

# Street Surfaces and Boundaries from Depth Image Sequences Using Probabilistic Models

## **Inaugural-Dissertation**

zur  
Erlangung des Grades  
Doktor-Ingenieur  
(Dr.-Ing.)  
der  
Hohen Landwirtschaftlichen Fakultät  
der  
Rheinischen Friedrich-Wilhelms-Universität  
zu Bonn

vorgelegt am 26. April 2013 von

**Jan Siegemund**

aus Köln

Referent: Prof. Dr.-Ing. Wolfgang Förstner

1. Korreferent: Prof. Dr. techn. Wolf-Dieter Schuh

2. Korreferent: Prof. Dr.-Ing. Bodo Rosenhahn

Tag der mündlichen Prüfung: 27. September 2013

Erscheinungsjahr: 2013

Diese Dissertation ist auf dem Hochschulschriften-  
server der ULB Bonn  
[http://hss.ulb.uni-bonn.de/diss\\_online](http://hss.ulb.uni-bonn.de/diss_online)  
elektronisch publiziert.



# **Zusammenfassung**

## **Schätzung von Straßenoberflächen und -begrenzungen aus Sequenzen von Tiefenkarten unter Verwendung probabilistischer Modelle**

Diese Arbeit präsentiert ein Verfahren zur Detektion und Rekonstruktion von Straßenoberflächen und -begrenzungen auf der Basis von Tiefenkarten.

Aktive Fahrerassistenzsysteme, welche mit der im Fahrzeug verbauten Sensorik die Umgebung erfassen, interpretieren und den Fahrer unterstützen, sind ein aktueller Forschungsschwerpunkt der Fahrzeugindustrie. Eine wesentliche Aufgabe dieser Systeme ist die Modellierung der statischen Fahrzeugumgebung. Dies beinhaltet die Bestimmung der vertikalen Neigungs- und Krümmungseigenschaften der Fahrbahn, sowie die robuste Detektion von Hindernissen und somit des befahrbaren Freiraumes. Hindernisse von geringer Höhe, wie z.B. Bordsteine, sind in diesem Zusammenhang von besonderem Interesse, da sie häufig die erste geometrische Begrenzung des Fahrbahnbereiches darstellen.

In diesem Kontext gewinnt die Verwendung von Tiefenkarten aus Stereo-Kamera-Systemen wegen der hohen Datenrate und relativ geringen Kosten des Sensors zunehmend an Bedeutung. Aufgrund des starken Messrauschens beschränken sich herkömmliche Verfahren zur Hinderniserkennung jedoch meist auf erhabene Objekte wie Fahrzeuge oder Leitplanken, oder aber adressieren einzelne Objektklassen wie Bordsteine explizit. Dazu werden häufig extrem restriktive Annahmen verwendet wie z.B. planare Straßenoberflächen.

Der Hauptbeitrag dieser Arbeit besteht in der Entwicklung, Analyse und Evaluation eines Verfahrens, welches den befahrbaren Freiraum im Nahbereich des Fahrzeugs detektiert und dessen Begrenzung mit Hilfe einer Spline-Kurve explizit modelliert. Das Verfahren berücksichtigt insbesondere Hindernisse geringer Höhe (größer als 10 cm) ohne Beschränkung auf bestimmte Objektklassen. Weiterhin ist das Verfahren in der Lage, mit verschiedenartigen Neigungs- und Krümmungseigenschaften der vor dem Fahrzeug liegenden Fahrbahnoberfläche umzugehen und diese durch Verwendung eines flexiblen Spline-Modells zu rekonstruieren.

Um trotz der hohen Flexibilität des Modells und des hohen Messrauschens robuste Ergebnisse zu erzielen, verwendet das Verfahren probabilistische Modelle zur Vorverarbeitung der Eingabedaten und zur Detektion des befahrbaren Freiraumes. Aus den Tiefenkarten wird unter Berücksichtigung der Strahlengänge und Unsicherheiten der Tiefenmessungen ein Höhenmodell berechnet. In einem iterativen Zwei-Schritt-Verfahren werden anhand dieses Höhenmodells der befahrbare Freiraum mit Hilfe eines Markov-Zufallsfeldes bestimmt sowie die Parameter der begrenzenden Spline-Kurve und Straßenoberfläche geschätzt. Ausreißer in den Höhendaten werden dabei explizit modelliert.

Die Leistungsfähigkeit des Gesamtverfahrens sowie der Einfluss zentraler Komponenten, wird im Rahmen von Experimenten auf synthetischen und realen Testszenen systematisch analysiert. Die Ergebnisse demonstrieren die Fähigkeit des Verfahrens, die Begrenzung des befahrbaren Freiraumes sowie die Fahrbahnoberfläche selbst in komplexen Szenarien mit multiplen Hindernissen oder starker Fahrbahnkrümmung akkurat zu modellieren. Weiterhin werden die Grenzen des Verfahrens aufgezeigt und detailliert untersucht.

# Summary

## Street Surfaces and Boundaries from Depth Image Sequences Using Probabilistic Models

This thesis presents an approach for the detection and reconstruction of street surfaces and boundaries from depth image sequences.

Active driver assistance systems which monitor and interpret the environment based on vehicle mounted sensors to support the driver embody a current research focus of the automotive industry. An essential task of these systems is the modeling of the vehicle's static environment. This comprises the determination of the vertical slope and curvature characteristics of the street surface as well as the robust detection of obstacles and, thus, the free drivable space (alias free-space). In this regard, obstacles of low height, e.g. curbs, are of special interest since they often embody the first geometric delimiter of the free-space.

The usage of depth images acquired from stereo camera systems becomes more important in this context due to the high data rate and affordable price of the sensor. However, recent approaches for object detection are often limited to the detection of objects which are distinctive in height, such as cars and guardrails, or explicitly address the detection of particular object classes. These approaches are usually based on extremely restrictive assumptions, such as planar street surfaces, in order to deal with the high measurement noise.

The main contribution of this thesis is the development, analysis and evaluation of an approach which detects the free-space in the immediate maneuvering area in front of the vehicle and explicitly models the free-space boundary by means of a spline curve. The approach considers in particular obstacles of low height (higher than 10 cm) without limitation on particular object classes. Furthermore, the approach has the ability to cope with various slope and curvature characteristics of the observed street surface and is able to reconstruct this surface by means of a flexible spline model.

In order to allow for robust results despite the flexibility of the model and the high measurement noise, the approach employs probabilistic models for the preprocessing of the depth map data as well as for the detection of the drivable free-space. An elevation model is computed from the depth map considering the paths of the optical rays and the uncertainty of the depth measurements. Based on this elevation model, an iterative two step approach is performed which determines the drivable free-space by means of a Markov Random Field and estimates the spline parameters of the free-space boundary curve and the street surface. Outliers in the elevation data are explicitly modeled.

The performance of the overall approach and the influence of key components are systematically evaluated within experiments on synthetic and real world test scenarios. The results demonstrate the ability of the approach to accurately model the boundary of the drivable free-space as well as the street surface even in complex scenarios with multiple obstacles or strong curvature of the street surface. The experiments further reveal the limitations of the approach, which are discussed in detail.

*To my grandparents*



# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
1.1	Motivation . . . . .	13
1.2	Related Work . . . . .	14
1.2.1	Driver Assistance Systems . . . . .	14
1.2.2	Detection and Modeling of the Street Surface and Street Boundaries .	16
1.2.2.1	Vision Based Free-Space Detection and Modeling . . . . .	16
1.2.2.2	Alternative Sensors and Sensor Fusion . . . . .	20
1.2.2.3	Detection of Low Obstacles Using Depth Maps . . . . .	20
1.3	Problem Statement and Achievements of the Thesis . . . . .	21
1.4	Organization of the Thesis . . . . .	21
<b>2</b>	<b>Technical Background</b>	<b>23</b>
2.1	Notation . . . . .	23
2.1.1	Representation of Geometric Entities . . . . .	23
2.1.1.1	Euclidean and Homogeneous Representation of Point Entities	23
2.1.1.2	Homogeneous Representation of 2D Lines . . . . .	24
2.1.2	Representation of Probabilistic Entities . . . . .	24
2.2	Sensor Model for Stereo Camera Systems . . . . .	25
2.2.1	Image Acquisition Model . . . . .	25
2.2.1.1	General Camera Model . . . . .	25
2.2.1.2	Ideal Camera Model . . . . .	27
2.2.2	Triangulation of 3D Points . . . . .	27
2.2.2.1	Normal Case of The Image Pair . . . . .	27
2.2.2.2	Stereo Rectification . . . . .	28
2.2.3	Simplified Notation . . . . .	28
2.2.4	Stereo Matching . . . . .	28
2.2.5	Camera Calibration . . . . .	29
2.3	Parameter Estimation . . . . .	30
2.3.1	Gauss-Markov Model . . . . .	30
2.3.2	Least Squares Estimation in Linear Models . . . . .	31
2.3.2.1	Maximum Likelihood (ML) Estimation . . . . .	31
2.3.2.2	Maximum a Posteriori (MAP) Estimation . . . . .	32
2.3.3	Least Squares Estimation in Non-Linear Models . . . . .	34
2.4	Probabilistic Graphical Models . . . . .	34
2.4.1	Markov Random Fields . . . . .	35
2.4.1.1	Conditional Independence Properties . . . . .	36
2.4.1.2	Utilized Variants of Markov Random Fields . . . . .	36
2.4.2	Inference in Markov Random Fields . . . . .	37

2.4.2.1	Exact Inference in Undirected Acyclic Graphs . . . . .	38
2.4.2.2	Approximate Inference in Undirected Graphs with Cycles . .	38
2.5	Logistic Regression . . . . .	38
2.5.1	Logistic Regression Task . . . . .	39
2.5.2	Logistic Regression Model . . . . .	39
2.5.3	Estimation of the model parameters . . . . .	40
2.5.3.1	Optimization via Newton's method . . . . .	40
2.5.3.2	Determination of the Precision of the Decision Boundary . .	42
2.5.3.3	Regularization . . . . .	43
2.5.4	Treatment of Non-Deterministic Targets . . . . .	44
2.6	The Expectation Maximization Algorithm . . . . .	45
2.6.1	General EM Algorithm . . . . .	45
2.6.2	EM Algorithm with Simplified Assumptions . . . . .	46
2.7	B-Splines . . . . .	47
2.7.1	B-Spline Curves . . . . .	47
2.7.1.1	Spline Parametrization . . . . .	47
2.7.1.2	Curvature of a B-Spline Curve . . . . .	48
2.7.2	B-Spline Surfaces . . . . .	49
2.7.2.1	Derivatives of a B-Spline Surface . . . . .	50
2.7.2.2	Curvature of a B-Spline Surface . . . . .	51
2.7.3	B-Spline Basis Functions . . . . .	51
2.7.4	Handling of Points beyond Outer Section Borders . . . . .	51
<b>3</b>	<b>Concept</b>	<b>53</b>
3.1	Overview . . . . .	53
3.2	Data Acquisition and Preprocessing . . . . .	59
3.2.1	Sensor Setup . . . . .	60
3.2.2	Probabilistic Elevation Map Computation . . . . .	60
3.2.2.1	Grid Structure of the Digital Elevation Map . . . . .	61
3.2.2.2	Local Elevation Estimation . . . . .	62
3.2.2.3	Stochastic Properties of the Elevation Map . . . . .	67
3.2.3	Ego-Motion . . . . .	69
3.3	Environment Model . . . . .	69
3.3.1	Definition of the Generative Environment Model . . . . .	70
3.3.2	Surface Model . . . . .	70
3.3.2.1	Elevation Model for Street Regions . . . . .	71
3.3.2.2	Elevation Model for Street Adjacent Regions . . . . .	73
3.3.3	Street Boundary Model . . . . .	73
3.3.3.1	Spline Parametrization . . . . .	74
3.3.3.2	Class Probabilities from Boundary Function . . . . .	74
3.3.4	A Priori Assumptions . . . . .	75
3.3.4.1	Prior Assumptions about the Street Surface . . . . .	77
3.3.4.2	Prior Assumptions about the Street Boundary . . . . .	77
3.4	Model Prediction . . . . .	78
3.4.1	Acquisition of Temporal Prior Knowledge . . . . .	78
3.4.1.1	Temporal Prior for the Street Surface Parameters . . . . .	79
3.4.1.2	Temporal Prior for the Street Boundary Parameters . . . .	80
3.4.2	Initial Values for the Model Parameters . . . . .	84
3.4.2.1	Initial Values for the Street Surface . . . . .	84

3.4.2.2	Initial Values for the Street Boundary . . . . .	85
3.4.2.3	Model Initialization . . . . .	86
3.5	Model estimation via EM-Algorithm . . . . .	87
3.5.1	Expectation Step (Posterior Distribution of the Latent Variables) . .	87
3.5.2	Minimization Step (Estimation of the Model Parameters) . . . . .	93
3.5.2.1	Estimation of the Street Surface Parameters . . . . .	94
3.5.2.2	Estimation of the Street Boundary Parameters . . . . .	96
3.5.3	Evaluation of the Estimated Model . . . . .	107
3.5.3.1	Local Evaluation of the Boundary Spline . . . . .	107
3.5.3.2	Termination Criteria . . . . .	108
<b>4</b>	<b>Experiments</b>	<b>111</b>
4.1	Evaluation Criteria . . . . .	112
4.1.1	Accuracy Assessment of the Image Projection of the Free-Space: . . .	113
4.1.2	Evaluation of the Spatial Accuracy of the Free-Space Boundary: . . .	114
4.2	Experiments on Synthetic Data . . . . .	116
4.2.1	Setup of Synthetic Sequences . . . . .	116
4.2.2	Evaluation of the outer precision at increasing noise . . . . .	119
4.2.3	Evaluation of the inner precision . . . . .	126
4.2.4	Evaluation of the influence of outliers . . . . .	127
4.2.5	Discussion of the Results on Synthetic Data . . . . .	130
4.3	Experiments on Real-World Data . . . . .	133
4.3.1	Data Acquisition and Setup of the Research Vehicle . . . . .	133
4.3.2	Benchmark Dataset . . . . .	134
4.3.2.1	Dataset Description . . . . .	134
4.3.2.2	Evaluation Methods . . . . .	134
4.3.3	Benchmark Evaluation Using the Proposed Model and Parameters . .	134
4.3.4	Evaluation of Selected Parameter and Model Modifications . . . . .	139
4.3.4.1	Evaluation of the Influence of the Maximum Number of EM Iterations . . . . .	139
4.3.4.2	Evaluation of the Usage of Simplified Street Surface Models	142
4.3.4.3	Evaluation of Changes of the CRF's Unary and Binary Terms	143
4.3.4.4	Evaluation of the Impact of the Temporal Prior Assumptions	145
4.3.4.5	Evaluation of the Impact of the Spatial Prior Assumptions .	146
4.3.4.6	Evaluation of the Impact of the Local Self-Evaluation Step .	147
4.3.4.7	Evaluation of a Naive DEM Computation Approach . . . . .	147
4.3.5	Falsely Detected Objects ( <i>Ghost Objects</i> ) . . . . .	149
4.3.6	Discussion of Real-World Results . . . . .	152
4.3.6.1	Applicability of the Approach in Real-World Traffic Scenarios	152
4.3.6.2	Influence of Single Model Components on the Performance of the Overall Approach . . . . .	153
4.3.6.3	Transferability of the Insights Achieved on Synthetic Data .	154
4.3.6.4	Challenges and Limits . . . . .	155
<b>5</b>	<b>Conclusion and Outlook</b>	<b>159</b>
<b>A</b>	<b>Construction of the Digital Elevation Map</b>	<b>163</b>
<b>B</b>	<b>Treatment of Sample Position Uncertainty</b>	<b>165</b>

<b>C</b>	<b>Error propagation for the Prediction Step</b>	<b>167</b>
C.1	Error propagation for the Prediction of the Street Surface . . . . .	167
C.2	Error propagation for the Prediction of the Street Boundary . . . . .	169
<b>D</b>	<b>Mathematical Derivations</b>	<b>171</b>
D.1	Simplified Computation of the Expectation in the EM-Algorithm . . . . .	171
D.2	Standard Deviation from Intersection of Gaussians . . . . .	172



# Notation Overview

## Mathematical Notation

Symbol	Description
<i>General notation:</i>	
$x$	scalar
$\mathbf{x}$	Euclidean vector
$\mathbf{x}$	homogeneous vector
$[\mathbf{x}]_i$	$i$ -th element of vector $\mathbf{x}$
$M$	Euclidean matrix
$M$	homogeneous matrix
$[M]_{ij}$	matrix element in $M$ at row $i$ and column $j$
$\tilde{\mathbf{x}}$	true value of uncertain parameter $\mathbf{x}$
$\hat{\mathbf{x}}$	best estimate for uncertain parameter $\mathbf{x}$
$\Sigma_{xx}$	covariance matrix of uncertain parameter $\mathbf{x}$
$\Sigma_{\hat{x}\hat{x}}$	covariance matrix of estimated parameter $\mathbf{x}$
<i>Mathematical operators:</i>	
$\text{Diag}(\mathbf{x})$	square matrix holding the elements of the vector $\mathbf{x}$ as diagonal elements
$\text{Diag}(M_1, M_2)$	block-diagonal matrix composed of the square matrices $M_1$ and $M_2$
$\delta(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{else} \end{cases}$	Kronecker delta
$\mathbf{a} \circ \mathbf{b}$	Hadamard product of the vectors $\mathbf{a}$ and $\mathbf{b}$
$\lceil x \rceil$	round up $x$ to the next integer value
$\lfloor x \rfloor$	round down $x$ to the next integer value
$\text{tr}(M)$	trace of matrix $M$
<i>Special entities:</i>	
$\mathbf{1}_n$	$(n \times 1)$ vector with all elements equal 1
$I_n$	$(n \times n)$ identity matrix
$t$	discrete point in time
$\mathbf{l}$	vector of discrete latent variables
$\mathbf{z}$	observation vector
$\mathbf{p}$	unknown parameter vector
$\varphi$	probability density function (PDF)
$P$	probability

Special entities (continued):

$\mathcal{N}(\mu, \sigma^2)$	normal distribution governed by mean $\nu$ and variance $\sigma^2$
$G(x \mid \mu, \sigma^2)$	PDF of normal distribution evaluated at position $x$
$\mathcal{U}([a, b])$	uniform distribution defined over the interval $[a, b]$
$U(x \mid [a, b])$	PDF of uniform distribution evaluated at position $x$

---

**Acronyms**

Acronym	Meaning
<i>ABS</i>	Anti-lock Braking System
<i>CPU</i>	Central Processing Unit
<i>CRF</i>	Conditional Random Field
<i>DEM</i>	Digital Elevation Map
<i>EBA</i>	Emergency Brake Assist
<i>EM</i>	Expectation Maximization
<i>ESC</i>	Electronic Stability Control
<i>FOV</i>	Field Of View
<i>FPGA</i>	Field-Programmable Gate Array
<i>GPS</i>	Global Positioning System
<i>GPU</i>	Graphics Processing Unit
<i>IPM</i>	Inverse Perspective Mapping
<i>IVM</i>	Import Vector Machines
<i>MAP</i>	Maximum A Posteriori
<i>ML</i>	Maximum Likelihood
<i>MRF</i>	Markov Random Field
<i>PDF</i>	Probability Density Function
<i>SGM</i>	Semi-Global Matching

---

# Chapter 1

## Introduction

### 1.1 Motivation

Automotive driver assistance systems which support and relieve the driver gained greater significance and credence in the last decade. Reasons are more stringent demands on safety and driving comfort due to the increasing traffic volume. Let us divide these systems in two groups.

On the one hand, there are conventional systems which support the driver in critical situations, such as the Anti-lock Braking System (ABS) or the Electronic Stability Control (ESC). During the last decades, these systems have widely established as standard equipment in today's cars.

On the other hand, more advanced systems aim to avoid critical situations beforehand. Vehicle mounted sensors are used to monitor and analyze the immediate vicinity of the car to increase and guide the driver's attention and assist the driver in planning decisions. Examples for these high-level applications are collision and lane departure warnings, vehicle and traffic sign recognition, or traffic jam and parking assistants. The majority of these advanced applications require a preprocessing of the sensor output to extract meta information about the local environment of the vehicle. Particularly, the robust detection and accurate modeling of the free drivable space (alias free-space) in front of the vehicle (see Figure 1.1) as well as of the street elevation is a vital prerequisite.



FIGURE 1.1: Two examples for free-space detection results. The image projection of the drivable free-space is marked by a green carpet. The left image shows the result of the approach presented by Badino et al. [2007]. The sidewalk and median strip are classified as road inliers due to their low height occurrence. The right image shows the desired result representing the free-space limited by the actual first geometric obstacle, including curbs.

Various vehicle mounted sensors are used in this context, such as laser scanners, time-of-flight cameras or radar. However, stereo camera systems are getting more and more affordable

and are used for versatile applications, since they provide appearance as well as dense depth map information at a high data rate using modern stereo vision algorithms.

Although the task of free-space detection from stereo vision depth maps is well addressed in the field of driver assistance and robotics, most of these approaches are limited to the detection of obstacles that are distinctive in height, such as cars, walls or guardrails. In general, the actual geometric boundary of the drivable area is defined by objects of low height, e.g. curbs or traffic isles, as illustrated in Figure 1.1. Due to the high geometric variability of the obstacles and their possibly low height occurrence compared to the measurement noise of stereo vision depth maps, the detection and modeling of this actual boundary remains a challenging task.

As key focus of this thesis, we address this task in a probabilistic way to obtain accurate reconstructions of the street boundary (alias free-space boundary) and the street surface from depth image sequences.

## 1.2 Related Work

Automotive driver assistance systems aim to enhance the safety and comfort of the vehicle occupants as well as driving economics. The scope and variety of these systems and their submodules is very extensive. A comprehensive overview is given e.g. in [Rajamani, 2012] or [Küçükay and Bergholz, 2004].

The field of autonomous driving, where the driver is replaced rather than relieved, forms the extreme case of this development and becomes increasingly topical. Pioneer work was done in the joint-venture program *PROMETHEUS*<sup>1</sup> from 1986 to 1995, a collaboration of European universities and automotive companies. The achievable performance capabilities using almost arbitrary combinations of high-tech sensors were shown in the *DARPA*<sup>2</sup> Grand Challenges in 2004/05 and the Urban Challenge 2007. A more recent example is the *VIAC*<sup>3</sup> of the University of Parma, where in 2010 four vehicles drove from Parma, Italy, to Shanghai, China, largely autonomously based on internal sensors and sensor data perceived by a preceding non-autonomous vehicle [Broggi et al., 2010].

The remainder of this section presents the integration of the approach proposed in this thesis in the overall context of driver assistance systems (Section 1.2.1) and provides a comprehensive overview on published approaches closely related to the topic of this thesis (Section 1.2.2).

### 1.2.1 Driver Assistance Systems

To specify the focus of this thesis let us consider the classification of driver assistance systems presented in Section 1.1 in more detail. We distinguished two groups by whether or not the system is based on information on the vehicle's environment perceived by car mounted sensors.

The first group comprises systems based on submodules monitoring the vehicle state or even the driver himself by inertial or inner car sensors. Examples are ABS and ESP to improve the driveability in critical situations as already mentioned, adaptive headlights which improve the illumination in curves, or Emergency Brake Assist (EBA) which ensure the maximum braking power in an emergency stop situation.

The systems of the second group support and relieve the driver in his planning tasks based on information about the environment of the vehicle or its scheduled path perceived from car

<sup>1</sup>PROgramme for a European Traffic of Highest Efficiency and Unprecedented Safety

<sup>2</sup>Defense Advanced Research Projects Agency

<sup>3</sup>VisLab Intercontinental Autonomous Challenge

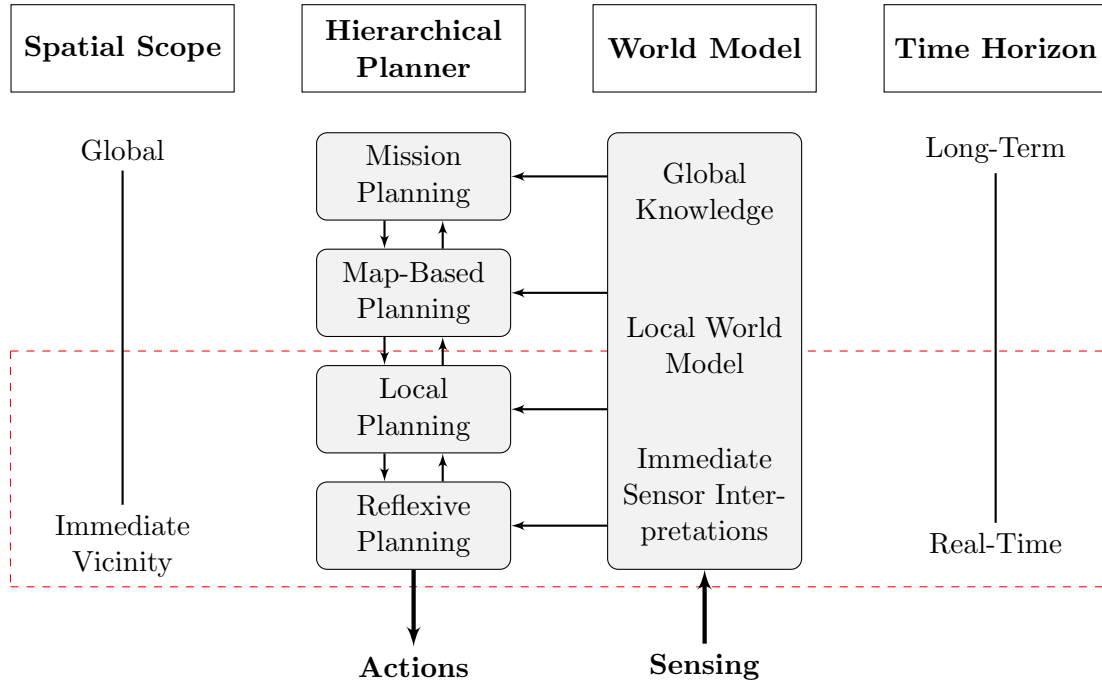


FIGURE 1.2: Hierarchical formulation of the overall planning effort for autonomous driving. The planning effort is subdivided in four single steps, where for each of them the alignment with respect to the spatial scope, the required world model knowledge and computation time horizon is symbolized. The dashed box encompasses the scope addressed in this thesis. The diagram is modified from [Arkin, 1998].

mounted sensors and map knowledge. Following the hierarchical decomposition of planning tasks for automotive control defined by Payton [1986] and Arkin [1998], we subdivide the overall planning effort in four separate steps:

- (a) **Mission Planning:** Determination of abstract mission goals in a geographic manner using global map knowledge (e.g. reach specific destination, search for next fuel station).
- (b) **Map-Based Planning:** Translation of geographic goals to specific route plans using map information and constraints (e.g. determine fastest route, avoid traffic jam).
- (c) **Local Planning:** Actual vehicle control aiming at the execution of a map-based plan by adaption to local conditions and unexpected changes based on sensor data (e.g. compliance with traffic rules, safety distance and lane keeping, overtaking and lane change maneuvers).
- (d) **Reflexive Planning:** Immediate reaction to unexpected events (e.g. collision avoidance, active brake assist or precrash systems).

Figure 1.2 illustrates the relationship of planning steps, required knowledge and time horizon.

The combinatorial support for the tasks (a) and (b) is tackled by GPS-based automotive navigation systems employing map-based knowledge as well as traffic information. In the last decade, these systems have established as a standard tool in today's cars.

Support for the tasks (c) and (d) requires knowledge on the local world model. This requires car mounted sensors and efficient algorithms to perceive and analyze the immediate vicinity of the vehicle in real-time. In particular, this includes the robust detection of the traversable street region and the accurate modeling of its boundary, which has been explored

by many researchers in the intelligent transportation systems community. One can categorize the developed approaches in two subgroups depending on their task.

The first category represents the detection of specific objects. This comprises for example the detection of vehicles [Sun et al., 2006] or more recent [Barth and Franke, 2010], pedestrian detection [Ess et al., 2009, Dollár et al., 2012, Byun et al., 2012], or the detection of curbs [Turchetto and Manduchi, 2003, Oniga et al., 2007b, Siegemund et al., 2010].

The second category comprises approaches for generic obstacle detection which aim to detect and model geometric limitations of the drivable free-space independent of their semantics. This task is referred to as *free-space detection* or *free-space computation* in the literature and represents the scope of this thesis.

## 1.2.2 Detection and Modeling of the Traversable Street Surface and Street Boundaries

The following paragraphs provide an overview on relevant publications addressing free-space computation, where we focus on vision based approaches including monocular and stereoscopic systems.

Unlike in case of scientific challenges such as the DARPA challenges, the kind and number of usable sensors is limited by cost and design properties when addressing systems for the automotive market. Thus, it is important to tackle multiple applications with a few low-cost sensors. This attracts the usage of affordable stereo camera systems which provide synchronous depth map and appearance information at a high data rate employing modern stereo vision algorithms.

For the sake of completeness, we also give examples for approaches based on alternative sensors, such as laser scanners, and approaches based on sensors fusion. Finally, we explicitly address the problem of detecting low obstacles, such as curbs and small traffic isles, and give a brief introduction to the few existing suitable methods.

### 1.2.2.1 Vision Based Free-Space Detection and Modeling

**Monocular Systems:** The set of algorithms based on a single camera is largely dominated by two basic ideas.

Approaches following the first idea classify the image pixels in a set of classes representing street and street-adjacent objects using appearance information, e.g. color and texture. The result is either given by the set of pixels classified to street itself (e.g. [Shioyama et al., 2003]), or the classification result is used to estimate the parameters of a predefined street model using simplifying assumptions and information on the camera orientation. In an early approach, Thorpe et al. [1988] perform a pixelwise Gaussian Maximum Likelihood classification to assign each single pixel to a road or non-road class. From the classification result the direction and vanishing point of the street is estimated assuming the road to be straight, flat and of a predefined width. In [Crisman and Thorpe, 1993] the authors present an extension of this idea in order to deal with intersections. In a similar approach, Kim et al. [2011] suggest the usage of alternative color-spaces to cope with illumination changes.

Broggi and Bert [1995] and more recent Chen and Liu [2010] propose an alternative idea, where they project the image to the assumed street plane using the known camera calibration and estimate the street borders by comparing the projected image to a set of road templates. Since no explicit geometric reconstruction of obstacles is performed, these methods are referred to as *road detection* algorithms rather than free-space detection algorithms in the literature.



FIGURE 1.3: Example for a road pavement that mimics an obstacle. The dark stripe on the left side appears similar to the curb on the right side. Consideration of its junction to the traffic isle shows that the stripe actually represents no geometric obstacle.

The second group of monoscopic approaches additionally applies 3d information determined by structure from motion. Brostow et al. [2008] and Sturgess et al. [2009] combine color and texture cues with geometric information such as height above the street plane, distance to the camera path and surface orientation in order to segment the image into regions representing street and street adjacent objects. To obtain a smooth segmentation, they employ a Markov Random Field (see Section 2.4.1) defined on the image lattice. However, although obtaining promising results, the approaches are far from being real-time capable.

