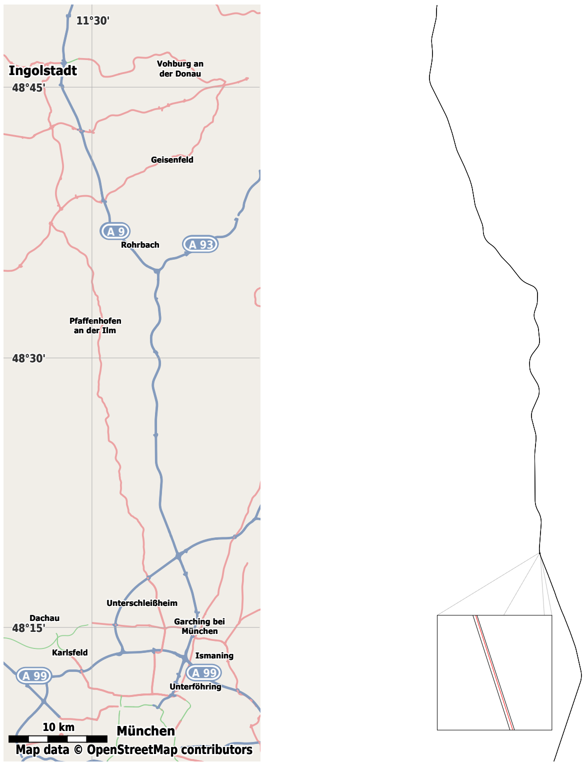


Figure 7.1.2: Illustration of the numbers of segments for the clothoid approximation results in table 7.1.1. Note that the horizontal axis has a logarithmic scale.

More details and results concerning different clothoid parameters and tolerance values are presented in [Schindler 11].

7.1.2 Map building on a highway

In order to evaluate the arc spline approximation for highways, a single track of the highway between Munich and Ingolstadt (cf. Figure 7.1.3) is considered with a length of $74,60\,\mathrm{km}$. The corresponding ground data set of the track is represented by a point sequence of $M = 37301$ two-dimensional points. In the following experiment, this point sequence is approximated with a smooth arc spline (SMAP) using the methods described in Section 5.4. As already explained there, the approximation error (regarding the maximum norm) which is controlled by the tolerance $\varepsilon \in \mathbb{R}^+$ (cf. Section 5.4.3.2) is varied to show the performance for different map accuracy levels.



Left: OpenStreetMap section of the highway (between Munich and Ingolstadt in Bavaria).
Right: SMAP approximation (red) of the input point sequence. The zoom shows the tolerance channel with $\varepsilon = 0.1\,\text{m}$.

Figure 7.1.3: Map generation on a highway section

Figure 7.1.4 shows the resulting segment numbers of the SMAP. Additionally, the segment number of the polygonal approximation of the point sequence with a Minimum Link Path (MLP) is depicted.

Table 7.1.2 summarizes some approximation results concerning the segment numbers depending on the tolerance ε. The compression factor *comp* is defined by the ratio $comp := \frac{2 \cdot M}{2 \cdot n_{SMAP} + 3}$ based on the total number of input points M and the SMAP segment number n_{SMAP}. The definition of *comp* reflects the discussion in Section 5.2.4, where it is shown that encoding a smooth arc spline with n_{SMAP} segments requires $2 \cdot n_{SMAP} + 3$ floating points, while the representation of the M input points results in $2 \cdot M$ floating points. The compression *comp* shows the significant data reduction when using curves instead of point sequences. Furthermore, the data volume in Byte per kilometer for storing the resulting arc spline approximation based on 64 Bit floating point numbers is indicated.

In principle, the segment number of the approximation result decreases if the tolerance is increased. For this experiment, the SMAP extension described in Section 5.4.3.2 is applied to detect straight line segments in a heuristic way. The latter avoids arc segments with large radii in favor of line segments. However, due to the heuristic integration strategy of line segments, the total number of spline segments n_{SMAP} can increase for larger tolerances ε (transition from $\varepsilon = 0.7$ to $\varepsilon = 0.8$), which would not be the case for the standard version of the SMAP algorithm. Regarding the ratio $\frac{n_{MLP}}{n_{SMAP}}$, one can see that the SMAP algorithm reduces the segment number significantly compared to the polygonal MLP results. When interpreting the segment numbers, it should be emphasized that a SMAP, in contrast to an MLP, additionally reconstructs the tangent information and provides an indicator for the curvature.

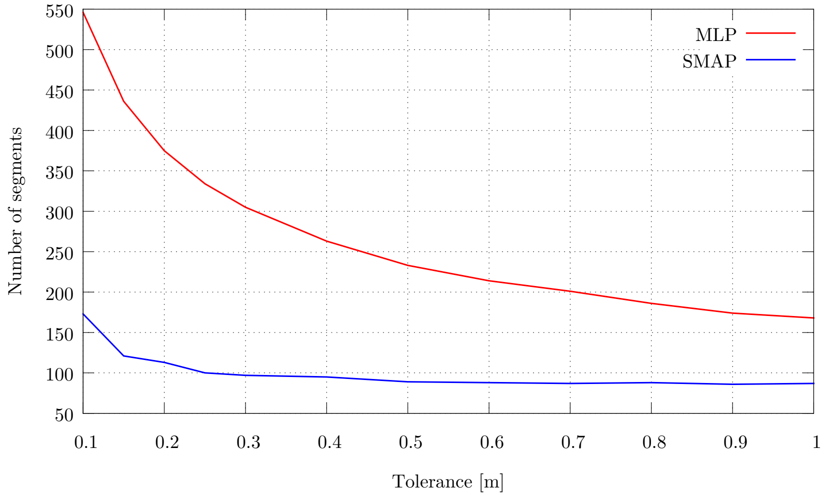


Figure 7.1.4: Segment number for different tolerance values on the highway track.

ε	n_{SMAP}	n_l	$\frac{Byte}{km}$	$comp$	n_{MLP}	$\frac{n_{MLP}}{n_{SMAP}}$	$RMSE$	l_{avg}	l_{max}
0.10	173	21	37.4	213.75	546	3.15	0.0574	431.21	1813.79
0.15	121	21	26.2	304.49	436	3.60	0.0850	616.52	2581.56
0.20	113	22	24.5	325.77	375	3.31	0.118	660.17	2805.16
0.25	100	22	21.7	367.49	334	3.34	0.148	746.00	4012.65
0.30	97	21	21.1	378.69	305	3.14	0.180	769.07	4010.55
0.40	95	23	20.6	386.53	263	2.76	0.248	785.26	4008.27
0.50	89	24	19.4	412.16	233	2.61	0.319	838.20	4006.45
0.60	88	24	19.1	416.77	214	2.43	0.392	847.72	4014.83
0.70	87	24	18.9	421.48	201	2.31	0.461	857.47	4002.81
0.80	88	23	19.1	416.77	186	2.11	0.523	847.72	4013.47
0.90	86	22	18.7	426.29	174	2.02	0.603	867.44	4229.46
1.00	87	24	18.9	421.48	168	1.93	0.676	857.47	3871.58

Table 7.1.2: Approximation results for the highway track. The table summarizes the approximation tolerance ε, the number of SMAP segments n_{SMAP} including n_l line segments, the data volume per kilometer for the SMAP, the compression factor $comp$, the number of MLP segments n_{MLP} and the ratio between the segment numbers of the SMAP and the MLP result. Furthermore, the root-mean-square fitting error $RMSE$ of the input points, the average and maximal length of the SMAP segments are given.

The value $RMSE \in \mathbb{R}$ represents the root-mean-square error between the input points $p_i \in \mathbb{R}^2$ $1 \leq i \leq M$ and the SMAP $s \in \mathfrak{S}^\infty$:

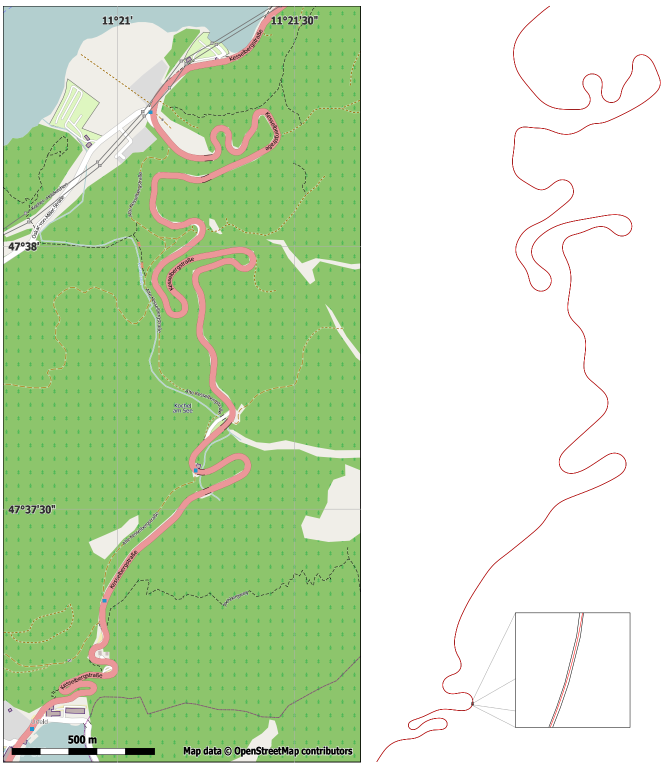
$$RMSE := \sqrt{\sum_{i=1}^{M} d(p_i, s)^2}, \tag{7.1.3}$$

where the distance d refers to definition (2.1.3). Obviously, the $RMSE$ error is smaller than ε which is defined by the maximum error. The values l_{avg} and l_{max} represent the average and maximal segment length (in meter) of the arc spline.

One can see that the segment number n_{SMAP} does not change significantly within the range of $0.50 \leq \varepsilon \leq 1.00$ though this interval represents a doubling of the approximation tolerance. This observation indicates that the resulting spline curves model the road section in a realistic way.

7.1.3 Map building on a mountain road

Analogous to the previous section, a track on a winding mountain (cf. Figure 7.1.5) road is evaluated regarding the approximation results with different tolerance values. The input point sequence consists of 21733 two-dimensional points on a track length of 5949 m.



Left: OpenStreetMap section of the winding mountain road (between Kochelsee and Walchensee in Bavaria).
Right: SMAP approximation (red) of the input point sequence. The zoom shows the tolerance channel with $\varepsilon = 0.1\,\text{m}$.

Figure 7.1.5: Map generation on a winding mountain road

Figure 7.1.6 shows the relation between the approximation tolerance and the resulting segment number for the approximation with the SMAP algorithm and the MLP algorithm. Again, one can see that the SMAP approximation reduces the number of segments significantly compared to the polygonal MLP approximation. For any tolerance $\varepsilon < 0.1\,\text{m}$, the resulting segment number

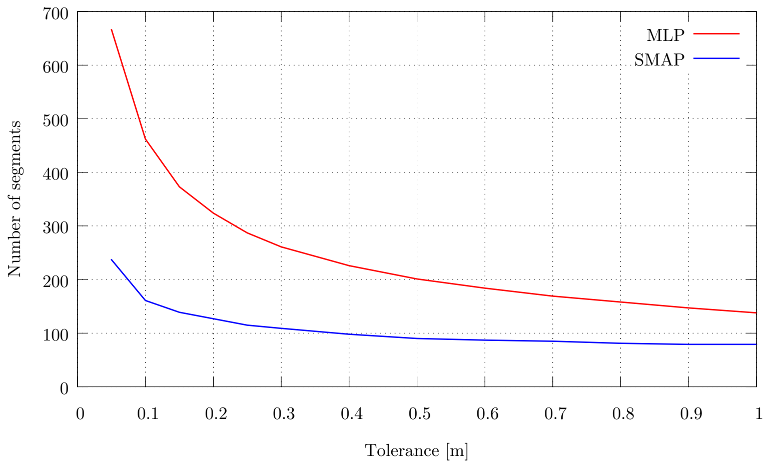


Figure 7.1.6: Segment number for different tolerance values on the mountain road track.

increases considerably. This effect is presumably due to the fact that the inaccuracy of the input points is in the same order of magnitude as ε.
The ratio $\frac{n_{MLP}}{n_{SMAP}}$ between the segment numbers increases when ε is decreased. However, it should be noted that the resulting segment number not only depends on the tolerance ε but it is also dependent on the density of the input points, as the following example shows:

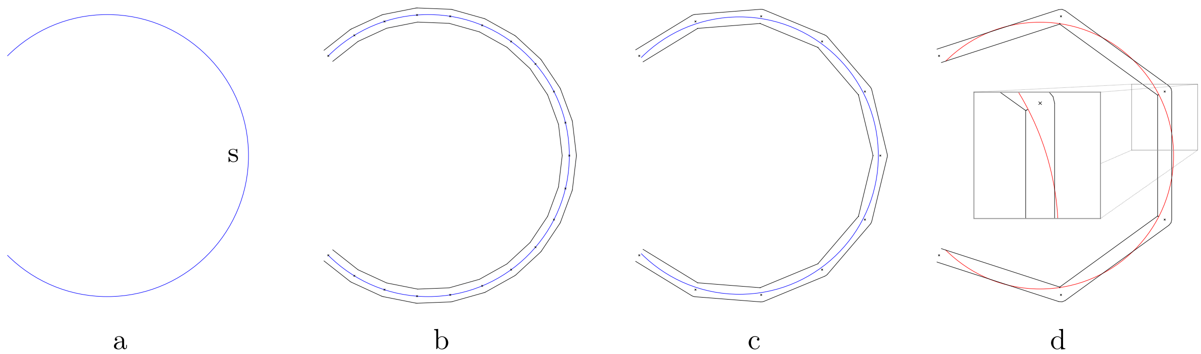


Figure 7.1.7: a: Circular arc s with radius $r(s) = 1.0$ as reference geometry
b: Tolerance channel based on a sampling sequence of s with 21 points and $\varepsilon = 0.05$
c: Tolerance channel based on a sampling sequence of s with 11 points and $\varepsilon = 0.05$
d: Tolerance channel based on a sampling sequence of s with 6 points and $\varepsilon = 0.05$. It is not possible to construct a single arc passing through the entire tolerance channel.

Figure 7.1.7.a shows a circular arc with radius $r(s) = 1.0$ that corresponds to a road turn. This arc is sampled with different numbers of points, which corresponds to different point densities (relation between the number of points and the track length) at the extraction process of raw measurement points as described in Section 5.4. Based on these point sequences, tolerance channels with a fixed tolerance of $\varepsilon = 0.05$ are built. It is shown that the existence of a singular arc passing through the tolerance channel depends on the point density. This example illustrates that the point density should be as high as possible at the measurement extraction process for map building.

ε	n_{SMAP}	n_l	$\frac{Byte}{km}$	$comp$	n_{MLP}	$\frac{n_{MLP}}{n_{SMAP}}$	$RMSE$	l_{avg}	l_{max}
0.05	237	1	641.4	91.12	666	2.81	0.0302	25.10	86.27
0.10	161	1	437.0	133.74	462	2.86	0.0636	36.95	139.56
0.15	139	1	377.8	154.68	373	2.68	0.0982	42.79	137.78
0.20	127	1	345.6	169.12	324	2.55	0.134	46.84	136.03
0.25	115	4	313.3	186.54	287	2.49	0.171	51.73	134.05
0.30	109	4	297.1	196.67	261	2.39	0.205	54.57	139.86
0.40	98	4	267.6	218.42	226	2.30	0.273	60.70	151.15
0.50	90	4	246.0	237.51	201	2.23	0.345	66.10	183.42
0.60	87	4	238.0	245.57	184	2.11	0.422	68.37	205.58
0.70	85	5	232.6	251.24	169	1.98	0.497	69.98	200.98
0.80	81	7	221.8	263.43	158	1.95	0.577	73.44	206.68
0.90	79	7	216.5	269.97	147	1.86	0.637	75.30	213.86
1.00	79	9	216.5	269.97	138	1.74	0.723	75.30	183.58

Table 7.1.3: Approximation results for the mountain road track. The table summarizes the approximation tolerance ε, the number of SMAP segments n_{SMAP} including n_l line segments, the data volume per kilometer for the SMAP, the compression factor *comp*, the number of MLP segments n_{MLP} and the ratio between the segment numbers of the SMAP and the MLP result. Furthermore, the root-mean-square fitting error $RMSE$ of the input points, the average and maximal length of the SMAP segments are given.