Carlsson and Eklundh [1990] and Cerri and Grisleri [2005] use information about the vehicle's ego-motion to employ a technique called *Inverse Perspective Mapping* (IPM). Two successively acquired images are projected on the ground plane and aligned considering the planar ego-motion of the vehicle. Comparison of the color values in both images allows for the detection of horizontal regions which represent moving obstacles. In case the ego-vehicle is moving, even static obstacles of distinct height may be detected considering variations caused by the changed perspective. Note that these approaches heavily depend on the validity of the flat ground plane assumption and the quality of the ego-motion estimation.

**Stereoscopic Approaches:** An alternative approach to monocular systems exploits the additional depth information provided by stereo camera systems. The derived spatial information is used to overcome challenges of purely appearance based methods which severely suffer from image artifacts, highlights and shadows mimicking obstacles. Furthermore, the road paving may confuse appearance based algorithms by forming obstacle-like line and color structures, as depicted in Figure 1.3, demonstrating the need for geometry information. This is particularly the case when using gray value cameras. In [Michalke et al., 2010] the authors demonstrate the benefit of the explicit detection of geometric obstacles for their road detection framework.

The general idea of geometry based approaches is to evaluate the relationship of 3d point or depth measurements with a street surface model. The free-space is extracted as the horizontal region incorporating mainly measurements which conform to this model. Existing approaches vary in the way the free-space is determined and the way the street surface and free-space boundary are modeled. Let us first consider some examples for street surface representations and then proceed with an overview on relevant publications categorized in approaches based on disparity map analysis, IPM and grid based methods.

**Street Surface Models:** The most common street surface model is given by the flat surface assumption. The street is modeled as ground plane either fixed to the camera system or governed by free slope parameters. Although comfortable for handling and computation effort, the flat surface assumption in general does not hold, e.g. in case of undulating terrain and in case of roads which are roof-shaped for water drainage. Thus, height thresholds and noise values are defined lavishly to cope with the model limitations which restricts the detection scope to objects of distinct height.

More sophisticated approaches comprise piecewise planar representations [Labayrade et al., 2002], quadratic surfaces [Nedevschi et al., 2008], or clothoid based models [Nedevschi et al., 2004]. Wedel et al. [2009] employ a cubic B-Spline (see Section 2.7) parameterized along the longitudinal axis of the vehicle to model the curvature in longitudinal direction. However, these models may still be insufficient for more complicated environments, e.g. when crossing the roof shaped median strip in highway construction sites.

Manduchi et al. [2005] propose an interesting, alternative approach which was originally designed for off-road navigation but also applied in the VIAC challenge [Broggi et al., 2011]. The approach does not rely on a specific street surface model. The triangulated 3d points are classified as obstacle or road related assuming an obstacle point to be always above a road point, whereas a small horizontal offset is admitted. Simply put, each point defines a cone open to the top with the vertex given by the point coordinates and a predefined aperture angle. Each point lying in the cone of another point with at least a certain vertical offset is classified as obstacle. Although very flexible, outlier detection becomes complicated due to the lack of a proper surface model. This makes the method unsuitable for the detection of low obstacles.

**Disparity Map Analysis:** So called  $v$ -disparity map approaches are based on the desire to compute the free-space from a disparity map without performing expansive triangulation for each pixel. Wedel et al. [2009] give a comprehensive review of  $v$ -disparity approaches.

The basic idea is that the ideal transformation from the domain spanned by the height and longitudinal axis of the camera system to the domain spanned by the image row  $v$  and disparity  $d$  is straight line preserving. Thus, assuming the bank angle of a flat ground plane to be zero, points on the ground plane are projected on a line in the  $(v, d)$ -domain. Labayrade et al. [2002] employ Hough transformation to extract a piecewise planar street surface model from the  $(v, d)$ -measurements. Note that this implies the assumption that the majority of pixels observes the street surface. All pixels are classified as road or non-road comparing their  $(v, d)$ -coordinates to the respective reference given by the surface model using a region growing algorithm to obtain a dense result. In [Labayrade and Aubert, 2003] an extension for non-zero bank angles is proposed. Jung et al. [2007] use this representation to detect preceding vehicles.

**Inverse Perspective Mapping:** Similar to the monocular IPM approaches, knowledge about the relative orientation of the cameras to the ground plane is used to warp the left and right image to a common view-port. This view-port is either given by the birds eye view [Bertozzi and Broggi, 1998, Hattori and Maki, 2000, Onoguchi et al., 1995] or by one of the camera view-ports [Okutomi and Noguchi, 1998]. In case the ground-plane assumption holds, corresponding pixels depicting the ground plane in both of the warped images should match in their color values. Thus, analyzing the matching of the corresponding pixels yields the free-space estimation. Hattori and Maki [2000] determine the free-space boundary by fitting a road template to the pixels classified as free-space similar to the monocular approaches.



Bertozzi and Broggi [1998] and Onoguchi et al. [1995] perform a radial sweep and search for the closest violation of the ground-plane assumption along each scan-line.

The local evaluation of the match of corresponding pixels can be insufficient considering image artifacts such as reflections and highlights or homogeneous regions. Kubota et al. [2007] propose a sophisticated approach based on a restrictive environment model. The objects delimiting the free-space are assumed to be embodied by building blocks which are vertical columns perpendicular to the ground plane whose widths are defined to span the same amount of image columns each. An extended version of IPM is performed. The pixels below the foot-points of the building blocks are warped via the ground plane as before, while all pixels above the foot-points are warped via the fronto-parallel vertical surface of the building blocks. Thus, different distances of the building blocks to the camera yield different matching costs. Dynamic programming is employed to find the solution that minimizes the costs.

**Probabilistic Occupancy Grids:** Grid based representations are an established method to model the environment and to efficiently store and evaluate 3d information. They are widely used for non-automotive modeling applications in computer vision and computer graphics. Examples are space carving techniques for 3d reconstruction introduced by Kutulakos and Seitz [2000] or visibility culling in synthetic scenes [Batagelo and Wu, 2002].

Elfes [1989] introduced so called *occupancy grids* for path-planning applications in robotics. The occupancy grid is designed as planar lattice in the horizontal Cartesian space holding a probabilistic statement for each cell whether to be occupied by an obstacle or not.

One can categorize occupancy grids in two groups. First, there are deterministic occupancy grids which can be understood as 2d histograms counting the number of 3d measurements aligned to a specific cell without modeling the noise properties [Franke et al., 1997, Nedeveschi et al., 2008]. On the other hand, stochastic or probabilistic occupancy grids considering the sensor uncertainty of incoming range measurements to update the occupancy statements of the cells. Thrun et al. [2005] give a comprehensive review on the computation of planar and spatial occupancy grids and their applications in robotics.

Existing methods for free-space detection employing probabilistic occupancy grids mainly differ in the design of the grid structure and the coordinate system it is aligned to. Badino et al. [2007] give an overview of planar occupancy grids comprising Cartesian grids, polar grids and grids aligned to the column-disparity domain of the stereo-camera system. While Cartesian grids fixed to the world system facilitate temporal integration [Murray and Little, 2000, Kohara et al., 2010], grids fixed to the camera system (polar and column-disparity) allow for an efficient tracing of the optical ray and, thus, evaluation of free-space and occlusion information induced by a depth measurement (see Section 3.2.2.1).

In their proposal, Badino et al. [2007] employ a polar grid and use an approach similar to [Kubota et al., 2007] to estimate the free-space boundary. Dynamic programming is performed to search for the respective cell in each longitudinal grid column which most likely incorporates the nearest obstacle. Thus, the boundary is given by vertical building blocks located at the determined cells. As a global constraint, solutions with small differences between the estimated obstacle distances in neighboring columns are favored.

In [Badino et al., 2009], the authors perform an additional dynamic programming step to estimate a single height value for each building block claiming neighboring blocks to be of similar height. The building blocks are referred to as so called *Stixel* (vertical STicks projected to the piXEL domain) and the overall model as *Stixel world*.

In recent publications, several improvements were proposed for the Stixel world. Wedel et al. [2009] introduce a B-spline based ground plane estimation to overcome violations of the flat ground plane assumption. Benenson et al. [2011] propose an approach to estimate

Stixels without prior computation of a depth map. In [Pfeiffer and Franke, 2011], the authors introduce multilayer Stixel where multiple consecutively arranged obstacles can be estimated in each longitudinal grid column.

### 1.2.2.2 Alternative Sensors and Sensor Fusion

Besides vision based approaches, several alternative sensors are employed for free-space detection. Although not scope of this thesis, let us consider selected publications employing alternative sensors for free-space detection in the automotive domain for the sake of completeness.

Ultrasonic devices and Time of Flight cameras have a very low detection range and, thus, are used for applications concerning the direct vicinity of the car, such as parking assistance [Pohl et al., 2006, Satonaka et al., 2006, Scheunert et al., 2007].

In [Kaliyaperumal et al., 2001] and [Nikolova and Hero, 2000] the authors describe how to obtain a synthetic top-view image of the terrain by analyzing the backscatter of a front mounted radar sensor. Free-space is determined similar to the monocular vision based approaches which analyze birds-eye view projections.

Laser based approaches generally either fit a parameterized road model to the measured point cloud data [Kirchner and Ameling, 2000, Wijesoma et al., 2004, Zhang, 2010] or employ occupancy grids especially when fused with other sensors [Lacaze et al., 2002, Wellington et al., 2006, Urmson et al., 2008, Homm et al., 2010].

Homm et al. [2010] propose an efficient algorithm for computing occupancy grids from laser and radar data using the power of modern GPU's.

Wellington et al. [2006] propose an outstanding approach addressing off-road scenarios based on laser, camera and infrared sensors. The sensor data is stored in a 3d Cartesian occupancy grid which is fixed to the world system. For each cell, material features are determined describing point density, remission strength and color. Each vertical column of the grid is decomposed in cells representing ground, vegetation, obstacle and free-space via a Markov Chain (see Section 2.4). Based on this decomposition the columns are classified in traversable and non-traversable classes considering neighborhood relations in a Markov Random Field.

### 1.2.2.3 Detection of Low Obstacles Using Depth Maps

All the listed depth map based approaches for free-space detection in the automotive domain are designed to detect obstacles which are distinctive in height (e.g. vehicles, walls, guardrails and trees) and are not suited for the detection of low (or in-distinctive) objects, such as curbs or small traffic isles.

Although these objects generally form the actual delimiter of the drivable free-space, to our knowledge only a few depth map based approaches proved suitable for their detection.

In [Nedevschi et al., 2008] and [Nedevschi et al., 2009] the authors employ a deterministic Cartesian occupancy grid. The grid cells are labeled as road, traffic isle or obstacle by threshold based classification considering the highest point per cell and the point density. A quadratic ground surface model as well as temporal integration are employed to enhance precision and robustness of the labeling result. The free-space boundary is determined by a radial sweep along the image columns which yields the nearest obstacle in specified column directions. The detected samples are lined up yielding a polygonal boundary representation.

Kang and Chung [2011] employ a probabilistic polar occupancy grid and a stereo vision system with a large baseline of 0.5 meters. Following the idea of multi-volume occupancy grids [Dryanovski et al., 2010] the 3d points in each single cell are arranged in vertical clusters

by thresholding the vertical point distance. For each single cluster information about the height above the ground plane and about the point density as well as height discontinuities to the longitudinal neighboring cells are evaluated in a probabilistic way to decide whether the cell is occupied or not. However, although achieving promising results, the robustness of the current implementation suffers from the flat ground plane assumption and the lack of temporal integration.

### 1.3 Problem Statement and Achievements of the Thesis

The goal of this thesis is to provide a novel approach for the robust detection and modeling of street surfaces and boundaries from depth map sequences for automotive applications. In this context, we refer to the term *street surface* as a continuous representation of the vertical elevation of the street. The term *street boundary* means a continuous representation of the first geometric obstacles in viewing direction delimiting the traversable (drivable) free-space in front of the vehicle, where we consider height discontinuities of more than 10 cm as non-traversable (non-drivable).

Thereby, we address urban scenarios implying a low velocity of the ego-vehicle ( $< 50$  km/h) on the one hand as well as complex structures of the street boundary, including small traffic isles, complicated intersections, parking and moving cars, etc. on the other hand. In some literature, such complex structures are referred to as *unstructured environment*.

The key achievement of this thesis is an approach based on a parameterized environment model describing the relations and stochastic properties of the model components, sensor data and measurement outliers in a probabilistic way. Thus, the approach is able to cope with a high amount of measurement noise, which for example occurs in depth maps computed from small baseline stereo vision. A flexible, spline based model of the street surface allows for the detection and modeling of even low obstacles (curbs, traffic isles) and undulating street surfaces up to distances of 16 m to the sensor. The continuous representation of the environment model based on B-splines allows for an evaluation of the street elevation and boundary at an arbitrary position and direction in the region of interest.

### 1.4 Organization of the Thesis

In Chapter 2 we start by introducing the mathematical concepts and used notation which provide the theoretical background of this thesis. This includes the fundamental concepts of binocular stereo vision for depth map computation as well as techniques employed for classification and parameter estimation tasks. The chapter further describes the concept and properties of B-splines, which are used in this thesis to define a flexible model for the surface and boundary of the street.

Chapter 3 provides an exhaustive definition of this model as well as a detailed description of the total workflow developed to estimate the model parameters from recorded depth map sequences. This further comprises data acquisition, model initialization and self-validation of the estimated model.

The proposed approach is evaluated in Chapter 4 by characterizing the performance properties and limitations of the overall concept and of selected submodules based on synthetic and real world scenarios.

Finally, Chapter 5 gives a conclusion and presents an outlook on possible future research referring to the proposed work.



## Chapter 2

# Technical Background

In this chapter, we introduce the mathematical concepts used for the detection and modeling of street boundaries and surfaces from depth image sequences as well as the utilized notation.

First, we give an introduction to the general notation regarding geometric and probabilistic entities. The second section provides the utilized model assumptions and mathematics for depth map acquisition from a stereo vision sensor. Further, the chapter gives an overview on the employed techniques for the detection and reconstruction of the street surface and boundary comprising least squares parameter estimation, graphical models, logistic regression and the Expectation Maximization algorithm. Finally, the concept of B-splines, which form the basis of the street surface and street boundary model, is sketched.

To give the reader the opportunity to assess the relevance of the single sections due to his personal prior knowledge, we provide a short hint on the purpose of the respective technique in the overall concept as well as key assumptions in a box at the beginning of each section excluding Section 2.1.

## 2.1 Notation

### 2.1.1 Representation of Geometric Entities

Throughout this thesis, we use both Euclidean and homogeneous coordinates to represent geometric entities such as 2d and 3d points and 2d lines as well as their relations. This section introduces the notation utilized to distinguish between Euclidean and homogeneous representation and the entity itself.

#### 2.1.1.1 Euclidean and Homogeneous Representation of Point Entities

Assume a point entity in the  $n$ -dimensional space to be given which we denote by a calligraphic symbol  $\mathfrak{x}$ . The representation of this entity in Euclidean coordinates is denoted by a bold slanted symbol  $\mathbf{x} \in \mathbb{R}^n$ , whereas the homogeneous representation is given by a bold upright symbol  $\mathbf{x} \in \mathbb{R}^{n+1}$ . Euclidean and homogeneous representation of the same point entity are related by

$$\mathbf{x} = \lambda \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \quad (2.1)$$

with an arbitrary non-zero scale factor  $\lambda$ , and vice versa via normalization

$$\mathbf{x} = \frac{1}{[\mathbf{x}]_{n+1}} \begin{bmatrix} [\mathbf{x}]_1 \\ \vdots \\ [\mathbf{x}]_n \end{bmatrix}. \quad (2.2)$$

To simplify the distinction, we use capital letters for the representation of 3d point entities, e.g.  $\mathcal{X}$ ,  $\mathbf{X}$  and  $\mathbf{X}$ . Single scalar coordinate values are always denoted by small letters, e.g.  $\mathbf{X} = [x, y, z]^\top$ , or by the combination of vector (or matrix) symbol and index, e.g.  $\mathbf{X} = [[\mathbf{X}]_1, [\mathbf{X}]_2, [\mathbf{X}]_3]^\top$ .

### 2.1.1.2 Homogeneous Representation of 2D Lines

For the representation of 2d lines, let us consider the Hessian normal form which claims that each point  $\mathbf{x}$  on the line satisfies  $\mathbf{n}^\top \mathbf{x} = d$ , where  $\mathbf{n}$  denotes the normal of the line and  $d$  its distance to the origin. This motivates the homogeneous representation

$$\mathbf{l} = \lambda \begin{bmatrix} \mathbf{n} \\ -d \end{bmatrix}, \quad (2.3)$$

with  $\lambda \in \mathbb{R} \setminus \{0\}$  and  $|\mathbf{n}| = 1$ , allowing for the simple test of the point-line incidence relation

$$\mathbf{l}^\top \mathbf{x} = 0. \quad (2.4)$$

Further, the intersection point of two lines  $\mathbf{l}_1$  and  $\mathbf{l}_2$  is given by their cross product

$$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2 \quad (2.5)$$

$$= S(\mathbf{l}_1)\mathbf{l}_2 = S(\mathbf{l}_2)\mathbf{l}_1, \quad (2.6)$$

with the skew symmetric matrix

$$S(\mathbf{l}) = \begin{bmatrix} 0 & -[\mathbf{l}]_3 & [\mathbf{l}]_2 \\ [\mathbf{l}]_3 & 0 & -[\mathbf{l}]_1 \\ -[\mathbf{l}]_2 & [\mathbf{l}]_1 & 0 \end{bmatrix}. \quad (2.7)$$

For detailed information on relations between geometric entities including treatment of uncertainties we refer to [Meidow et al., 2009].

## 2.1.2 Representation of Probabilistic Entities

This section introduces the notation used to represent probabilistic entities including random variables as well as discrete probabilities and probability density functions. Assume  $a$  to be a discrete random variable which can take values  $a \in \{\dots, c_i, \dots\}$ , with  $i = 1, \dots, I$ .

We use the symbol  $P(a = c_i)$  to denote the probability that  $a$  takes the value  $c_i$ . Note, we will use this pedantic notation in spatial cases to avoid ambiguity, though in general we will refrain from explicit distinction of random variable and assigned value in order to simplify the notation. Thus, in case the interpretation is provided by the context, we simply write  $P(a)$  to denote the distribution of the random variable and  $P(c_i)$  to denote the evaluated probability for a certain value  $c_i$ . In an analogous manner, we employ this simplified notation for conditional and joint distributions of multiple random variables.

Finally, in case  $l$  represent a continuous random variable, we use the symbol  $\wp(l)$  to represent its probability density function (PDF).

## 2.2 Sensor Model for Stereo Camera Systems

Cameras are passive sensors that project the three dimensional environmental space on a two dimensional image plane. Intensities of light rays emitted or reflected from visible objects are measured by light sensitive material, such as photographic film (analog cameras) or electronic sensors (digital cameras). Exploitation of the exposure geometry and radiometric relations of images taken from two different viewpoints enables to estimate the 3d position of the ray emitting point making the camera system a 3d point cloud sensor.

In this section we summarize the mathematical formalisms and assumptions applied for modeling the geometric component of the imaging process (Section 2.2.1) as well as the reconstruction process (Section 2.2.2). Section 2.2.3 introduces the utilized notation and the usage of the terms *camera system* and *image coordinates* in the remainder of this thesis. Finally, Section 2.2.4 and Section 2.2.5 present the employed techniques for stereo matching and camera calibration.

We follow [McGlone et al., 2004, chapter 3.2] and [Hartley and Zisserman, 2003], which we refer to for more detailed information.

Within this thesis, we use the *pinhole camera model* to represent the image acquisition process [Hartley and Zisserman, 2003, chapter 6.1]. The utilized stereo camera system is assumed to be accurately calibrated, i.e. the interior and relative orientation of the cameras is known and rectified to the *normal case of the image pair* [McGlone et al., 2004, chapter 3.2.2.5]. Image coordinates are given in the rectified camera system, which we briefly refer to as *camera system* in the subsequent chapters. For dense stereo matching, we employ the method of Gehrig et al. [2009] which embodies an implementation of the Semi Global Matching approach presented in [Hirschmüller, 2005].

### 2.2.1 Image Acquisition Model

The general projection model for image acquisition comprises three euclidean, right handed coordinate systems (see Figure 2.1(a)). These are:

- A 3d world system  $S_w$  holding object coordinates in the environmental space.
- A local 3d camera system  $S_c$  describing object coordinates relative to the camera's orientation. The negative local  $^c z$  axis defines the camera's viewing direction, while  $^c y$  points upwards.
- A 2d sensor system  $S_s$  parallel to the camera system's  $xy$ -plane holding the measurable pixel coordinates of projected points.

Modeling the projection process means to mathematically describe the mapping of the world coordinates  $^w \mathbf{X}$  of a point entity  $\mathcal{X}$  to the sensor coordinates  $^s \mathbf{x}$  of the corresponding image point  $\chi$ .

For this purpose, we employ the geometry of a *pinhole camera* assuming all projection rays to pass through a single point called the *projection center*  $O$ .

#### 2.2.1.1 General Camera Model

We model the overall projection process  $^s \mathbf{x} = {}^s \mathcal{P}_w({}^w \mathbf{X})$  by four successive transformations

$$^s \mathbf{x} = {}^s \mathcal{P}_w({}^w \mathbf{X}) = {}^s \mathcal{P}_s \circ {}^s \mathcal{P}_c \circ {}^c \mathcal{P}_c \circ {}^c \mathcal{P}_w({}^w \mathbf{X}). \quad (2.8)$$

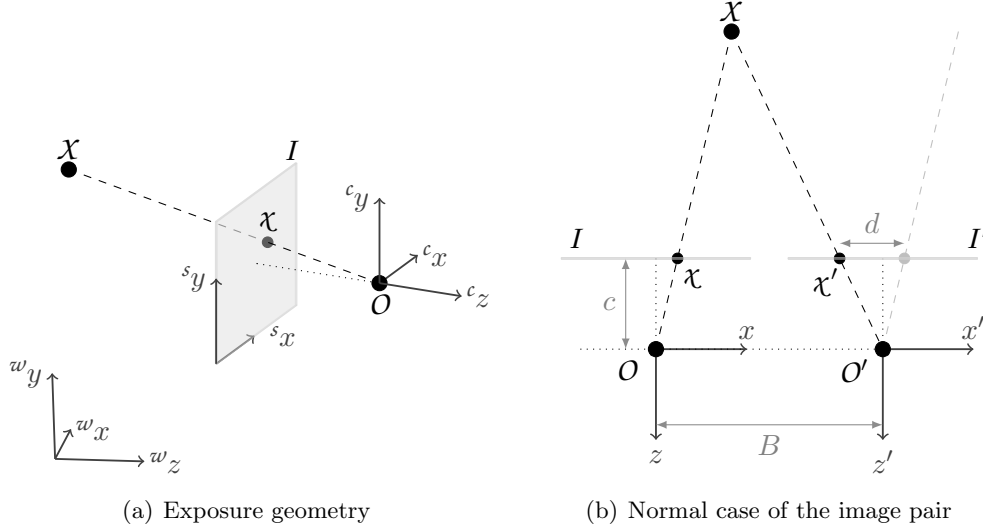


FIGURE 2.1: Utilized geometric model for a single camera (a) and a stereo camera pair (b). In (b), we use the shorthand notation  $a'$  to mark entities related to the second camera. Note, that the image plane, in contrast to the physical setup, is drawn in viewing direction to get an upright image.

These are:

1. The transformation  ${}^c\mathbf{X} = {}^c\mathcal{P}_w({}^w\mathbf{X})$  of  $X$  from the world system into the local camera system.
2. The perspective projection  ${}^c\mathbf{x} = {}^c\mathcal{P}_c({}^c\mathbf{X})$  of  $X$  onto the image plane  $I$ .
3. The affine transformation into the sensor system yielding the ideal image point  ${}^s\bar{\mathbf{x}} = {}^s\mathcal{P}_c({}^c\mathbf{x})$ .
4. A nonlinear transformation modeling straight line perturbing parts, such as design related influences like lens distortion, yielding the observable image point  ${}^s\mathbf{x} = {}^s\mathcal{P}_s({}^s\bar{\mathbf{x}})$ .

In homogeneous coordinates the combination of the first three transformations read as

$${}^s\bar{\mathbf{x}} = {}^s\mathbf{P}_w {}^w\mathbf{X} \quad (2.9)$$

$$= \underbrace{{}^s\mathbf{K}_c}{{}^s\mathcal{P}_c} \underbrace{{}^c\mathbf{K}_c [I_3 \mid \mathbf{0}]}{{}^c\mathcal{P}_c} \underbrace{{}^c\mathbf{M}_w}{{}^c\mathcal{P}_w} {}^w\mathbf{X}, \quad (2.10)$$

which defines the so called *projective camera model*.

**Exterior Orientation:** The relative orientation  ${}^c\mathcal{P}_w$  of world and camera system is governed by six *extrinsic parameters*. These are namely the three parameters defining the rotation matrix  $R$  and the euclidean coordinates of the projection center  $\mathbf{X}_O$ , which in combination define the transformation of the point coordinates from the world system into the camera system

$${}^c\mathbf{M}_w = \begin{bmatrix} R & -R\mathbf{X}_O \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (2.11)$$



**Interior Orientation:** The orientation of the image plane in the camera system as well as the relation of the sensor coordinate system to the camera system is given by five *intrinsic parameters*. The transformation

$${}^cK_c = \begin{bmatrix} c & 0 & 0 \\ 0 & c & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

is defined by the *principle distance*  $c$ . The principle distance embodies the distance of projection center and image plane in the ideal case.

The affine transformation into the sensor system

$${}^sK_c = \begin{bmatrix} 1 & s & x_0 \\ 0 & 1+m & y_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

is governed by the position of the principle point  $[x_0, y_0]^T$ , i.e. the foot point of the projection center onto the image plane, and by the potential scale difference  $m$  and shear  $s$  of the coordinate axes of  $S_s$ .

Additional intrinsic parameters are required to model the straight line perturbing parts  ${}^sP_s$  to compensate projection errors. The number of this parameters depends on the utilized camera and the intended level of precision. We refer to [McGlone et al., 2004, pp. 227 ff.] and [Hartley and Zisserman, 2003, chapter 7.4].

### 2.2.1.2 Ideal Camera Model

In case the intrinsic parameters are known, image coordinates can be reduced to  ${}^c\mathbf{x}$ . The projection model facilitates to

$${}^c\mathbf{x} = {}^cP_c \circ {}^cP_w({}^w\mathbf{X}) \quad (2.14)$$

$$= {}^cK_c R [I_3 \mid -\mathbf{X}_O] {}^w\mathbf{X} \quad (2.15)$$

$$= {}^cP_w {}^w\mathbf{X}, \quad (2.16)$$

which is called the *ideal camera model*.

Note that  ${}^c\mathbf{x}$  refers to a 2d point in the image plane rather than to a 3d point in the camera system.

## 2.2.2 Triangulation of 3D Points

Using pictures taken from two different viewpoints enables to reconstruct the world coordinates of a depicted point  $X$  from the point coordinates measured in both images  ${}^c\mathbf{x}$  and  ${}^{c'}\mathbf{x}$ . Assuming the interior and exterior orientations of both cameras to be known, the 3d-coordinates of  ${}^w\mathbf{X}$  can be reconstructed determining the intersection point of the respective projection rays. In case of calibrated cameras, the reconstruction problem facilitates to the matching problem of finding the corresponding point  ${}^{c'}\mathbf{x}$  to a given image point  ${}^c\mathbf{x}$ . We can break this down to a one dimensional search along image rows introducing further assumptions about the exposure geometry.

### 2.2.2.1 Normal Case of The Image Pair

We talk about a *normal case of the image pair* in case both cameras are ideal and identical and the relative orientation is given by a translation along the first camera's  $x$ -axis, i.e. both

cameras share the same image plane (see Figure 2.1(a)). The *relative orientation* of the cameras  $\bar{c}'\mathcal{M}_{\bar{c}} = \bar{c}'\mathcal{M}_w \circ {}^w\mathcal{M}_{\bar{c}}$  is represented by the coordinate transformation describing the motion of the first camera system into the second camera system

$$\bar{c}'\mathcal{M}_{\bar{c}} = \begin{bmatrix} 1 & 0 & 0 & -B \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.17)$$

where the *base length*  $B$  defines the distance between the projection centers.

Note that in the current and the subsequent paragraph we temporally use  $\bar{c}$  to distinguish the rectified camera system from the system  $c$  describing the general case. Following this, Section 2.2.3 presents the simplified final notation used in the remainder of this thesis.

### 2.2.2.2 Stereo Rectification

If the exterior orientations of the cameras is known, the general case can be reduced to the normal case by warping the images using homographies  $\bar{c}\mathbf{x} = \mathbf{H}c\mathbf{x}$  and  $\bar{c}'\mathbf{x} = \mathbf{H}'c'\mathbf{x}$  to fulfill the projection models

$$\bar{c}\mathbf{x} = \underbrace{\begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\bar{c}\mathbf{P}_w} \bar{c}\mathcal{M}_w {}^w\mathbf{X} \quad (2.18)$$

$$\bar{c}'\mathbf{x} = \underbrace{\begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\bar{c}'\mathbf{P}_w} \bar{c}'\mathcal{M}_{\bar{c}} \bar{c}\mathcal{M}_w {}^w\mathbf{X}. \quad (2.19)$$

The estimation of  $\mathbf{H}$  and  $\mathbf{H}'$  is called *stereo rectification* in the literature. Note that we will not discuss corresponding estimation techniques in this thesis.

### 2.2.3 Simplified Notation

In the remainder of this thesis, we assume the cameras to be accurately calibrated and rectified to the normal case and image points to be given by their coordinates in the rectified system  $\bar{c}\mathbf{x}$  rather than by their raw measurements  ${}^s\mathbf{x}$  or general case coordinates  $c\mathbf{x}$ .

Thus, to simplify the notation, we use the symbol  $c$  to address the system of the first rectified camera  $\bar{c}$  and use the shorthand notation  $\mathbf{x} = [u, v, 1]^T := \bar{c}\mathbf{x}$  for the rectified left image as well as  $\mathbf{x}' = [u', v', 1]^T := \bar{c}'\mathbf{x}$  for the rectified right image.

### 2.2.4 Stereo Matching

The correspondence problem in the normalized case facilitates to a matching problem along image rows. This means, given a certain interest point  $\mathbf{x} = [u, v]^T$  in the left image, the corresponding point in the right image  $\mathbf{x}' = [u + d, v]^T$  is defined by an image row offset  $d$  called *disparity*. The reconstructed 3d coordinates in the camera system of the first camera

are given by

$${}^c x = \frac{uB}{d} \quad (2.20)$$

$${}^c y = \frac{vB}{d} \quad (2.21)$$

$${}^c z = \frac{cB}{d}. \quad (2.22)$$

There exist various algorithms for stereo matching in the literature. For a detailed review on stereo matching algorithms we refer to [Brown et al., 2003] or more recent [Lazaros et al., 2008].

We can roughly subdivide existing stereo algorithms in two groups. On the one hand, *local stereo matching* methods which estimate correspondences for a set of interest points matching local statistics computed from the points neighborhood, e.g. [Zinner et al., 2008]. These local approaches bear advantages in computation time, local precision and robustness, but they fail in image regions with weak local statistics such as poorly textured regions. This results in sparse disparity estimates and may be insufficient for applications requiring a dense disparity map.

On the other hand, *global stereo matching* algorithms aim to compute a correspondence for nearly every pixel in the reference image. Prior information concerning global coherence in the estimated disparity field, e.g. smoothness or ordering constraints, is introduced to estimate correspondences even for poorly textured image regions. Local statistics and prior information are combined in a global cost function. Recent methods, e.g. [Taguchi et al., 2008], [Papadakis and Caselles, 2010] or [Pock et al., 2010], differ in modeling this cost function and the chosen optimization technique.

However, targeting a global coherent solution implicates significant increase in computation time. This brings rise to so called *scanline* algorithms which demand coherence for each image row independently, e.g. [Birchfield and Tomasi, 1996] or [Meerbergen et al., 2002]. Unfortunately, these methods suffer from streaking artifacts due to the unmodeled scanline perpendicular correlations. This means that the estimated disparities tend to be smooth in scanline direction but inconsistent in perpendicular direction.

The *Semi-Global Matching* (SGM) approach of Hirschmüller [2005] yields a compromise between coherence level and computational effort. The cost function is optimized along a specific number of scanlines uniformly rotated to each other. The work of Gehrig et al. [2009] proposes a implementation of an extended SGM approach, yielding a real-time capable hardware solution on a field-programmable gate array (FPGA) platform.

A performance study on SGM in comparison to other matching algorithms can be found in [Steingrube et al., 2009] or on the middlebury evaluation platform <sup>1</sup>. Figure 2.2(b) illustrates an example of a disparity map computed with the Semi-Global Matching algorithm.

### 2.2.5 Camera Calibration

The calibration of the stereo camera system comprises the estimation of the intrinsic parameters for both cameras defining  ${}^s\mathcal{P}_s$ ,  ${}^{s'}\mathcal{P}_{s'}$  and  ${}^s\mathcal{P}_c$ ,  ${}^{s'}\mathcal{P}_{c'}$ , the estimation of the relative orientation  ${}^c\mathcal{M}_{c'}$ , as well as the rectification step computing  $\mathbf{H}$  and  $\mathbf{H}'$ .

We use a calibration system similar to the tool presented by Bouguet [2007]. A planar calibration rig showing a checkerboard pattern of known size is captured from different positions, as depicted in Figure 2.2(a). The unknown calibration parameters are estimated from automatically detected correspondences in the respective checkerboard projections.

---

<sup>1</sup><http://middlebury.edu/stereo/>

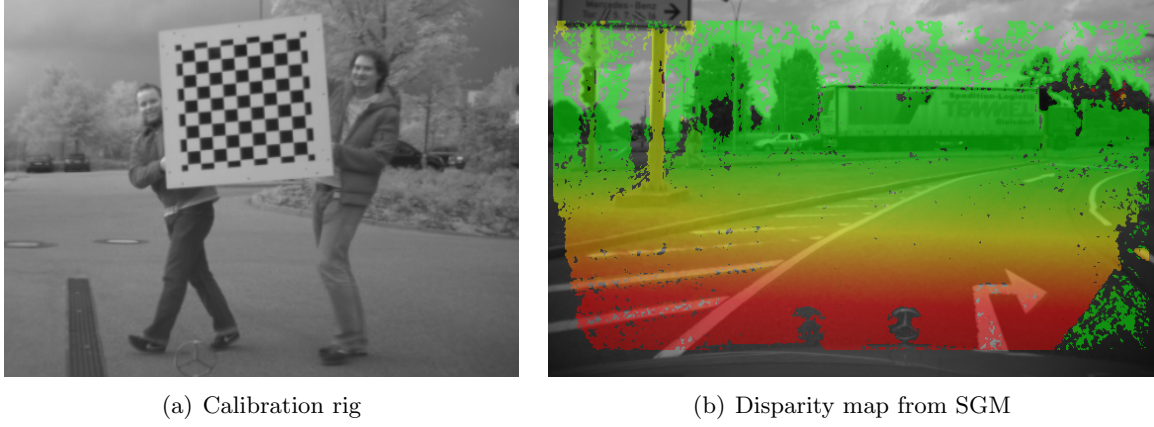


FIGURE 2.2: Planar rig with checkerboard pattern used for camera calibration (a) and exemplary disparity map computed with the SGM algorithm (b). Red color encodes large disparities (near distances), while green color encodes small disparity values (far distances).

## 2.3 Parameter Estimation

In this section we sketch the concept of parameter estimation from uncertain measurements in an overdetermined system. The underlying model which characterizes the relation and stochastic properties of measurements and unknown parameters is introduced in Section 2.3.1. The calculus for parameter estimation in linear models is presented in Section 2.3.2 comprising maximum likelihood estimation as well as the consideration of additional prior information to achieve the maximum a posteriori estimate. Section 2.3.3 describes the utilized procedure to deal with non-linear models. For additional information, it is referred to [McGlone et al., 2004, chapter 2.2.4] or [Koch, 2000, chapter 4], which provide the basis for this section.

The parameters governing the street surface and boundary in the proposed approach are estimated in a weighted least squares sense assuming observations and unknown parameters to be Gaussian distributed. Bayesian estimation is performed by introducing temporal and spatial prior information via fictitious observations. Further, we use the Gauss-Newton method to deal with the non-linearity of the estimation problem.

### 2.3.1 Gauss-Markov Model

Assume a vector of uncertain observations  $\mathbf{z} \in \mathbb{R}^N$  to be given which embodies a random sample drawn from the Gaussian distribution  $\mathbf{z} \sim \mathcal{N}(\tilde{\mathbf{z}}, \Sigma_{zz})$ . Note, that we temporarily use a detailed notation for a better understanding that symbolically distinguishes between the true value  $\tilde{\mathbf{z}}$ , the stochastic variable  $\mathbf{z}$  and the actually observed sample value  $\mathbf{z}$ .

From the observations, a vector of unknown parameters  $\mathbf{p} \in \mathbb{R}^U$  shall be estimated. Further, the nonlinear *functional model*  $\mathbf{f} : \mathbb{R}^U \rightarrow \mathbb{R}^N$  describes the relation of the real observation and parameter values  $\tilde{\mathbf{z}}$  and  $\tilde{\mathbf{p}}$  such that

$$\tilde{\mathbf{z}} = \mathbf{f}(\tilde{\mathbf{p}}). \quad (2.23)$$

Since the observations are uncertain, the equality holds merely for the expected value

$$E(\mathbf{z}) = \mathbf{f}(\tilde{\mathbf{p}}) \quad (2.24)$$

or respectively by adding a *residual* vector  $\mathbf{v}$  modeling the negative measurement deviations

$$\mathbf{z} + \mathbf{v} = \mathbf{f}(\tilde{\mathbf{p}}). \quad (2.25)$$

As stated above, we assume that the mean of the observations corresponds to the true observations, i.e. the distribution of the residuals is given by the Gaussian distribution

$$\tilde{\mathbf{z}} - \mathbf{z} = \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Sigma_{zz}), \quad (2.26)$$

which we refer to as *stochastic model*.

The combination of the functional model (2.23) and the Gaussian stochastic model (2.26) forms the so called *Gauss-Markov model* [see e.g. Koch, 1997, chapter 3.2].

In order to keep the notation uncluttered, we refrain from an explicit symbolic distinction of stochastic variables and their sample values in the remainder of this thesis and use the simplified notation

$$\mathbf{z} + \mathbf{v} = \mathbf{f}(\mathbf{p}), \quad \mathbf{v} \sim \Sigma_{zz} \quad (2.27)$$

to represent estimation tasks via the Gauss-Markov model.

### 2.3.2 Least Squares Estimation in Linear Models

Let us now consider how to determine estimates  $\hat{\mathbf{p}}$  for the true parameters  $\tilde{\mathbf{p}}$  based on the introduced model. We start with the case of a linear<sup>2</sup> functional model, i.e. the function  $\mathbf{f}(\mathbf{p}) = \mathbf{A}\mathbf{p}$  is defined by the so called *design matrix*  $\mathbf{A} \in \mathbb{R}^{N \times U}$ . The treatment of non-linear models is discussed subsequently in Section 2.3.3.

Section 2.3.2.1 describes the principle estimation procedure. Section 2.3.2.2 discusses the consideration of additional prior information.

#### 2.3.2.1 Maximum Likelihood (ML) Estimation

From a probabilistic point of view, the model represents the likelihood of the unknown parameters

$$\wp(\mathbf{z} | \mathbf{p}) = G(\mathbf{z} | \mathbf{A}\mathbf{p}, \Sigma_{zz}), \quad (2.28)$$

where  $G$  represents the Gaussian PDF. The task is to find the parameter assignment  $\hat{\mathbf{p}}$  which is most likely, i.e. which yields the maximum argument for the likelihood function

$$\hat{\mathbf{p}} = \underset{\mathbf{p}}{\operatorname{argmax}} [\wp(\mathbf{z} | \mathbf{p})], \quad (2.29)$$

referred to as *maximum likelihood estimate*.

Considering the negative logarithm of the likelihood function, the maximum likelihood estimate is given by the minimum argument

$$\hat{\mathbf{p}} = \underset{\mathbf{p}}{\operatorname{argmin}} \left[ (\mathbf{A}\mathbf{p} - \mathbf{z})^\top \Sigma_{zz}^{-1} (\mathbf{A}\mathbf{p} - \mathbf{z}) \right]. \quad (2.30)$$

The determination of the minimum argument represents a weighted least squares estimation task. A detailed derivation is given in [Koch, 1997, pp. 174 ff.].

---

<sup>2</sup>W.l.o.g., the linear model may be replaced by an affine model by adding a constant term.

As a necessary criterion, the first derivative with respect to the unknown parameters needs to vanish at the optimum, i.e.

$$A^T \Sigma_{zz}^{-1} A \hat{\mathbf{p}} - A^T \Sigma_{zz}^{-1} \mathbf{z} \stackrel{!}{=} 0. \quad (2.31)$$

This yields  $\hat{\mathbf{p}}$  to be the solution of the linear *normal equation system*

$$\underbrace{A^T \Sigma_{zz}^{-1} A}_{\mathbf{N}} \hat{\mathbf{p}} = \underbrace{A^T \Sigma_{zz}^{-1} \mathbf{z}}_{\mathbf{h}} \quad (2.32)$$

with the *normal equation matrix*  $\mathbf{N}$  and the *right hand side of the normal equation*  $\mathbf{h}$ .

The theoretical precision of the estimated parameters defined by this model is given by

$$\Sigma_{\hat{\mathbf{p}}\hat{\mathbf{p}}} = \mathbf{N}^{-1}. \quad (2.33)$$

### 2.3.2.2 Maximum a Posteriori (MAP) Estimation

In addition to the information provided by the observations via the likelihood (2.28), prior knowledge  $\wp(\mathbf{p})$  on the unknown parameters may be available, e.g. derived from temporal or spatial smoothness constraints. The first paragraph of this section sketches the consideration of such prior knowledge in the estimation process yielding the *maximum a posteriori estimate* of the unknown parameters. The second paragraph describes how to treat prior information which is given by samples of a function of the unknown parameters rather than about the parameters themselves.

For detailed information we refer to [Koch, 2000, chapter 4.2].

**MAP Estimation in the Gauss-Markov Model** The maximum a posteriori estimate  $\hat{\mathbf{p}}_M$  is defined as the maximum argument of the posterior distribution of the unknown parameters. The posterior distribution combines the likelihood (2.28) and prior information via the Bayes theorem

$$\wp(\mathbf{p} | \mathbf{z}) = \frac{\wp(\mathbf{z} | \mathbf{p}) \wp(\mathbf{p})}{\wp(\mathbf{z})}, \quad (2.34)$$

with  $\wp(\mathbf{z}) = \int \wp(\mathbf{z} | \mathbf{p}) \wp(\mathbf{p}) d\mathbf{p}$ . Considering that the denominator does not depend on  $\mathbf{p}$ , the maximum a posteriori estimate is given by

$$\hat{\mathbf{p}}_M = \underset{\mathbf{p}}{\operatorname{argmax}} \wp(\mathbf{p} | \mathbf{z}) \quad (2.35)$$

$$= \underset{\mathbf{p}}{\operatorname{argmax}} [\wp(\mathbf{z} | \mathbf{p}) \wp(\mathbf{p})] \quad (2.36)$$

or, if defined in terms of the negative logarithm, by the minimum argument

$$\hat{\mathbf{p}}_M = \underset{\mathbf{p}}{\operatorname{argmin}} [-\ln \wp(\mathbf{z} | \mathbf{p}) - \ln \wp(\mathbf{p})]. \quad (2.37)$$

Assume uncertain prior knowledge on the unknown parameters is given by a Gaussian distribution

$$\wp(\mathbf{p}) = G(\mathbf{p} | \mathbf{p}_0, \Sigma_{p_0 p_0}), \quad (2.38)$$

i.e. the expected value of  $\mathbf{p}$  is a priori assumed to be given by a specific value  $\mathbf{p}_0$  and its precision by the covariance matrix  $\Sigma_{p_0 p_0}$ . The maximum a posteriori estimate can be derived via the procedure presented in the previous section using the extended Gauss-Markov model

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{p}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{v} \\ \mathbf{v}_{p_0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{I}_U \end{bmatrix} \hat{\mathbf{p}}_M, \quad \begin{bmatrix} \mathbf{v} \\ \mathbf{v}_{p_0} \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma_{zz} & \mathbf{0} \\ \mathbf{0} & \Sigma_{p_0 p_0} \end{bmatrix}\right), \quad (2.39)$$

see [Koch, 2000, chapter 4.2.6]. Thus,  $\hat{\mathbf{p}}_M$  is given by the solution of the extended normal equation system

$$\left(\mathbf{A}^\top \Sigma_{zz}^{-1} \mathbf{A} + \Sigma_{p_0 p_0}^{-1}\right) \hat{\mathbf{p}}_M = \mathbf{A}^\top \Sigma_{zz}^{-1} \mathbf{z} + \Sigma_{p_0 p_0}^{-1} \mathbf{p}_0. \quad (2.40)$$

**Prior Information from fictitious observations:** In this thesis, we exploit prior knowledge represented by uncertain information about a function  $\mathbf{k}(\mathbf{p})$  of the parameters rather than about the parameters themselves. The information is given by an additional set of observations  $\mathbf{z}_k$  which are independent to  $\mathbf{z}$  and connected to the unknown parameters  $\mathbf{p}$  via a linear Gaussian model

$$\mathbf{z}_k + \mathbf{v}_k = \mathbf{k}(\mathbf{p}) := \mathbf{A}_k \mathbf{p}, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \Sigma_{kk}). \quad (2.41)$$

This model provides prior information about the unknown parameters considering the maximum likelihood estimate

$$\hat{\mathbf{p}}_k = \underset{\mathbf{p}_k}{\operatorname{argmax}} [\wp(\mathbf{z}_k | \mathbf{p}_k)] \quad (2.42)$$

obtained from (2.32). This yields the prior distribution

$$\wp(\mathbf{p}) = G\left(\mathbf{p} | \hat{\mathbf{p}}_k, \left(\mathbf{A}_k^\top \Sigma_{kk}^{-1} \mathbf{A}_k\right)^{-1}\right). \quad (2.43)$$

By substituting  $\Sigma_{p_0 p_0}^{-1} = \mathbf{A}_k^\top \Sigma_{kk}^{-1} \mathbf{A}_k$  and  $\mathbf{A}_k \mathbf{p}_0 = \mathbf{z}_k$  in (2.40), the MAP estimate is given by the solution of the extended normal equation system

$$\underbrace{\left(\mathbf{A}^\top \Sigma_{zz}^{-1} \mathbf{A} + \mathbf{A}_k^\top \Sigma_{kk}^{-1} \mathbf{A}_k\right)}_{\mathbf{N}} \hat{\mathbf{p}}_M = \underbrace{\mathbf{A}^\top \Sigma_{zz}^{-1} \mathbf{z} + \mathbf{A}_k^\top \Sigma_{kk}^{-1} \mathbf{z}_k}_{\mathbf{h}}. \quad (2.44)$$

Note that this corresponds to the solution directly obtained from the combined model

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{z}_k \end{bmatrix} + \begin{bmatrix} \mathbf{v} \\ \mathbf{v}_k \end{bmatrix} = \begin{bmatrix} \mathbf{A} \\ \mathbf{A}_k \end{bmatrix} \mathbf{p}_M, \quad \begin{bmatrix} \mathbf{v} \\ \mathbf{v}_k \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \Sigma_{zz} & \mathbf{0} \\ \mathbf{0} & \Sigma_{kk} \end{bmatrix}\right) \quad (2.45)$$

treating actual observations  $\mathbf{z}$  and prior information  $\mathbf{z}_k$  as two sets of independently distributed observations. Thus, there is no need for an explicit computation of  $\hat{\mathbf{p}}_k$ .

Throughout this thesis, we utilize the model (2.45) to formulate MAP estimation tasks. We use the term *fictitious observations* for the observations  $\mathbf{z}_k$  corresponding to the prior in order to distinguish them from the actual observations  $\mathbf{z}$ .

### 2.3.3 Least Squares Estimation in Non-Linear Models

In case  $\mathbf{f}$  (or  $\mathbf{k}$ ) is a non-linear function we employ the Gauss-Newton algorithm [see e.g. Alt, 2002] treating the non-linear problem as a sequence of linear problems that are solved using the method presented in the last section. For simplicity, we explain the procedure by only considering the function  $\mathbf{f}$ . Non linear MAP estimation is treated analogously.

The non-linear function  $\mathbf{f}$  is approximated using first degree Taylor expansion at a linearization point  $\mathbf{p}^{(0)}$  neglecting higher order terms

$$\mathbf{f}(\mathbf{p}) = \mathbf{f}(\mathbf{p}^{(0)} + \Delta\mathbf{p}) \approx \mathbf{f}(\mathbf{p}^{(0)}) + \underbrace{\frac{\partial \mathbf{f}(\mathbf{p})}{\partial \mathbf{p}} \bigg|_{\mathbf{p}=\mathbf{p}^{(0)}}}_{\mathbf{A}^{(0)}} \Delta\mathbf{p} \quad (2.46)$$

to obtain the *linearized Gauss-Markov model*

$$\Delta\mathbf{z} + \mathbf{v} = \mathbf{A}^{(0)}\Delta\mathbf{p}, \quad \mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Sigma_{zz}), \quad (2.47)$$

with the reduced observations  $\Delta\mathbf{z} := \mathbf{z} - \mathbf{f}(\mathbf{p}^{(0)})$ .

The unknown parameters  $\hat{\mathbf{p}}$  are estimated in an iterative manner. Starting with approximate values  $\hat{\mathbf{p}}^{(0)} := \mathbf{p}^{(0)}$  and  $\nu := 0$ , we repeat the steps:

1. Linearize  $\mathbf{f}$  at the current parameter estimate  $\hat{\mathbf{p}}^{(\nu)}$  to achieve  $\mathbf{A}^{(\nu)}$ , using (2.46);
2. Compute the reduced observations  $\Delta\mathbf{z}^{(\nu)} = \mathbf{z} - \mathbf{f}(\hat{\mathbf{p}}^{(\nu)})$ ;
3. Estimate the correction  $\widehat{\Delta\mathbf{p}}^{(\nu)}$  by solving  $\mathbf{N}^{(\nu)}\widehat{\Delta\mathbf{p}}^{(\nu)} = \mathbf{h}^{(\nu)}$ ;
4. Update the parameter estimate  $\hat{\mathbf{p}}^{(\nu+1)} = \hat{\mathbf{p}}^{(\nu)} + \widehat{\Delta\mathbf{p}}^{(\nu)}$ ;
5. Increase iteration counter  $\nu := \nu + 1$ ;

until a termination criterion is fulfilled, e.g. until the maximal relative update

$$\max_i \left( \frac{|\widehat{\Delta p}_i^{(\nu)}|}{\sqrt{[\Sigma_{\widehat{\mathbf{p}}\widehat{\mathbf{p}}}^{(\nu)}]_{ii}}} \right), \quad i = 1, \dots, U, \quad (2.48)$$

falls below a certain threshold.

In case  $\widehat{\nu}$  denotes the last iteration of this procedure, the final parameter estimate is given by  $\hat{\mathbf{p}}^{(\widehat{\nu})}$  and its precision by

$$\Sigma_{\widehat{\mathbf{p}}\widehat{\mathbf{p}}}^{(\widehat{\nu})} = \mathbf{N}^{(\widehat{\nu})^{-1}}. \quad (2.49)$$

## 2.4 Probabilistic Graphical Models

*Probabilistic graphical models*, also called *graphical models*, are diagrammatic representations used to model the conditional independence properties of a set of random variables. The representation is usually given by a graph whose nodes represent the random variables while its edges encode the dependency relation between them. The decision whether using a directed or undirected graph for the representation distinguishes the two major groups of graphical models given by *Bayesian Networks*, also known as *directed* graphical models, and *Markov*



*Random Fields* (MRFs), also known as *undirected* graphical models [see Kindermann and Snell, 1980]. In this thesis, we restrict to the undirected version.

The remainder of this chapter provides a brief introduction of MRFs including their properties and variations utilized in this thesis (Section 2.4.1). Further, we mention the techniques employed for inference which denotes the determination of the posterior probabilities of the assignment of the random variables based on the assumptions defined by the MRF's structure (Section 2.4.2). The introduction is based on [Bishop, 2006, chapter 8] and [Hastie et al., 2009, chapter 17.2], which we refer to for more detailed information.

We employ graphical models based on Conditional Random Fields (CRFs) for two tasks. First, models based on an acyclic chain structure are used to capture the elevation at discrete horizontal positions in front of the vehicle. The second task embodies the determination of a probabilistic statement about which of these positions belong to the drivable street region and which do not. The horizontal positions are arranged in a 4-connected horizontal grid, and we employ a CRF aligned to this grid structure in order to model the respective probabilistic properties the statement depends on. For inference, we employ max-sum algorithm for the chains and Loopy Belief Propagation for the grid.

### 2.4.1 Markov Random Fields

A Markov Random Field is represented by an undirected graph which is defined by a set of nodes  $i \in \mathcal{I}$ , each connected to a discrete random variable  $l_i$ , as well as a set of edges representing the neighborhood structure within the graph. Two nodes  $i, j \in \mathcal{I}$  are said to be neighbored if an edge between both corresponding nodes exists. We use the symbol  $\mathcal{N}_i$  to describe the set of nodes which are neighbored to node  $i$ . The vector holding the full set of random variables is denoted by  $\mathbf{l} = [\dots, l_i, \dots]^T$ . Figure 2.3(a) depicts an example for a graphical model of a MRF with four variables. In the following, we will use the term node to additionally address the respective connected random variable.

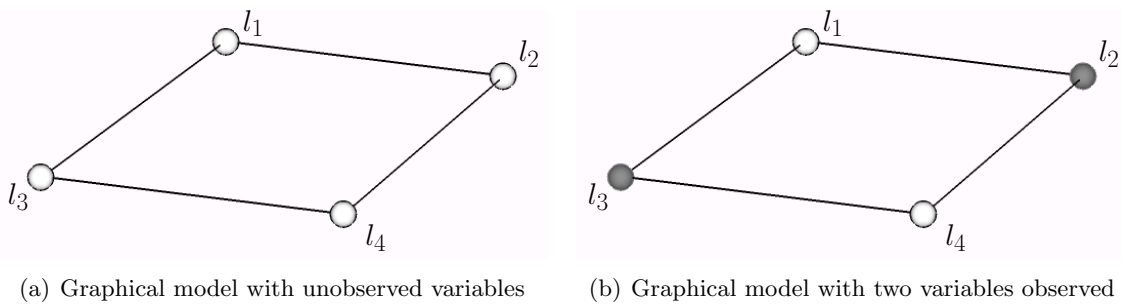


FIGURE 2.3: Graphical model for an MRF representing the conditional independence properties of four variables  $l_1, \dots, l_4$ . In Figure (a) all variables are treated as unobserved represented by bright shaded nodes. In Figure (b) the variables  $l_2$  and  $l_3$  are treated as observed, i.e. their specific value is known, which is represented by dark shaded nodes. Via the global Markov property, the graph structure implies that each variable is conditional independent of its diagonal opponent given the value of its connected neighbors since there exists no direct link between them. For example, considering the observation state in (b), the class assignment of  $l_1$  is conditional independent of the assignment of  $l_4$  given the value of  $l_2$  and  $l_3$ . This is because there exists no path between  $l_1$  and  $l_4$  that is not ‘blocked’ by an observed node. This means  $P(l_1 \mid l_2, l_3, l_4) = P(l_1 \mid l_2, l_3)$ .

### 2.4.1.1 Conditional Independence Properties

The graph structure implies the conditional independence properties of the random variables via the so called *global Markov property*. The property states that given its neighbors a node is conditionally independent to all nodes of the graph not included in its neighborhood, i.e.  $P(l_i | \mathbf{l}_{\setminus i}) = P(l_i | \mathcal{N}_i)$ ,  $\forall i \in \mathcal{I}$ . The symbol  $\mathbf{l}_{\setminus i}$  denotes the vector  $\mathbf{l}$  without element  $i$ . See Figure 2.3(b) for an illustration of this property.

This insight allows for a simplification of the joint probability  $P(\mathbf{l})$  via the *Hammersley-Clifford* theorem [Clifford, 1990]. The theorem states that the set of factorizations of  $P(\mathbf{l})$  meeting the requirements defined by the Markov property for a specific graph equals the set of distributions that factorize into positive functions  $\Psi_C(\mathbf{l}_C)$  over the maximal cliques  $C \in \mathcal{C}$  of the graph

$$P(\mathbf{l}) = \frac{1}{\zeta} \prod_{C \in \mathcal{C}} \Psi_C(\mathbf{l}_C). \quad (2.50)$$

A maximal clique is a fully connected subset of nodes which are unable to add an additional node without losing the property of full connectivity. The entity  $\zeta$  denotes the normalization constant

$$\zeta = \sum_{\mathbf{l} \in \mathcal{L}} \prod_{C \in \mathcal{C}} \Psi_C(\mathbf{l}_C), \quad (2.51)$$

which is known as *partition function*. The vector  $\mathbf{l}_C$  holds the subset of random variables associated to the respective clique. The functions  $\Psi_C(\mathbf{l}_C)$  are referred to as *potential functions*.

For example, the graphical model presented in Figure 2.3(a) implies the factorization  $P(l_1, l_2, l_3, l_4) \sim \Psi_{12}(l_1, l_2) \Psi_{23}(l_2, l_3) \Psi_{34}(l_3, l_4) \Psi_{41}(l_4, l_1)$ . Note that this factorization should be understood as a kind of upper limit of the possible dependency of the variables, i.e. the potential functions must not depend on variables which are not part of the respective cliques, but they may depend only on a subset of the clique or further factorize into products of subsets.

### 2.4.1.2 Utilized Variants of Markov Random Fields

There exist several special variants of MRFs which mainly differ in the complexity of the underlying graph structure or the definition of the potential functions. In this thesis, we utilize so called *Conditional Random Fields* (CRFs) introduced by Lafferty et al. [2001]. CRFs explicitly model the dependency of the random variables  $\mathbf{l}$  to a set of observations  $\mathbf{z}$ , such as pixel color values in an image segmentation task, i.e. the task of assigning each pixel to one out of a set of specific image segments. This dependency is modeled explicitly by defining the potential functions to depend on both  $\mathbf{l}$  and  $\mathbf{z}$

$$P(\mathbf{l} | \mathbf{z}) = \frac{1}{\zeta} \prod_{C \in \mathcal{C}} \Psi_C(\mathbf{l}_C | \mathbf{z}). \quad (2.52)$$

More precisely, following [Lafferty et al., 2001] the set  $\{\mathbf{l}, \mathbf{z}\}$  represents a CRF in case the variables  $\mathbf{l}$  obey the Markov property with respect to the graph when conditioned on  $\mathbf{z}$ . Note that there exists some inconsistency in the exact definition of CRFs in different communities.

Graphically the observations are introduced as additional nodes in the graph structure which are treated as observed.

The structure of the CRFs employed in the approach proposed in this thesis are restricted to two cases. The first of them are *linear-chain CRFs*, whereby the name follows from the

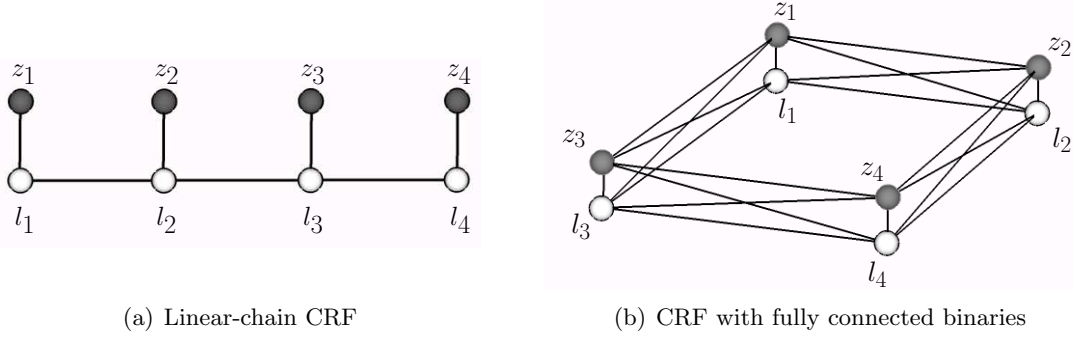


FIGURE 2.4: Examples for the graph structures of CRFs relevant for this thesis. Figure (a) illustrates the first case in which the nodes for the class labels are lined up in a chain structure. Figure (b) demonstrates the second case in which the nodes are arranged in a 4-connected grid. For simplicity only a  $2 \times 2$  subset is represented. In both cases, the upper layer of observed nodes represents the observations  $z_i$ , while the lower layer of unobserved nodes represents the class labels  $l_i$ . Note that in comparison to (a), the nodes for observations and class labels of grid neighbors in (b) are fully connected and, thus, the binary potential functions may also depend on the respective observations.

graph structure since all nodes  $l_i$  are lined up in one row, as illustrated in Figure 2.4(a). This structure implies the factorization of the joint probability in unary and binary terms

$$P(\mathbf{l} \mid \mathbf{z}) = \frac{1}{\zeta} \prod_{i=1}^I \Psi(l_i \mid z_i) \prod_{i=2}^I \Phi(l_i, l_{i-1}), \quad (2.53)$$

where  $\Psi$  defines the unary potentials and  $\Phi$  the binary potential functions. As described in the subsequent section, the simple structure of these models allows for exact inference. Note that this roughly corresponds to the concept of Markov chains [see Bishop, 2006, chapter 13], which are usually represented via directed graphical models.

The graph structure of the second case is governed by a planar graph with 4-connected neighborhood, see Figure 2.4(b). The nodes of labels and observations of direct grid neighbors are fully connected and, thus, form the cliques of the graph. This structure implies that the single potential functions depend on no more than the observations  $z_i, z_j$  and class labels  $l_i, l_j$  of direct grid neighbors  $i$  and  $j$ . In order to separately model the individual local class decisions based on the observations and the influence of the neighborhood relation, we will assume the joint probability to decompose into a product of unary and binary terms

$$P(\mathbf{l} \mid \mathbf{z}) = \frac{1}{\zeta} \prod_{i \in \mathcal{I}} \Psi(l_i \mid z_i) \prod_{(i,j) \in \mathcal{N}_4} \Phi(l_i, l_j \mid z_i, z_j). \quad (2.54)$$

The symbol  $\mathcal{N}_4$  denotes the set of index tuples  $(i, j)$  which define the neighbors out of the 4-neighborhood of the grid.