7.1.4 Digital map in Ko-PER

This section summarizes some results for the digital map generated in the Ko-PER project. In particular, it is focused on the rural part of the map in contrast to the urban intersections, which are treated within the project as well. The rural digital map covers some road sections located in the north of Munich. Figure 7.1.8 shows five parts of the considered terrain, which offers a relatively high diversity concerning the natural environment (wood and grassland), the elevation profile (cambers and depressions) and, in particular, concerning the availability of road elements (road markings and landmarks).

In contrast to the approximation experiments in Section 7.1.2 and 7.1.3, where the input data consists of a point sequence, the digital map in Ko-PER has been created using the whole processing chain described in Section 5.4 based on the data acquisition with the experimental vehicle and the sensor configuration listed in the Appendix. Regarding the curve approximations, a tolerance value of $\varepsilon = 0.1\,\text{m}$ was used for the arc spline fit.

Table 7.1.4 and 7.1.5 show some statistics on the resulting map elements. Twelve lanes are used to cover the five road sections in both driving directions, including some gaps due to road intersections or junctions. The high number of road markings is justified by the fact that each individual dashed road marking in the center of the road is modeled separately. The average length of road markings results from considering both dashed and continuous road markings on the road edge. As indicated in Section 5.4.3.2, individual dashed road markings are represented as line segments, which explains the short average length of 3.25 m.

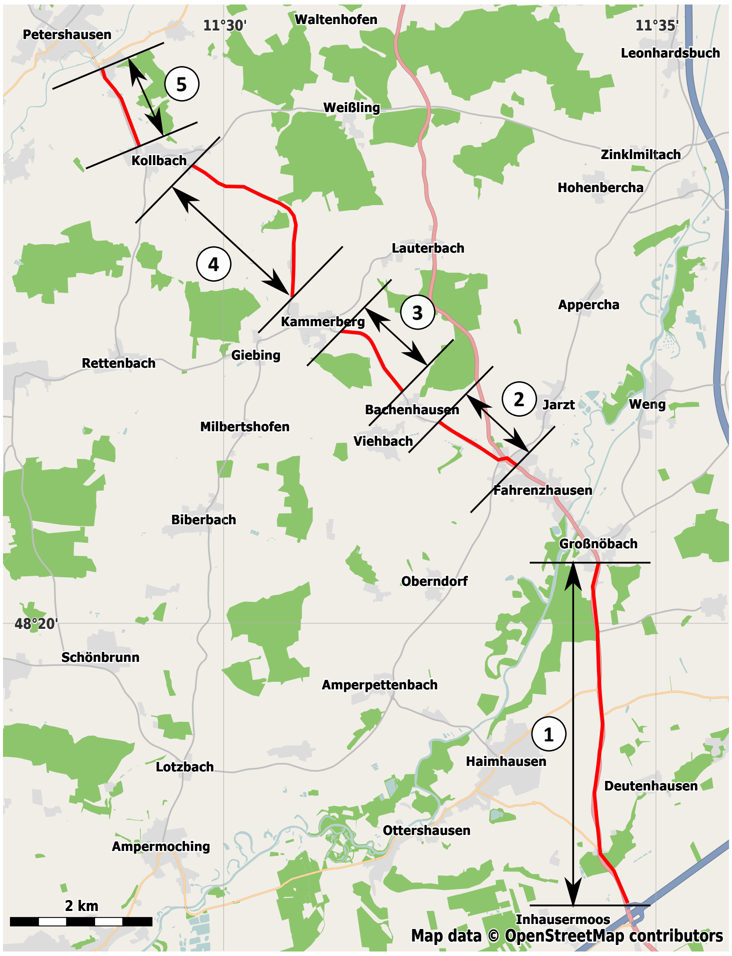


Figure 7.1.8: The rural digital map in Ko-PER with individual road sections.

	quantity	l_{total}	l_{avg}	l_{max}
Lanes	12	24436.85	2036.40	5136.36
Road markings	1331	16496.80	12.39	2244.27

Table 7.1.4: Number and lengths (total, average and maximal lengths in meter) statistics of smooth arc splines modeling individual lanes and road markings.

	Line segments			Line segments		
	quantity	l_{avg}	l_{max}	quantity	l_{avg}	l_{max}
Lanes	0	-	-	327	74.73	316.14
Road markings	1313	3.25	45.01	185	66.07	315.20

Table 7.1.5: Number and lengths (average and maximal lengths in meter) statistics of spline segments modeling individual lanes and road markings.

Furthermore, 1331 landmarks, partially including their semantic classification, were captured. This comprises traffic signs, reflections posts and trees along the way.
The different road sections 1 to 5 differ in the availability of road markings on the road edge. While in section 1, markings are present in both the middle of the road and the road edge, in section 2 to 4 only dashed road markings are available, separating two lanes. Additionally, section 5 has some continuous road markings on the lane in north direction. This diversity allows evaluating the self-localization approach for different configurations of map contents.

7.1.4.1 Evaluation of the map accuracy

In order to evaluate the accuracy of the map, a sequence high-precision reference points have been captured using the stationary geodetic RTK-GPS measurement system specified in the Appendix A.6. The reference points have an average global accuracy of 0.017 m according to the internal quality measure of the geodetic device. A total of 38 reference points have been captured on road markings and 41 reference points for landmarks (reflection posts) on both sides of the road and spread over the road sections 1, 3 and 5 (cf. Figure 7.1.8). The reference points are associated with the best approximating points on the nearest map elements as depicted in Figure 7.1.9.

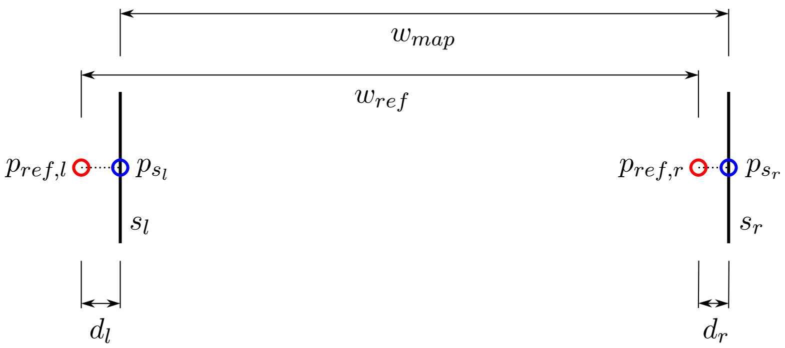


Figure 7.1.9: Reference points $p_{ref,l}$ and $p_{ref,r}$ with best approximating point p_{s_l} and p_{s_r} on left and right road markings s_l and s_r. The distances d_l and d_r are integrated in the root-mean-square error of the global map accuracy. The reference road width w_{ref} is compared to the road width in the map w_{map} for the evaluation of the relative map error.

Regarding the road markings, the root-mean-square error (RMSE) of the Euclidean distance between all pairs of reference points and map correspondences was measured to 0.23 m. Some further investigations showed that there was systematic error of the road markings in road section 1. By computing a translation that minimizes the RMSE in a least squares sense, the road markings in that section were corrected, finally leading to a RMSE of 0.10 m.
Likewise, the accuracy of the landmarks has been evaluated, resulting in a RMSE of 0.32 m. Apart from the inaccuracy of the geodetic reference system, the computed RMSE errors of the map elements reflect a global absolute map error. However, this high accuracy can only be stated for the map elements next to the captured reference measurements.
There are several potential reasons for the remaining global inaccuracy of the map: During

the extraction process of raw measurement points for the map generation based on the lane recognition system described in Section 4.2, reconstruction errors may occur due to rotational movements (pitching, rolling) of the vehicle relative to the road (cf. Section 3.1.3). This problem could be handled using accurate ego-motion techniques as discussed in Section 3.2.1.

Furthermore, the accuracy of the RTK-GPS, which is used for the map building, directly influences the resulting map precision. Figure 7.1.10 visualizes the global accuracy of the RTK-GPS unit on a part of road section 1. It can be seen that the accuracy is below 0.1 m for the most part. The increase of the inaccuracy shown in the zooms is probably related to losses of the differential GPS correction signal.

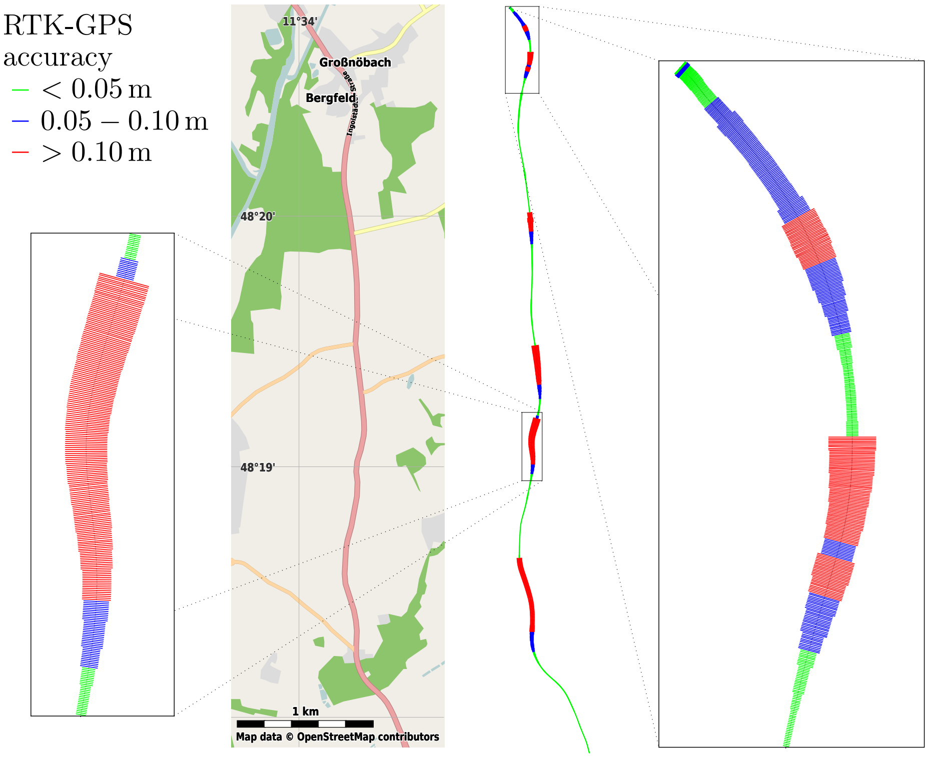


Figure 7.1.10: Visualization of the RTK-GPS accuracy. The length of the colored line segments perpendicular to the road course corresponds to the accuracy value at that point.

The relative accuracy of the map has been investigated by comparing the road width of the map w_{map} with the distance of two corresponding reference measurements w_{ref} on opposite road sides (cf. Figure 7.1.9). The road width of the map elements is determined by the distance of the best approximating points of the reference measurements. As a result, an RMSE of the road width differences $\|w_{ref} - w_{map}\|$ regarding all pairs of reference measurements has been determined to 0.16 m.

The remaining relative error of the map elements can be explained by two main effects: The arc spline approximation is controlled by a tolerance channel with $\varepsilon = 0.1$ m, which corresponds to an error with respect to the maximum norm. Since the individual road markings on both sides of the road are approximated independently, a maximum deviation of the resulting road width w_{map} is in the magnitude of 0.2 m.

Furthermore, it should be noted that the outer road markings for opposite driving directions in a map section are generated based on the processing of different sensor datasets. As discussed in Section 5.4.2, a global postprocessing of the raw measurement points could enhance the relative accuracy of the map significantly.
The average length of 287 dashed road markings was measured to 3.96 m, where the line segment length is 4.0 m according to the road construction regulation in [BFV 93].

7.2 Self-localization results

7.2.1 Implementation notes

Before scoping the self-localization results, some notes on the reference implementation are given.

- Within the reference implementation of the particle filter, the vehicle state space (cf. Section 3.2) is divided into the pose parameters x, y, ψ and the dynamics parameters v, c, β. While the pose parameters form the state space of the particle filter, the individual dynamics parameters are equal for all particles. This is motivated by the fact that the dynamics parameters are directly determined by the intrinsic vehicle measurements (cf. Section 6.5.3) but they are not influenced by any other observation model. All of the described dynamic and observation model can be applied analogously. The resulting reduction of the particle filter state space reduces the number of particles needed for an adequate approximation of the state probability distribution, which in turn reduces the computational complexity significantly.
- The yaw angle ψ is represented internally by its sine and cosine value, which simplifies calculations of rotations in terms of the computational complexity. For instance, this is the case in the motion model of the vehicle in (3.2.8).
- In order to reduce the search space for the calculation of best approximating points on road markings, the caching strategy described in Section 6.4 is modified in the following way: The set of road markings M_r within a region of interest $r \subset \mathbb{R}^2$ does not contain entire arc splines representing individual road markings as indicated in (6.4.11) but M_r only reflects the subset of spline segments having non-empty intersection with r.
- Due to several software optimizations like the examples mentioned above, the overall localization approach is fully realtime capable using the hardware available in the experimental vehicle as specified in the Appendix A.1. In numbers, this means that, based on a system cycle time of 0.08 s (corresponding to one data acquisition of the synchronized camera and laser scanner with a frequency of 12.5 Hz), the processing time of the localization strategy is below 0.01 s. Thereof, the image processing methods (lane recognition) represent about 12 % of the time, the access to a map section is below 1 % and applying the observation models of the particle filter including the map association is about 60 % of the time (42 % for the road markings, 17 % for landmarks and 1 % for GPS and the vehicle dynamics). The remaining 27 % of the time is spent for preprocessing and filtering of sensor data as well as the organization and resampling of the particle filter.

7.2.2 Evaluation method

The subsequent accuracy evaluations of the presented self-localization approach is based on the RTK-GPS reference measurements, which are available at each frame. Since the frequency of RTK-GPS measurements (100 Hz) is higher than the frequency of the video camera and the laser scanner (both 12.5 Hz due to synchronization), a linear interpolation between the RTK-GPS measurements is applied using the corresponding sensor time stamps in order to associate an estimated vehicle pose with the reference data at the same point in time.

Let $p_k^{ref} = (x_k^{ref}, y_k^{ref})^T \in \mathbb{R}^2$ and $\psi_k^{ref} \in [0, 2\pi]$ be the reference position and yaw angle of the vehicle at time t_k in the X, Y-plane of the navigation frame. Likewise, let $p_k = (x_k, y_k)^T \in \mathbb{R}^2$ and $\psi_k \in [0, 2\pi]$ denote the estimated vehicle pose parameters using the self-localization approach presented in the previous chapter. Some essential quantities for the evaluation are defined subsequently and depicted in Figure 7.2.11.

The vectors

$$d_{k,lon}^{ref} := \begin{pmatrix} \sin(\psi_k^{ref}) \\ \cos(\psi_k^{ref}) \end{pmatrix} \in \mathbb{R}^2 \tag{7.2.4}$$

and

$$d_{k,lat}^{ref} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} d_{k,lon}^{ref} \in \mathbb{R}^2 \tag{7.2.5}$$

represent the longitudinal and lateral[2] reference orientation of the vehicle according to the longitudinal and lateral vehicle axes. The term

$$\Delta_k := p_k - p_k^{ref} \in \mathbb{R}^2 \tag{7.2.6}$$

denotes the difference vector between the localization estimation and the reference position.