### 2.4.2 Inference in Markov Random Fields

Inference in MRFs denotes the task to determine the posterior probabilities for the class assignments of all nodes, or a subset of nodes, given the conditional independence assumptions defined by the MRFs structure. Thereby, a subset of the nodes may be clamped to observed values, such as the observations  $\mathbf{z}$  in a CRF.

With respect to the CRFs employed in this thesis, we restrict to the two tasks of searching for:

- (a) The most probable assignment of all random variables en bloc

$$\hat{\mathbf{l}} = \underset{\mathbf{l}}{\operatorname{argmax}} P(\mathbf{l} | \mathbf{z}). \quad (2.55)$$

- (b) The evaluation of the marginals

$$P(l_i | \mathbf{z}) = \sum_{\mathbf{l}_{\setminus l_i}} P(\mathbf{l} | \mathbf{z}) \quad (2.56)$$

for all  $l_i$ , with  $i \in \mathcal{I}$ . The short notation under the sum means that the  $i$ th element is omitted when summing over the possible assignments of  $\mathbf{l}$ .

The naive method of evaluating all possible assignments is computational expensive or even intractable also for small sized problems due to the sheer number of combinations to be evaluated. There exist several methods for inference exploiting the simplifying assumptions given by the CRF's structure.

#### 2.4.2.1 Exact Inference in Undirected Acyclic Graphs

For tackling problem (a) in acyclic graphs (chains in our case), we use the *max-sum* algorithm [Bishop, 2006, chapter 8.4.5]. This method is based on a message passing schedule, propagating probability information from the graph's leave nodes to a selected root node. The method keeps track of the respective node assignments that gave rise to the most probable assignment of the root node. This tracked set of assignments represents the desired most probable assignment of all random variables.

There exists a similar method, which encounters problem (b) called *sum-product* algorithm [see MacKay, 2003, pp. 334-340]. The method also relies on a message passing technique but uses the messages for marginalization rather than for maximization.

Both message passing techniques are also known as *belief propagation* in literature.

#### 2.4.2.2 Approximate Inference in Undirected Graphs with Cycles

There exist techniques for exact inference in graphs with cycles, such as *Graph-Cuts* [Boykov et al., 2001]. However, they are restricted to certain model classes, e.g. the  $l_i$  need to be binary variables and the potential functions need to be submodular [see Kolmogorov and Zabini, 2004], in order to guarantee a global optimal solution making them unsuitable for our applications.

To tackle problem (b) in cyclic graphs we employ *loopy belief propagation* [Frey and MacKay, 1998]. The idea is to apply the sum-product algorithm as for acyclic graphs and sequentially pass messages to the neighbors until convergence or a maximum number of iterations is reached. Note that this technique also yields approximative results but bears the advantage to be highly parallelizable and, thus, is suitable for real-time applications.

## 2.5 Logistic Regression

In the later sections we encounter the problem to estimate a *classification model* from sample-sets of two predefined classes in a one dimensional feature space. Put simply, a set of samples is given which describes the appearance of two classes with respect to a single real-valued feature. The task is to estimate a model that given an arbitrary feature value, i.e. an arbitrary position in the addressed feature space, provides probabilities for both classes to be

assigned. Further, the model shall provide an explicit formulation of the *decision boundary*. The decision boundary partitions the feature space into regions differing in which of the both classes is more probable, i.e. both classes are equiprobable on the decision boundary. For the estimation of the classification model, we employ logistic regression, an established probabilistic classifier.

The fundamental concept of logistic regression is summarized in Section 2.5.1 and Section 2.5.2, which specify the task and the logistic regression model. Section 2.5.3 describes the estimation of the model focusing on the determination of the decision boundary and its accuracy. Further, we show how we tread uncertainties in the sample positions. Finally, Section 2.5.4 depicts the modifications of the procedure for the case that the class assignments of the input samples are non-deterministic, i.e. the information is provided by the probability of a class-membership instead of a definite membership statement.

For detailed information, we refer to [Hastie et al., 2009, chapter 4.4] and [Bishop, 2006, chapter 4.3], which provide the basis for this section.

We employ logistic regression to estimate local decision boundaries along discrete viewing directions in the horizontal plane, i.e. to estimate the local separation of ‘street regions’ from ‘non-street regions’. We assume this decision boundaries to be adequate sample measurements of the horizontal free-space boundary.

### 2.5.1 Logistic Regression Task

Assume a number of samples  $j = 1, \dots, J$  was drawn from a set of classes, and their position with respect to a feature space  $y$  is known. This samples form the so called *training set*  $\{\mathbf{y} = [\dots, y_j, \dots]^T, \mathbf{t} = [\dots, t_j, \dots]^T\}$  which is defined by the position of the samples in the feature space  $y_j$  and their respective class assignments  $t_j$ , also referred to as *targets*.

The task is to estimate the parameters  $\mathbf{w}$  of a model  $g(\mathbf{w}, y)$  from the training set which allows to evaluate the posterior probability  $P(t_i | y_i, \mathbf{w})$  for an arbitrary class assignment  $t_i$  at an arbitrary feature position  $y_i$ . The model parameters shall be optimal in the sense that the posterior probability of the class assignments given by the training set is maximized

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{j=1}^J P(t_j | y_j, \mathbf{w}). \quad (2.57)$$

In this thesis, we restrict to the case of a binary classification task over an one dimensional feature space  $y$ . This means only two classes exist and the class assignment can be encoded by the target values  $t_j \in \{0, 1\}$ . Further, the decision boundary is given by a single scalar  $y_0$  which represents the most probable separation of both classes. We use an alternative formulation of (2.57) for the binary classification task

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \prod_{j=1}^J P(0 | y_j, \mathbf{w})^{1-t_j} P(1 | y_j, \mathbf{w})^{t_j}. \quad (2.58)$$

### 2.5.2 Logistic Regression Model

The model employed by logistic regression is based on a *logistic function*  $g$ , which we also refer to as *sigmoid function*, which models the posterior probability of the classes 0 and 1

with respect to the feature space  $y$

$$P(1 | y, \mathbf{w}) = g(\mathbf{w}, y) := \frac{1}{1 + \exp(-(w_1 y + w_0))} \quad (2.59)$$

$$P(0 | y, \mathbf{w}) = 1 - g(\mathbf{w}, y) = \frac{\exp(-(w_1 y + w_0))}{1 + \exp(-(w_1 y + w_0))}. \quad (2.60)$$

The sigmoid function is governed by the model parameters  $\mathbf{w} = [w_0, w_1]^\top$ . Since  $w_1$  controls the slope of the sigmoid function, we will refer to it as *slope parameter* in the remainder of this thesis.

The decision boundary  $y_0$  between the classes is given by the 0.5-crossing of the sigmoid function

$$y_0 = -\frac{w_0}{w_1}. \quad (2.61)$$

### 2.5.3 Estimation of the model parameters

In the following, we describe the procedure for the estimation of the model parameters  $\mathbf{w}$ . The estimation task embodies a non-linear optimization problem. The utilized iterative optimization procedure based on Newton's method is presented in Section 2.5.3.1.

Beside the model parameters and the decision boundary, we also demand information about the precision of the decision boundary  $\sigma_{y_0}^2$ . Section 2.5.3.2 describes an approach to determine this information by exploiting the similarity of the iterative optimization procedure and the iterative weighted least squares scheme presented in Section 2.3.3. The mathematical derivations in this section are largely based on [Bishop, 2006, pp. 207-208].

Finally, Section 2.5.3.3 describes a regularization scheme utilized to prevent from numeric instabilities in case the best solution would be a sigmoid function of infinite steepness.

#### 2.5.3.1 Optimization via Newton's method

**Formulation of the Cost Function:** Instead of searching for the maximum argument of (2.58), we solve for the minimum argument

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} C(\mathbf{w}) \quad (2.62)$$

of the respective negative logarithm. This yields the cost function of the non-linear optimization task

$$C(\mathbf{w}) = \sum_{j=1}^J (\mathbf{t}_j - 1) \ln P(0 | y_j, \mathbf{w}) - \mathbf{t}_j \ln P(1 | y_j, \mathbf{w}). \quad (2.63)$$

In order to facilitate the determination of the Jacobian and Hessian of the cost function presented in the next paragraph, we rearrange the cost function to a simplified representation. Substituting (2.59) and (2.60) and exploiting the identity  $1 - g(\mathbf{w}, y) = \exp(-(w_1 y + w_0))g(\mathbf{w}, y)$  results in

$$C(\mathbf{w}) = \sum_{j=1}^J (\mathbf{t}_j - 1) \ln (1 - g(\mathbf{w}, y_j)) - \mathbf{t}_j \ln g(\mathbf{w}, y_j) \quad (2.64)$$

$$= \sum_{j=1}^J -(\mathbf{t}_j - 1) (w_1 y_j + w_0) + (\mathbf{t}_j - 1) \ln g(\mathbf{w}, y_j) - \mathbf{t}_j \ln g(\mathbf{w}, y_j). \quad (2.65)$$

Finally, after summarizing we achieve the simplified formulation of the cost function

$$C(\mathbf{w}) = \sum_{j=1}^J (1 - \mathbf{t}_j) (w_1 y_j + w_0) - \ln g(\mathbf{w}, y_j). \quad (2.66)$$

**Iterative Update Scheme:** The parameters are estimated in an iterative scheme using Newton's method. Each update step

$$\hat{\mathbf{w}}^{(\nu+1)} = \hat{\mathbf{w}}^{(\nu)} - H_C^{-1}(\hat{\mathbf{w}}^{(\nu)}) J_C(\hat{\mathbf{w}}^{(\nu)}) \quad (2.67)$$

is governed by the Jacobian  $J_C$  and Hessian  $H_C$  holding the first and second derivatives of the cost function with respect to the parameters  $\mathbf{w}$ .

The derivation of the Jacobian and Hessian facilitates when bringing to mind that the partial derivatives of the sigmoid function  $g$  can in turn be expressed in terms of the sigmoid function itself

$$\frac{\partial}{\partial w_0} g(\mathbf{w}, y_j) = \frac{\exp(-(w_1 y + w_0))}{(1 + \exp(-(w_1 y + w_0)))^2} = g(\mathbf{w}, y_j) (1 - g(\mathbf{w}, y_j)) \quad (2.68)$$

$$\frac{\partial}{\partial w_1} g(\mathbf{w}, y_j) = \frac{\exp(-(w_1 y + w_0)) y_j}{(1 + \exp(-(w_1 y + w_0)))^2} = g(\mathbf{w}, y_j) (1 - g(\mathbf{w}, y_j)) y_j. \quad (2.69)$$

Using this insight, we obtain the Jacobian

$$J_C(\mathbf{w}) = \sum_{j=1}^J (1 - \mathbf{t}_j) \begin{bmatrix} 1 \\ y_j \end{bmatrix} - \frac{\partial \ln g(\mathbf{w}, y_j)}{\partial g} \begin{bmatrix} \frac{\partial}{\partial w_0} g(\mathbf{w}, y_j) \\ \frac{\partial}{\partial w_1} g(\mathbf{w}, y_j) \end{bmatrix} \quad (2.70)$$

$$= \sum_{j=1}^J (g(\mathbf{w}, y_j) - \mathbf{t}_j) \begin{bmatrix} 1 \\ y_j \end{bmatrix}. \quad (2.71)$$

The second derivatives follow from the derivation of the components of the Jacobian with respect to  $w_0$  and  $w_1$ . They again facilitate to the partial derivation of the sigmoid function yielding the Hessian

$$H_C(\mathbf{w}) = \sum_{j=1}^J \begin{bmatrix} 1 \\ y_j \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial w_0} g(\mathbf{w}, y_j) & \frac{\partial}{\partial w_1} g(\mathbf{w}, y_j) \end{bmatrix} \quad (2.72)$$

$$= \sum_{j=1}^J g(\mathbf{w}, y_j) (1 - g(\mathbf{w}, y_j)) \begin{bmatrix} 1 \\ y_j \end{bmatrix} \begin{bmatrix} 1 & y_j \end{bmatrix}. \quad (2.73)$$

Inserting Jacobian and Hessian into (2.67) yields the update step of the estimation procedure. The update step can be formulated in a compact matrix representation

$$\hat{\mathbf{w}}^{(\nu+1)} = \hat{\mathbf{w}}^{(\nu)} + \left( \mathbf{Y}^\top \text{Diag}(\mathbf{k}^{(\nu)}) \mathbf{Y} \right)^{-1} \mathbf{Y}^\top (\mathbf{t} - \mathbf{g}^{(\nu)}) \quad (2.74)$$

in terms of the feature matrix  $\mathbf{Y} := [\mathbf{y}, \mathbf{1}_J]$  and the current posterior probabilities

$$\mathbf{g}^{(\nu)} := [g(\hat{\mathbf{w}}^{(\nu)}, y_1), \dots, g(\hat{\mathbf{w}}^{(\nu)}, y_J)]^\top. \quad (2.75)$$

The short hand notation  $\mathbf{k}^{(\nu)}$  denotes the Hadamard product

$$\mathbf{k}^{(\nu)} := \mathbf{g}^{(\nu)} \circ (1 - \mathbf{g}^{(\nu)}). \quad (2.76)$$

### 2.5.3.2 Determination of the Precision of the Decision Boundary

The decision boundary  $y_0$  follows from the negative fraction of the estimated parameters (2.61). Information about its precision  $\sigma_{y_0}^2$  can be achieved via error propagation in case the precision of the estimated parameters  $\Sigma_{\hat{w}\hat{w}}$  is known. To obtain this information, let us consider the similarity of the Newton update step (2.74) and the update step for the weighted least squares optimization in non-linear models presented in Section 2.3.3. We will find that we can re-interpret the optimization task of logistic regression as least squares optimization task and, thus, obtain  $\Sigma_{\hat{w}\hat{w}}$  by the inverse of the respective normal equation matrix.

The first paragraph of this section describes the analogy between the tasks as well as the utilized procedure for the computation of the desired precision information. The second paragraph describes how to tread the case that the training sample positions  $y_j$  are uncertain, i.e. are given by Gaussian distributed random variables rather than by fixed values. A scheme is presented to incorporate the known precision  $\sigma_{y_j}^2$  in the estimation process.

**Optimization via Iterative Reweighted Least Squares:** We can re-interpret the Newton update step by reshaping (2.74) to

$$\hat{\mathbf{w}}^{(\nu+1)} = \hat{\mathbf{w}}^{(\nu)} + \underbrace{\left( A^{(\nu)\top} \Sigma_{\mathbf{t}\mathbf{t}}^{(\nu)-1} A^{(\nu)} \right)^{-1}}_{\mathbf{N}^{(\nu)}} \underbrace{A^{(\nu)\top} \Sigma_{\mathbf{t}\mathbf{t}}^{(\nu)-1} (\mathbf{t} - \mathbf{g}^{(\nu)})}_{\mathbf{h}^{(\nu)}}. \quad (2.77)$$

This corresponds to a Gauss-Newton update step of the weighted least squares problem

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmin}} \left[ (\mathbf{t} - \mathbf{g}(\mathbf{w}, \mathbf{y}))^\top \Sigma_{\mathbf{t}\mathbf{t}}^{(\nu)-1} (\mathbf{t} - \mathbf{g}(\mathbf{w}, \mathbf{y})) \right], \quad (2.78)$$

with the Jacobian (i.e. the design matrix)

$$A^{(\nu)} = \left. \frac{\partial \mathbf{g}^{(\nu)}}{\partial \mathbf{w}} \right|_{\mathbf{w}=\hat{\mathbf{w}}^{(\nu)}} = \operatorname{Diag}(\mathbf{k}^{(\nu)}) \mathbf{Y} \quad (2.79)$$

and the covariance matrix of the targets  $\Sigma_{\mathbf{t}\mathbf{t}}^{(\nu)} = \operatorname{Diag}(\mathbf{k}^{(\nu)})$ . Thus, we can think of the optimization procedure as estimation of the parameters  $\hat{\mathbf{w}}$  in a non-linear Gauss-Markov model, as described in Section 2.3.3, with the functional and the stochastic model given by

$$\mathbf{t} + \mathbf{v}_{\mathbf{t}} = \mathbf{g}(\mathbf{w}, \mathbf{y}), \quad \mathbf{v}_{\mathbf{t}} \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{t}\mathbf{t}}^{(\nu)}) \quad (2.80)$$

using the short hand notation  $\mathbf{g}(\mathbf{w}, \mathbf{y}) = [g(\mathbf{w}, y_1), \dots, g(\mathbf{w}, y_J)]^\top$ . The only difference is given by the fact that  $\Sigma_{\mathbf{t}\mathbf{t}}^{(\nu)}$  is recomputed in every iteration (reweighted least squares).

Following (2.49), the precision of the estimated parameters is given by the inverse of the normal equation matrix in the final iteration step  $\hat{\nu}$

$$\Sigma_{\hat{w}\hat{w}} = \mathbf{N}^{(\hat{\nu})-1}. \quad (2.81)$$

Finally, linear error propagation yields the desired precision of the decision boundary

$$\sigma_{y_0}^2 = \left[ -\frac{1}{\hat{w}_1}, \frac{\hat{w}_0}{\hat{w}_1^2} \right] \Sigma_{\hat{w}\hat{w}} \left[ -\frac{1}{\hat{w}_1}, \frac{\hat{w}_0}{\hat{w}_1^2} \right]^\top. \quad (2.82)$$



For a better interpretation of the obtained precision, let us examine the stochastic meaning of  $\Sigma_{\mathbf{tt}}^{(\nu)}$ . Consider the expected value and variance of a class assignment  $t \in \{0, 1\}$  based on the currently estimated model parameters, i.e. governed by the posterior probabilities (2.59) and (2.60), which read as

$$E(\mathbf{t}) = 0 \cdot P(\mathbf{t} = 0 \mid y, \mathbf{w}^{(\nu)}) + 1 \cdot P(\mathbf{t} = 1 \mid y, \mathbf{w}^{(\nu)}) = g(\mathbf{w}^{(\nu)}, y) \quad (2.83)$$

$$\sigma_{\mathbf{t}}^2 = E(\mathbf{t}^2) - E(\mathbf{t})^2 = g(\mathbf{w}^{(\nu)}, y) - g(\mathbf{w}^{(\nu)}, y)^2. \quad (2.84)$$

Note that the evaluation of the variance (2.84) for the training samples  $y_j$  equals the components of  $\mathbf{k}^{(\nu)}$ , which in turn embody the diagonal elements of  $\Sigma_{\mathbf{tt}}^{(\nu)}$ . Put simply,  $\Sigma_{\mathbf{tt}}^{(\nu)}$  represents the variance of the class assignments based on the currently estimated model rather than the variance of the target values of the training set.

**Treatment of Sample Position Uncertainty:** Let us now assume that the sample positions  $y_j$  are Gaussian distributed random variables with the variance  $\sigma_{y_j}^2$  rather than fixed values. This yields the estimation task

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} \varphi(\mathbf{t}, \mathbf{y} \mid \mathbf{w}). \quad (2.85)$$

We use the local derivative of  $\mathbf{g}$  with respect to  $y$  to propagate the uncertainty in the space of  $\mathbf{g}$ . In combination with (2.84) we obtain the new covariance matrix of the targets

$$\overline{\Sigma}_{\mathbf{tt}}^{(\nu)} = \operatorname{Diag} \left( [\dots, \overline{\sigma}_{\mathbf{t}_j}^{(\nu)^2}, \dots]^T \right), \quad (2.86)$$

with

$$\overline{\sigma}_{\mathbf{t}_j}^{(\nu)^2} = \sigma_{\mathbf{t}_j}^{(\nu)^2} + \left( \left. \frac{\partial g(\hat{\mathbf{w}}^{(\nu)}, y_j)}{\partial y} \right|_{y_j} \right)^2 \sigma_{y_j}^2 \quad (2.87)$$

$$= \sigma_{\mathbf{t}_j}^{(\nu)^2} + \left( k_j^{(\nu)} \hat{w}_1 \right)^2 \sigma_{y_j}^2, \quad (2.88)$$

where  $k_j$  denotes the  $k$ th component of (2.76).

With the uncertainty transferred to the targets, we treat  $\mathbf{y}$  as certain and achieve the same initial situation as stated for the original task (2.58). Thus, we approximate (2.85) by solving

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\operatorname{argmax}} P(\mathbf{t} \mid \mathbf{y}, \mathbf{w}) \quad (2.89)$$

by the procedure presented in the previous sections using the extended uncertainty (2.88).

### 2.5.3.3 Regularization

In case of separable training data, i.e. in case a decision boundary can be determined without any misclassified training sample, the optimal model is represented by a sigmoid function of infinite steepness. This would cause numerical instabilities in the estimation process. Thus, the slope parameter  $w_1$  is regularized using the prior assumption that the slope parameter is of small value  $w_1 \sim \mathcal{N}(0, \frac{1}{\lambda})$ . The variance is governed by a regularization parameter  $\lambda$  which embodies a weighting parameter of small value, say  $\lambda = 10^{-6}$ . The prior is introduced to the estimation procedure by appending the negative logarithm of the respective Gaussian function  $G(w_1 \mid 0, \frac{1}{\lambda})$  to the cost function (2.66). With the first and second derivative

$$\frac{\partial (-\ln G(w_1 | 0, \frac{1}{\lambda}))}{\partial w_1} = \lambda w_1 \quad \text{and} \quad \frac{\partial^2 (-\ln G(w_1 | 0, \frac{1}{\lambda}))}{\partial w_1^2} = \lambda, \quad (2.90)$$

the linearized regularization term extends the parameter update in Equation (2.77) to

$$\hat{\mathbf{w}}^{(\nu+1)} = \hat{\mathbf{w}}^{(\nu)} + \left( N^{(\nu)} + \text{Diag} \left( \begin{bmatrix} 0 \\ \lambda \end{bmatrix} \right) \right)^{-1} \left( \mathbf{h}^{(\nu)} - \begin{bmatrix} 0 \\ \lambda \hat{w}_1^{(\nu)} \end{bmatrix} \right). \quad (2.91)$$

Analogous to (2.80), we can re-interpret the newton update for the regularized model as reweighted least squares update step. The respective Gauss-Markov model is given by

$$\begin{bmatrix} \mathbf{t} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_t \\ v_w \end{bmatrix} = \begin{bmatrix} \mathbf{g}(\mathbf{w}, \mathbf{y}) \\ w_1 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{v}_t \\ v_w \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \bar{\Sigma}_{\mathbf{t}\mathbf{t}}^{(\nu)} & \mathbf{0} \\ \mathbf{0}^\top & \frac{1}{\lambda} \end{bmatrix} \right), \quad (2.92)$$

with the covariance matrix of the targets defined by (2.86).

#### 2.5.4 Treatment of Non-Deterministic Targets

In this thesis, we face the situation that the information provided by the targets of the training set is given by the probabilities of the class memberships  $P(\mathbf{t}_j = 0 | y_j)$  and  $P(\mathbf{t}_j = 1 | y_j)$  instead of a definite membership statement.

With the short hand notations  $P_{j0} := P(\mathbf{t}_j = 0 | y_j)$  and  $P_{j1} := P(\mathbf{t}_j = 1 | y_j)$ , the cost function for the respective minimization task reads as

$$C(\mathbf{w}) = \sum_{j=1}^J -P_{j0} \ln P(0 | y_j, \mathbf{w}) - P_{j1} \ln P(1 | y_j, \mathbf{w}) \quad (2.93)$$

$$= \sum_{j=1}^J (P_{j1} - 1) \ln P(0 | y_j, \mathbf{w}) - P_{j1} \ln P(1 | y_j, \mathbf{w}). \quad (2.94)$$

Considering the similarity of (2.94) to the original cost function (2.63), we can determine the model parameters by the procedure described in Section 2.5.3 by simply substituting  $P_{j1}$  for  $\mathbf{t}_j$ . Therefore, the parameters are estimated in an iterative re-weighted least squares procedure based on the functional model

$$\begin{bmatrix} \vdots \\ P_{j1} \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{v}_t \\ v_w \end{bmatrix} = \begin{bmatrix} \vdots \\ \mathbf{g}(\mathbf{w}, y_j) \\ \vdots \\ w_1 \end{bmatrix}, \quad (2.95)$$

which follows from the adaption of (2.92). The stochastic model

$$\begin{bmatrix} \mathbf{v}_t \\ v_w \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \bar{\Sigma}_{\mathbf{t}\mathbf{t}}^{(\nu)} & \mathbf{0} \\ \mathbf{0}^\top & \frac{1}{\lambda} \end{bmatrix} \right) \quad (2.96)$$

is governed by the covariance matrix of the targets (2.86) and the regularization parameter  $\lambda$ .

## 2.6 The Expectation Maximization Algorithm for Maximum a Posteriori Estimation

In parameter estimation tasks, the statistical model which defines the relationship between unknown parameters and observations may depend on additional latent (unobserved) variables that are vital for the estimation process. The Expectation Maximization (EM) algorithm [see Dempster et al., 1977] is a general technique to deal with this problem by alternately searching for the latent variables and unknown parameters.

This section describes the usage of the EM algorithm for maximum a posteriori estimation tasks. The formalisms are mainly based on [Bishop, 2006, Chapter 9.3].

In the proposed approach we encounter the problem that the detection task, i.e. the decision which areas of the considered world region belong to the street and which do not, and the estimation of the street surface and boundary parameters mutually depend on each other's results. Thus, we use the EM-algorithm to tackle both tasks simultaneously.

### 2.6.1 General EM Algorithm

Assume a joint distribution  $\wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta})$  over a set of observations  $\mathbf{z}$  and discrete latent variables  $\mathbf{l}$  that is controlled by unknown parameters  $\boldsymbol{\theta}$  to be given. Let the latent variables  $\mathbf{l} = [\dots, l_i, \dots]^T \in \mathcal{L}$  be discrete values  $l_i \in \{c_1, \dots, c_K\}$ , with  $i = 1, \dots, I$ . Assume the only knowledge on  $\mathbf{l}$  to be given by the posterior probability  $P(\mathbf{l} | \mathbf{z}, \boldsymbol{\theta})$ . Further, a prior distribution  $\wp(\boldsymbol{\theta})$  for the unknown parameters is known.

We address the task of finding the MAP estimate

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \wp(\boldsymbol{\theta} | \mathbf{z}). \quad (2.97)$$

We can reformulate this problem in terms of the known distributions  $\wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta})$  and  $\wp(\boldsymbol{\theta})$ . Therefore, we exploit the proportionality  $\wp(\boldsymbol{\theta} | \mathbf{z}) \sim \wp(\mathbf{z} | \boldsymbol{\theta}) \wp(\boldsymbol{\theta})$  defined by the Bayes' theorem as well as the law of total probability  $\wp(\mathbf{z} | \boldsymbol{\theta}) = \sum_{\mathbf{l} \in \mathcal{L}} \wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta})$  to achieve the reformulated task

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[ \sum_{\mathbf{l} \in \mathcal{L}} \wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta}) \wp(\boldsymbol{\theta}) \right]. \quad (2.98)$$

The evaluation of the likelihood term requires information about both the observations and latent variables, i.e. about the set  $\{\mathbf{z}, \mathbf{l}\}$ . This set is called *complete* dataset in order to distinguish from the *incomplete* dataset formed by the observations  $\mathbf{z}$  solely. Thus, we also use the term *complete-data likelihood* to refer to  $\wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta})$ .

According to the problem specific definition of the likelihood term, this optimization problem may be quite challenging or even infeasible. In particular, if the joint distribution  $\wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta})$  belongs to the exponential family (e.g. a Gaussian distribution as in the least squares problems presented in Section 2.3.2) the technique of searching for the maximum argument of the logarithm as alternative to (2.98), i.e.

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[ \ln \sum_{\mathbf{l} \in \mathcal{L}} \wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta}) + \ln \wp(\boldsymbol{\theta}) \right], \quad (2.99)$$

will not yield the anticipated simplification of the task since the sum appears inside the logarithm.

The EM algorithm addresses this problem by iteratively estimating the unknown parameters  $\hat{\boldsymbol{\theta}}^{(\nu+1)}$  from the expected value  $\xi(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\nu)})$  of the complete-data log likelihood  $\ln \wp(\mathbf{z}, \mathbf{l} | \boldsymbol{\theta})$  under the posterior distribution of the latent variables  $P(\mathbf{l} | \mathbf{z}, \hat{\boldsymbol{\theta}}^{(\nu)})$ , as described in Algorithm 1.

Choose an initial setting for the parameters  $\boldsymbol{\theta}^{(0)}$ ;

**repeat**

**Estimation step**

        Evaluate  $P(\mathbf{l} | \mathbf{z}, \hat{\boldsymbol{\theta}}^{(\nu)})$  for all possible assignment of  $\mathbf{l}$ ;

**Maximization step**

        Estimate

$$\hat{\boldsymbol{\theta}}^{(\nu+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[ \xi(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\nu)}) + \ln \wp(\boldsymbol{\theta}) \right], \quad (2.100)$$

        with the expectation of the complete-data log likelihood

$$\xi(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\nu)}) = \sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l} | \mathbf{z}, \hat{\boldsymbol{\theta}}^{(\nu)}) \ln \wp(\mathbf{l}, \mathbf{z} | \boldsymbol{\theta}); \quad (2.101)$$

**until** convergence criterion is fulfilled ;

**Algorithm 1:** EM algorithm for MAP estimation

In case the algorithm is used for a minimization problem, e.g. when searching for the minimum argument of the negative log likelihood, the maximization step simply turns into a minimization step, i.e. (2.100) changes to

$$\hat{\boldsymbol{\theta}}^{(\nu+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[ -\xi(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}^{(\nu)}) - \ln \wp(\boldsymbol{\theta}) \right]. \quad (2.102)$$

Note that the EM algorithm is a local optimization procedure and, thus, in general is not guaranteed to find the global optimal solution.

### 2.6.2 EM Algorithm with Simplified Assumptions

If we assume the probabilities in (2.101) to factorizes into products

$$P(\mathbf{l} | \mathbf{z}, \hat{\boldsymbol{\theta}}^{(\nu)}) = \prod_{i=1}^I P(l_i | \mathbf{z}, \hat{\boldsymbol{\theta}}^{(\nu)}) \quad (2.103)$$

and

$$\wp(\mathbf{l}, \mathbf{z} | \boldsymbol{\theta}) = \prod_{i=1}^I \wp(l_i, z_i | \boldsymbol{\theta}), \quad (2.104)$$

the maximization step simplifies to

$$\hat{\boldsymbol{\theta}}^{(\nu+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[ \sum_{i=1}^I \sum_{k=1}^K P(l_i = c_k | \mathbf{z}, \hat{\boldsymbol{\theta}}^{(\nu)}) \ln (\wp(l_i = c_k, z_i | \boldsymbol{\theta}) + \ln \wp(\boldsymbol{\theta})) \right]. \quad (2.105)$$

A proof is given in Section D.1.

## 2.7 B-Splines

B-splines have shown to be a suitable tool to model complex curves and surfaces of locally varying shape and are widely used in the field of computer graphics. The basic idea is to divide the entity, e.g. a curve or a surface, into segments which are locally modeled by a polynomial representation of low degree, say quadratic or cubic. This bears advantages in terms of complexity compared to a single polynomial segment as well as in terms of locality compared to several alternative spline representations, e.g. Bézier curves and surfaces [Piegl and Tiller, 1997, Chapter 1]. We will find the latter aspect particularly helpful when estimating the spline parameters for the street boundary curve and the street surface (see Section 3.5.2) since the locality property yields a sparse relation between observations and unknown parameters.

In the following, we give a short introduction of B-splines and in particular B-spline curves and surfaces. We focus on the chosen parametrization and properties that are exploited in this thesis. For the sake of completeness, after describing curves and surfaces in Section 2.7.1 and 2.7.2, we give a formal definition of the B-spline basis functions in Section 2.7.3 and describe how we deal with parameter values beyond the defined parameter range in Section 2.7.4. The theory presented in this section is largely based on [Piegl and Tiller, 1997, Chapter 3].

Within this thesis, we utilize uniform B-spline functions to model the elevation characteristics of the street and the horizontal delimiter of the free-space. The street surface is represented by a B-spline surface defined over a bidirectional grid of control points located in the horizontal space. A planar B-spline curve is used to represent the horizontal boundary of the street.

### 2.7.1 B-Spline Curves

A planar B-spline curve is a parametric curve  $\mathbf{x}(\tau) = [x(\tau), y(\tau)]^\top$  governed by the curve parameter  $\tau \in [\tau_{\min}, \tau_{\max}]$  which specifies the position along the curve. Each point on the curve is computed by a weighted linear combination of a set of control points  $\mathbf{x}_m = [x_m, y_m]^\top$ , with  $m = 1 \dots M$ , such that

$$\mathbf{x}(\tau) = \sum_{m=1}^M \mathcal{B}_{m,q}(\tau) \mathbf{x}_m. \quad (2.106)$$

The respective weights  $\mathcal{B}_{m,q}(\tau)$  for each control point are sampled from polynomial basis functions of degree  $q$ . A mathematical definition of this B-spline basis functions and their derivatives is given in Section 2.7.3. This basis functions differ from zero only in a finite interval. This yields the locality property, i.e. the spline curve is subdivided in  $M - q$  so called *spline sections*. These sections are each controlled by a subset of  $q + 1$  control points  $\{\mathbf{x}_j, \dots, \mathbf{x}_{j+q}\}$ , where  $j$  denotes the respective section number. Thus, (2.106) expands to a polynomial of degree  $q$  over the parameter  $\tau$  whose coefficients are fully described by the respective control point subset. We refer to the respective subset as the spline section's *active* control points. This property is illustrated in Figure 2.5.

Another property that we exploit states that each curve section lies inside the convex hull of its respective active control points.

#### 2.7.1.1 Spline Parametrization

The borders of the intervals in which the basis functions differ from zero are defined by a set of so called *knots*  $\boldsymbol{\tau} = [\tau_1 \dots \tau_{M+q+1}]^\top$ . In case the knots are equally distributed on  $\tau$ , the

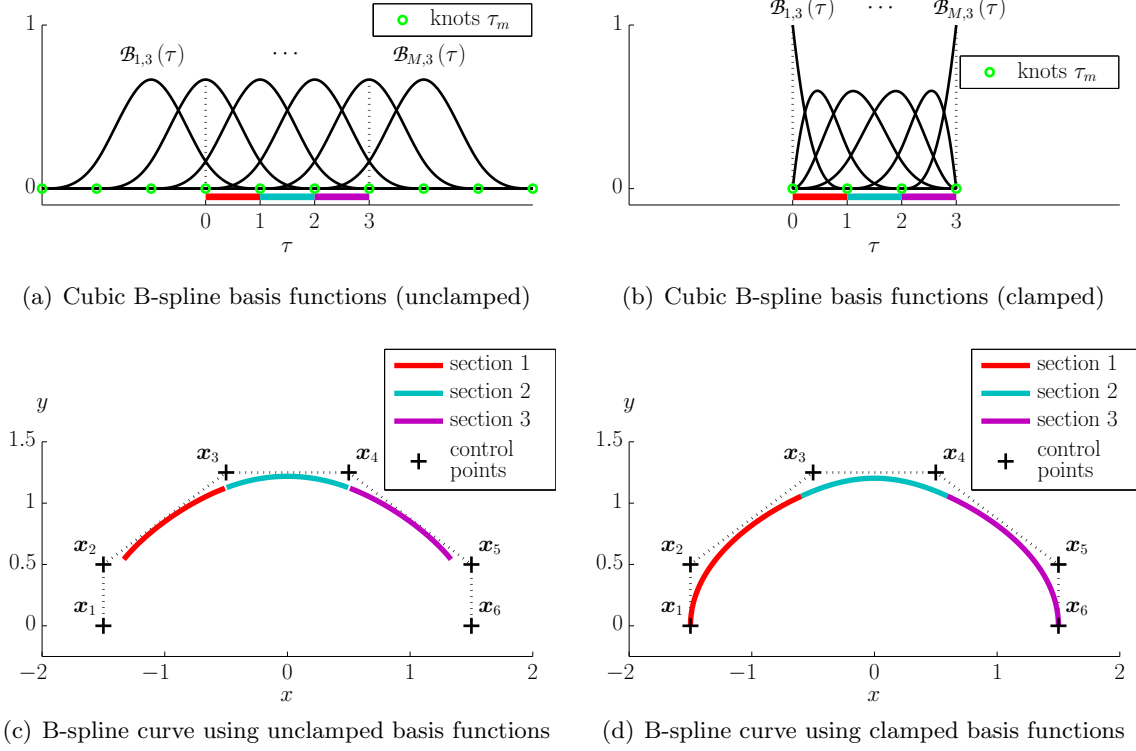


FIGURE 2.5: The first row illustrates  $M = 6$  cubic B-spline basis functions covering the parameter range  $\tau \in [0, 3]$  over three spline sections. For Figure (a) a uniform knot distribution was chosen. Figure (b) shows the *clamped* version, i.e. the first and last knot are repeated degree + 1 times. This gives the first and the last basis function complete control at  $\tau_{\min}$  and  $\tau_{\max}$ , respectively. The second row illustrates B-spline curves constructed from a set of control points  $\mathbf{x}_m$  using the respective basis function set depicted the first row. In (d) we can see the property of the *clamped* version to force the curve to reach the first and last control point at  $\tau_{\min}$  and  $\tau_{\max}$ , respectively.

basis functions are just shifted copies of each other and we speak of a *uniform* spline. In case the first and last knots are repeated  $q + 1$  times, the endpoints of the curve match the first and last control point, as shown in Figure 2.5(d), and we speak of a *clamped* spline.

In this thesis, we use the uniform and unclamped definition. In order to describe the position along the spline, we choose a parametrization which is  $\tau_{\min} := 0$  at the start of the first spline section and  $\tau_{\max} := M - q$  at the end of the last spline section. This means that each section has a parameter range of 1.

### 2.7.1.2 Curvature of a B-Spline Curve

In this work, we make use of prior assumptions about the curvature of the B-spline curve for regularization issues. Following [Bronstein et al., 2001, p. 239], the local curvature of a planar parameterized curve at a certain parameter position  $\tau_0$  is defined as

$$\kappa(\tau_0) = \frac{\begin{vmatrix} x'(\tau_0) & x''(\tau_0) \\ y'(\tau_0) & y''(\tau_0) \end{vmatrix}}{|x'(\tau_0)|^3} = \frac{x'(\tau_0)y''(\tau_0) - x''(\tau_0)y'(\tau_0)}{(x'(\tau_0)^2 + y'(\tau_0)^2)^{3/2}}. \quad (2.107)$$

This corresponds to the inverse of the radius of the osculating circle, i.e. the circle that touches the curve at the considered point  $\mathbf{x}(\tau_0)$  with the same tangent and curvature.

In terms of the control point coordinates, the  $k$ th derivative of a B-spline curve with respect to the spline parameter  $\tau$  is defined as

$$\mathbf{x}^{(k)}(\tau_0) = [x^{(k)}(\tau_0), y^{(k)}(\tau_0)]^\top = \sum_{m=1}^M \mathcal{B}_{m,q}^{(k)}(\tau_0) \mathbf{x}_m, \quad (2.108)$$

where  $\mathcal{B}_{m,q}^{(k)}(\tau_0)$  denotes the  $k$ th derivative of the  $m$ th basis function as defined in (2.123). Inserting (2.108) into (2.107), we achieve the local curvature as function of the control point coordinates

$$\kappa_{\tau_0}([\mathbf{x}_1, \dots, \mathbf{x}_M]^\top) = \frac{\left( \sum_{m=1}^M \mathcal{B}'_{m,q}(\tau_0) x_m \right) \left( \sum_{m=1}^M \mathcal{B}''_{m,q}(\tau_0) y_m \right) - \left( \sum_{m=1}^M \mathcal{B}''_{m,q}(\tau_0) x_m \right) \left( \sum_{m=1}^M \mathcal{B}'_{m,q}(\tau_0) y_m \right)}{\left( \left( \sum_{m=1}^M \mathcal{B}'_{m,q}(\tau_0) x_m \right)^2 + \left( \sum_{m=1}^M \mathcal{B}'_{m,q}(\tau_0) y_m \right)^2 \right)^{3/2}}. \quad (2.109)$$

In Section 3.5.2.2, this curvature measure is utilized as regularization term in a non-linear optimization procedure. For linearization issues, we require the first derivatives with respect to the control point coordinates given by

$$\frac{\partial \kappa_{\tau_0}}{\partial x_m} = \frac{y''(\tau_0) \mathcal{B}'_{m,q}(\tau_0) - y'(\tau_0) \mathcal{B}''_{m,q}(\tau_0)}{|\mathbf{x}'(\tau_0)|^3} - \frac{2x'(\tau_0) \mathcal{B}'_{m,q}(\tau_0) \begin{vmatrix} x'(\tau_0) & x''(\tau_0) \\ y'(\tau_0) & y''(\tau_0) \end{vmatrix}}{|\mathbf{x}'(\tau_0)|^5} \quad (2.110)$$

and

$$\frac{\partial \kappa_{\tau_0}}{\partial y_m} = \frac{x'(\tau_0) \mathcal{B}''_{m,q}(\tau_0) - x''(\tau_0) \mathcal{B}'_{m,q}(\tau_0)}{|\mathbf{x}'(\tau_0)|^3} - \frac{2y'(\tau_0) \mathcal{B}'_{m,q}(\tau_0) \begin{vmatrix} x'(\tau_0) & x''(\tau_0) \\ y'(\tau_0) & y''(\tau_0) \end{vmatrix}}{|\mathbf{x}'(\tau_0)|^5}. \quad (2.111)$$

### 2.7.2 B-Spline Surfaces

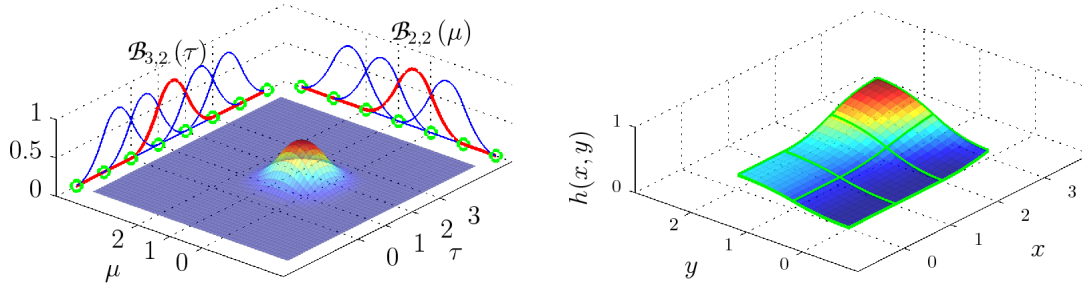
Similar to curves, B-splines are used to define piecewise polynomial 2.5d surfaces, i.e. surfaces defined as a height function  $h(x, y)$  over the coordinate axes  $x$  and  $y$ . The height function is embodied by the weighted linear combination of a bidirectional grid of  $N = I \times J$  control point heights  $h_{i,j}$  and reads as

$$h(x, y) = \sum_{i=1}^I \sum_{j=1}^J \mathcal{B}_{i,q}(\tau(x)) \mathcal{B}_{j,q}(\mu(y)) h_{i,j}. \quad (2.112)$$

The 2d B-spline basis functions are given by the product of two 1d basis functions governed by the parameters  $\tau(x) \in [0, I - q]$  and  $\mu(y) \in [0, J - q]$ , respectively. An illustration is provided by Figure 2.6(a).

In the remainder of this thesis, we restrict to the case that the knot points are arranged in a bidirectional grid with constant spacing  $d_x = \frac{\partial \tau}{\partial x}$  and  $d_y = \frac{\partial \mu}{\partial y}$ , respectively, which is located at  $[x_{\min}, y_{\min}]^\top$  as illustrated in Figure 2.6. Thus, the relation between world coordinates and parameters is given by the linear transformation

$$\begin{bmatrix} \tau(x) \\ \mu(y) \end{bmatrix} = \begin{bmatrix} \frac{(x - x_{\min})}{d_x} \\ \frac{(y - y_{\min})}{d_y} \end{bmatrix}. \quad (2.113)$$

(a) 2d quadratic B-spline basis function  $\mathcal{B}_{10,2}(\tau, \mu)$ 

(b) Quadratic B-spline surface

FIGURE 2.6: Figure (a) shows an example of a 2d B-spline basis function constructed from two 1d basis functions  $\mathcal{B}_{10,2}(\tau, \mu) = \mathcal{B}_{3,2}(\tau) \mathcal{B}_{2,2}(\mu)$ . Additionally, the set of one dimensional basis functions for both axes is plotted. The green circles mark the knot positions. The dotted lines in the  $\tau, \mu$  plane show that six spline sections are created in the interval  $[0, 3] \times [0, 2]$ . Figure (b) shows an example for a B-spline surface in this interval constructed from the basis functions depicted in (a). The section borders on the surface are marked by green lines. Note that in this example the spline parametrization equals the axes parametrization, i.e.  $\tau = x$  and  $\mu = y$ .

Analog to B-spline curves, the surface is subdivided in two dimensional splines sections which are fully described by a subset of the control point heights ( $3 \times 3$  in case of quadratic basis functions). An example for this subdivision is given in Figure 2.6(b).

For ease of readability we introduce the shorter notation

$$h(x, y) = \sum_{n=1}^N \mathcal{B}_{n,q}(x, y) h_n, \quad (2.114)$$

with  $\mathcal{B}_{n,q}(x, y) := \mathcal{B}_{i,q}(\tau(x)) \mathcal{B}_{j,q}(\mu(y))$  and the new indices  $n = (i-1)J + j$ .

### 2.7.2.1 Derivatives of a B-Spline Surface

Let us now consider the  $k$ th derivatives of the surface with respect to the grid coordinates, where the order  $k = a + b$  composes of the order of the partial derivative in  $x$  direction  $a$  and the order of the partial derivative in  $y$  direction  $b$ . The derivatives evaluated at the position  $[x_0, y_0]^T$  read as

$$\frac{\partial^k}{\partial x^a \partial y^b} h(x_0, y_0) = \frac{1}{d_x^a d_y^b} \sum_i^I \sum_j^J \mathcal{B}_{i,q}^{(a)}(\tau(x_0)) \mathcal{B}_{j,q}^{(b)}(\mu(y_0)) h_{i,j} \quad (2.115)$$

and in the shorter notation

$$\frac{\partial^k}{\partial x^a \partial y^b} h(x_0, y_0) = \sum_{n=1}^N \mathcal{B}_{n,q}^{(a,b)}(x_0, y_0) h_n, \quad (2.116)$$

with

$$\mathcal{B}_{n,q}^{(a,b)}(x_0, y_0) = \frac{\mathcal{B}_{i,q}^{(a)}(\tau(x_0)) \mathcal{B}_{j,q}^{(b)}(\mu(y_0))}{d_x^a d_y^b}. \quad (2.117)$$



### 2.7.2.2 Curvature of a B-Spline Surface

Analog to the curves, we introduce prior knowledge on smoothness when estimating the street spline surfaces in Section 3.5.2.1.

As measure of the local curvature in the  $xy$  space at the position  $[x_0, y_0]^T$ , we use the quadratic variation given by the trace of the squared Hessian

$$\text{tr} (H_h^2(x_0, y_0)) = \left( \frac{\partial^2}{\partial x^2} h(x_0, y_0) \right)^2 + 2 \left( \frac{\partial^2}{\partial x \partial y} h(x_0, y_0) \right)^2 + \left( \frac{\partial^2}{\partial y^2} h(x_0, y_0) \right)^2. \quad (2.118)$$

The trace of the squared Hessian yields an approximation for the *mean squared curvature*

$$\overline{\kappa^2}_{x_0, y_0} \approx \frac{1}{2} \text{tr} (H_h^2(x_0, y_0)) \quad (2.119)$$

since we assume the slope of the street surface to be small, i.e. we assume the first derivatives of  $h$  with respect to  $x$  and  $y$  to vanish. We refer to Grimson [1981] for further details.

Inserting Equation (2.116), we obtain a formulation of the local curvature measure as function of the control point heights

$$\begin{aligned} \overline{\kappa^2}_{x_0, y_0}([h_1, \dots, h_N]^T) \approx \\ \frac{1}{2} \left( \sum_{n=1}^N \mathcal{B}_{n,q}^{(2,0)}(x_0, y_0) h_n \right)^2 + \left( \sum_{n=1}^N \mathcal{B}_{n,q}^{(1,1)}(x_0, y_0) h_n \right)^2 + \frac{1}{2} \left( \sum_{n=1}^N \mathcal{B}_{n,q}^{(0,2)}(x_0, y_0) h_n \right)^2. \end{aligned} \quad (2.120)$$

### 2.7.3 B-Spline Basis Functions

We use the recursive definition of the B-spline basis functions given by the Cox-de Boor recursion formula [De Boor, 1972]. The  $m$ th B-spline basis function of degree  $q$  governed by the knot vector  $\boldsymbol{\tau} = [\tau_1, \dots, \tau_{M+q+1}]^T$  is defined as

$$\mathcal{B}_{m,q}(\tau) = \frac{\tau - \tau_m}{\tau_{m+q} - \tau_m} \mathcal{B}_{m,q-1}(\tau) + \frac{\tau_{m+q+1} - \tau}{\tau_{m+q+1} - \tau_{m+1}} \mathcal{B}_{m+1,q-1}(\tau) \quad (2.121)$$

$$\mathcal{B}_{m,0}(\tau) = \begin{cases} 1 & \text{if } \tau_m \leq \tau_{m+1} \\ 0 & \text{otherwise.} \end{cases} \quad (2.122)$$

The  $k$ th derivative of the B-spline basis function is given by the closed form

$$\mathcal{B}_{m,q}^{(k)}(\tau) = q \left( \frac{\mathcal{B}_{m,q-1}^{(k-1)}(\tau)}{\tau_{m+q} - \tau_m} - \frac{\mathcal{B}_{m+1,q-1}^{(k-1)}(\tau)}{\tau_{m+q+1} - \tau_{m+1}} \right). \quad (2.123)$$

### 2.7.4 Handling of Points beyond Outer Section Borders

When discussing the temporal filtering of the street surface model in Section 3.4.1.1, we will encounter the problem of dealing with point coordinates aligned with parameter values beyond the outer section borders (beyond the outer green borders in Figure 2.6(b)), i.e.  $[\tau, \mu]^T \notin \{[0, I - q] \times [0, J - q]\}$ . We treat those points as if their parameter is included in the closest outer section when evaluating the case differentiation (2.122). In simple terms, the local surface of the respective most outer section is expanded in the undefined area.



## Chapter 3

# Concept of an Approach for Identification and Reconstruction of Street Surfaces and Boundaries

In this chapter, we present the concept of an approach for the detection and reconstruction of street surfaces and boundaries from depth image sequences captured from a moving platform. The approach is designed to match the requirements and conditions formulated in the problem statement given in Section 1.3.

The subsequent section provides an overview which sketches the objective and the total workflow of the approach. The remainder of this chapter gives a detailed description of the computation steps, mathematical entities and model assumptions.

### 3.1 Overview

This section gives a brief description of the input data, objective and workflow of the approach including its single processing and key ideas. Figure 3.1 provides a diagrammatic illustration of the total workflow. We use the roman numerals introduced in the figure to refer to the single processing steps.

**Input Data:** The proposed algorithm is performed for each acquisition time step of a continuous data stream. The input data is given by depth maps containing 3d information about the area in front of the vehicle. In our examples they are extracted from image pairs of a stereo camera system via the Semi-Global Matching (SGM) approach introduced in Section 2.2.4. The stereo matching procedure corresponds to processing step (I) in Figure 3.1.

Further, information about the 3d ego-motion of the test vehicle and, thus, of the cameras is provided. Both data sources are required to be synchronized. Section 3.2 gives a detailed summary of the data acquisition, related precomputation steps and coordinate systems.

**Objective:** The key objective of the proposed approach is to determine and model the drivable free-space in front of the vehicle for every depth map acquisition time step. This comprises the detection and reconstruction of the street surface and the horizontal street boundary, i.e. the estimation of the parameters of a continuous representation of both entities. The term *street surface* defines the vertical elevation of the street, while the term *street boundary* defines the first geometric delimitation of the drivable street area, i.e. height differences of at least 10 cm in positive or negative direction. Without providing any further

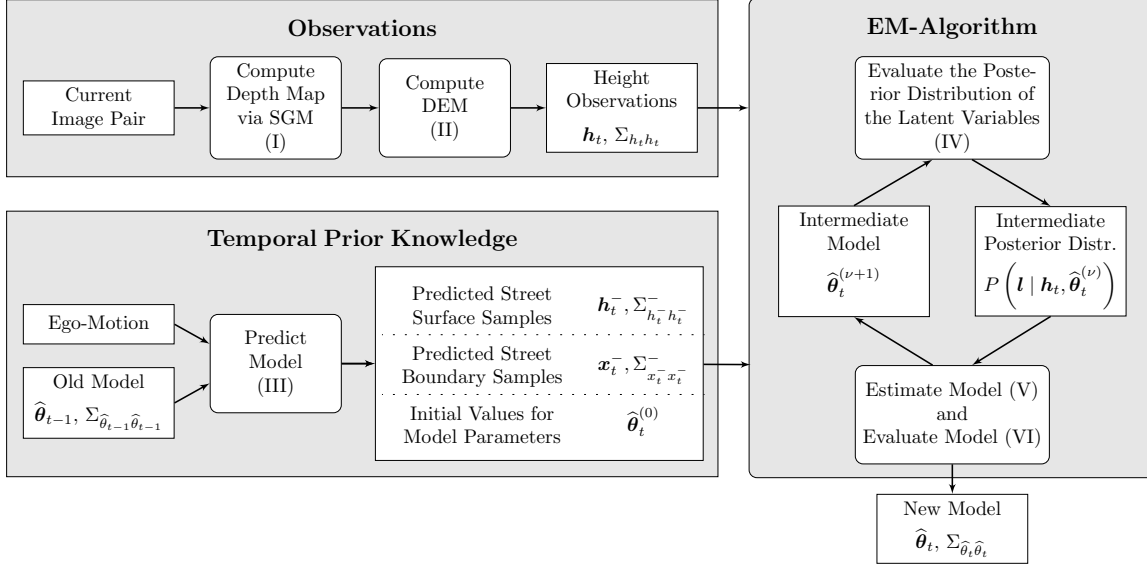


FIGURE 3.1: Workflow illustrating the main processing steps (boxes with rounded corners) at each image acquisition time step  $t$ . The rectangular shaped boxes present the involved entities such as the current input data, transfer parameters and results as well as their respective precisions are represented by covariance matrices. The total workflow is divided in three steps: (a) The extraction of the digital elevation map (DEM) from the image data. The height values  $\mathbf{h}_t$  stored in the DEM form the observations for the estimation of the model parameters  $\boldsymbol{\theta}$  and the evaluation of the posterior distribution of the latent variables  $\mathbf{l}$ ; (b) The processing of temporal prior information for temporal filtering as well as approximate values for the model parameters  $\hat{\boldsymbol{\theta}}_t^{(0)}$  employed to initialize the EM-Algorithm. The prior information is computed based on the estimation result of the previous time step  $\hat{\boldsymbol{\theta}}_{t-1}$  and information of the vehicle's egomotion. It consists of predicted sample points on the street surface  $\mathbf{h}_t^-$  and boundary  $\mathbf{x}_t^-$ ; (c) The estimation of the current model parameters  $\hat{\boldsymbol{\theta}}_t$  via the EM-algorithm. The symbol  $\nu$  represents the iteration counter of the EM-algorithm; The remainder of this chapter gives a detailed description of the single steps and meanings of the symbols. We will use the roman numerals assigned to the rounded boxes to refer to the respective processing steps.

details on the specific definition at this juncture, let us denote the unknown model parameters by the parameter vector  $\boldsymbol{\theta}$  which is composed of the parameters of the street surface  $^S\boldsymbol{\theta}$  and boundary  $^B\boldsymbol{\theta}$ .

The approach shall not be limited to a special type of delimiting obstacles, e.g. curbs or vehicles, but be generic enough to model the geometric delimitation of the free-space caused by arbitrary obstacles in short-range. The experiments in Section 4 show the actual detection range to be limited to distances between 10 and 20 m to the sensor. The detection range is governed by the ratio of the accuracy of the sensor data and the height of the free-space delimiting obstacle.

Note that we do not aim to reconstruct the surface structures in regions beyond the street boundary.

Furthermore, in the context of driver assistance systems the approach should be real-time capable to allow for online processing. Figure 3.2 illustrates an example for a reconstruction result in order to provide a visual explanation of the meaning of drivable and non-drivable regions, street surface and street boundary.

**Workflow and Key Idea:** This paragraph gives a brief description of the algorithm's total workflow and sketches the key idea. It further provides an explicit definition of the overall optimization task and introduces simplifying assumptions. For this purpose, the major components of the estimation model are roughly introduced. A detailed description is

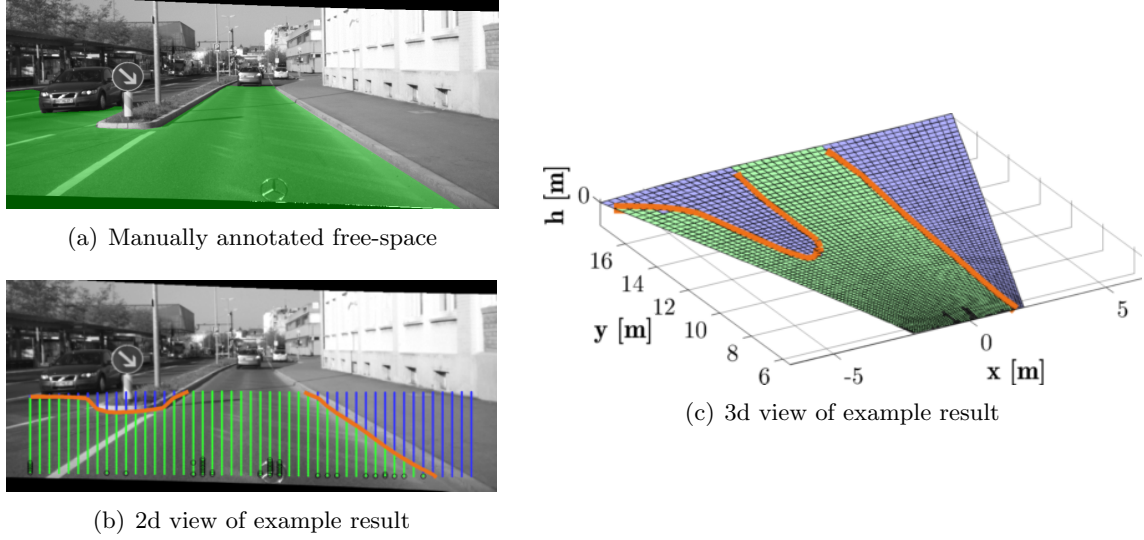


FIGURE 3.2: Example for a reconstruction result. The left input image is shown in (a). The true drivable free-space is manually annotated by a green carpet. In (c), a 3d model of the reconstruction result is illustrated. The plotted surface represents the reconstructed street surface (extended into non-street regions). The colors encode the estimated most probable assignment, i.e. green colored dots mark *drivable* regions (alias street regions), while blue colored dots mark *non drivable* regions (alias non-street regions). The orange line represents the reconstructed horizontal street boundary. Figure (b) depicts the backprojection of the result to the left input image.

provided later on in Section 3.3.

**Preprocessing:** Two preprocessing steps are performed for each acquisition time step  $t$ . First, an elevation model is computed from the incoming depth measurements (processing step (II)). It is represented by a *Digital Elevation Map* (DEM)  $\mathcal{D}$  providing discrete height measurements  $\mathbf{h}_t = [\dots, h_{ij,t}, \dots]^\top$  together with their precision  $\Sigma_{h_t h_t}$ . The height measurements are aligned to the cell centers  $\mathbf{x} = [\dots, \mathbf{x}_{ij}, \dots]^\top$  of a horizontal grid, with  $(ij) \in \mathcal{D}$ . The grid is located in front of the vehicle with the horizontal cell coordinates fixed to the camera system. Thus, observed from the camera, the horizontal position of the height measurements do not change over time. However, they possess an uncertainty caused by the grid discretization that we model by  $\Sigma_{xx}$ .

In addition to the height measurements, the estimated model parameters of the previous time step  $\hat{\boldsymbol{\theta}}_{t-1}$  together with their precision  $\Sigma_{\hat{\boldsymbol{\theta}}_{t-1} \hat{\boldsymbol{\theta}}_{t-1}}$  are known. This model is predicted to the current time step in the second preprocessing step (processing step (III)) in order to yield both temporal prior knowledge and approximate values for the estimation of the current model parameters.

**Optimization Task:** This rough introduction of the model and its involved entities allows us to formulate the objective stated in the previous paragraph as optimization task. We define the desired model parameters to be embodied by the most probable parameter assignment given the introduced information

$$\hat{\boldsymbol{\theta}}_t = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[ \wp \left( \boldsymbol{\theta} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_{t-1}, \mathbf{x}, \Sigma_{h_t h_t}, \Sigma_{xx}, \Sigma_{\hat{\boldsymbol{\theta}}_{t-1} \hat{\boldsymbol{\theta}}_{t-1}} \right) \right]. \quad (3.1)$$

In order to keep the notation uncluttered, we omit the explicit representation of the dependency on the covariance matrices and the constant grid poses  $\mathbf{x}$  in the following. Using

this simplified notation, we reformulate (3.1) as a minimization task employing the negative logarithm

$$\hat{\theta}_t = \underset{\theta}{\operatorname{argmin}} \left[ -\ln \wp \left( \theta \mid \mathbf{h}_t, \hat{\theta}_{t-1} \right) \right] \quad (3.2)$$

$$= \underset{\theta}{\operatorname{argmin}} \left[ -\ln \wp \left( \mathbf{h}_t \mid \theta \right) - \ln \wp \left( \theta \mid \hat{\theta}_{t-1} \right) \right]. \quad (3.3)$$

The second transformation step follows from the proportionality  $\wp(a \mid b) \sim \wp(b \mid a)\wp(a)$  yielded by Bayes' theorem. It provides a representation of the task in terms of a likelihood and a prior for the unknown model parameters. Note that we assume the likelihood term to be independent of the parameters estimated in the previous time step, i.e.  $\wp(\mathbf{h}_t \mid \theta, \hat{\theta}_{t-1}) = \wp(\mathbf{h}_t \mid \theta)$ .

The prior term encodes model knowledge of temporal and spatial smoothness. By 'temporal smoothness' we mean that the new model should be consistent with the samples taken from the previous model in processing step (III).