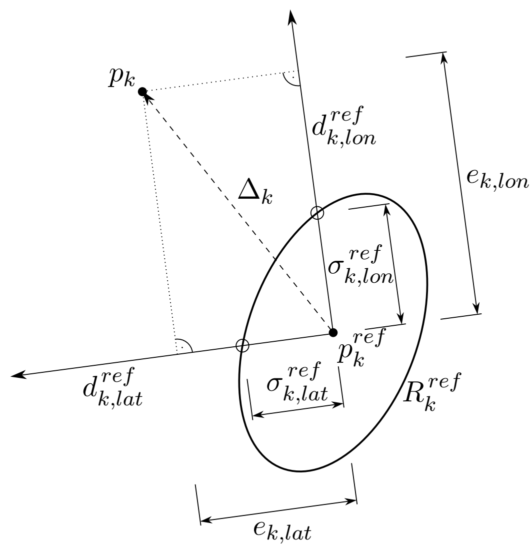


Figure 7.2.11: Quantities of the evaluation method.

[2]The terms *longitudinal* and *lateral* refer to the axes of the vehicle and should not be confounded with the geographic longitude and latitude in Section 3.1.1.

The (signed) lateral and longitudinal localization error $e_{k,lon}$ and $e_{k,lat}$ as well as the orientation error $e_{k,yaw}$ at time t_k are defined as follows:

$$e_{k,lon} := \left\langle d_{k,lon}^{ref} \middle| \Delta_k \right\rangle, \tag{7.2.7}$$

$$e_{k,lat} := \left\langle d_{k,lat}^{ref} \middle| \Delta_k \right\rangle \quad \text{and} \tag{7.2.8}$$

$$e_{k,yaw} := \psi_k - \psi_k^{ref}. \tag{7.2.9}$$

Considering a whole data sequence of $n \in \mathbb{N}$ observation frames, the (unsigned) absolute errors are defined by

$$\bar{e}_{lon} := \frac{1}{n} \sum_{k=1}^{n} \left| \left\langle d_{k,lon}^{ref} \middle| \Delta_k \right\rangle \right|, \tag{7.2.10}$$

$$\bar{e}_{lat} := \frac{1}{n} \sum_{k=1}^{n} \left| \left\langle d_{k,lat}^{ref} \middle| \Delta_k \right\rangle \right| \quad \text{and} \tag{7.2.11}$$

$$\bar{e}_{yaw} := \frac{1}{n} \sum_{k=1}^{n} \left| \psi_k - \psi_k^{ref} \right|. \tag{7.2.12}$$

Since all error values defined above directly depend on the accuracy of the RTK-GPS reference system, its uncertainty must be considered as well.
Let $R \in \mathbb{R}^{2\times 2}$ be the covariance matrix of a normally distributed, two-dimensional random variable. Then, the ellipse defined by

$$E := \left\{ q \in \mathbb{R}^2 \,\middle|\, q^T R^{-1} q = 1 \right\} \subset \mathbb{R}^2 \tag{7.2.13}$$

is commonly called *uncertainty ellipse*. The lengths of its half-axes correspond to the square roots of the eigenvalues of R. In order to express an uncertainty value regarding R in a given normalized direction $v \in \mathbb{S}_1(0)$, we consider the Euclidean distance d_v of the intersection between E and a ray from the origin through v. The latter means solving the equation

$$(d_v \cdot v)^T R^{-1} (d_v \cdot v) = 1 \tag{7.2.14}$$

for d_v resulting in

$$d_v = \frac{1}{\sqrt{v^T R^{-1} v}}. \tag{7.2.15}$$

It is assumed that the measurements of the RTK-GPS system are normally distributed with measurement noise covariance matrix R_k^{ref} concerning the position in X,Y-plane of the navigation frame. Using the direction vectors $d_{k,lon}^{ref}$ and $d_{k,lat}^{ref}$ ((7.2.4) and (7.2.5)), the longitudinal and lateral accuracy of the reference system can be expressed by the standard deviations

$$\sigma_{k,lon}^{ref} = \frac{1}{\sqrt{(d_{k,lon}^{ref})^T (R_k^{ref})^{-1} (d_{k,lon}^{ref})}} \quad \text{and} \quad \sigma_{k,lat}^{ref} = \frac{1}{\sqrt{(d_{k,lat}^{ref})^T (R_k^{ref})^{-1} (d_{k,lat}^{ref})}}. \tag{7.2.16}$$

7.2.3 Results of the self-localization evaluation

In the following, several scenarios are evaluated regarding the self-localization results. Therefore, the quantities defined in the previous section are plotted over time. One unit on the time axis corresponds to one internal system cycle of 0.08 s. The plots of the localization accuracies show the errors (7.2.7), (7.2.8) and (7.2.9) over time, including the corresponding RTK-GPS uncertainty (7.2.16). Furthermore, the average absolute errors (7.2.10), (7.2.11) and (7.2.12) as well as their standard deviations are indicated in the description under the plots.
It should be noted that all subsequent error values refer to the global accuracy of the self-localization approach and not to a relative error regarding the digital map.

Scenario	Section	Direction	Comment
1	1	north	Road markings on all road sides
2	1	south	Opposite direction of scenario 1
3	2	north	Poor map content with dashed markings in the road mid
4	3	north	Poor map content with cambers on the road
5	-	south	Challenging scenarios
6	-	-	Investigations on a longitudinal localization error
7	2	north	Localization without laser scanner landmarks
8	1	north	Importance of precise sensor timestamps

Table 7.2.6: Scenario overview for the self-localization evaluation. The section number refers to one of the five road sections in Figure 7.1.8.

For each scenario, data sets of several passages with the experimental vehicle are available, such that a statistically relevant evaluation of the localization results can be realized.

7.2.3.1 Scenario 1

In this scenario, road markings are available on both sides as well as in the middle of the road. The average absolute errors in lateral and longitudinal direction are in a range below one meter. The orientation error is significantly below 1°. In general, the main causes for remaining localization errors are

- global inaccuracies of digital map elements,
- outliers in the reconstructed measurement points and landmarks due to the environment perception methods,
- association errors between the perceived objects and map elements.

In the highlighted areas of Figure 7.2.13a, 7.2.13b and 7.2.13c, an overtaking maneuver was driven. The relative stability of the error plots in these regions show the robustness of the localization approach regarding lane changes.

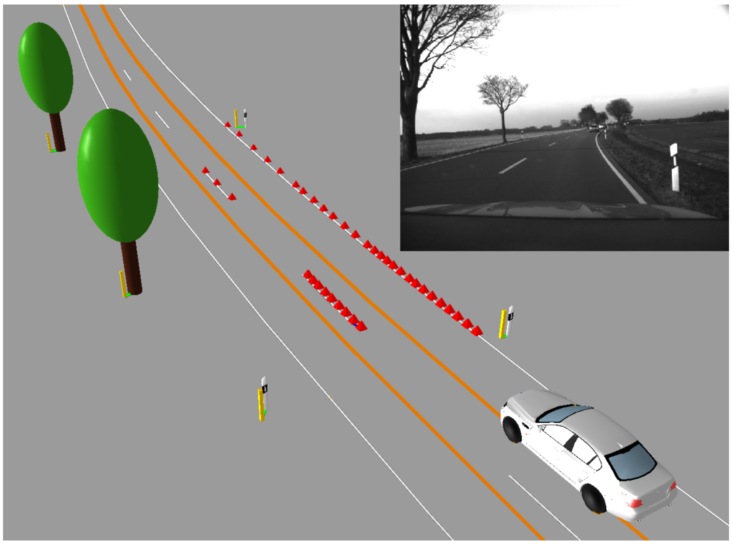
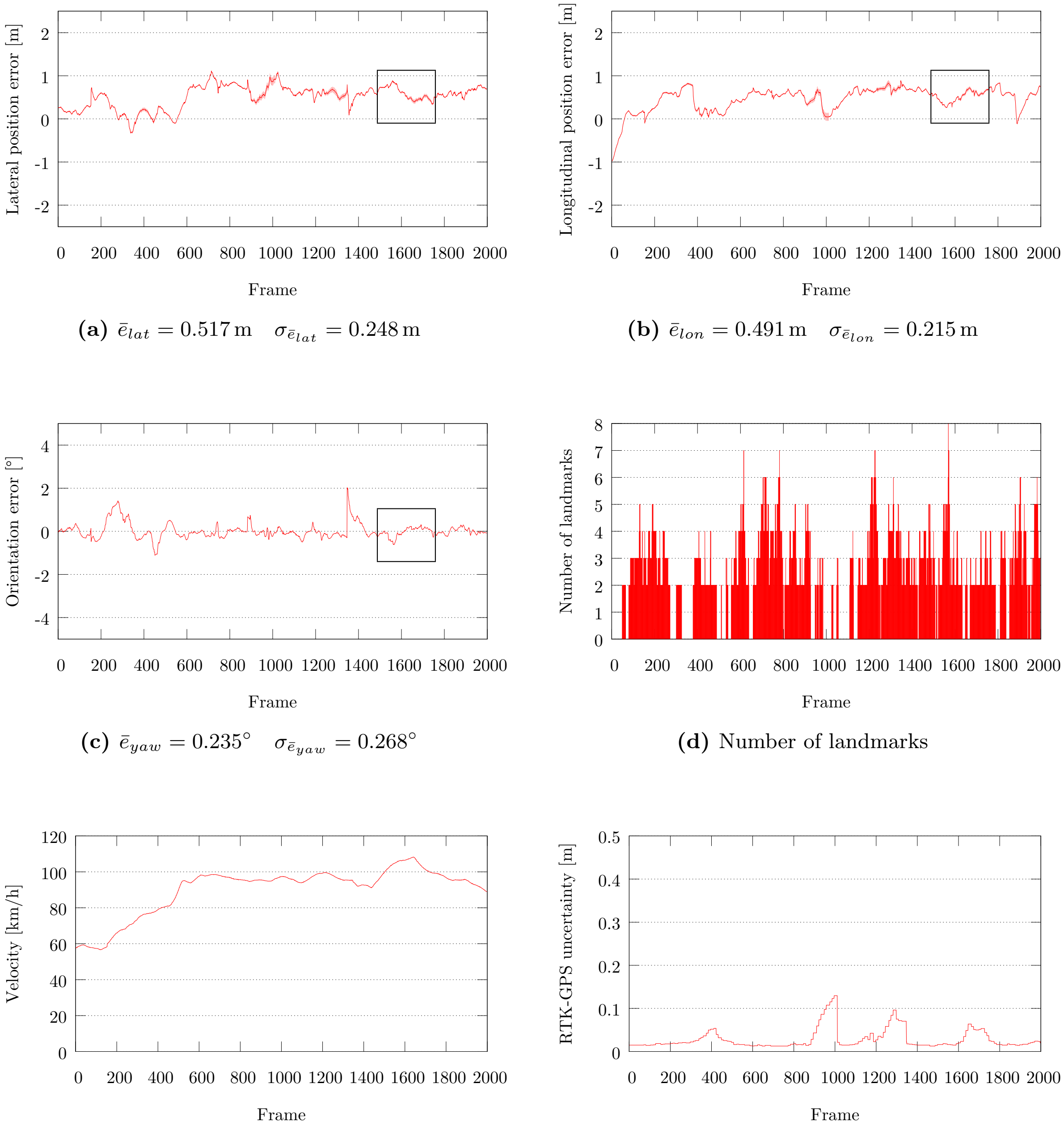


Figure 7.2.12: Position and orientation of the vehicle corresponds to the estimated pose of the self-localization strategy based on the association of perceived objects with elements of the digital map. Individual lanes are marked in orange. Extracted measurement points of the lane recognition (red pyramids) are associated with road markings. Landmark hypotheses of the laser scanner (yellow) can be related to corresponding landmarks in the map (trees, reflection posts). The camera image is displayed on the right top.

For both directions (longitudinal and lateral), one can see a systematic error. Concerning the lateral error, the sensor data processing of the camera and the laser scanner landmarks as well as the association with map elements works well at this section (cf. Figure 7.2.12). Furthermore, the uncertainty of the RTK-GPS reference is relatively low (mostly below 0.1 m), which justifies putting confidence in the stated error values. In all comparable data sets for this scenario, the same systematic error characteristics are visible, like the increase of the lateral error in the highlighted areas in Figure 7.2.14, which shows a comparison of several passages of scenario 1.



(a) $\bar{e}_{lat} = 0.517\,\text{m} \quad \sigma_{\bar{e}_{lat}} = 0.248\,\text{m}$

(b) $\bar{e}_{lon} = 0.491\,\text{m} \quad \sigma_{\bar{e}_{lon}} = 0.215\,\text{m}$

(c) $\bar{e}_{yaw} = 0.235° \quad \sigma_{\bar{e}_{yaw}} = 0.268°$

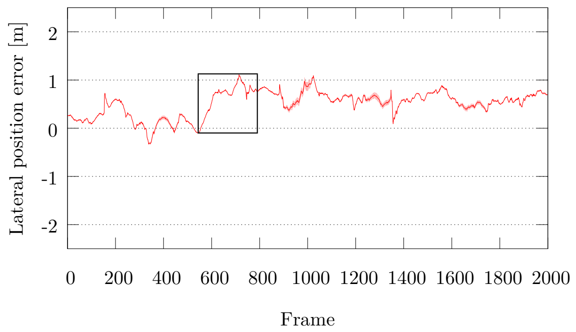
(d) Number of landmarks

(e) velocity

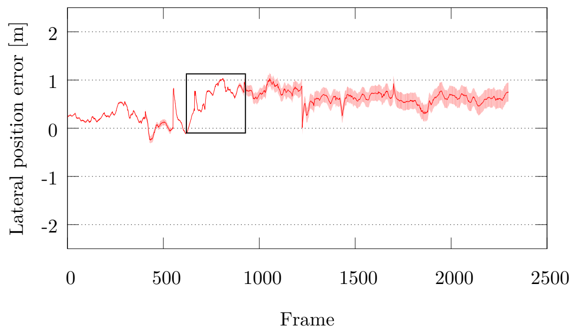
(f) RTK-GPS uncertainty

Figure 7.2.13: Localization results of scenario 1

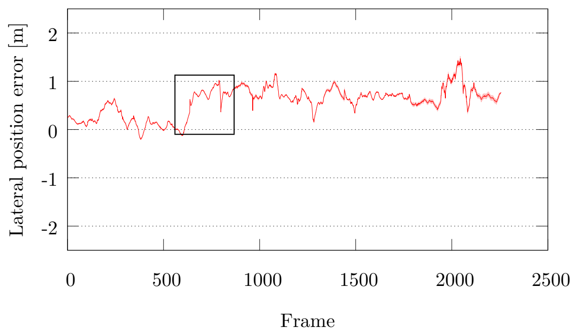
This leads to the assumption that the lateral localization error can be traced back to a global inaccuracy of the road markings in this map section. In fact, no map reference measurement points (cf. Section 7.1.4.1) were available in this part, so that the global map accuracy could not be evaluated in that region. The systematic longitudinal error is investigated in Section 7.2.3.6.