The likelihood encodes the relation of model parameters and observed height values. The idea is to estimate the street surface from the height observations of all cells located in the drivable street area. The street boundary shall be estimated as border which separates the cells located in the drivable area from those located in the non-drivable area, as depicted in Figure 3.2. For this purpose, we need additional information about the affiliation of each cell with respect to the drivable area in order to evaluate the likelihood in (3.3). Assume this information is encoded in the discrete latent variables  $\mathbf{l} = [\dots, l_{ij}, \dots]^\top$  and the postulated relations between the model parameters and the *complete* dataset  $\{\mathbf{h}_t, \mathbf{l}\}$  are encoded in the *complete-data likelihood*  $\wp(\mathbf{h}_t, \mathbf{l} \mid \theta)$ . We obtain the final formulation of the task exploiting that  $\wp(\mathbf{h}_t \mid \theta)$  follows from marginalization of the complete-data likelihood over  $\mathbf{l}$

$$\hat{\theta}_t = \underset{\theta}{\operatorname{argmin}} \left[ -\ln \sum_{\mathbf{l} \in \mathcal{L}} \wp(\mathbf{h}_t, \mathbf{l} \mid \theta) - \ln \wp \left( \theta \mid \hat{\theta}_{t-1} \right) \right], \quad (3.4)$$

where  $\mathcal{L}$  denotes the set of all possible assignments for  $\mathbf{l}$ . The specific definitions of the likelihood and prior terms are described in the later sections. In advance, Figure 3.3(a) provides a graphical representation of the conditional independence properties which are assumed for the entities involved in the optimization task.

**Optimization Procedure:** The exact optimization of (3.4) is infeasible considering the background of a real-time application. This is because the logarithm acts on the sum rather than directly on the likelihood terms. Since we employ normal distributions to model the densities  $\wp(\mathbf{h}_t, \mathbf{l} \mid \theta)$  the solution for (3.4) results in a quite complex expression. In order to overcome this challenges, we approximate the solution using the iterative procedure of the Expectation Maximization algorithm (EM-algorithm) presented in Section 2.6. This means that the log likelihood  $\ln \wp(\mathbf{h}_t \mid \theta)$  is approximated by its expected value under the posterior distribution of the latent variables  $P(\mathbf{l} \mid \mathbf{h}_t, \hat{\theta}_t)$

$$\ln \wp(\mathbf{h}_t \mid \theta) \approx \sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l} \mid \mathbf{h}_t, \hat{\theta}_t) \ln \wp(\mathbf{h}_t, \mathbf{l} \mid \theta) \quad (3.5)$$

rather than computed by exact marginalization over  $\mathbf{l}$  as in (3.4). Since the desired parameters  $\hat{\theta}_t$  are needed for the evaluation of the posteriors, the estimation is performed in an

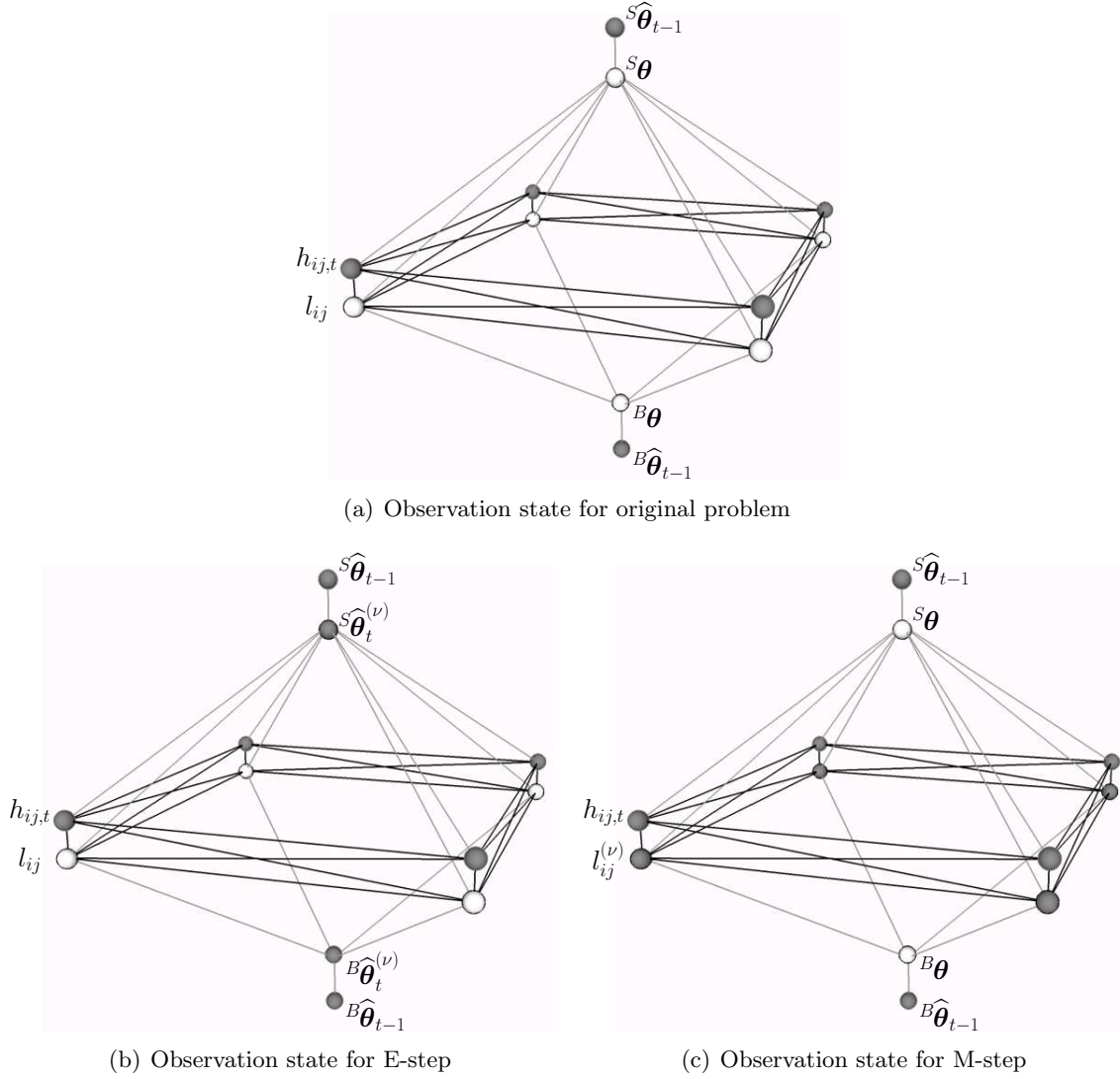


FIGURE 3.3: Graphical representation of the conditional independence properties of the entities involved in the parameter estimation process at acquisition time step  $t$  in the manner of a Conditional Random Field. For simplification, only a  $2 \times 2$  subset of the DEM's grid cells is depicted. Dark shaded nodes are considered as observed, i.e. the value of the corresponding variable is known and fixed. Bright shaded nodes are considered as unobserved. The different colors of the edges are used to improve visibility and have no further meaning. Figure (a) depicts the initial situation with labels and model parameters unobserved. The remaining figures illustrate the observation state for the single steps of the EM-algorithm. In the E-step (Figure (b)), the model parameters are treated as observed using the values  $S_{\hat{\theta}_t}^{(\nu)}$  and  $B_{\hat{\theta}_t}^{(\nu)}$  estimated in the M-step of the previous iteration. Considering the global Markov property (Section 2.4.1.1), the graph structure implies that the conditional distribution of each variable  $l_{ij}$  conditioned on all other variables only depends on the model parameters, its corresponding height observation as well as the latent variables and height observations of its direct grid neighbors. In the M-step (Figure (c)), the latent variables are treated as observed using subsequently each possible assignment  $l_{ij} \in \mathcal{L}$ . The new parameter estimate is achieved by the expected value resulting from all assignments, as defined in (3.6). The graph structure demonstrates that the parameter sets  $S_{\theta}$  and  $B_{\theta}$  are independent given the  $h_{ij,t}$  and  $l_{ij}$  and, thus, can be estimated separately in the M-step.

iterative manner

$$\hat{\boldsymbol{\theta}}_t^{(\nu+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[ - \sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l} | \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln \wp(\mathbf{h}_t, \mathbf{l} | \boldsymbol{\theta}) - \ln \wp(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}_{t-1}) \right]. \quad (3.6)$$

The symbol  $\nu$  represents the iteration counter.

Starting with an initial parameter guess  $\hat{\boldsymbol{\theta}}_t^{(0)}$  achieved from the processing step (III), three iterative steps are alternately accomplished.

In the Estimation step (processing step (IV)), briefly denoted as E-step, the posterior probabilities  $P(\mathbf{l} | \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})$  for all possible assignments of the latent variables  $\mathbf{l} \in \mathcal{L}$  are evaluated. For this purpose, the parameters estimated in the previous iteration  $\hat{\boldsymbol{\theta}}_t^{(\nu)}$  are kept fixed. In order to efficiently evaluate the posterior probabilities in an approximative manner, we employ the inference tools for CRFs mentioned in Section 2.4.

The second step (processing step (V)) of the iterative procedure is embodied by the Minimization step, briefly denoted as M-step. It embodies the estimation of the parameters via (3.6) based on the posterior probabilities evaluated in the E-step. As opposed to the literature, we refer to this step as ‘Minimization step’ rather than as ‘Maximization step’ since we address a minimization task. We will find that the total estimation task decomposes into two independent tasks for estimating the parameters of the street surface  $^S\boldsymbol{\theta}$  and street boundary  $^B\boldsymbol{\theta}$ , respectively. This becomes explicit referring to Figure 3.3(c). The figure provides an overview of which entities are treated as unknown and which as observed during the M-step. The parameter sets  $^S\boldsymbol{\theta}$  and  $^B\boldsymbol{\theta}$  are conditional independent given the latent variables  $\mathbf{l}$  and heights  $\mathbf{h}$  are observed since there exist no connecting path through unobserved variables. Thus, we solve (3.6) by estimating both parameter sets independently based on the ideas roughly sketched in the following. The street surface parameters are estimated in a least squares sense using the MAP estimation procedure presented in Section 2.3.2.2. The idea regarding the street boundary is as follows. We estimate a classification model that provides probabilities for the affiliation to the drivable area or non-drivable area for arbitrary horizontal positions in front of the vehicle. This model is based on the logistic regression model presented in Section 2.5 and is estimated from the posterior probabilities obtained in the E-step. The street boundary is then given by the decision boundary of the classification model which represents the separation of drivable from non-drivable areas.

As a final step for each iteration, a self diagnostic check (processing step (VI)) on the estimated environment model is performed after the M-step. Its purpose is to check for convergence of the iterative procedure and to detect local or global violations of the model assumptions in order to evaluate the validity of the result.

**Organization of the Chapter:** The single components and processing steps of the overall approach are introduced in the following sections.

First, Section 3.2 describes the procedure for the acquisition and preprocessing of the depth map data obtained from processing step (I). This includes the definition of the involved coordinate systems and the computation of the elevation model (processing step (II)). Further, the section depicts the acquisition of ego-motion information.

Section 3.3 provides the definition of the geometric and probabilistic properties of the environment model which forms the basis for the estimation process. This includes a formal definition of the model parameters and their relation to the observations and latent variables. Further, the section describes the structure of the prior distribution  $\wp(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}_{t-1})$  which



composes of prior assumptions about the spatial and temporal behavior of the model. The specific definition of the spatial prior is given at this point.

The specific definition of the temporal prior is addressed in Section 3.4 within the scope of the prediction step (processing step (III)). This section further comprises the prediction of the current model for the initialization of the EM-algorithm based on the results estimated in the previous acquisition time step.

Finally, Section 3.5 describes the estimation of the model parameters governed by the EM-algorithm and provides the specific definition of the likelihood  $\wp(\mathbf{h}_t, \mathbf{l} \mid \boldsymbol{\theta})$  and the posterior  $P(\mathbf{l} \mid \mathbf{h}_t, \boldsymbol{\theta})$ . The section is subdivided in the description of the Estimation step (processing step (IV)), the Minimization step (processing step (V)) and the evaluation of the estimated result (processing step (VI)).

## 3.2 Data Acquisition and Preprocessing

This section gives a detailed description of the representations and algorithms used to extract the relevant input data for a single acquisition time step. The approach presented in this thesis is mainly based on the interpretation of an elevation model of the respective observed world region. The elevation model is extracted from synchronously captured depth maps, i.e. all measurements in the map are acquired simultaneously. The depth maps are obtained from a stereo camera system using the SGM algorithm for depth computation. Other sensors providing synchronously captured depth maps, such as time-of-flight cameras, would also be suitable. Figure 3.8 shows an example scenario including the triangulated point cloud. For comparison, the respective extracted elevation model is depicted in Figure 3.8(c).

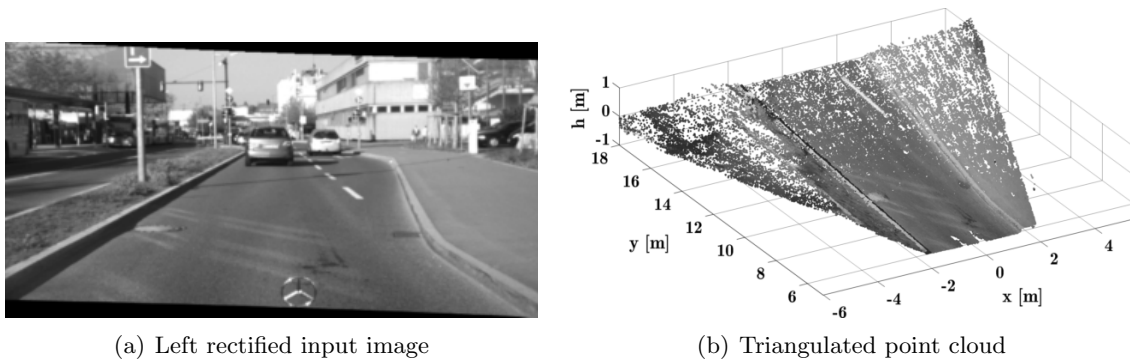


FIGURE 3.4: Left image of a stereo pair (a) and the corresponding point cloud triangulated from SGM depth measurements (b) plotted in image gray values. The task is to detect and model the street surface as well as the horizontal boundary of the street.

The section is organized as follows. After a brief description of the sensor setup in Section 3.2.1, we present the extraction of the elevation model from the captured depth map in Section 3.2.2. This corresponds to processing step (II) in the total workflow (see Figure 3.1) and includes the definition of the underlying assumptions and requirements, involved coordinate systems and structures, as well as the estimation of the elevation and elevation uncertainties. Section 3.2.3 finally describes the acquisition of information about the ego-vehicle's three dimensional motion between successive time steps, which is required for temporal filtering issues.

### 3.2.1 Sensor Setup

The vehicle's environment is captured by a stereo camera system mounted behind the windshield. The cameras are assumed to be calibrated and rectified to the normal case, as described in Section 2.2. The intrinsic parameters as well as the position and orientation of the camera relative to the vehicle is assumed to be known and accurate. Thus, 3d coordinates can be computed from the depth map using (2.20) to (2.22).

The viewing direction of the camera approximately matches the driving direction of the vehicle. Further, it is approximately leveled to the local ground surface, i.e. the deviation of the upward direction of the rectified camera system  $c_y$  to the normal of the plane through the vehicles axles is assumed to be small.

Figure 3.5(a) illustrates the mounting of the stereo camera system.

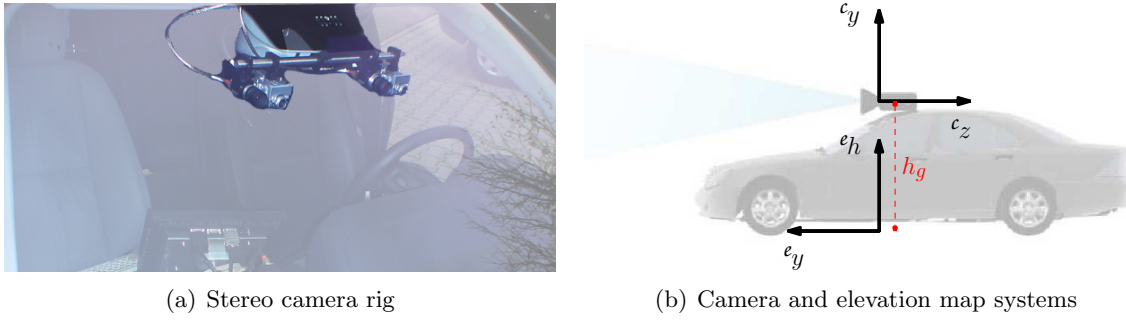


FIGURE 3.5: Utilized stereo camera system (a) mounted behind the windshield of the car. Figure (b) illustrates the relative orientation between the camera system  $c$  and the elevation map system  $e$ . The  $x$ -axes of both systems point into the image. The distance  $h_g$  denotes the height of the camera above the ground level.

### 3.2.2 Probabilistic Elevation Map Computation

The representation used to capture the elevation should meet the following requirements:

- (a) It needs to be *efficient* in both computation and access on neighborhood relations.
- (b) It shall be *dense* which means that elevation information is provided homogeneously in the observed region of interest.
- (c) It shall be *robust* in dealing with outliers and measurement uncertainties.
- (d) It needs to be *accurate* enough to model even small obstacles, such as curbs of at least 10 cm in height.

To meet these requirements, we compute a digital elevation map (DEM)  $\mathcal{D}$  from the depth information achieved from SGM stereo computation, i.e. a discrete 2.5d model storing unique height values for the cells of a regular grid which is parallel to the horizontal plane of the camera system. Grid based data structures are widely used for environment modeling and obstacle detection tasks since they are efficient in handling and allow for a direct access to neighborhood relations. The local height values are estimated by means of a probabilistic model which evaluates occlusion and free-space information yielded by the depth measurements. This brings two major advantages. First, assuming the elevation of occluded areas to match the elevation of the occluding object, the extracted elevation information becomes more dense. Second, free-space information allows to outvote the erroneous elevation statement induced by measurement outliers.

In the following section, we give a detailed description of the utilized DEM structure and introduce the corresponding notation relevant for the succeeding sections. Note that the design of the DEM's grid structure is customized for the usage of a stereo camera system and may be altered to be more suitable when employing alternative sensors.

Section 3.2.2.2 describes the probabilistic approach employed for the estimation of the local cell elevations.

Finally, we discuss the stochastic properties of the DEM in Section 3.2.2.3.

### 3.2.2.1 Coordinate System and Grid Structure of the Digital Elevation Map

The local coordinate system  $S_e$  (the index  $e$  stands for elevation map) containing the elevation map is defined to be parallel to the horizontal plane of the coordinate system of the left camera, as illustrated in Figure 3.5(b). In simple terms, the local  $y$ -axis is defined by the cameras viewing direction, whereas the  $x$ -axis points right. The right handed system is completed by the  $h$ -axis representing the elevation coordinate (alias height coordinate). The origin is located on ground level, straight under the projection center of the left camera. Thus, the position of the camera in the elevation map system is fully described by its height above the ground  $h_g$ . Note that we omit the superscript  $e$  at the axes for a shorter notation and, thus, all axis specifications refer to this system if not otherwise stated.

The DEM's grid structure is aligned to the  $xy$ -plane. The cell indices  $(ij) \in \mathcal{D}$  count from near to far  $j = 1 \dots J$  in longitudinal direction and from left to right  $i = 1 \dots I$  in lateral direction. We use the term *longitudinal column* (or *y-column*) to address all cells with the same value in  $i$  and analogously *lateral column* (or *x-column*) for all cells with the same value in  $j$ .

The coordinates of the cell centers are denoted by  $\mathbf{x}_{ij} = [x_{ij}, y_{ij}]^T$  and collected in the vector  $\mathbf{x} = [\dots, \mathbf{x}_{ij}^T, \dots]^T$ . The corresponding estimated height values are given by the vector  $\mathbf{h} = [\dots, h_{ij}, \dots]^T$ . The aggregate of all three coordinates for a single point is denoted by the 3-vector  $\mathbf{X}_{ij} = [x_{ij}, y_{ij}, h_{ij}]^T$ .

Instead of arranging the grid cells paraxial to the Cartesian axes, we design the grid structure to be regular in the image columns with increasing cell sizes in  $y$  direction resulting in the trapezoidal shape presented in Figure 3.6(a). This choice is motivated by three reasons:

- It allows for an efficient tracing of projection rays through the grid cells in order to extract free-space and occlusion information.
- It provides a suitable coverage of the field of view (FOV) of the camera, i.e. it does not hold any cells that lie outside the FOV.
- The density and precision of depth measurements decreases with increasing distance to the camera favoring the usage of increasing cell sizes.

The property 'regular in the image columns' means that each cell center on the grid's longitudinal column  $i$  projects back to the same image column  $u_i$  yielding the relation

$$\frac{x_{ij}}{y_{ij}} = \frac{u_i}{c}, \quad (3.7)$$

where  $c$  denotes the principle distance of the camera. Further, the grid's  $x$ -spacing  $d_{x_j}$  is governed by a constant pixel spacing  $d_u$

$$d_{x_j} := \frac{d_u}{c} y_{ij}. \quad (3.8)$$

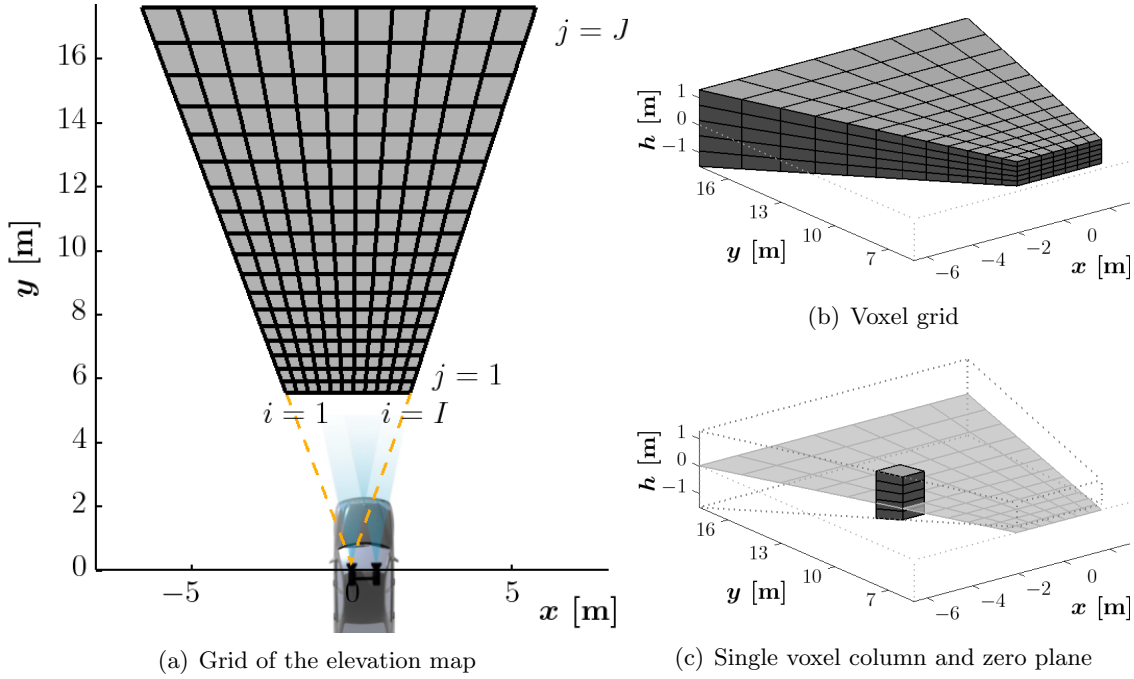


FIGURE 3.6: Grid structure of the DEM (a), voxel grid (b) and single voxel column (c). The left camera is positioned at the origin. The  $y$ -axis corresponds to the camera's viewing direction. Note, the resolution is scaled down for illustration issues.

To make a compromise between measurement per cell and a homogeneous allocation of the grid cells, we design the grid cells to grow equally in both dimensions with increasing distance to the camera. Thus, the  $y$ -spacing  $d_{y_j}$  is defined to match the local  $x$ -spacing at the cell center

$$d_{y_j} := d_{x_j}, \quad (3.9)$$

to form nearly isotropic cells, as illustrated in Figure 3.6(a).

The limits of the grid's extent in lateral longitudinal direction are denoted by  $[x_{\min}, x_{\max}]$  and  $[y_{\min}, y_{\max}]$  respectively. The extent in image columns is denoted by  $[u_{\min}, u_{\max}]$ .

We refer to Section A for a detailed description of the computation of the cell center coordinates and the cell assignment of depth measurements.

### 3.2.2.2 Local Elevation Estimation

There exist various strategies to compute the local elevation values  $h_{ij}$  for the cells  $(ij)$  of the DEM from depth measurements. The straightforward way is to extract the height value from the triangulated heights of all measurements assigned to a cell due to their triangulated horizontal position. A common height value for the cell could then be determined by the mean of the assigned measurements or by the highest value as in [Nedevski et al., 2007].

However, in addition to the triangulated point position, depth estimates yield information about free-space and occlusion by tracing the optical ray's path. Similar to the work of Wellington et al. [2006], we use this information to determine the local elevations in a probabilistic way. The approach is similar to the idea of space carving techniques and 3d occupancy grids, see Section 1.2.2.1.

To model free-space and occlusion information, we extend the horizontal grid structure to a voxel grid by assigning a vertical stack of  $\mathcal{V}$  voxels (alias *voxel column*) to each cell of

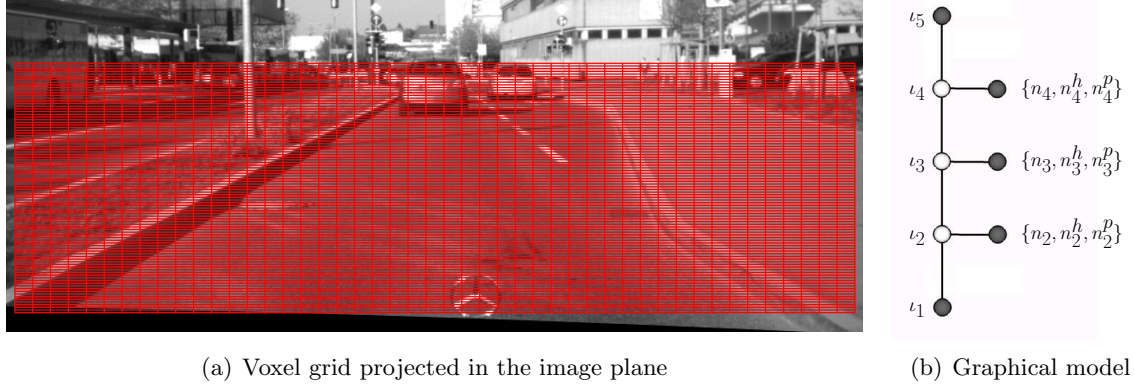


FIGURE 3.7: Figure (a): Voxel grid projected in the image plane. The spacings in this example are chosen by  $d_u = 20$  pel and  $d_v = 3$  pel. Figure (b): Graphical model for the estimation of the most probable assignment of the classes *solid*, *surface*, and *free-space* to the voxels  $v \in \{1, \dots, \mathcal{V}\}$  of a single voxel column. For visualization issues, a simplified model with  $\mathcal{V} = 5$  is depicted. Further, the respective numbers of valid observations  $n_v$  as well as the numbers of hits  $n_v^h$  and pass-throughs  $n_v^p$  are collected in a single node connected to the respective class label  $\iota_v$ . Note that the lowest and highest nodes are observed since the lowest voxel  $v = 1$  is permanently assigned to *solid* and the highest voxel  $v = \mathcal{V}$  is permanently assigned to *free-space*.

the elevation grid, as illustrated in Figure 3.6(b). The vertical voxel spacing is defined to be regular in the image rows governed by a constant spacing  $d_v$ . Figure 3.7(a) depicts the projection of the voxel grid to the image.

The idea for the elevation estimation for a specific cell is to determine the voxel of the respective vertical column which most likely contains the local surface, i.e. the vertical separation between solid ground and free-space. The vertical coordinate of the corresponding voxel center embodies the desired elevation value. The basic concept for this approach, which is based on a probabilistic model, is presented in the following paragraph. The subsequent paragraph describes the learning procedure for the related probabilistic entities, i.e. likelihoods and transition probabilities.

**Concept for the Probabilistic Estimation of the Local Elevation:** The basic approach for the estimation of the local elevation of a DEM cell is as follows. We decide for each voxel in the respective vertical column whether its volume contains *free-space* or is *solid*, i.e. blocked or occluded by an obstacle. The local surface elevation is then obtained from the voxel which most likely represents the vertical boundary between *free-space* and *solid* voxels, i.e. the upper boundary of the local surface structure. We explicitly denote this boundary by the class *surface*. This explicit representation by a separate class allows to model the appearance of the *surface* voxels with respect to the observations and obviates the discussion of whether the highest *solid* voxel or the lowest *free-space* voxel would be more suitable to describe the surface.

The labeling decision for a certain voxel is governed by the depth measurements whose optical ray paths coincide with the voxel volume, i.e. whose pixel coordinates are included in the image projection of the voxel (see Figure 3.7(a)). There are three possible ways a depth measurement may affect a voxel. It may exactly fall in this voxel causing a *hit*, may hit a more distant voxel causing a *pass-through*, or hit a nearer voxel implying the current voxel to be *occluded*. The respective numbers of hits, pass-throughs and occlusions for the voxels  $v = 1, \dots, \mathcal{V}$  form the observations for the classification task and are denoted by  $n_v^h$ ,  $n_v^p$ , and  $n_v^o$ . Note that the number of states and thus the amount of incorporated information differs from the work of Wellington et al. [2006] where only binary states (*hit* or *pass-through*)

are considered.

In summary, the state  $\iota_v$  of a voxel  $v$  is given by either *solid*, *surface* or *free-space*. For each voxel column we search for the most probable path from the lowest voxel of the column, permanently assigned to *solid*, to the highest voxel, permanently assigned to *free-space*. Forcing the label *surface* to appear exactly once per voxel column, the desired elevation values  $h_{ij}$  are given by the height coordinates of the voxels assigned to *surface*. This classification is done for each voxel column independently employing the linear-chain CRF model presented in (2.53). This means the joint probability conditioned on the observed numbers and the fixed class assignments of the lowest and highest voxels decomposes into a product of unary and binary terms

$$P(\iota_2, \dots, \iota_{\mathcal{V}-1} \mid \iota_1, \iota_{\mathcal{V}}, \mathbf{n}^o, \mathbf{n}^h, \mathbf{n}^p) \sim \prod_{v=2}^{\mathcal{V}-1} \wp\left(\frac{n_v^h}{n_v}, \frac{n_v^p}{n_v} \mid \iota_v\right) \prod_{v=2}^{\mathcal{V}} P(\iota_v \mid \iota_{v-1}), \quad (3.10)$$

whereas  $\mathbf{n}^o$ ,  $\mathbf{n}^h$  and  $\mathbf{n}^p$  collect the respective numbers  $n_v^o$ ,  $n_v^h$  and  $n_v^p$  for  $v = 2, \dots, \mathcal{V} - 1$ .