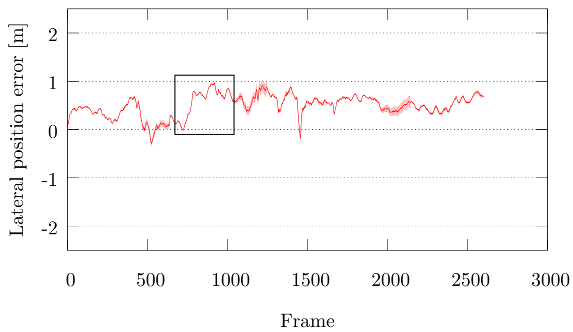
(a) $\bar{e}_{lat} = 0.517\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.248\,\mathrm{m}$



(b) $\bar{e}_{lat} = 0.531\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.243\,\mathrm{m}$



(c) $\bar{e}_{lat} = 0.578\,\mathrm{m} \quad \sigma_{\bar{e}_latyaw} = 0.293\,\mathrm{m}$

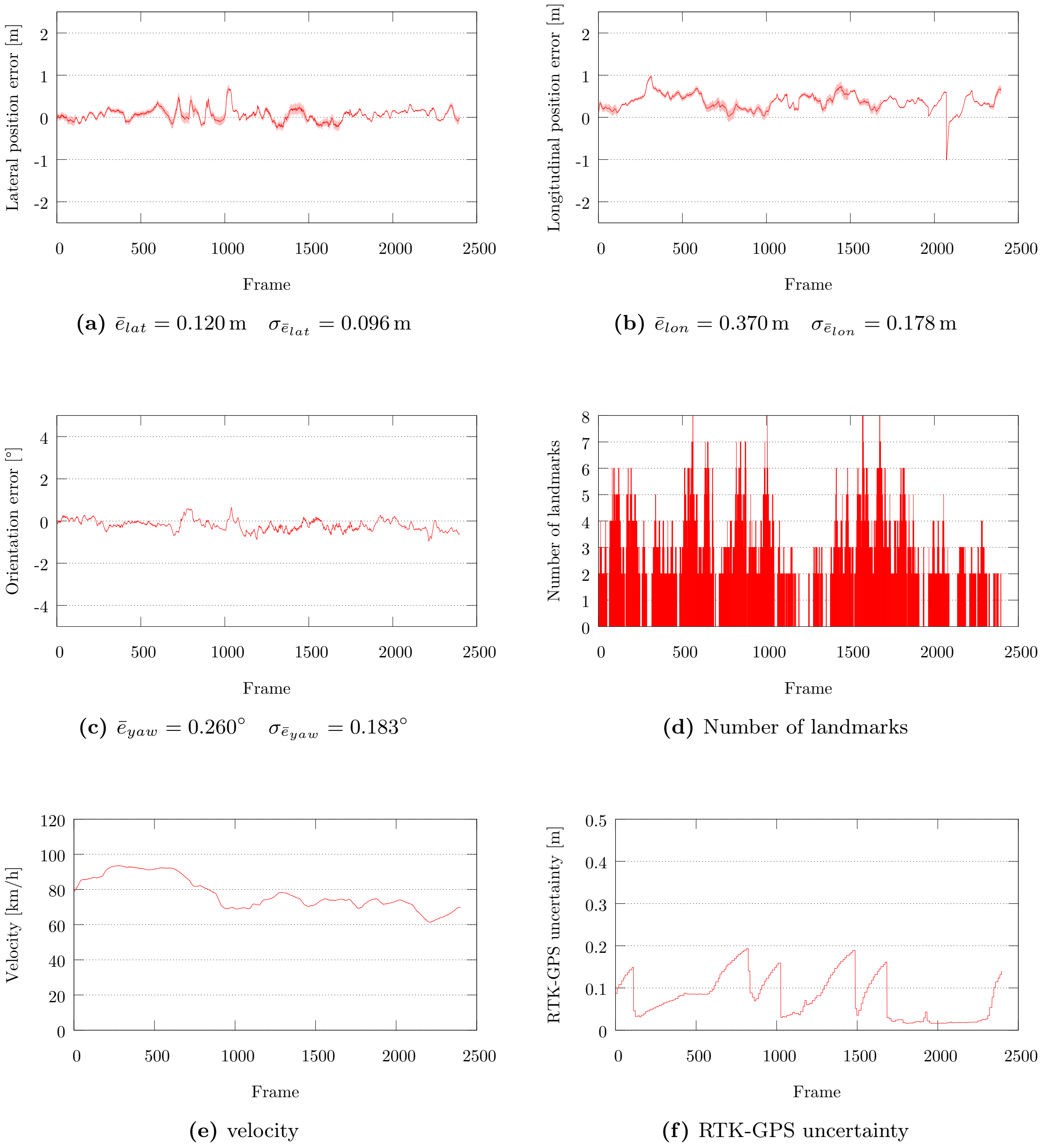


(d) $\bar{e}_{lat} = 0.501\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.211\,\mathrm{m}$

Figure 7.2.14: Comparison of several passages of scenario 1 focusing on the lateral localization error. The boxes highlight a section of the passage from where the lateral error increases in all considered data sets. This effect is probably due to a global inaccuracy of the road markings in this map section.

7.2.3.2 Scenario 2

This scenario represents the passage of road section 1 in south direction. In this case, the localization results are even superior to the north direction and they show rarely systematical lateral errors. This can be explained by the fact that the individual lanes and road markings for the north and south direction are mapped based on different data sets, which were captured by passages of the experimental vehicle in the corresponding driving directions.



(a) $\bar{e}_{lat} = 0.120\,\mathrm{m}$ $\sigma_{\bar{e}_{lat}} = 0.096\,\mathrm{m}$

(b) $\bar{e}_{lon} = 0.370\,\mathrm{m}$ $\sigma_{\bar{e}_{lon}} = 0.178\,\mathrm{m}$

(c) $\bar{e}_{yaw} = 0.260°$ $\sigma_{\bar{e}_{yaw}} = 0.183°$

(d) Number of landmarks

(e) velocity

(f) RTK-GPS uncertainty

Figure 7.2.15: Localization results of scenario 2

7.2.3.3 Scenario 3

In road section 2 (cf. Figure 7.1.8), dashed road markings in the road mid are available but there are no road markings on the sideways road edges as depicted in Figure 7.2.16. Hence, measurement points extracted by the lane recognition can only be associated with the road markings in the middle of the road. Nevertheless, the resulting localization errors remain relatively small, as detailed in the plots of Figure 7.2.17.

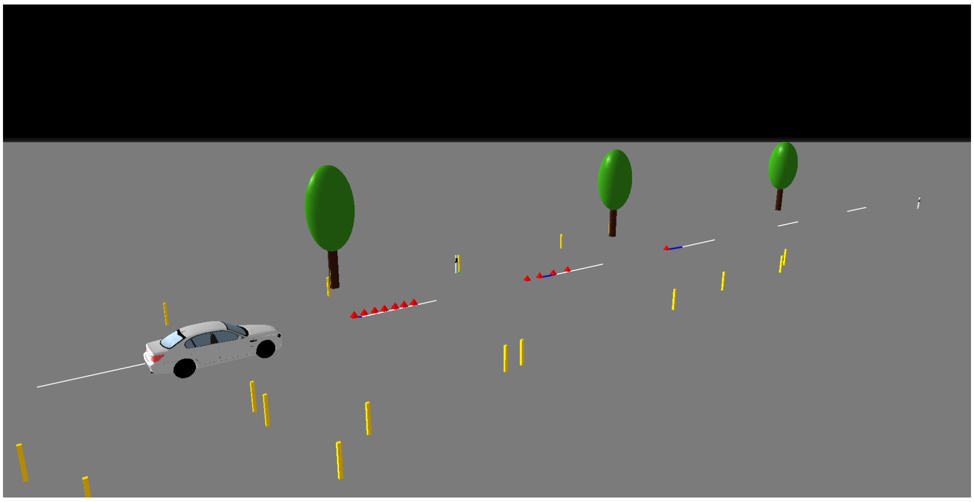
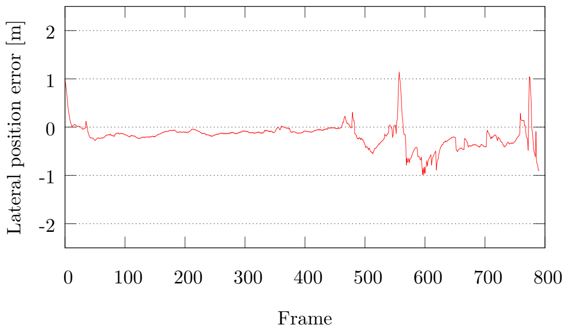


Figure 7.2.16: Association of perceived objects with elements of the digital map. Measurement points of the lane recognition (red pyramid) are associated with the dashed road markings in the middle of the road. The landmark hypotheses of the laser scanner (yellow) are associated with landmarks (trees, reflection posts) in the map if applicable.

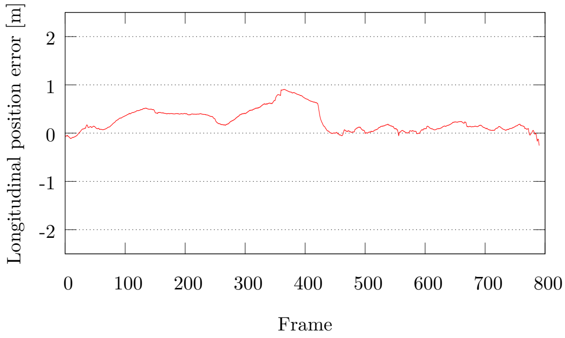
In this scenario, it is remarkable that there are no landmark hypotheses between frame 100 and 400 within the data set, which is possibly due to a sensor failure. Thus, the map-based localization strategy is purely video-based within this section. The longitudinal localization error increases, since no associations of landmark hypotheses and corresponding map elements can be realized. However, the lateral localization error remains small (in average about 0.1 m for this section).

The inaccuracies starting from frame 450 can be attributed to two reasons: At first, the lane recognition extracts partly erroneous measurement points due to reflections on the road surface. Then, these measurement points are associated misleadingly with the only available dashed road markings in the middle of the road, resulting in a localization error regarding the lateral position and the orientation.

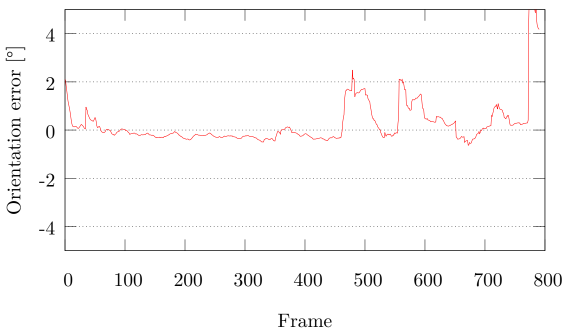
The uncertainty of the RTK-GPS reference is relatively low all the time (about 0.02 m), which allows authoritative statements on the localization errors. The overall localization results are significantly below 1 m regarding the position errors and below 1° regarding the orientation error.



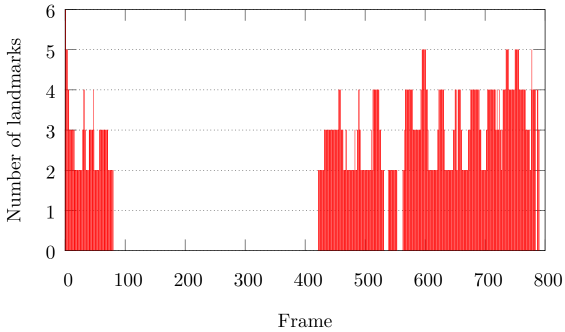
(a) $\bar{e}_{lat} = 0.223\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.201\,\mathrm{m}$



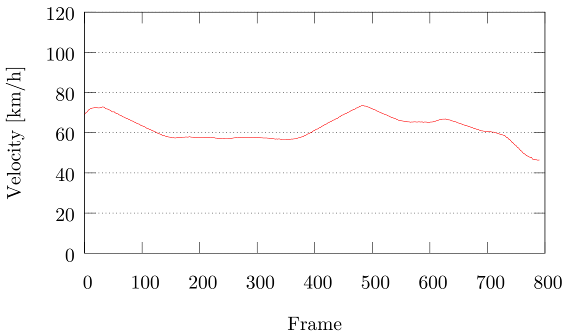
(b) $\bar{e}_{lon} = 0.264\,\mathrm{m} \quad \sigma_{\bar{e}_{lon}} = 0.229\,\mathrm{m}$



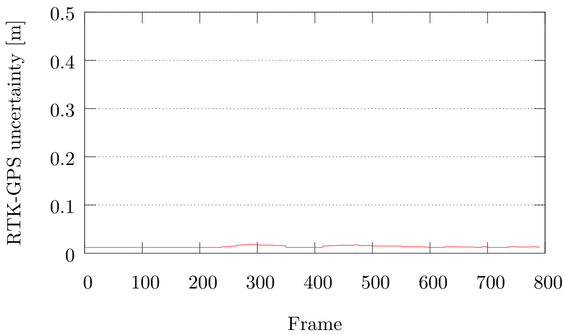
(c) $\bar{e}_{yaw} = 0.533° \quad \sigma_{\bar{e}_{yaw}} = 0.826°$



(d) Number of landmarks



(e) velocity



(f) RTK-GPS uncertainty

Figure 7.2.17: Localization results of scenario 3

7.2.3.4 Scenario 4

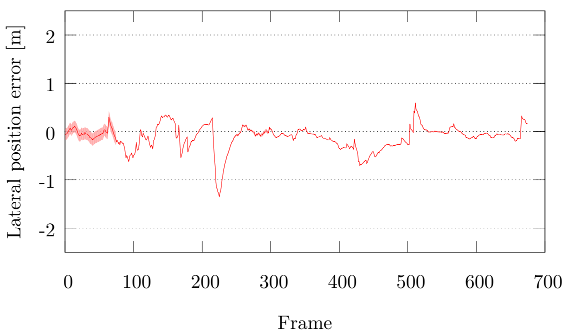
Similar to road section 2 and 4, only dashed markings in the middle of the road are available in road section 3. After the relatively high uncertainty of the RTK-GPS reference ($0.15 - 0.20\,\mathrm{m}$) in the beginning of the sequence, the positional localization results are in a range below $1\,\mathrm{m}$ except for outliers. The orientation error between frame 50 and 140 can be traced back to a camber on the road (cf. Figure 7.2.18g) which is combined with a right turn. Due to the rise, the vehicle's pitch angle relative to the upcoming road section changes and oscillates for a few seconds, resulting in a modified relative view of the camera on the road. As a consequence, the measurement points of the video-based lane recognition are reconstructed erroneously, which, in turn, leads to errors in the estimation of the orientation by associating the measurement points with road markings. This kind of error could potentially be reduced when considering the elevation profile of the lanes (cf. Section 5.3.4.1) at the reconstruction step of measurement points as well as an adequate pitch compensation.

7.2.3.5 Scenario 5, Challenging scenarios

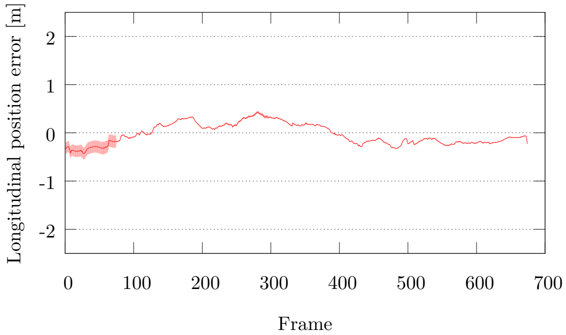
The quality of the localization results depends decisively on the performance of the environment perception methods for the extraction of landmarks and the lane recognition. In some test sequences in winter time, the vehicle moved in the direction of the low sun. The strong insolation can cause total reflections on the wet road surface and glare effects in the camera image as depicted in Figure 7.2.19. These effects worsen the conditions for the image processing methods within the lane recognition.

As shown in Figure 7.2.19, strong glaring leads to overload effects of the camera. The sharp contrast along the solar ray causes the extraction of measurement points in the lane recognition in that region, which, then, are associated with road markings in the digital map. As a result, errors regarding the pose estimation occur in the self-localization. Furthermore, the snow on the road edge builds a higher contrast than the line markings on the road, causing an erroneous extraction of measurements as well.