Let us have a closer look at the potential functions embodying the unary and binary terms, which are both learned from reference data as described in the subsequent paragraph. We define the unary term for a voxel  $v$  by the likelihood

$$\wp\left(\frac{n_v^h}{n_v}, \frac{n_v^p}{n_v} \mid \iota_v\right), \quad \iota_v \in \{\text{solid}, \text{surface}, \text{free-space}\}, \quad (3.11)$$

governed by the ratio of the number of hits  $n_v^h$  and the number of pass-throughs  $n_v^p$  regarding the total number of pixels  $n_v = n_v^h + n_v^p + n_v^o$  that observe the voxel and have a valid depth value attached. Note that due to the usage of  $n_v$  the number of occlusions  $n_v^o = n_v - n_v^h - n_v^p$  is implicitly modeled in (3.11). DEM cells with less than 20% valid depth observations regarding all assigned pixels of the whole voxel column (e.g. originating from data gaps in the depth map) as well as DEM cells where the whole voxel column does not include any *hits* (completely occluded structures) are marked as invalid and are not used for the estimation of the street surface (see Section 3.5.2.1). The utilized class likelihoods are illustrated in Figure 3.11.

The binary terms  $P(\iota_v \mid \iota_{v-1})$  of (3.10) are represented by the class transition probabilities given in Table 3.1. Note that the probability that a *solid* voxel and a *free-space* voxel are neighbored as well as the probability that a *surface* voxel is neighbored to another *surface* voxel equals zero. Thus, with the lowest and highest voxel permanently assigned to *solid* and *free-space*, each assignment of  $\iota_2, \dots, \iota_{\mathcal{V}-1}$  yielding non-zero probability contains exactly one appearance of class *surface*.

The most probable assignment  $\hat{\iota}_2, \dots, \hat{\iota}_{\mathcal{V}-1}$  and, thus, the position of the *surface* voxel is determined via the max-sum algorithm, see [Bishop, 2006, Chapter 13.2].

An example for an estimated elevation map is illustrated in Figure 3.8(c). Further, Figure 3.9 shows a comparison to alternative, less complex strategies for elevation estimation. The probabilistic approach 3.9(b) demonstrates the ability to cope with the depth measurement errors scattering from the traffic sign due to the evaluation of pass-through information which provides a distinct benefit compared to 3.9(c) and 3.9(d). Moreover, although the effect of height discretization is observable in 3.9(b), the resolution appears to be sufficient to model the elevation of the curb (approximately located at  $y = 10.4$  m). Note, the elevation in the region behind the traffic sign follows the optical ray over the top of the occluding object since the voxels behind the sign are dominated by *occluded* votes. Of course, this prevents a correct representation of the surface behind the traffic sign but, since we only search for the closest obstacle, the approach does not suffer from this issue.

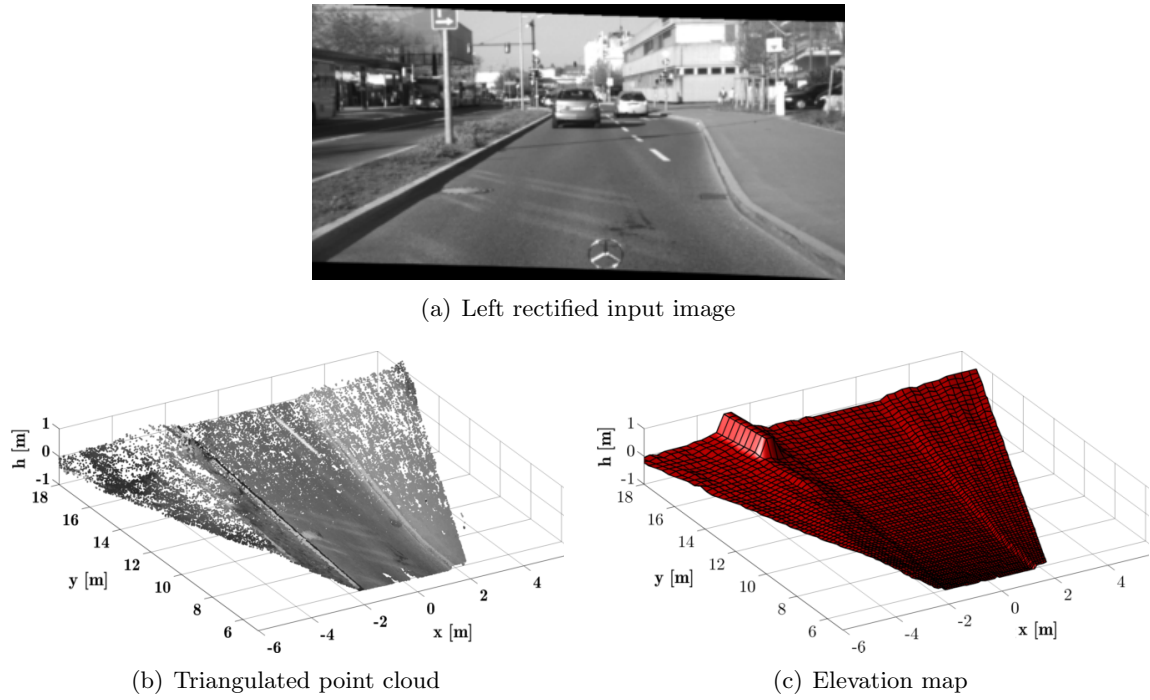


FIGURE 3.8: Figure (c) illustrates the DEM which is computed from depth map information received via SGM from the scenario depicted in (a). For comparison, the triangulated point cloud is depicted in (b) plotted in image gray values.

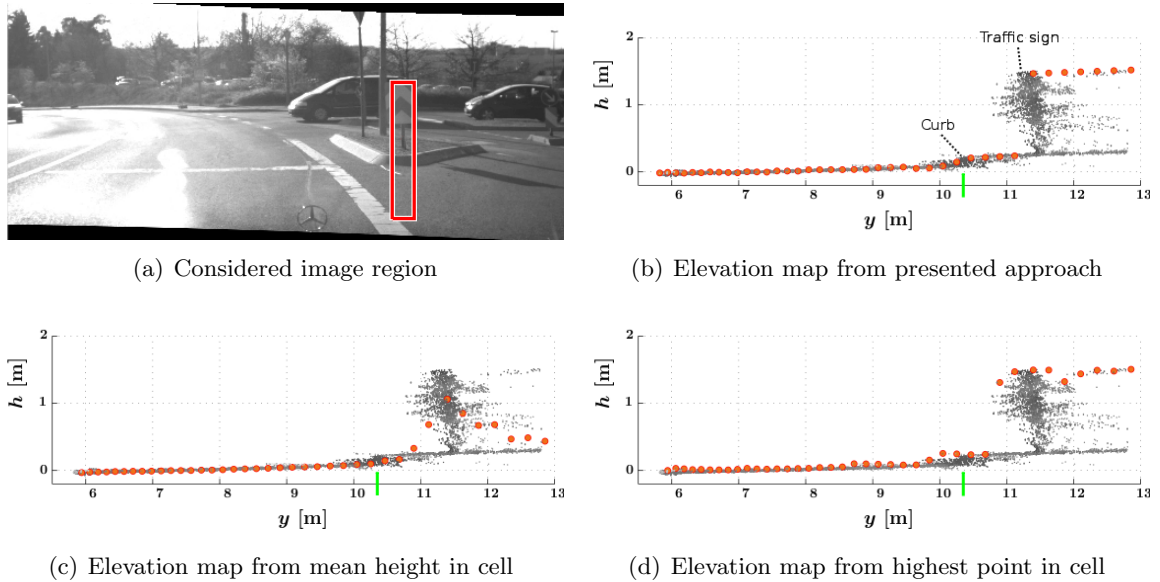


FIGURE 3.9: Elevation maps computed via different strategies for the small horizontal patch framed in (a). The Figures (b)-(d) show the cross section through the point cloud corresponding to the local elevation map cells in camera system coordinates. The computed heights of the cell centers are marked by orange dots. Figure (b) shows the result of the presented probabilistic approach. For comparison the result from taking the mean height (c) and maximum height (d) of all triangulated points assigned to the respective cell is illustrated. The green markings on the horizontal axis indicate the position of the curb.



**Learning of Class Likelihoods and Transition Probabilities:** This paragraph proposes the learning procedure utilized for the determination of the required likelihoods and transition probabilities for the classes *solid*, *surface* and *free-space*.

The class likelihoods and transition probabilities are learned from a set of about 450 reference images containing various obstacles, such as parking cars, walls and pedestrians.

Reference data is obtained from a Velodyne LIDAR HDL-64E<sup>1</sup> scanner mounted on the rooftop of the test vehicle, see Figure 3.10. The sequences are recorded frame by frame in stop and go motion to avoid motion drift in the lidar point cloud and, thus, to allow for a stable calibration to the stereo camera system. For detailed information on the test setup we refer to Pfeiffer et al. [2010].

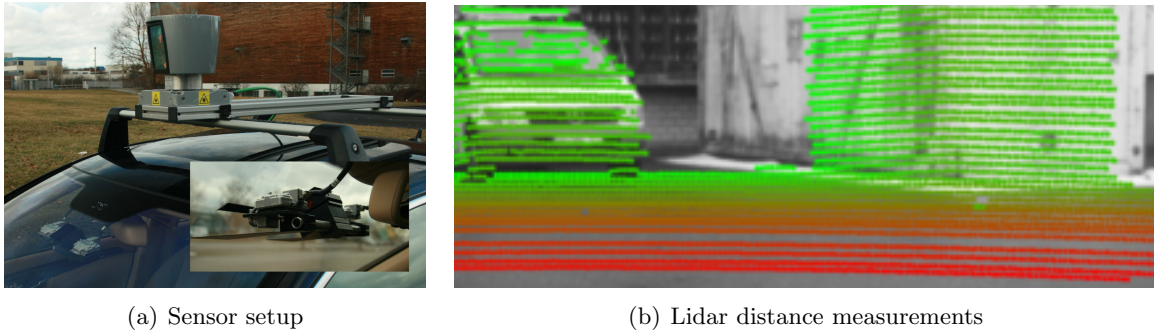


FIGURE 3.10: Illustration of the utilized sensor setup (a). The Velodyne lidar scanner is mounted on a roof rack, while the stereo camera rig is installed behind the windshield of the test vehicle (image source: Pfeiffer et al. [2010]). In (b), an example for a lidar point cloud projected into the image plane is shown. The colors encode the distance to the camera. Red dots represent small distances and green dots large distances. For visibility issues the point cloud is cropped at 18 meter distance.

Ground truth labels are derived as follows. The lidar measurements are assigned to the respective voxels of the grid. For each voxel column containing at least one measurement, the highest voxel containing a lidar point is assigned to *surface*, while all voxels lying above are assigned to *free-space* and voxels below to *solid*. Voxel columns with no lidar measurement are excluded from the learning process.

The class likelihoods are learned based on the ground truth labels and the observed numbers of *hits*, *path-throughs* and *occlusions* obtained from the stereo camera system using kernel density estimation [see Bishop, 2006, chapter 2.5.1]. Thereby we employ symmetric Gaussian kernels governed by a standard deviation of  $\sigma = 0.04$ . Since the domain of the hit and pass-through ratio is bounded, the resulting likelihoods are renormalized over the interval  $\{[0, 1] \times [0, 1]\} \cap \{\frac{n^h}{n} + \frac{n^p}{n} \leq 1\}$ . Moreover, to allow for an efficient access when computing the DEM, we store the likelihoods evaluated at discrete positions in a  $d_u d_v \times d_u d_v$  grid rather than evaluating online from the total set of kernels.

The logarithms of the learned likelihoods are illustrated in Figure 3.11. Let us consider two examples to simplify the interpretation of the figures. A voxel with a low hit ratio of  $\approx 0$  and a high pass-through ratio of  $\approx 1$  obtains likelihood values listed in the upper left corners of the respective plots. Comparison to the colorbar reveals that the voxel is most likely assigned to the class *free-space*. A voxel which shows low values in both ratios obtains the values in the lower left edges. Thus, it is most likely assigned to *solid* since the respective plot yields the highest values for the stated ratios. This seems reasonable because low hit and pass-through ratios indicate a high ratio of occlusions.

<sup>1</sup><http://velodynelidar.com/lidar/lidar.aspx>



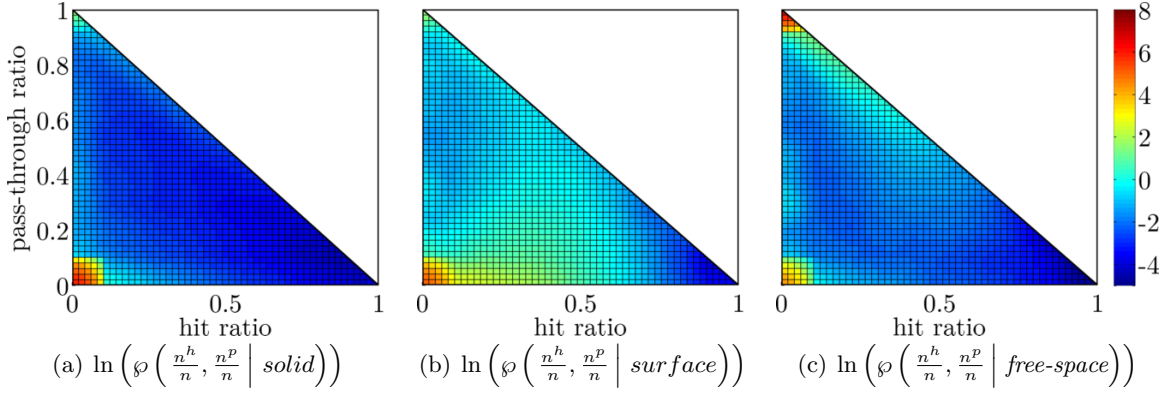


FIGURE 3.11: Logarithms of learned class likelihoods for the classes *solid* (a), *surface* (b) and *free-space* (c) evaluated in a grid with  $50 \times 50$  cells. The triangle structure is caused by the relation  $\frac{n^h}{n} + \frac{n^p}{n} \leq 1$ . We use the terms *hit ratio* and *pass-through ratio* for  $\frac{n^h}{n}$  and  $\frac{n^p}{n}$  respectively.

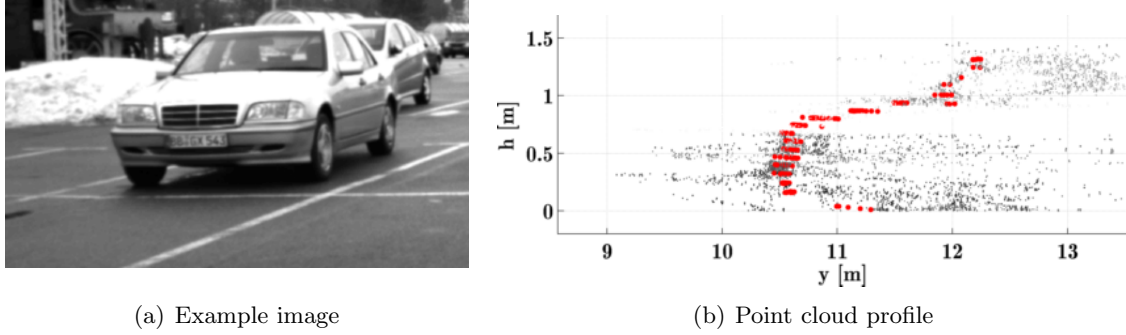


FIGURE 3.12: Example showing a vehicle front (a) and the respective point cloud profile (b) which demonstrates the depth scattering of SGM. The point cloud derived from SGM is plotted in gray values, while the lidar measurements are plotted by red points. The scattering effect appears in particular at horizontal image structures, such as the radiator grill of the car.

A particularly conspicuous aspect is the relatively high amount of occlusions, i.e.  $\frac{n^h}{n} \approx \frac{n^p}{n} \approx 0$ , in the likelihood of the class *free-space*. This can be explained by strong scattering effects in the SGM depth measurements, as shown in Figure 3.12. The erroneous points count as occlusions for all *free-space* voxels on the path between the point's triangulated position and the actual boundary of the obstacle.

The class transition probabilities are derived by simply counting the observed class transitions in the test set, which yields the values presented in Table 3.1.

### 3.2.2.3 Stochastic Properties of the Elevation Map

Let us now discuss the stochastic properties of the elevation measurements concerning their vertical and horizontal precision. We define the horizontal components of the DEM's elevation samples to be given by the respective cell centers. The cell centers therefore act as representatives of the whole cell. This sample position is uncertain with respect to the discretization of the DEM grid, i.e. an arbitrary position inside a cell might be a better representation for the actual object that yields the determined elevation.

The horizontal uncertainties caused by the local discretization are modeled by the vari-

TABLE 3.1: Transition probabilities of a voxel assigned to class  $b$  to its higher neighbor assigned to class  $a$ , where  $a$  and  $b$  take the values *solid*, *surface* or *free-space*. For a better understanding, let us assume a certain voxel is assigned to class *solid*. The conditional probability of its next higher neighboring voxel to belong the same class is given by 0.95, whereas the conditional probability to belong to class *surface* is given by 0.05.

$P(a   b)$	$b = \text{solid}$	$b = \text{surface}$	$b = \text{free-space}$
$a = \text{solid}$	0.95	0	0
$a = \text{surface}$	0.05	0	0
$a = \text{free-space}$	0	1	1

ances

$$\sigma_{x_{ij}}^2 := E(x_{ij}^2) - \underbrace{E(x_{ij})^2}_0 \quad (3.12)$$

$$= \frac{1}{d_{x_j}} \int_{-\frac{d_{x_j}}{2}}^{\frac{d_{x_j}}{2}} x^2 dx = \frac{d_{x_j}^2}{12} \quad (3.13)$$

$$\sigma_{y_{ij}}^2 := \sigma_{x_{ij}}^2 \quad (3.14)$$

and collected in the covariance matrix  $\Sigma_{xx} = \text{Diag}([\dots, \sigma_{x_{ij}}^2, \sigma_{y_{ij}}^2, \dots]^T)$ .

In addition, we compute local height variances for the estimated height values  $\mathbf{h} = [\dots, h_{ij}, \dots]^T$ . The variances are modeled by the sum of three components

$$\sigma_{h_{ij}}^2 = \frac{1}{12} \left( \frac{y_{ij}}{c} d_v \right)^2 + \frac{1}{12} \left( \frac{h_{ij} - h_g}{y_{ij}} d_x \right)^2 + \left( \frac{(h_{ij} - h_g) y_{ij}}{Bc} \right)^2 \sigma_d^2. \quad (3.15)$$

Keep in mind that the symbols  $B$  and  $c$  denote the baseline and principle distance of the camera system. The first and second term describe the effect of the vertical and horizontal discretization. The third term represents the precision of the height measurements propagated from the precision of the disparity measurements  $\sigma_d^2$  via the triangulation concept. From the geometric constellation illustrated in Figure 3.13 and from the relation of disparity and distance (2.22) considering that the camera system's  $z$ -axis corresponds to the negative  $y$ -axis of the elevation map system (see Figure 3.5(b)), we obtain the identities

$$\frac{v}{h - h_g} = \frac{c}{y} = -\frac{d}{B}. \quad (3.16)$$

The identities yield the relations

$$(i) : h = -\frac{vB}{d} + h_g, \quad (ii) : v = \frac{(h - h_g)c}{y}, \quad (iii) : d = -\frac{Bc}{y}. \quad (3.17)$$

From relation (i), we obtain the relation of the precision of the height and the precision of the disparity via non-linear error propagation

$$\sigma_h^2 = \left( \frac{vB}{d^2} \right)^2 \sigma_d^2. \quad (3.18)$$

Finally, substituting the relations (ii) and (iii) yields the third term of the total height precision (3.15).

We assume a precision of  $\sigma_d = 0.5$  pel for the disparities computed from SGM algorithm. The single elevation variances are collected in the covariance matrix  $\Sigma_{hh} = \text{Diag}([\dots, \sigma_{h_{ij}}^2, \dots]^T)$ .

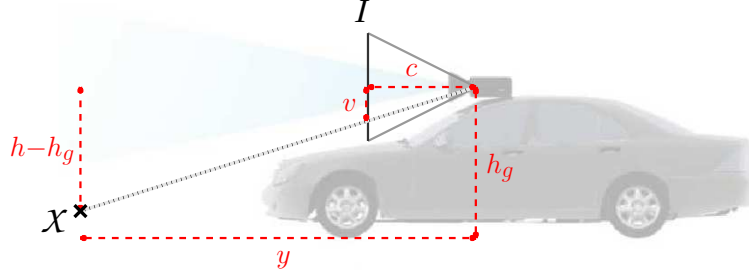


FIGURE 3.13: Relation of the longitudinal and vertical coordinates  $y$  and  $h$  of a world point  $X$  to the horizontal coordinate  $v$  of its image projection and the principle distance  $c$ . The geometric constellation shows the ratio of  $c$  and  $v$  to be identical with the ratio of  $y$  and  $h - h_g$ , whereas  $h_g$  describes the height of the camera's projection center over ground level. Note that the image plane  $I$ , in contrast to the physical setup, is drawn in viewing direction to get an upright image.

### 3.2.3 Ego-Motion

Formally, we use the term ego-motion to describe the six parameter coordinate transformation from the local grid coordinate systems of the previous time step  $S_{e_{t-1}}$  into the current time step  $S_{e_t}$ . The transformation is given by the homogeneous transformation matrix

$${}^tM_{t-1}(\mathbf{u}_t) = \begin{bmatrix} R(\mathbf{u}_t) & T\mathbf{u}_t \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (3.19)$$

The 6-vector

$$\mathbf{u}_t = \begin{bmatrix} R\mathbf{u}_t \\ T\mathbf{u}_t \end{bmatrix} \quad (3.20)$$

contains the respective three parameters of rotation  $R\mathbf{u}_t$  and translation  $T\mathbf{u}_t$ . The corresponding precision is represented by covariance matrix  $\Sigma_{\mathbf{u}_t\mathbf{u}_t}$ . The rotation parameters are given in Rodriguez representation. For a detailed description of this representation we refer to [McGlone et al., 2004, chapter 2.1.2]. We use the algorithm of [Badino, 2007] to estimate the ego-motion parameters. Other vision based approaches would also be suitable.

## 3.3 Environment Model

This section presents a detailed definition of the geometric and probabilistic properties of the environment model which form the basis for the estimation process. The environment model acts as a generative model describing the measured elevation of observed street regions, street adjacent regions as well as the street boundary. As introduced in Section 3.1, we define the considered task of street surface and boundary detection and reconstruction by a MAP estimation of the respective model parameters incorporating temporal and spacial prior information. The model is chosen to fulfill several requirements which we assume to be crucial to allow for an adequate estimation result:

- (a) The model needs to be *flexible* and *general* enough to model even complex scenarios, such as multiple obstacles and small traffic isles.
- (b) It needs to be *robust* regarding erroneous and sparse measurements.
- (c) It shall be based on a *continuous* representation, to allow for evaluation in the entire observed region of interest.

To fulfill these requirements, we assume the observed elevation model to be generated by a B-spline surface for street regions and to be drawn from a uniform distribution for street adjacent regions providing a high generality of the model. The street boundary which represents the separation of street and street adjacent regions is modeled by a B-spline curve. The usage of B-splines yields a continuous and flexible representation. Robustness is increased by explicit modeling of outliers and simplified topology assumptions, i.e. the street boundary is unique in image column direction. Further, we claim spatial and temporal smoothness as prior assumptions to cope with sparse and noisy measurements.

The subsequent sections give a detailed definition of the individual components of the utilized environment model starting with a general overview in Section 3.3.1. Subsequently, Section 3.3.2 and Section 3.3.3 yield the definition for the street surface and boundary model. Section 3.3.4 provides additional prior assumptions about the spatial and temporal behavior of the model.

### 3.3.1 Definition of the Generative Environment Model

As stated above, the environment model is governed by two functions, namely a B-spline surface  $^Sf$  which defines the elevation model of the street surface and a B-spline curve  $^Bf$  which defines the horizontal boundary of the street, i.e. the separation of street regions and street adjacent regions. Figure 3.14(d) illustrates an example for the environment model. As pointed out in Section 2.7 B-splines bear the property of locality and, thus, embody a flexible and continuous representation. The parameter vectors governing the functions  $^Sf$  and  $^Bf$  are denoted by  $^S\theta$  and  $^B\theta$ , respectively, forming the joint parameter set  $\theta = [^S\theta^\top, ^B\theta^\top]^\top$ . Their uncertainty is represented by the block diagonal covariance matrix  $\Sigma_{\theta\theta} = \text{Diag}(^S\Sigma_{\theta\theta}, ^B\Sigma_{\theta\theta})$ .

To interrelate elevation map cells and model functions, we assign each cell to one out of three classes which define the whole horizontal semantics of the environment model. The classes are given by  $\mathcal{C}^h = \{\textit{street}, \textit{outlier}, \textit{adjacent}\}$ . The class assignments of the DEM's cells are denoted by the labels  $\mathbf{l} = [\dots, l_{ij}, \dots]^\top$ , with  $l_{ij} \in \mathcal{C}^h$ . The set of all possible labelings  $\mathbf{l}$  is given by  $\mathcal{L}$ .

### 3.3.2 Surface Model

We assume the total elevation model to be given by the mixture distribution

$$\begin{aligned} h_{ij} \sim & \frac{1 - \alpha_{ij}}{2} \mathcal{N} \left( ^Sf(\mathbf{x}_{ij}, ^S\theta), ^S\sigma_{h_{ij}}^2 \right) \\ & + \frac{1 - \alpha_{ij}}{2} \mathcal{N} \left( ^Sf(\mathbf{x}_{ij}, ^S\theta), ^O\sigma_{h_{ij}}^2 \right) \\ & + \alpha_{ij} \mathcal{U}(\mathbf{h}), \end{aligned} \quad (3.21)$$

where the symbols  $\mathcal{N}$  and  $\mathcal{U}$  represent the normal and uniform distribution. The symbols  $^S\sigma_{h_{ij}}$  and  $^O\sigma_{h_{ij}}$  denote the height uncertainty of cells assigned to the classes *street* and *outlier*, respectively. The weighting factors  $\alpha_{ij}$  describe the cell's prior probabilities to belong to the class *adjacent* and are governed by the street boundary function  $^Bf$  defined in (3.27). A detailed definition of the  $\alpha_{ij}$  is given in Section 3.3.3.2.

Let us now take a closer look at the meaning of the elevation models for the different classes.

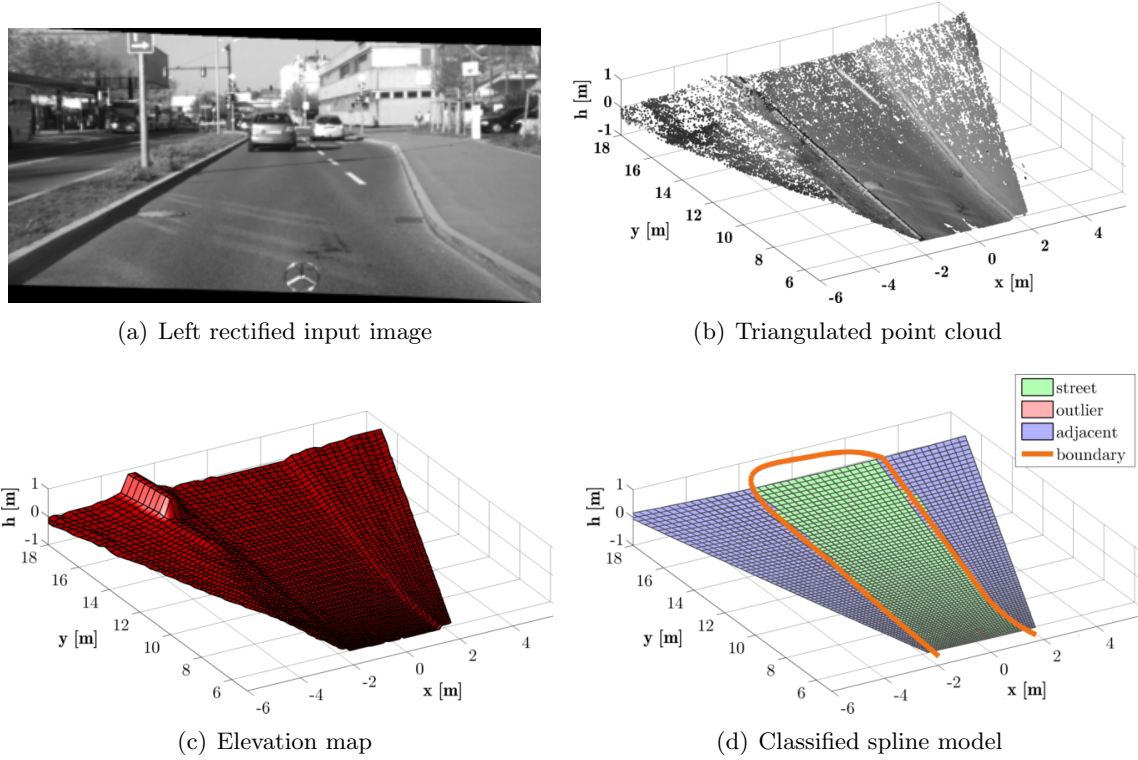


FIGURE 3.14: Figure (d) presents the environment model derived from the DEM plotted in (c). The model is represented by estimated street surface, colored due to the estimated most probable class assignments, and the estimated street boundary, which is represented by the orange line. The input image and the respective triangulated point cloud are plotted in (a) and (b).

### 3.3.2.1 Elevation Model for Street Regions

Existing methods for free-space detection generally use planar or second order surfaces to describe the street surface. Although adequate for most scenarios, results are unsatisfying when the street curvature is just locally constant, e.g. in case of for roof-shaped street surfaces often used for drainage of rainwater.

Instead of using higher degree surfaces, we assume the street surface to be sufficiently describable by a second degree, uniform B-spline surface

$$Sf([x, y]^T, S\theta) = \sum_{n=1}^N \mathcal{B}_{n,2}(x, y) S h_n \quad (3.22)$$

$$= [\mathcal{B}_{1,2}(x, y), \dots, \mathcal{B}_{N,2}(x, y)] S\theta. \quad (3.23)$$

This enables the system to model locally different curvatures. The functions  $\mathcal{B}_{n,2}(x, y)$  denote the B-spline basis functions of second degree providing the weights for the control points  $S h_n$ , see Section 2.7.2. The set of surface parameters, namely the control point heights, is collected in the vector  $S\theta = [S h_1, \dots, S h_N]^T$ . This allows for the representation of the surface as vector product (3.23).