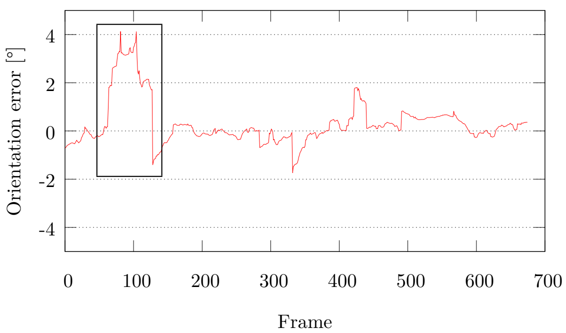
To cope with these problems, the lane recognition could be enhanced with a more sophisticated strategy for the detection of road markings in order to robustly keep track of the lane even under bad conditions. However, extremal weather and environment conditions probably remain challenging for any video-based perception system. Therefore, sensor data fusion concepts, which combine the advantages of several different kinds of sensors, represent a promising way for dealing with more challenging scenarios.



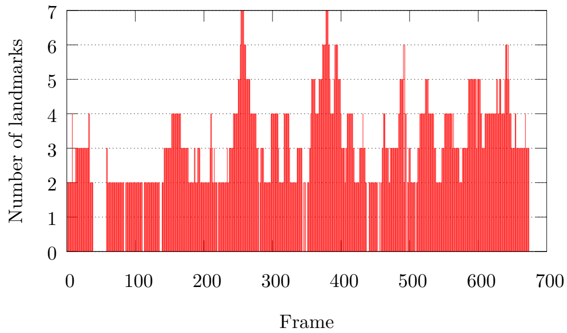
(a) $\bar{e}_{lat} = 0.191\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.202\,\mathrm{m}$



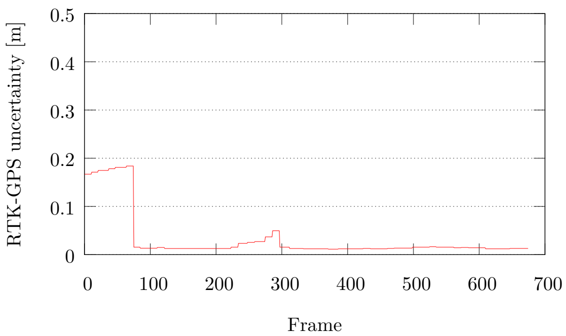
(b) $\bar{e}_{lon} = 0.194\,\mathrm{m} \quad \sigma_{\bar{e}_{lon}} = 0.094\,\mathrm{m}$



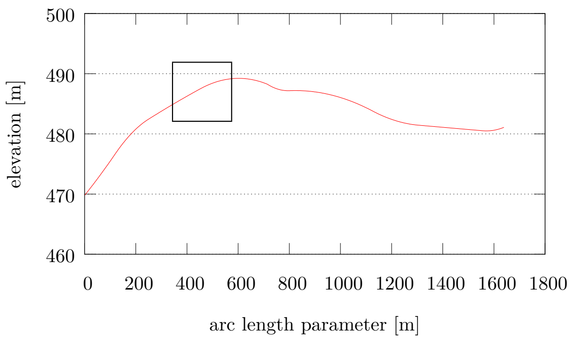
(c) $\bar{e}_{yaw} = 0.583° \quad \sigma_{\bar{e}_{yaw}} = 0.782°$



(d) Number of landmarks



(e) RTK-GPS uncertainty



(f) Elevation profile of the corresponding lane

(g) Camber with right turn

The area of the orientation error marked with a box in Figure 7.2.18c corresponds to a rise on a camber (box in Figure 7.2.18f) which is combined with a right turn. The oscillating pitch movement of the car results in a suboptimal reconstruction of road marking measurement points.

Figure 7.2.18: Localization results of scenario 4

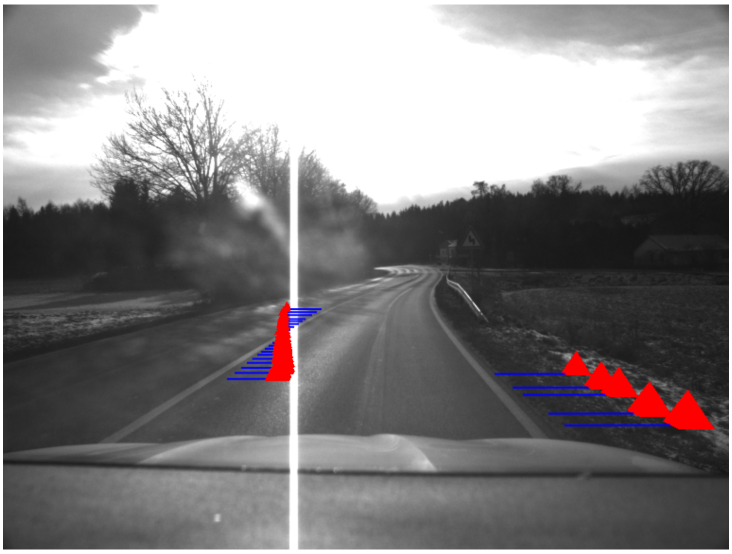


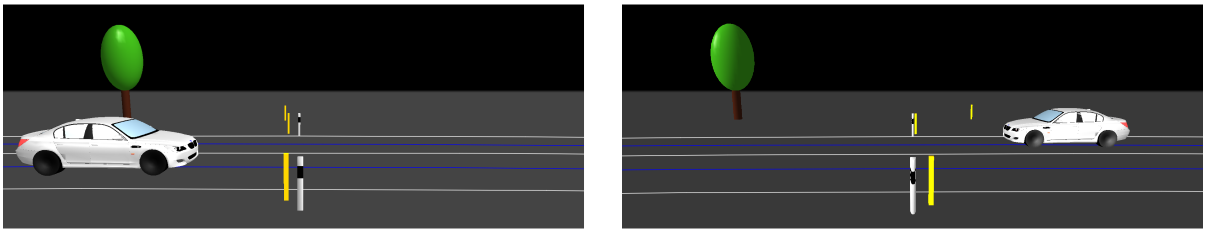
Figure 7.2.19: Strong reflections and glare effects due to the specular road surface and the direct solar radiation. The measurement points (red pyramids) extracted by the lane recognition lead to localization errors when associating with road markings in the digital map.

7.2.3.6 Scenario 6, Longitudinal localization error

In most of the considered data sequences, a systematic longitudinal localization error could be observed in the results. More precisely, the position estimation is located in front of the RTK-GPS reference position in driving direction ($e_{k,lon} > 0$ in (7.2.7)). This effect is observable for both driving directions, to the north and to the south. As a result of the localization strategy, the estimation of the longitudinal position is mainly determined by the association of landmarks rather than by the association of road markings. The latter, in turn, define primarily the estimation of the lateral position and the orientation. Thus, it is supposed that the longitudinal localization error is related to the landmark association.

In order to investigate this effect, the following experiment has been run: In each frame, the positions of the landmarks detected by the laser scanner were compared to the landmarks available in the digital map, where the vehicle pose was determined by the RTK-GPS and not by the localization approach above. In an ideal case, these landmark correspondences should coincide. However, it could be observed that in both cases of driving directions (north and south), the landmark hypothesis of the laser scanner were located closer to the vehicle than the landmarks in the digital map. Figure 7.2.20a and 7.2.20b visualize the same map section including two reflection posts on the road sides. The images show the passage of the vehicle in north and south direction. In both cases, the landmark hypotheses (yellow) are located closer to the vehicle compared to the landmarks in the map. This observation is consistent with the longitudinal localization error which results when associating these landmarks.

The reason for this error could not be identified ultimately but the observed longitudinal difference could be related to an error within the temporal interpretation of the landmark hypotheses. The effect would be explainable if these landmarks are associated with a too early timestamp within the raw sensor data. A longitudinal difference of $0.30 - 0.40\,\mathrm{m}$ corresponds to a times-



(a) south direction **(b)** north direction

Figure 7.2.20: In both driving directions, the landmark hypotheses (yellow) are located closer to the vehicle compared to the landmarks in the map.

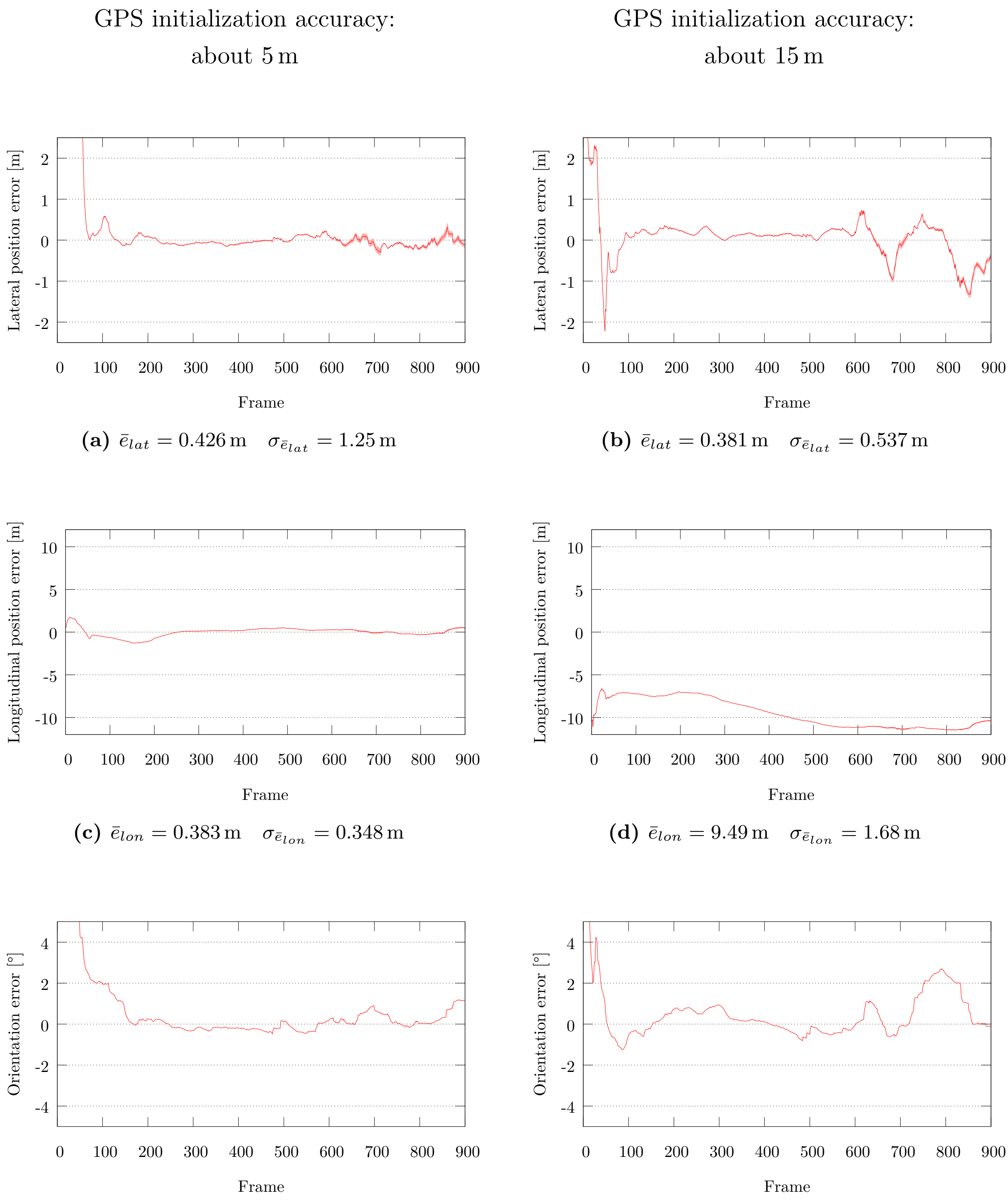
tamp inaccuracy of $0.012 - 0.016\,\mathrm{s}$ when driving $90.0\,\mathrm{km/h}$. However, it should be recalled that a global localization accuracy in the magnitude of half a meter widely fulfills the requirements on the localization quality defined on behalf of this work.

7.2.3.7 Scenario 7, Localization without laser scanner landmarks

In this experiment, the performance of the self-localization approach is investigated regarding the case of ignoring all laser scanner landmark hypotheses. This modification simulates the absence of the laser scanner and results in a map-based localization strategy that is purely based on video, GPS and the vehicular intrinsic measurements.

Figure 7.2.21 summarizes the localization results of this modified approach on the same road section for two different values of the GPS initialization accuracy. One can see, that the quality of the GPS initialization influences the localization results significantly. In both cases (Figure 7.2.21a and 7.2.21b), the lateral localization error decreases after a about 60 frames (4.8 s) which is mainly due to the map-based resampling strategy described in Section 6.6.2 which explicitly uses a priori knowledge on the individual lanes. For the rest of the sequence, the lateral localization error remains relatively small. In the last third of the sequence in Figure 7.2.21b, the lateral position accuracy is disturbed by wrong associations of measurement points and road markings due to the orientation error (cf. Figure 7.2.21f).

The most interesting observation is that the GPS initialization accuracy has a significant influence on the longitudinal localization error. This can be explained by the fact that the longitudinal localization is mainly determined by the laser scanner landmarks which are, in fact, missing in this experiment. It is evident that the observation model for road markings cannot distinguish between subsequent dashed road markings and hence it is not able to correct the initial longitudinal localization error (Figure 7.2.21d) in case of bad GPS initialization within this example. However, the experiment shows that if the positional initialization accuracy is high (in this example about 5 m were enough) then the localization approach performs well even in the absence of the laser scanner.



(a) $\bar{e}_{lat} = 0.426\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 1.25\,\mathrm{m}$

(b) $\bar{e}_{lat} = 0.381\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.537\,\mathrm{m}$

(c) $\bar{e}_{lon} = 0.383\,\mathrm{m} \quad \sigma_{\bar{e}_{lon}} = 0.348\,\mathrm{m}$

(d) $\bar{e}_{lon} = 9.49\,\mathrm{m} \quad \sigma_{\bar{e}_{lon}} = 1.68\,\mathrm{m}$

(e) $\bar{e}_{yaw} = 0.834^\circ \quad \sigma_{\bar{e}_{yaw}} = 1.64^\circ$

(f) $\bar{e}_{yaw} = 0.832^\circ \quad \sigma_{\bar{e}_{yaw}} = 1.23^\circ$

Figure 7.2.21: Comparison of the localization results without laser scanner landmarks for different GPS initialization accuracies. The longitudinal localization error largely depends on the quality of the position initialization. The high standard deviations suffer from the positional and orientational error at the starting phase.

7.2.3.8 Scenario 8, Localization without precise timestamps

In the localization approach described in this thesis, the sensor data is temporally interpreted according the associated timestamps related to the sensor acquisition time (cf. Section 4.1). These sensor timestamps are depicted in Figure 7.2.22 in the rows for the laser scanner, the camera and the RTK-GPS reference. In order to show the importance of the correct temporal interpretation of the sensor data, the following experiment is run:

In a modified localization approach, the sensor timestamps are replaced by standard timestamps defined by the start of the next system cycle, as illustrated in Figure 7.2.22 in the row for the system cycle. The dotted cross lines between the sensor acquisition time and the start time of the next system cycle symbolizes the sensor specific delay caused by the internal processings of raw data within the sensors as well as the transmission of data and handling within the hard- and software frameworks.

The negative effect of these delays can be seen in Figure 7.2.23, where the standard version of the localization (with sensor timestamps) is compared to the modified version (without sensor timestamps). One can see that both the lateral positional error (Figure 7.2.23a and 7.2.23b) and the orientation error (Figure 7.2.23e and 7.2.23f) do not change significantly when ignoring the sensor timestamps. However, the longitudinal error is substantially worse when not using the precise sensor timestamps. Furthermore, the increase of jitter can be traced back to the inability of exactly associating the RTK-GPS reference with the pose estimation for lack of the sensor timestamps.

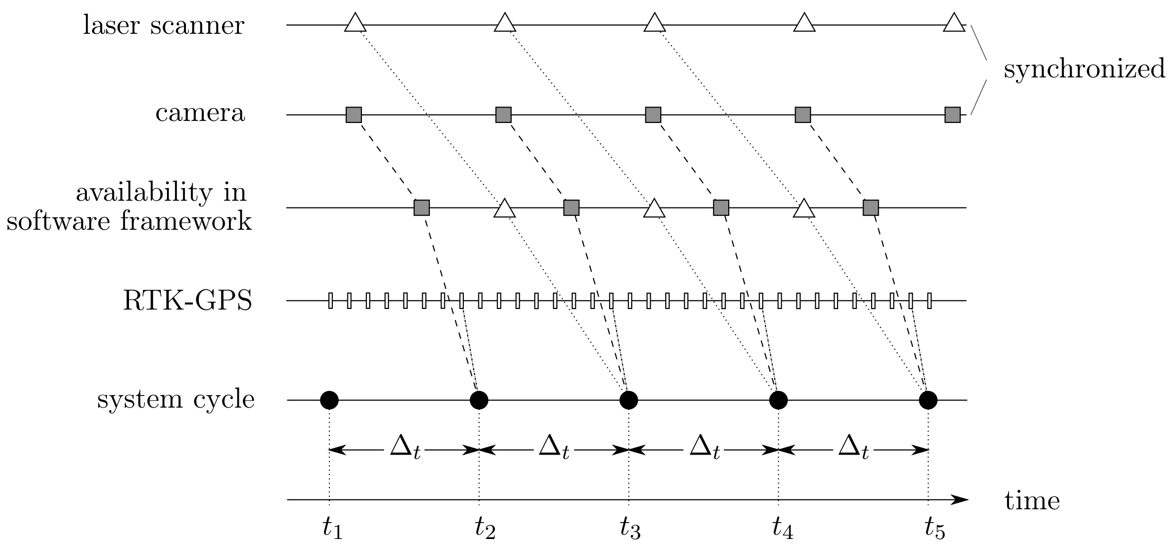
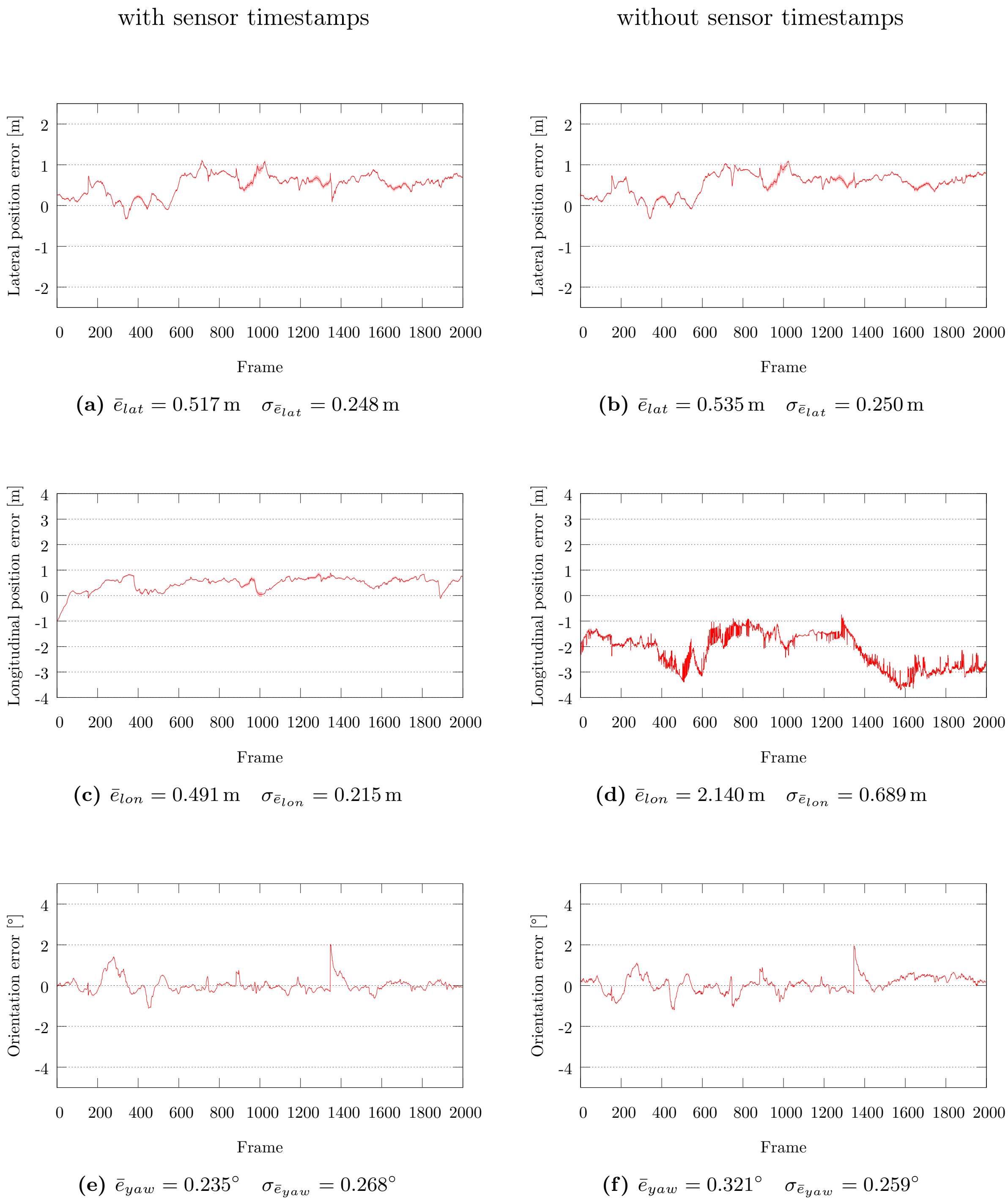


Figure 7.2.22: Illustration of sensor timestamps and delays

The experiment shows that it is worth spending the effort to the precise generation and association of timestamps since the localization results can be improved significantly regarding the accuracy and the robustness.



(a) $\bar{e}_{lat} = 0.517\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.248\,\mathrm{m}$

(b) $\bar{e}_{lat} = 0.535\,\mathrm{m} \quad \sigma_{\bar{e}_{lat}} = 0.250\,\mathrm{m}$

(c) $\bar{e}_{lon} = 0.491\,\mathrm{m} \quad \sigma_{\bar{e}_{lon}} = 0.215\,\mathrm{m}$

(d) $\bar{e}_{lon} = 2.140\,\mathrm{m} \quad \sigma_{\bar{e}_{lon}} = 0.689\,\mathrm{m}$

(e) $\bar{e}_{yaw} = 0.235° \quad \sigma_{\bar{e}_{yaw}} = 0.268°$

(f) $\bar{e}_{yaw} = 0.321° \quad \sigma_{\bar{e}_{yaw}} = 0.259°$

Figure 7.2.23: Comparison of the localization results with and without using the sensor timestamp.

7.3 Résumé

In this thesis, methods and models for a map-based vehicle self-localization approach have been presented. The proposed digital map model, which is based on smooth arc splines, shows several positive characteristics regarding efficient calculations of best approximating points, offset curves, curvature information and the lengths of curve segments. Furthermore, the integration of elevation profiles allows modeling a 3D representation in a compact way. Regarding the map generation, for any given maximal tolerance, the applied curve approximation method generates a smooth arc spline with a minimum number of segments. These properties are most valuable for digital maps since they imply the checkability of accuracy of map elements as well as the minimization of data volume required for storing the map. Also, the advantages are profitable not only for the self-localization observation models defined in this work, but they represent an additional value for further automotive applications.

The suitability of the presented mapping strategy has been demonstrated on both simulated and real data. By means of an extensive evaluation for different kinds of roads and approximation accuracy levels, it has been shown that the approach developed in this thesis generally outperforms the widely-used map modeling with polygons or other arc spline approximations when judged by criteria like efficiency of map calculations, data volume and information content.

Based on the presented map model, a vehicle self-localization approach has been proposed. The basic idea is to associate information from the vehicular environment perception with corresponding elements of the digital map in order to deduce the vehicle's position. The probabilistic self-localization strategy fuses data from a video camera, laser scanner, GPS and intrinsic vehicular measurements in a particle filter framework. It has been shown that a global localization accuracy in both lateral and longitudinal direction significantly below one meter and an orientation accuracy below one degree can be reached even at a speed up to 100 km/h in real-time using the methods presented. Thus, the localization strategy satisfies the accuracy requirements defined by the advanced driver assistance applications. A series of experiments demonstrated the robustness of the approach regarding different levels of details in the map, altering sensor configurations and environment conditions.

It has been shown that the mapping approach and the self-localization strategy presented in this work represent a promising concept for future systems within the field of active traffic safety.

7.4 Possible future work

Since the performance of the environment perception methods influences the quality of the self-localization results significantly, it is recommended to further improve the sensor data processing like the lane recognition. The robustness and reliability of the environment perception system could be enhanced by integrating other sensors or processing methods on behalf of the sensor data fusion.

For future approaches, the accuracy of the digital map can be enhanced by mapping methods based on several passages of road sections. It is assumed that, on behalf of a preprocessing step of the extracted raw measurement points, a SLAM-based optimization is appropriate for filtering outliers and increase the relative accuracy of the measurement points. Due to this preprocessing, applying the presented methods for the subsequent generation of continuous map elements yields digital maps of even higher quality. Since the accuracy of the map determines the precision of the map-based self-localization strategy, it es expected that the above mentioned optimization technique leads to better positioning results as well.

Appendix A

Appendix

A.1 Specification of the experimental vehicle

Figure A.1.1: BMW 528i experimental vehicle

Name	BMW 528i
Type	Sedan
Onboad CPU	Intel Core i7, 2.8 GHz
Memory	4 GB DDR3 RAM

Table A.1.1: Technical specification of the experimental vehicle

A.2 Specification of the IDS uEYE camera

Figure A.2.2: IDS uEYE camera

Name	IDS uEYE UI-6220SE-M-GL camera
Measuring principle	Frame-Shutter
Sensor chip	1/2" CCD
Frequency range	400 − 700 nm
Dynamic range	high dynamic range
Distance range	greater than 2 m
Opening angle	horizontal: 46.7°, vertical: 38.6°
Angular resolution	0.06°
Pixel resolution	768 × 576
Aspect ratio	4 : 3
Measuring frequency	maximal: 52 Hz used: 12.5 Hz due to synchronization with laser scanner
Exposure time	50 μs − 10 min in trigger mode
Supply voltage	12 V
Interface	Ethernet
Dimensions	34.00 × 44.00 × 43.50 mm
Mass	108 g

Table A.2.2: Technical specification of the IDS uEYE camera.

A.3 Specification of the SICK LD-MRS laser scanner

Figure A.3.3: SICK LD-MRS laser scanner

Name	SICK LD-MRS-400001
Measuring principle	Laser scanner with 4 scan layers
Frequency range	905 nm
Distance range	0.3-200 m
Opening angle	horizontal: 100° vertical: 0.8° between two scan layers, 3.2° in total
Distance resolution	4 cm
Angular resolution	0.125°
Measuring frequency	12.5 Hz
Supply voltage	12 V
Interface	Ethernet
Dimensions	85.00 × 128.00 × 93.00 mm
Mass	1 kg

Table A.3.3: Technical specification of the SICK LD-MRS laser scanner.

A.4 Specification of the Novatel OEMV GPS receiver

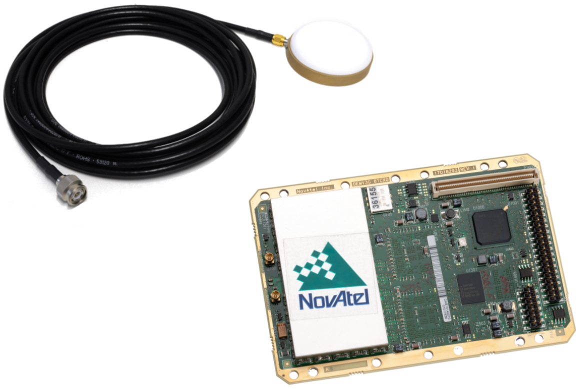


Figure A.4.4: Novatel OEMV GPS receiver

Name	Novatel OEMV Family
Measuring principle	GPS receiver
Position accuracy	1.5 m
Velocity accuracy	0.1 m/s
Code-phase noise	0.20 m
Measuring frequency	10 Hz
Supply voltage	12 V
Interface	Serial interface (RS-232)

Table A.4.4: Technical specification of the Novatel OEMV GPS receiver.

A.5 Specification of the OXTS RT3003 reference positioning system

Figure A.5.5: OXTS RT3003 reference system

Name	OXTS RT3003
Measuring principle	Inertial RTK-GPS measurement system
Antenna	2 × G5Ant-2AMNS1
Differential correction	ASCOS (axionet) RTCM 3.1 [ASCOS 12]
Position accuracy	2 cm
Velocity accuracy	0.05 km/h
Yaw accuracy	0.1°
Roll/Pitch accuracy	0.03°
Slip angle accuracy	0.15° (at 50 km/h)
Measuring frequency	100 Hz
Supply voltage	9 – 18 V
Dimensions	234.00 × 120.00 × 80.00 mm
Mass	2.4 kg

Table A.5.5: Technical specification of the OXTS RT3003 reference positioning system. All accuracy values refer to one standard deviation.

A.6 Specification of the Leica GPS1200 system

Figure A.6.6: Leica GPS1200 geodetic system

Name	Leica GX1230+ GNNS
Measuring principle	Geodetic RTK-GPS measurement system
Differential correction	Leica SmartNet, RTCM v3 or ASCOS (axionet) RTCM 3.1 [ASCOS 12]
Position accuracy	1 cm
Supply voltage	12 V
Dimensions	$212.00 \times 166.00 \times 79.00$ mm
Mass	1.2 kg

Table A.6.6: Technical specification of the Leica GPS1200 system. All accuracy values refer to one standard deviation.

List of Figures

Index

Bibliography

[Ahlers 09] F. Ahlers & D. Westhoff. *Precise Laserscanner-based positioning on intra-urban crossings.* In SAFESPOT Technical presentations to the 16th World Congress on Intelligent Transportation Systems, 2009. 74

[Arulampalam 02] M. Arulampalam, S. Maskell, N. Gordon & T. Clapp. *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking.* IEEE Transactions on Signal Processing, vol. 50, no. 2, pages 174–188, 2002. 18

[ASCOS 12] ASCOS - satellite positioning services. http://www.ascos.de/, 2012. 40, 123, 124

[Aurenhammer 91] F. Aurenhammer. *Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure.* ACM Computing Surveys, vol. 23, no. 3, pages 345–405, 1991. 13

[Aycard 10] O. Aycard. *Contribution to Perception for Intelligent Vehicles.* PhD thesis, Université de Grenoble, 2010. 46, 74

[Bar-Shalom 88] Y. Bar-Shalom & T. E. Fortmann. Tracking and data association. Mathematics in science and engineering. Academic Press, 1988. 17

[Bar-Shalom 95] Y. Bar-Shalom & X. R. Li. Multitarget-Multisensor Tracking: Principles and Techniques. YBS Publishing, 1995. 17, 85

[Bar-Shalom 09] Y. Bar-Shalom, F. Daum & J. Huang. *The probabilistic data association filter.* Control Systems, IEEE, vol. 29, no. 6, pages 82–100, 2009. 17, 85

[Bergman 99] N. Bergman. *Recursive Bayesian Estimation.* Navigation and tracking applications Linkoping Studies in Science and Technology. Dissertations, no. 579, 1999. 15

[Besl 92] P. J. Besl & H. D. McKay. *A method for registration of 3-D shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pages 239–256, February 1992. 80, 82

[Betaille 10] D. Betaille & R. Toledo-Moreo. *Creating Enhanced Maps for Lane-Level Vehicle Navigation.* In IEEE Transactions on Intelligent Transportation Systems, pages 786–798, 2010. 47

[BFV 93] Richtlinien für die Markierung von Straßen. Bundesministerium für Verkehr, 1993. 43, 48, 52, 55, 58, 90, 100

[Bosch 06] S. Bosch. Algebra. Springer-Lehrbuch. Springer, 2006. 49

[Cao 03] F. Cao. Geometric Curve Evolution and Image Processing. Springer, Berlin Heidelberg, 2003. 7

[Carmo 92] D. Carmo. Riemannian Geometry. Birkhäuser, Basel, 1992. 7

[Chang 83] K.-C. Chang & Y. Bar-Shalom. *Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements.* In American Control Conference, 1983, pages 466–471, 1983. 17

[Chen 10] A. Chen, A. Ramanandan & J.A. Farrell. *High-precision lane-level road map building for vehicle navigation.* In Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS), pages 1035–1042, 2010. 47, 49

[De Berg 08] M. De Berg, O. Cheong & M. Van Kreveld. Computational geometry: algorithms and applications. Springer-Verlag New York Inc, 2008. 13, 76

[De Boor 01] C. De Boor. A Practical Guide to Splines. Springer, Berlin, 2001. 49

[DESTATIS 11] DESTATIS - Deutsches statistisches Bundesamt. http://www.destatis.de/, 2011. 3

[Dickmanns 92] E. D. Dickmanns & B. D. Mysliwetz. *Recursive 3-d road and relative ego-state recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pages 199–213, 1992. 33

[Dieudonné 60] J. Dieudonné. Foundations of Modern Analysis. Academic Press, London, 1960. 7, 65

[DIN 01] Straßenfahrzeuge; Fahrzeugdynamik und Fahrverhalten; Begriffe (ISO 8855:1991, modifiziert). 2001. 23

[Dogruer 08] C. Dogruer, B. Koku & M. Dolen. *Global Urban Localization of Outdoor Mobile Robots using Satellite Images.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3927–3932, 2008. 74

[Donner 97] K. Donner. *Image Interpretation Based on Local Transform Characterization.* Pattern Recognition and Image Analysis, vol. 7, no. 4, 1997. 80, 81, 82

[Doucet 98] A. Doucet. *On Sequential Monte Carlo Sampling Methods for Bayesian Filtering*, 1998. 18

[Douglas 73] D. H. Douglas & T. K. Peucker. *Algorithms for the reduction of the number of points required to represent a digitized line or its caricature.* Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, no. 2, pages 112–122, 1973. 47

[Drysdale 08] R. L. S. Drysdale, G. Rote & A. Sturm. *Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs.* Comput. Geom. Theory Appl., vol. 41, pages 31–47, October 2008. 64

[Durrant-Whyte 06] H. Durrant-Whyte & T. Bailey. *Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms.* IEEE Robotics and Automation, vol. 2, 2006. 74

[Eykhoff 74] P. Eykhoff. System identification: parameter and state estimation. Wiley-Interscience, 1974. 37, 78

[Farin 02] G. Farin, J. Hoschek & M.-S. (eds.) Kim. Handbook of Computer Aided Geometric Design. Elsevier, Amsterdam, 2002. 49

[Fischler 81] M. A. Fischler & R. C. Bolles. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography.* Commun. ACM, vol. 24, no. 6, pages 381–395, 1981. 62

[Forster 11] O. Forster. Analysis 2: Differentialrechnung im $\mathbb{R}^n$, gewöhnliche Differentialgleichungen. Grundkurs Mathematik. Vieweg+Teubner Verlag, 2011. 55

[Franke 07] U. Franke, H. Loose & C. Knoppel. *Lane recognition on country roads.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 99–104, 2007. 33

[Fuerstenberg 05] K. Fuerstenberg & T. Weiss. *Feature-Level Map Building and Object Recognition for Intersection Safety Application.* In Proceedings of IEEE Intelligent Vehicles Symposium, 2005. 63, 74

[Gerlach 09] K. Gerlach & M. Meyer zu Horste. *A precise digital map for GALILEO-based train positioning systems.* In Proceedings of the 9th International Conference on Intelligent Transport Systems Telecommunications, pages 343–347, oct. 2009. 47

[Ghosh 91] S. K. Ghosh. *Computing the visibility polygon from a convex set and related problems.* Journal of Algorithms, vol. 12, no. 1, pages 75–95, 1991. 47

[GIDAS 08] GIDAS - German In-Depth Accident Study. http://www.gidas.org/, 2008. 3

[Gillespie 92] T. D. Gillespie. Fundamentals of Vehicle Dynamics. Society of Automotive Engineers, 1992. 25

[GML 12] *Geography Markup Language (GML).* http://www.opengeospatial.org/standards/gml, 2012. 50

[Golban 09] C. Golban, C. Mitran & S. Nedevschi. *A practical method for ego vehicle motion estimation from video.* In Proceedings of the 5th IEEE International Conference on Intelligent Computer Communication and Processing, pages 87–94, aug. 2009. 28

[Goodwin 77] G. C. Goodwin & R. L. Payne. Dynamic system identification: Experiment design and data analysis, volume 136. Academic Press, 1977. 37, 78

[GPS 08] Global Positioning System Standard Positioning Service Performance Standard. Department of Defense USA & GPS NAVSTAR, 2008. 4

[Graupe 72] D. Graupe. Identification of systems. Van Nostrand Reinhold, 1972. 37, 78

[Grewal 01] Grewal, Weill & Andrews. Global Positioning Systems, Inertial Navigation and Integration. John Wiley & Sons Inc., 2001. 39, 46

[Großmann 76] W. Großmann. Geodätische Rechnungen und Abbildungen in der Landesvermessung. Wittwer, 1976. 22, 23

[Groves 07] P. D. Groves. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. Artech House, 2007. 40, 46

[Grünwald 05] P. Grünwald. *Introducing the minimum description length principle.* Advances in minimum description length: theory and applications, page 3, 2005. 46

[Gutmann 99] J.-S. Gutmann & K. Konolige. *Incremental Mapping of Large Cyclic Environments.* In Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation, 1999. 74

[Haas 00] J. Haas. Echtzeit-Korrespondenzprobleme in Bildsequenzen und Subpixelgenauigkeit. Shaker, 2000. 32

[Hanning 05] T. Hanning, S. Graf & M. Kellner. *Re-projective vs. projective camera calibration: effects on 3D-reconstruction.* In IEEE International Conference on Image Processing, pages 1170–1173, 2005. 32

[Hanning 10] T. Hanning. High Precision Camera Calibration. Vieweg + Teubner research. Vieweg+Teubner Verlag, 2010. 31, 32

[Haralick 92] R. M. Haralick & L. G. Shapiro. Computer and Robot Vision. Addison-Wesley, 1992. 32

[Hartshorne 00] R. Hartshorne. Geometry. Springer, New York, 2000. 7

[HAVEit 11] HAVEit. http://www.haveit-eu.org/, 2011. 3

[Heck 03] B. Heck. Rechenverfahren und Auswertemodelle der Landesvermessung, klassische und moderne Methoden. Herbert Wichmann Verlag, 2003. 23

[Heenan 05] A. Heenan, C. Shooter, M. Tucker, K. Fuerstenberg & T. Kluge. *Feature-Level Map Building and Object Recognition for Intersection Safety Applications.* In 9th International Forum on AMAA, 2005. 39, 63

[Heimlich 08] M. Heimlich & M. Held. *Biarc Approximation, Simplification and Smoothing of Polygonal Curves by Means of Voronoi-Based Tolerance Bands.* Int. J. Comput. Geometry Appl., vol. 18, no. 3, pages 221–250, 2008. 64

[Held 05] M. Held & J. Eibl. *Biarc approximation of polygons within asymmetric tolerance bands.* Computer-Aided Design, vol. 37, no. 4, pages 357–371, 2005. 64

[Herold 05] M. Herold & D. Roberts. *Mapping Asphalt Road Conditions with Hyperspectral Remote Sensing.* In Joint Symposia URBAN - URS 2005, 2005. 46

[Heywood 06] I. Heywood, S. Cornelius & S. Carver. An Introduction to Geographical Information Systems. Prentice Hall, 2006. 50

[Hofmann-Wellenhof 03] B. Hofmann-Wellenhof & K. Legat. Navigation: Principles of Positioning and Guidance. Springer, 2003. 39

[Huck 11] T. Huck, A. Westenberger, M. Fritzsche, T. Schwarz & K. Dietmayer. *Precise timestamping and temporal synchronization in multi-sensor fusion.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 242–247, 2011. 30

[interactIVe 13] interactIVe - accident avoidance by active intervention for Intelligent Vehicles. http://www.interactive-ip.eu/, 2013. 3

[INTERSAFE-2 11] INTERSAFE-2. http://www.intersafe-2.eu/, 2011. 3, 74

[Irle 09] P. Irle, L. Gröll & M. Werling. *Zwei Zugänge zur Projektion auf 2d-Kurven für die Bahnregelung autonomer Fahrzeuge (Two Approaches for the Projection onto 2d-Curves for the Path Control of Autonomous*

Cars). Automatisierungstechnik, vol. 57, no. 8, pages 403–410, 2009. 13

[Janda 11] F. Janda. Verkehrszeichenerkennung im automobilen Umfeld basierend auf einer Grauwertkamera. Master's thesis, Universität Passau, 2011. 39

[Jazwinski 70] A. H. Jazwinski. Stochastic processes and filtering theory, volume 63. Academic Pr, 1970. 14

[Julier 95] S. J. Julier, J. K. Uhlmann & H. F. Durrant-Whyte. *A new approach for filtering nonlinear systems.* In Proceedings of the American Control Conference, volume 3, pages 1628–1632, 1995. 17

[Julier 97] S. J. Julier & J. K. Uhlmann. *A New Extension of the Kalman Filter to Nonlinear Systems.* In Proceedings of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls, pages 182–193, 1997. 17

[Kailath 00] T. Kailath, A. H. Sayed & B. Hassibi. *Linear estimation.* Upper Saddle River, NJ, 2000. 16

[Kalman 60] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems.* Transactions of the ASME-Journal of Basic Engineering, vol. 82, no. Series D, pages 35–45, 1960. 16

[Kellner 06] M. Kellner. *Zeitliche Konturgraphverfolgung.* PhD thesis, Universität Passau, 2006. 32

[Ko-FAS 13] Ko-FAS - Fahrerassistenz und präventive Sicherheit mittels kooperativer Perzeption. http://www.ko-fas.de/, 2013. 3

[Kolesnikov 03] A. Kolesnikov. *Efficient Algorithms for Vectorization and Polygonal Approximation.* PhD thesis, University of Joensuu, 2003. 47

[Konrad 12] M. Konrad, D. Nuss & K. Dietmayer. *Localization in digital maps for road course estimation using grid maps.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 87–92, 2012. 74

[Koutaki 06] G. Koutaki, K. Uchimura & Z. Hu. *Network Active Shape Model for Updating Road Map from Aerial Images.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 325–330, 2006. 49

[Kümmerle 09] R. Kümmerle, B. Steder, C. Dorhege, A. Kleiner, G. Grisetti & W. Burgard. *Large Scale Graph-based SLAM using Aerial Images as Prior Information.* In Proceedings of Robotics: Science and Systems, 2009. 61

[Levinson 11] J. S. Levinson. *Automatic laser calibration, mapping, and localization for autonomous vehicles.* PhD thesis, Standford University, 2011. 74

[Li 05] M. Li & D. S. Meek. *Smoothing an arc spline.* In Computers & Graphics, pages 576–587, 2005. 59

[Liu 98] J. S. Liu & R. Chen. *Sequential Monte Carlo methods for dynamic systems.* Journal of the American statistical association, pages 1032–1044, 1998. 18

[López-Pérez 12] D. López-Pérez & R. Toledo-Moreo. *An Approach to Road Enhanced Maps in Urban Areas.* In Workshop on Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles, IEEE Intelligent Vehicle Symposium, 2012. 47

[Lowe 99] D. G. Lowe. *Object Recognition from Local Scale-Invariant Features.* In Proceedings of the International Conference on Computer Vision, 1999. 74

[Lu 97] F. Lu & E. E. Milios. *Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans.* Journal of Intelligent and Robotic Systems, vol. 18, no. 3, pages 249–275, 1997. 85

[Maguire 97] D. J. Maguire, Goodchild M. F. & Rhind D. W. Geographic Information Systems: principles, and applications. Longman Scientific and Technical, 1997. 50

[Maier 10] G. Maier. *Smooth Minimum Arc Paths - Contour Approximation with Smooth Arc Splines.* PhD thesis, Universität Passau, 2010. 7, 10, 49, 64, 65, 67, 70

[Maier 11a] G. Maier, S. Pangerl & A. Schindler. *Real-time detection and classification of arrow markings using curve-based prototype fitting.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 442–447, 2011. 37

[Maier 11b] G. Maier & A. Schindler. *Visual Quality Control of Planar Working Pieces: A Curve Based Approach Using Prototype Fitting.* In Beniamino Murgante, Osvaldo Gervasi, Andrés Iglesias, David Taniar & Bernady O. Apduhan, editors, ICCSA (4), volume 6785 of *Lecture Notes in Computer Science*, pages 91–102. Springer, 2011. 81

[Maier 13] G. Maier, A. Schindler, F. Janda & S. Brummer. *Detecting Straight Sections and Optimal Arc Spline Approximation.* In Beniamino Murgante, Osvaldo Gervasi, Andrés Iglesias, David Taniar & Bernady O. Apduhan, editors, ICCSA, Lecture Notes in Computer Science. Springer, to be published, 2013. 69

[Mattern 08] N. Mattern, R. Schubert & G. Wanielik. *Image landmark based positioning in road safety applications using high accurate maps.* In Position, Location and Navigation Symposium, 2008 IEEE/ION, pages 1008–1013, 2008. 74

[Mattern 09] N. Mattern, R. Schubert & G. Wanielik. *Lane Level Positioning using Line Landmarks and High Accurate Maps.* In Proceedings of the 16th World Congress on Intelligent Transportation Systems, 2009. 74

[Mattern 10a] N. Mattern, R. Schubert & G. Wanielik. *High-accurate vehicle localization using digital maps and coherency images.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 462–469, 2010. 47

[Mattern 10b] N. Mattern & G. Wanielik. *Camera-based Vehicle Localization at Intersections using Detailed Digital Maps.* In Position Location and Navigation Symposium, 2010. 74

[McCall 06] J. C. McCall & M. M. Trivedi. *Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation.* IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 1, pages 20–37, 2006. 33

[Meek 90] D. S. Meek & D. J. Walton. *Offset curves of clothoidal splines.* Computer-Aided Design, vol. 22, no. 4, pages 199–201, 1990. 48

[Meek 04] D. S. Meek & D. J. Walton. *An arc spline approximation to a clothoid.* Journal of Computational and Applied Mathematics, vol. 170, no. 1, pages 59–77, 2004. 48, 90

[Meintrup 05] D. Meintrup & S. Schäffler. Stochastik. Springer, 2005. 14

[Mitschke 04] M. Mitschke & H. Wallentowitz. Dynamik Der Kraftfahrzeuge. Vdibuch. Springer, 2004. 25

[Montemerlo 02] M. Montemerlo, S. Thrun, D. Koller & B. Wegbreit. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem.* In Rina Dechter & Richard S. Sutton, editors, AAAI/IAAI, pages 593–598. AAAI Press / The MIT Press, 2002. 74

[Musicki 94] D. Musicki, R. Evans & S. Stankovic. *Integrated probabilistic data association.* IEEE Transactions on Automatic Control, vol. 39, no. 6, pages 1237–1241, 1994. 17, 85

[Musicki 02] D. Musicki & R. Evans. *Joint Integrated Probabilistic Data Association - JIPDA.* In Proceedings of the Fifth International Conference on Information Fusion, volume 2, pages 1120–1125, 2002. 17

[Napier 10] A. Napier, G. Sibley & P. Newman. *Real-Time Bounded-Error Pose Estimation for Road Vehicles using Vision.* In Proceedings of the 13th IEEE International Conference on Intelligent Transportation Systems, pages 1141–1146, 2010. 74

[Nocedal 99] J. Nocedal & S. J. Wright. Numerical Opitimization. Springer, New York, 1999. 82

[Noyer 08] U. Noyer, J. Schomerus, H. H. Mosebach, J. Gacnik, C. Loper & K. Lemmer. *Generating high precision maps for advanced guidance support.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 871–876, 2008. 47

[Nüchter 10] A. Nüchter. 3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom. Springer Berlin Heidelberg, 2010. 46, 62, 74

[Nürnberger 89] G. Nürnberger. Approximation by Spline Functions. Springer, 1989. 49

[OSM 12] *OpenStreetMap (OSM) project.* http://www.openstreetmap.org/, 2012. 47, 70

[Pangerl 10] S. Pangerl. Videobasierte Detektion und Erkennung von Landmarken. Master's thesis, Universität Passau, 2010. 39

[Pink 10] O. Pink & C. Stiller. *Automated map generation from aerial images for precise vehicle localization.* In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, pages 1517–1522, 2010. 47, 74

[Pink 11] O. Pink. *Bildbasierte Selbstlokalisierung von Straßenfahrzeugen.* PhD thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2011. 61, 74, 85

[Pisinger 03] G. Pisinger. Lokale Stützstrukturen zur Transformationspassung von Bildern. Shaker, 2003. 32

[Polychronopoulos 07] A. Polychronopoulos, M. Tsogas, A.J. Amditis & L. Andreone. *Sensor fusion for predicting vehicles' path for collision avoidance systems.* IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 3, pages 549–562, 2007. 33

[PReVENT 08] PReVENT. http://www.prevent-ip.org/, 2008. 3

[Rauch 13] S. Rauch. Kollektives Erlernen hochgenauer Straßenmodelle mit der Fahrzeugflotte. TU Braunschweig, to be published, 2013. 62

[Raudies 12] F. Raudies & H. Neumann. *A review and evaluation of methods estimating ego-motion.* Computer Vision and Image Understanding, vol. 116, no. 5, pages 606–633, 2012. 28

[Ress 08] C. Ress, D. Balzer, A. Bracht, S. Durekovic & J. Löwenau. *ADASIS Protocol For Advanced in-vehicle Applications.* In ADASIS Forum, 2008. 47

[Richter 08] D. Richter & M. Heindel. Straßen- und Tiefbau. Teubner Verlag, 2008. 48, 90

[Riekert 40] P. Riekert & T. E. Schunck. *Zur Fahrmechanik des gummibereiften Kraftfahrzeugs.* Archive of Applied Mechanics, vol. 11, no. 3, pages 210–224, 1940. 25

[Rissanen 85] J. Rissanen. Minimum-Description-Length Principle. Wiley Online Library, 1985. 46

[Rosenhahn 06] B. Rosenhahn, T. Brox, D. Cremers & H. Seidel. *A comparison of shape matching methods for contour based pose estimation.* In Combinatorial Image Analysis, LNCS 4040, pages 263–276, 2006. 80

[Rössler 06] B. Rössler, K. Fuerstenberg & U. Lages. *Intersection Safety — the EC Project INTERSAFE.* In J. Valldorf & W. Gessner, editors, Advanced Microsystems for Automotive Applications 2006, VDI-Buch, pages 437–449. Springer Berlin Heidelberg, 2006. 74

[Rusinkiewicz 01] S. Rusinkiewicz & M. Levoy. *Efficient Variants of the ICP Algorithm.* In Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling, pages 145–152, 2001. 80

[SAFESPOT 09] SAFESPOT. http://www.safespot-eu.org/, 2009. 74

[Samet 90] H. Samet. Applications of spatial data structures - computer graphics, image processing, and GIS. Addison-Wesley, 1990. 13, 76

[Schindler 09] A. Schindler & V. Lauren. *Video-based recognition of unmarked lanes via texture interpretation and N-level-set-fitting.* In Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, pages 1–6, 2009. 33

[Schindler 11] A. Schindler, G. Maier & S. Pangerl. *Exploiting Arc Splines for Digital Maps.* In Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems, pages 1–6, 2011. 64, 91

[Schindler 12] A. Schindler, G. Maier & F. Janda. *Generation of high precision digital maps using circular arc splines.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 246–251, 2012. 50, 59

[Schmidt 66] S. F. Schmidt. *Application of state-space methods to navigation problems.* Advances in Control Systems, vol. 3, pages 293–340, 1966. 16

[Schön 06] T. Schön. *Estimation of Nonlinear Dynamic Systems.* PhD thesis, Linköping University, 2006. 15, 16, 18

[Schubert 08] R. Schubert, E. Richter & G. Wanielik. *Comparison and Evaluation of Advanced Motion Models for Vehicle Tracking.* In Proceedings of the 11. International Conference on Information Fusion, 2008. 25

[Schubert 09] R. Schubert, N. Mattern, M. Schlingelhof, F. Ahlers & D. Westhoff. *Vehicle positioning for cooperative systems - the SAFESPOT approach.* In Proceedings of the 16th World Congress on Intelligent Transportation Systems, 2009. 74

[Se 02] S. Se, D. G. Lowe & J. J. Little. *Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks.* I. J. Robotic Res., vol. 21, no. 8, pages 735–760, 2002. 74

[Seidel 06] H. Seidel. Die Mathematik der Gauß-Krüger-Abbildung. online: http://henrik-seidel.gmxhome.de/gausskrueger.pdf, 2006. 22

[Smith 07] S. M. Smith & J. M. Brady. *SUSAN — A New Approach to Low Level Image Processing.* In International Journal of Computer Vision 23(1), pages 45–78, 2007. 32

[Soderstrom 89] T. Soderstrom & P. Stoica. *System Identification*, 1989. 37, 78

[Stone 11] T. C. Stone. Digitale Karten für sicherheitskritische Fahrerassistenzsysteme. Master's thesis, Universität Passau, 2011. 77

[Strang 08] T. Strang, F. Schuber, S. Thölert & R. Oberweis. Lokalisierungsverfahren. Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Kommunikation und Navigation, 2008. 39

[Suri 86] S. Suri. *A linear time algorithm with minimum link paths inside a simple polygon.* Comput. Vision Graph. Image Process., vol. 35, pages 99–110, July 1986. 47, 90

[Tatschke 08] T. Tatschke & J. Urlhardt. *Model-based road lane extraction from monocular image sequences.* PReVENT Fusion Forum eJournal Vol. 2, 2008. 33

[Tatschke 11] T. Tatschke. *Methoden und Modelle der frühen Sensordatenfusion zur Umgebungserfassung für Fahrerassistenzsysteme.* PhD thesis, Universität Passau, 2011. 14, 18, 19, 26, 31, 33, 37

[Thrun 98] S. Thrun, W. Burgard & D. Fox. *A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots.* In Machine Learning and Autonomous Robots, 1998. 74

[Thrun 00] S. Thrun, D. Fox, W. Burgard & F. Dellaert. *Robust Monte Carlo Localization for Mobile Robots.* Artificial Intelligence, vol. 128, no. 1-2, pages 99–141, 2000. 74

[Thrun 02] S. Thrun. *Robotic Mapping: A Survey.* In G. Lakemeyer & B. Nebel, editors, Exploring Artificial Intelligence in the New Millennium. Morgan Kaufmann, 2002. 74

[Thrun 05] S. Thrun, W. Burgard & D. Fox. Probabilistic robotics. MIT Press, 2005. 74

[Torge 03] W. Torge. Geodäsie, volume 22. De Gruyter, 2003. 22

[Veltkamp 99] Remco C. Veltkamp & Michiel Hagedoorn. *State-of-the-Art in Shape Matching.* Rapport technique, Principles of Visual Information Retrieval, 1999. 80

[Vogel 07] K. Vogel. *High-Accuracy Reference Data Acquisition for Evaluation of Active Safety Systems by means of a RTK-GNSS-Surveying System.* In Proceedings of the 6th European Congress and Exhibition on ITS, 2007. 40

[Walter 97] E. Walter & L. Pronzato. *Identification of parametric models.* Communications and Control Engineering, 1997. 37, 78

[Walton 90] D. J. Walton & D. S. Meek. *Clothoidal splines.* Computers & Graphics, vol. 14, no. 1, pages 95–100, 1990. 48

[Walton 05] D. J. Walton & D. S. Meek. *A controlled clothoid spline.* Computers & Graphics, vol. 29, no. 3, pages 353–363, 2005. 48

[Wang 01] L. Z. Wang, K. T. Miura, E. Nakamae, T. Yamamoto & T. J. Wang. *An approximation approach of the clothoid curve defined in the interval* $[0, \pi/2]$ *and its offset by free-form curves.* Computer-Aided Design, vol. 33, no. 14, pages 1049–1058, 2001. 48

[Wang 11] Z. Wang, S. Huang & G. Dissanayake. Simultaneous Localization and Mapping: Exactly Sparse Information Filters. World Scientific Pub Co, 2011. 46, 74

[Weiss 05a] T. Weiss, N. Kaempchen & K. Dietmayer. *High Accuracy Ego-Localization in Complex Dynamic Environments using Laserscanners.* In Proceedings of the 12th World Congress on Intelligent Transport Systems, 2005. 74

[Weiss 05b] T. Weiss, N. Kaempchen & K. Dietmayer. *Precise ego-localization in urban areas using Laserscanner and high accuracy feature maps.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 284–289, 2005. 74, 85

[Weiss 07] T. Weiss & K. Dietmayer. *Automatic Detection of Traffic Infrastructure Objects for the Rapid Generation of Detailed Digital Maps using Laser Scanners.* In Proceedings of the IEEE Intelligent Vehicle Symposium, 2007. 39, 63

[Weiss 11] T. Weiss. *Hochgenaue Positionierung und Kartographie mit Laserscannern für Fahrerassistenzsysteme.* PhD thesis, Universität Ulm, 2011. 63, 74, 85

[Wendel 07] J. Wendel. Integrierte Navigationssysteme. Oldenbourg, 2007. 40

[Wertheimer 13] R. Wertheimer, R. Krzikalla, A. Schindler & M. Wankerl. *Improving Safety by Self-Localization using Satellites and Landmarks.* In ATZ - Automobiltechnische Zeitschrift Volume 115, Issue 1, pages 48–52. Springer Vieweg Verlag, 2013. 3

[Westenberger 11] A. Westenberger, T. Huck, M. Fritzsche, T. Schwarz & K. Dietmayer. *Temporal synchronization in multi-sensor fusion for future driver assistance systems.* In Proceedings of the International IEEE Symposium on Precision Clock Synchronization for Measurement Control and Communication, pages 93–98, sept. 2011. 30

[Wimmer 11] A. Wimmer, R. Graf & K. Dietmayer. *Matching highly accurate maps to local environmental perception at road construction sites.* In Proceedings of the IEEE Intelligent Vehicles Symposium, pages 975–980, 2011. 74

[Xu 96] Y. Xu, V. Sasse & K. Harms. *The european digital road map MultiMap and its applications.* Google Scholar, 1996. 47

Supervised student works

Thomas Christian Stone: *Digitale Karten für sicherheitskritische Fahrerassistenzsysteme*, Master thesis, Universität Passau, 2011.

Florian Janda: *Verkehrszeichenerkennung im automobilen Umfeld basierend auf einer Grauwertkamera*, Master thesis, Universität Passau, 2011.

Sebastian Pangerl: *Videobasierte Detektion und Erkennung von Landmarken*, Diploma thesis, Universität Passau, 2010.

Tobias Knerr: *3D-Visualisierung von OpenStreetMap Daten*, Student work, 2011.

Publications

Reiner Wertheimer, Roland Krzikalla, Andreas Schindler, Matthias Wankerl: *Improving Safety by Self-Localization using Satellites and Landmarks.* In: ATZ – Automobiltechnische Zeitschrift 1-2013, Volume 115, Issue 1, 48-52. Springer Vieweg Verlag, 2013.

Andreas Schindler: *High Precision Digital Maps for Advanced Driver Assistance Systems.* In: Workshop on Navigation, Perception, Accurate Positioning and Mapping for Intelligent Vehicles, IEEE Intelligent Vehicles Symposium (IV), 2012.

Andreas Schindler, Georg Maier, Florian Janda: *Generation of High Precision Digital Maps using Circular Arc Splines.* In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV) 246-251. IEEE Publishing, 2012.

Andreas Schindler, Georg Maier, Sebastian Pangerl: *Exploiting Arc Splines for Digital Maps.* In: Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC) 1-6. 2011.

Georg Maier, Andreas Schindler, Florian Janda, Stephan Brummer: *Detecting Straight Sections and Optimal Arc Spline Approximation.* In: Beniamino Murgante et al. (eds.), ICCSA, Lecture Notes in Computer Science. Springer, to be published, 2013.

Georg Maier, Andreas Schindler, Florian Janda, Stephan Brummer: *Integration von Strecken in die Kreisbogensplinepassung mit optimaler Segmentzahl.* In: tm - Technisches Messen. Oldenbourg Verlag, to be published in 2013.

Georg Maier, Florian Janda, Andreas Schindler: *Minimum Description Length Arc Spline Approximation of Digital Curves.* In: Proceedings of IEEE 19th International Conference on Image Processing (ICIP) 1869-1872, IEEE Publishing, 2012.

Georg Maier, Andreas Schindler, Florian Janda, Stephan Brummer: *Integration von Strecken in die Kreisbogensplinepassung mit optimaler Segmentzahl.* In: Puente Leon, F. and Heizmann, M. (eds.): Forum Bildverarbeitung 95-106. KIT Scientific Publishing, Karlsruhe, 2012.

Georg Maier and Andreas Schindler: *Effiziente Prototyp-Passung mittels Kreisbogensplines und Anwendungsbeispiele aus dem Automotive-Bereich.* In: tm - Technisches Messen 78(9): 406-411. Oldenbourg Verlag, 2011.

Georg Maier, Sebastian Pangerl, Andreas Schindler: *Real-time Detection and Classification of Arrow Markings using Curve-based Prototype Fitting.* In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV) 442-447. IEEE Publishing, 2011.

Georg Maier, Andreas Schindler: *Visual Quality Control of Planar Working Pieces. A Curve based approach using Prototype Fitting.* In: Murgante et al. (eds.): Computational Science and its Applications - ICCSA 2011. Lecture Notes in Computer Science 91-102. Springer, 2011.

Georg Maier, Andreas Schindler: *Kreisbogensplines für die Prototyp-Passung Anwendungen im Automotive-Bereich.* In: Puente Leon, F. and Heizmann, M. (eds.): Forum Bildverarbeitung 35-46. KIT Scientific Publishing, Karlsruhe, 2010.

Andreas Schindler, Georg Maier: *Object Detection in gray scale Images based on Invariant Polynomial Features.* In: Proceedings of IEEE 17th International Conference on Image Processing (ICIP) 4633-4636. IEEE Publishing, 2010.

Andreas Schindler: *Model-based detection and tracking of objects using a 3D-camera.* In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV). IEEE Publishing, 2010.

Andreas Schindler, Verena Lauren: *Video-based Recognition of Unmarked Lanes via Texture Interpretation and N-Level-Set-Fitting.* In: Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems (ITSC) 114-119. 2009.