The spline parameters are aligned to the  $x$  and  $y$  coordinates of the DEM. The spline sections are arranged in an equidistant Cartesian grid (not to be confused with the non-equidistant DEM grid) which is fixed relative to the horizontal plane of the elevation map system  $S_e$ , as illustrated in Figure 3.15(a). The grid center is located at the DEM coordinates  $[0, \frac{y_{\max} - y_{\min}}{2} + y_{\min}]^T$ . In our experiments, we decided to use four sections in lateral

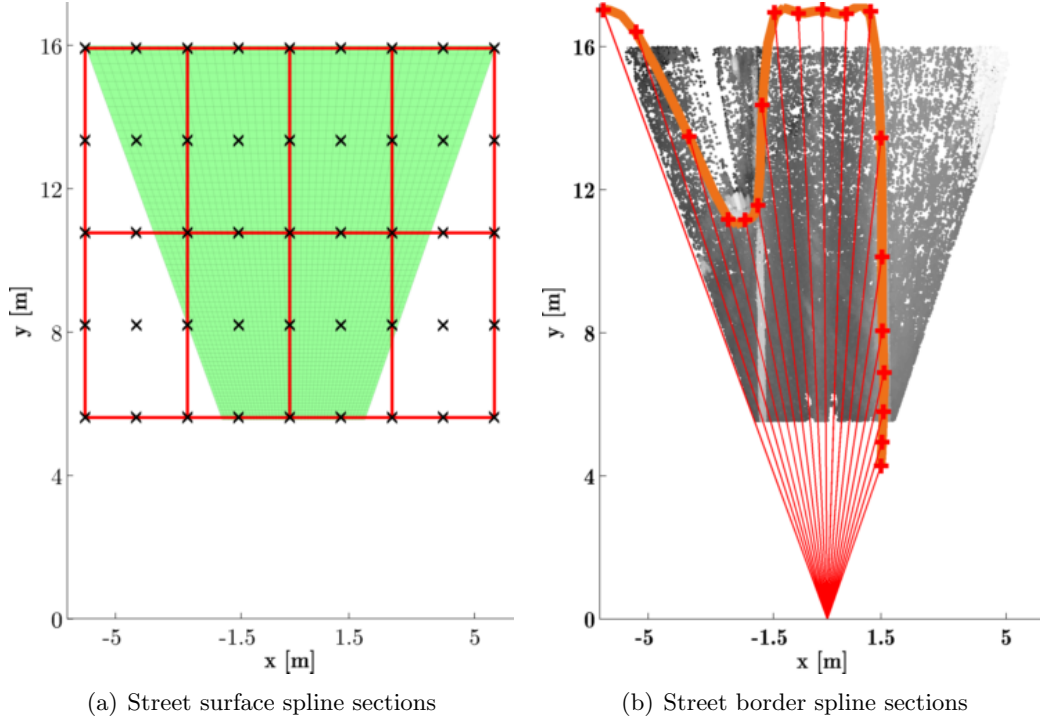


FIGURE 3.15: Spline sections of the street surface compared to the grid structure of the DEM (a). The borders of the surface sections are marked by red lines, while the DEM is shaded in green color. The black crosses mark the horizontal sample positions  $\mathbf{x}_q$ ,  $q = 1, \dots, Q$  used for spatial and temporal filtering. In (b) the sections of the boundary spline are illustrated. The spline itself is plotted by a thick orange line, the section bounds are marked by red crosses. The section limits with respect to the image columns are marked by red lines. The background is shaded by the colored point cloud computed from SGM and corresponds to the scene depicted in Figure 3.14(a).

and two in longitudinal direction, with the section lengths defined to cover the whole area of the DEM. This choice allows to model up to two different vertical curvature values in longitudinal direction. In lateral direction it allows for different curvatures for the transitions from the vehicle's current driving lane surface to the lateral neighboring surfaces assuming an approximative lane width of 3 m. In special situations a different choice may be useful.

To separate reliable observations from outliers, we split the street region in the two classes *street* and *outliers*. Thus, we assume the generative model for the heights of DEM cells belonging to the street region to be given by the Gaussian mixture model defined by the first two summands in Equation 3.21. This yields the class likelihood functions for the DEM cells  $(ij) \in \mathcal{D}$

$$\wp(h_{ij} | l_{ij} = \text{street}, {}^S\boldsymbol{\theta}) = G\left(h_{ij} \mid {}^Sf(\mathbf{x}_{ij}, {}^S\boldsymbol{\theta}), {}^S\sigma_{h_{ij}}^2\right) \quad (3.24)$$

and

$$\wp(h_{ij} | l_{ij} = \text{outlier}, {}^S\boldsymbol{\theta}) = G\left(h_{ij} \mid {}^Sf(\mathbf{x}_{ij}, {}^S\boldsymbol{\theta}), {}^O\sigma_{h_{ij}}^2\right). \quad (3.25)$$

The variance  ${}^S\sigma_{h_{ij}}^2$  corresponding to the class *street* models two effects, i.e. the influence of the uncertainty of the DEM's height measurements  $\sigma_{h_{ij}}^2$  defined in (3.15) and the influence of the uncertainty caused by horizontal discretization defined in (3.13) and (3.14). The horizontal uncertainty is propagated to the vertical component via the local slope of the

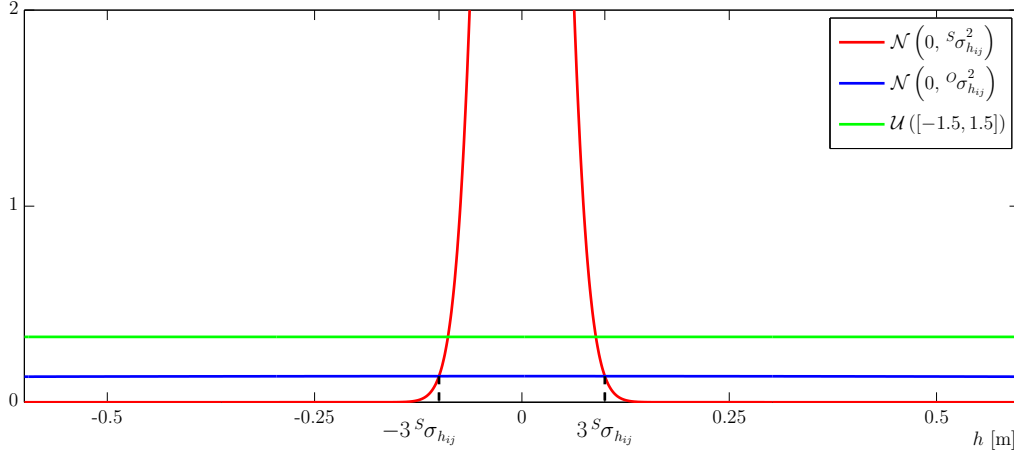


FIGURE 3.16: Example for density functions defining measurement noise (red line) and outlier (blue) distribution, which are centered at  ${}^Sf([x, y]^T, {}^S\theta) = 0$  for simplification. Height measurements exceeding the  $3 {}^S\sigma$  interval are more likely to belong to the class *outlier* than to *street* and vice versa. The density function of the class *adjacent* is plotted as green line. Note, that *adjacent* is always preferred over *outlier*.

street surface estimated in the last frame, as described in Section B, which yields a good approximation assuming the curvature of  ${}^Sf$  to be small. This allows us to treat the grid cell positions  $\mathbf{x}_{ij}$  as certain when estimating  ${}^S\theta$  and, thus, to estimate the surface parameters in a linear model which significantly reduces the computational effort.

The second summand in Equation (3.21) models *outliers* in the vertical distribution of the street measurements. We assume the outliers to originate from a normal distribution located at the proper mean but governed by a much larger variance  ${}^O\sigma_{h_{ij}}^2 = \rho {}^S\sigma_{h_{ij}}^2$ . The constant factor  $\rho$  is chosen such that both density functions intersect at  ${}^Sf([x, y]^T, {}^S\theta) \pm 3 {}^S\sigma_{h_{ij}}$ , as illustrated in Figure 3.16. Note that comparing both density functions when performing maximum likelihood estimation results in an implicit three sigma test, i.e. height measurements exceeding the  $3 {}^S\sigma_{h_{ij}}$  interval around the local surface elevation  ${}^Sf([x, y]^T, {}^S\theta)$  are more likely assigned to *outlier* than to *street*. The computation of the factor  $\rho$  is described in Section D.2.

### 3.3.2.2 Elevation Model for Street Adjacent Regions

The elevation structure of street adjacent regions may be arbitrary ranging from vertical structures through to horizontal pavement surfaces. Thus, we assume the height measurements drawn from street adjacent regions to follow a uniform distribution over the considered height interval  $\hat{h} = [-1.5, 1.5]$  m.

This class is denoted by *adjacent* and yields the likelihood

$$\wp(h_{ij} \mid l_{ij} = \textit{adjacent}, {}^S\theta) = U(h_{ij} \mid \hat{h}). \quad (3.26)$$

Note that the likelihood does not depend on  ${}^S\theta$  and, with this, has no effect on the street surface estimation step described in Section 3.5.2.1.

### 3.3.3 Street Boundary Model

The boundary function  ${}^Bf$  embodies a continuous representation of the perimeter of the drivable free-space, i.e. explicitly models the separation of the class set  $\{\textit{street}, \textit{outlier}\}$  from the class *adjacent*.

We introduce a simplified *topology assumption* which states that for every image row coordinate there exists a unique class change from the free-space region (classes *street* and *outlier*) to the non-free-space region (class *adjacent*) in longitudinal direction. This explicitly models the intention to detect the first and only the first object in viewing direction delimiting the drivable free-space.

We utilize a third order B-spline curve defined in the horizontal plane in order to allow for a flexible but smooth representation of the free-space boundary

$${}^B\mathbf{f}(\beta, {}^B\boldsymbol{\theta}) = \sum_{m=1}^M \mathcal{B}_{m,3}(\beta) {}^B\mathbf{x}_m \quad (3.27)$$

$$= \begin{bmatrix} \mathcal{B}_{1,3}(\beta) & 0 & \cdots & \mathcal{B}_{M,3}(\beta) & 0 \\ 0 & \mathcal{B}_{1,3}(\beta) & \cdots & 0 & \mathcal{B}_{M,3}(\beta) \end{bmatrix} {}^B\boldsymbol{\theta}. \quad (3.28)$$

Similar to the street surface model, the set of 2d control points defines the parameter vector for the street boundary  ${}^B\boldsymbol{\theta} = [{}^B\mathbf{x}_1^\top, \dots, {}^B\mathbf{x}_M^\top]^\top$ . This allows for the definition of the boundary via the matrix-vector product (3.28). The  $x$  and  $y$  components of  ${}^B\mathbf{f}$  are denoted by  ${}^Bf_x$  and  ${}^Bf_y$ .

### 3.3.3.1 Spline Parametrization

The parameter  $\beta \in [0, {}^Bn]$  defines the position on the spline curve and the local influence of the control points  ${}^B\mathbf{x}_m = [{}^Bx_m, {}^By_m]^\top$  via the basis functions  $\mathcal{B}_{m,3}$ . The entity  ${}^Bn = M - 3$  defines the number of spline sections (see Section 2.7.1). The parameter intervals are distributed uniformly along the  $u$ -axis of the image, i.e. the back projections of the spline sections cover the same number of image columns each. The parameter values aligned to the longitudinal columns of the DEM are denoted by  $\beta_i$ . An example illustration for the boundary spline sections is given in Figure 3.15(b).

There exist several alternative heuristics for the choice of the spline parametrization, e.g. based on the chord length or curvature. For a discussion of different parameterizations, we refer to [Piegl and Tiller, 1997, chapter 9.2] or [Farin, 2002, chapter 9.6].

However, we selected the parametrization along the image columns due to several reasons. Although a curvature based parametrization allows for more exact reconstruction results at edge structures, it appeared to be very sensitive to measurement errors and tends to oversampling in sections with strong curvature. In our experiments, the chosen approach appeared to be the best compromise between flexibility and robustness. It explicitly supports the topology assumption and guarantees a balanced distribution of observations per spline section. Furthermore, newly observed boundary segments can directly be integrated in the parametrization without the need for any prior information, which is particularly important in sharp turning maneuvers.

### 3.3.3.2 Class Probabilities from Boundary Function

The street boundary plays a central role in the definition of the elevation model represented by the mixture distribution (3.21). The relative position of a sample to the boundary, i.e. within or beyond the encircled region, indicates its affiliation to the class set  $\{\textit{street}, \textit{outlier}\}$  (free-space) or to the class *adjacent* (non-free-space), respectively. This affiliation is represented by the weighting factors  $\alpha_{ij}$  which embody assignment probabilities rather than a fixed



assignments since the boundary is uncertain, i.e.

$$\frac{1 - \alpha_{ij}}{2} = P(l_{ij} = \textit{street} \mid {}^B\boldsymbol{\theta}) \quad (3.29)$$

$$\frac{1 - \alpha_{ij}}{2} = P(l_{ij} = \textit{outlier} \mid {}^B\boldsymbol{\theta}) \quad (3.30)$$

$$\alpha_{ij} = P(l_{ij} = \textit{adjacent} \mid {}^B\boldsymbol{\theta}). \quad (3.31)$$

The assignment probability is modeled similar to the logistic regression model yielding the weighting factors by the local evaluation

$$\alpha_{ij} := g_i({}^B\boldsymbol{\theta}, y_{ij}) \quad (3.32)$$

of the sigmoid function

$$g_i({}^B\boldsymbol{\theta}, y) := \frac{1}{1 + \exp(-w_i(y - {}^Bf_y(\beta_i, {}^B\boldsymbol{\theta})))}. \quad (3.33)$$

The longitudinal component of the street boundary function  ${}^Bf_y$  determines the inflection point of the sigmoid and, thus, the decision boundary between free-space and non-free-space (see Figure 3.17). The usage of the sigmoid model reflects the simplified topology assumption which states that for every image row coordinate there exists a unique class change from the free-space region to the non-free-space region in  $y$  direction.

Note that the boundary function is evaluated at the respective DEM grid column  $i$ , i.e. at the parameter position  $\beta_i$ , indicating that we use one sigmoid model per longitudinal column of the DEM. The individual slopes of the sigmoids are controlled by the *slope parameters*  $\boldsymbol{w} = [w_1, \dots, w_I]^\top$ . Figure 3.17(b) illustrates the variation of the sigmoid slopes for different grid rows.

One can think of  $\alpha_{ij}$  as spatial prior for the class assignment, i.e. for the latent variables  $\boldsymbol{l}$ , not to be confused with the spatial prior for the model parameters defined in the next section. The prior states that the unique class change (i.e. the decision boundary) between *adjacent* and the remaining classes along grid column  $i$  is located at  ${}^Bf_y(\beta_i, {}^B\boldsymbol{\theta})$ . In this context, the slope parameter  $w_i$  defines the influence of the prior. The properties of this term are discussed in detail later on when defining the posterior probability  $P(\boldsymbol{l} \mid \boldsymbol{h}_t, \widehat{\boldsymbol{\theta}}_t^{(\nu)})$  for the estimation step of the EM-algorithm (see Section 3.5.1).

### 3.3.4 A Priori Assumptions

This section introduces the prior knowledge about the model parameters which is combined with the previously introduced likelihood functions to estimate the model parameters via the optimization function defined by (3.6).

We assume the prior knowledge for the surface and boundary parameters to be independent such that the total prior decomposes into

$$\wp(\boldsymbol{\theta} \mid \widehat{\boldsymbol{\theta}}_{t-1}) := \wp({}^S\boldsymbol{\theta} \mid {}^S\widehat{\boldsymbol{\theta}}_{t-1}) \wp({}^B\boldsymbol{\theta} \mid {}^B\widehat{\boldsymbol{\theta}}_{t-1}), \quad (3.34)$$

whereas both parameter sets only depend on their respective counterpart in  $\widehat{\boldsymbol{\theta}}_{t-1}$ . The subsequent sections describe the structure of both terms.



(a) Image projection of the estimated street boundary

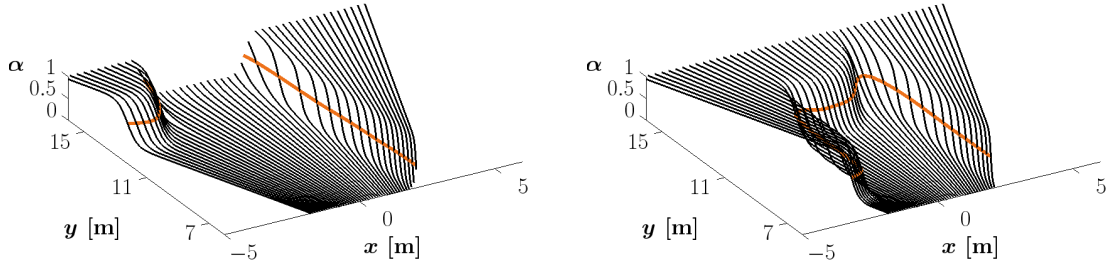
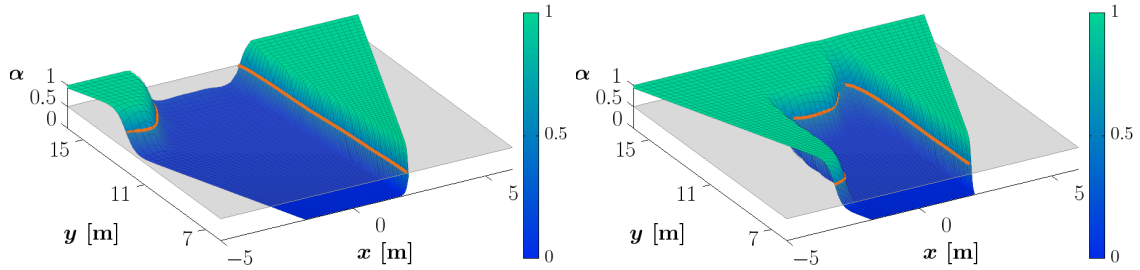
(b) Weighting factors  $\alpha_{ij}$  plotted as sigmoid functions along each DEM column(c) Weighting factors  $\alpha_{ij}$  plotted as 2.5d sigmoidal surface

FIGURE 3.17: Two examples scenarios demonstrating the relation of the street boundary spline  $Bf_y(\beta_i, {}^B\theta)$ , plotted in orange color, and the weighting factors  $\alpha_{ij}$ . The first row depicts the example scenarios as well as the image projection of the estimated street boundary. The second row shows the sigmoid functions defining the  $\alpha_{ij}$  plotted against each longitudinal column of the DEM. Note the different slopes of the sigmoid functions that are best visible in the right image around the gap between the truck and the curb. The street boundary is defined by the 0.5-level-sets of the sigmoid functions (orange line), which correspond to the respective inflection points. The weighting factors are plotted as 2.5d surface using color encoding for clarification, while the 0.5-level is plotted as transparent plane.

### 3.3.4.1 Prior Assumptions about the Street Surface

As a priori assumption for the street surface we claim temporal and spatial smoothness, i.e. the estimated surface is assumed to be rigid over time and of low curvature. We use the procedure described in Section 2.3.2.2 to model the prior  $\wp \left( {}^S\boldsymbol{\theta} \mid \widehat{{}^S\boldsymbol{\theta}}_{t-1} \right)$  in terms of fictitious observations. These observations compose of two sets  ${}^S\mathbf{z}_t^{\text{temp}}$  and  ${}^S\mathbf{z}^{\text{spat}}$  which define the temporal (temp) and spatial (spat) prior knowledge about the parameters via the functional relationship

$$\begin{bmatrix} {}^S\mathbf{z}_t^{\text{temp}} \\ {}^S\mathbf{z}^{\text{spat}} \end{bmatrix} + \begin{bmatrix} {}^S\mathbf{v}_t^{\text{temp}} \\ {}^S\mathbf{v}^{\text{spat}} \end{bmatrix} = \begin{bmatrix} {}^S\mathbf{k}^{\text{temp}} \left( {}^S\boldsymbol{\theta}, \widehat{{}^S\boldsymbol{\theta}}_{t-1} \right) \\ {}^S\mathbf{k}^{\text{spat}} \left( {}^S\boldsymbol{\theta} \right) \end{bmatrix}, \quad (3.35)$$

where the residuals  ${}^S\mathbf{v}_t^{\text{temp}}$  and  ${}^S\mathbf{v}^{\text{spat}}$  are independent and normal distributed.

The mathematical formulation of the observations  ${}^S\mathbf{z}_t^{\text{temp}}$  and the function  ${}^S\mathbf{k}^{\text{temp}}$  as well as the stochastic properties of  ${}^S\mathbf{v}_t^{\text{temp}}$  are given in Section 3.4.1.1 in the context of the model prediction step.

For spatial smoothness, we claim that each of the three components of the mean squared curvature defined in (2.120) shall vanish at a set of sample positions  $\mathbf{x}_q$ , with  $q = 1, \dots, Q$ , evenly distributed in the considered region of interest, as depicted in Figure 3.15(a). This yields three fictitious observations per sample. Summarized, the spatial prior information is defined by the linear functional model

$$\left\{ \begin{array}{c} \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \end{array} \right\}_{3Q} + {}^S\mathbf{v}^{\text{spat}} = \underbrace{\begin{bmatrix} \vdots \\ \dots \frac{1}{\sqrt{2}} \mathcal{B}_{n,2}^{(2,0)}(x_q, y_q) \dots \\ \dots \mathcal{B}_{n,2}^{(1,1)}(x_q, y_q) \dots \\ \dots \frac{1}{\sqrt{2}} \mathcal{B}_{n,2}^{(0,2)}(x_q, y_q) \dots \\ \vdots \end{bmatrix}}_{{}^S\mathbf{k}^{\text{spat}}({}^S\boldsymbol{\theta})} \begin{bmatrix} \vdots \\ {}^Sh_n \\ \vdots \end{bmatrix}. \quad (3.36)$$

The influence of the spatial smoothness term is governed by the parameter  ${}^S\lambda$  introduced via the stochastic model

$${}^S\mathbf{v}^{\text{spat}} \sim \mathcal{N} \left( \mathbf{0}, \frac{1}{{}^S\lambda} I_{3Q} \right). \quad (3.37)$$

Note that  ${}^S\lambda$  is a tuning parameter which is determined empirically.

### 3.3.4.2 Prior Assumptions about the Street Boundary

Analogous to the street surface, the prior assumptions for the street boundary parameters  ${}^B\boldsymbol{\theta}$  is defined by two separate sets of fictitious observations modeling the temporal and spatial assumptions

$$\begin{bmatrix} {}^B\mathbf{z}_t^{\text{temp}} \\ {}^B\mathbf{z}^{\text{spat}} \end{bmatrix} + \begin{bmatrix} {}^B\mathbf{v}_t^{\text{temp}} \\ {}^B\mathbf{v}^{\text{spat}} \end{bmatrix} = \begin{bmatrix} {}^B\mathbf{k}^{\text{temp}} \left( {}^B\boldsymbol{\theta}, \widehat{{}^B\boldsymbol{\theta}}_{t-1} \right) \\ {}^B\mathbf{k}^{\text{spat}} \left( {}^B\boldsymbol{\theta} \right) \end{bmatrix}. \quad (3.38)$$

Again, the temporal term is described in Section 3.4.1.2 when discussing the prediction step.

The spatial term claims low curvature. For each spline section, the curvature  $\kappa$  is given by a third order polynomial in the spline parameter governed by the respective active control points, as defined by (2.107). Thus, forcing the curvature to vanish at at least four distinct sample positions within a single spline section is sufficient to suppress the curvature in the complete section interval. Given  $B_n$  spline sections in our boundary spline model, we choose  $A = 4^{B_n} + 1$  sample positions  $\beta_a$ , with  $a = 1, \dots, A$ , distributed equidistant in the spline parameter space by  $\beta_a = \frac{1}{4}(a-1)$ . To regularize the spline control points  ${}^B\boldsymbol{\theta}$  we constrain the local curvature measure  $\kappa_{\beta_a}({}^B\boldsymbol{\theta})$  defined in (2.109) at each sample point to vanish. Collecting this  $A$  constraints in a single equation system, we obtain the nonlinear Gaussian model

$$A \left\{ \underbrace{\begin{bmatrix} \vdots \\ 0 \\ \vdots \end{bmatrix}}_{{}^B\mathbf{z}^{\text{spat}}} + {}^B\mathbf{v}^{\text{spat}} = \underbrace{\begin{bmatrix} \vdots \\ \kappa_{\beta_a}({}^B\boldsymbol{\theta}) \\ \vdots \end{bmatrix}}_{{}^B\mathbf{k}^{\text{spat}}({}^B\boldsymbol{\theta})}, \quad (3.39)$$

where again the variance

$${}^B\mathbf{v}^{\text{spat}} \sim \mathcal{N}\left(\mathbf{0}, \frac{1}{B\lambda} I_A\right) \quad (3.40)$$

is governed by a tuning parameter  ${}^B\lambda$ .

### 3.4 Model Prediction

We assume the variation of the observed street boundary and surface characteristics to be small between successive frames. Thus, using information about the vehicle's ego-motion, the model parameters estimated in the previous acquisition time step  $t-1$  can be used to predict the model at the current time step  $t$ . This information transfer provides both prior information and initial values for the current model parameters due to our assumption of temporal rigidity of the model. The acquisition of the prior terms and initial values embodies processing step (III) of the total workflow depicted in Figure 3.1.

The following sections provide a detailed description of processing step (III), also referred to as prediction step. First, Section 3.4.1 depicts the acquisition of the temporal prior knowledge based on the previously estimated model  $\hat{\boldsymbol{\theta}}_{t-1}$  and the ego-motion information introduced in Section 3.2.3. This comprises the formulation of the temporal components of the prior distributions  $\wp({}^S\boldsymbol{\theta} \mid {}^S\hat{\boldsymbol{\theta}}_{t-1})$  and  $\wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1})$  previously introduced in Section 3.3.4. The temporal prior terms for the street surface parameters are defined in Section 3.4.1.1 and those for the street boundary parameters in Section 3.4.1.2.

Subsequently, Section 3.4.2 demonstrate how to obtain approximate values  $\hat{\boldsymbol{\theta}}_t^{(0)}$  for the model parameters to initialize the very first EM iteration of the each time step. Again, the discussion is subdivided in terms of the surface parameters (Section 3.4.2.1) and the boundary parameters (Section 3.4.2.2). Further, the section describes the initialization of the model for the very first time step or in case of reinitialization (Section 3.4.2.3).

#### 3.4.1 Acquisition of Temporal Prior Knowledge

Keep in mind that we defined the parametrization of both the surface and the boundary spline relative to the local elevation map system in order to reduce the computational effort,

e.g. to allow for the precomputation of the spline weights. Unfortunately, this complicates the definition of a proper system model for the control points since the control point positions and the spline parametrization with respect to a stationary world system change for each acquisition time step. Thus, we employ a sampling based method to predict the current model parameters.

Simply put, temporal filtering is performed by demanding that the current estimated model needs to be able to describe a set of points sampled from the previous model. These predicted samples embody the fictitious observations  $^S \mathbf{z}_t^{\text{temp}}$  and  $^B \mathbf{z}_t^{\text{temp}}$  for the temporal prior terms in (3.35) and (3.38). In the following we describe the acquisition of these observations and formulate the functional dependency to the respective model parameter set represented by  $^S \mathbf{k}^{\text{temp}}(^S \boldsymbol{\theta}, ^S \hat{\boldsymbol{\theta}}_{t-1})$  and  $^B \mathbf{k}^{\text{temp}}(^B \boldsymbol{\theta}, ^B \hat{\boldsymbol{\theta}}_{t-1})$ .

### 3.4.1.1 Temporal Prior for the Street Surface Parameters

The street surface estimated in the previous time step is sampled at evenly arranged horizontal positions in the coordinate system  $S_{e_{t-1}}$ . The sample positions are defined by the corner points and half positions of the spline sections  $\mathbf{x}_q = [x_q, y_q]^\top$ , with  $q = 1, \dots, Q$ , as illustrated in Figure 3.15(a). This yields  $Q$  samples of the form

$$\mathbf{X}_{q,t-1} = \begin{bmatrix} x_q \\ y_q \\ Sf(\mathbf{x}_q, ^S \hat{\boldsymbol{\theta}}_{t-1}) \end{bmatrix}. \quad (3.41)$$

We transform the samples to the current system  $S_{e_t}$  by means of ego-motion information and obtain the predicted sample points

$$\mathbf{X}_{q,t}^- = \begin{bmatrix} \mathbf{x}_{q,t}^- \\ h_{q,t}^- \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^t \mathbf{M}_{t-1}(\mathbf{u}_t) \begin{bmatrix} \mathbf{X}_{q,t-1} \\ 1 \end{bmatrix}. \quad (3.42)$$

The matrix  ${}^t \mathbf{M}_{t-1}(\mathbf{u}_t)$  represents the coordinate transformation which projects the coordinates from the old system  $S_{e_{t-1}}$  to the current system  $S_{e_t}$  governed by the ego-motion parameter vector  $\mathbf{u}_t$ . See Section 3.2.3 for further details about the utilized ego-motion information.

We collect the predicted samples in the vector  $\mathbf{X}_t^- = [\dots, \mathbf{X}_{q,t}^-, \dots]^\top$  and denote their joint precision by the covariance matrix  $\Sigma_{X_t^- X_t^-}^-$ . The covariance matrix is derived via error propagation from the precisions of the ego-motion parameters  $\Sigma_{u_t u_t}$  and the precision of the previously estimate surface parameters  $^S \Sigma_{\hat{\boldsymbol{\theta}}_{t-1} \hat{\boldsymbol{\theta}}_{t-1}}$ . This derivation is lengthy and therefore presented separately in Section C.1.

The idea for the formulation of the prior term is to demand that the new model sufficiently describes the sample points, i.e.  $h_{q,t}^- \stackrel{!}{=} Sf(\mathbf{x}_{q,t}^-, ^S \hat{\boldsymbol{\theta}}_t)$  should hold for all  $q$  within the precision of the samples. Note, since all three coordinates of the predicted 3d samples are uncertain, this formulation represents an implicit function of the observations. To allow for an efficient estimation in a linear model, we use a procedure analogous to the calculus presented in Section B. This means we propagate the horizontal uncertainties in  $\Sigma_{X_t^- X_t^-}^-$  to the vertical component in order to treat the horizontal sample positions as certain. The local derivatives with respect to the horizontal coordinates at the sample positions are adopted from the surface parameters estimated in the previous frame. Neglecting the effect of the

vehicle's pitch and roll movement, which is assumed to be small, the derivatives read as

$$\frac{\partial}{\partial x} S f_q = \sum_{n=1}^N \mathcal{B}_{n,2}^{(1,0)}(x_q, y_q) \hat{h}_{n,t-1} \quad (3.43)$$

$$\frac{\partial}{\partial y} S f_q = \sum_{n=1}^N \mathcal{B}_{n,2}^{(0,1)}(x_q, y_q) \hat{h}_{n,t-1}, \quad (3.44)$$

where we use the short form notation  $S f_q := S f(\mathbf{x}_q, S \hat{\boldsymbol{\theta}}_{t-1})$ . Following (B.6), the precision of the heights of the samples including the predicted vertical uncertainty and propagated horizontal accuracy is given by

$$\Sigma_{h_t^- h_t^-} = \begin{bmatrix} \ddots & & & \\ & -\frac{\partial}{\partial x} & -\frac{\partial}{\partial y} & 1 \\ & & \ddots & \\ & & & \ddots \end{bmatrix} \Sigma_{X_t^- X_t^-} \begin{bmatrix} \ddots & & & \\ & -\frac{\partial}{\partial x} & -\frac{\partial}{\partial y} & 1 \\ & & \ddots & \\ & & & \ddots \end{bmatrix}^\top. \quad (3.45)$$

Finally, the temporal prior assumption which forms the second block of (3.35) is given by the linear Gaussian model

$$Q \left\{ \underbrace{\begin{bmatrix} \vdots \\ h_{q,t}^- \\ \vdots \end{bmatrix}}_{S \mathbf{z}_t^{\text{temp}}} + \underbrace{S \mathbf{v}_t^- + S \mathbf{r}_t^-}_{S \mathbf{v}_t^{\text{temp}}} = \underbrace{\begin{bmatrix} S f(\mathbf{x}_{q,t}^-, S \boldsymbol{\theta}) \\ \vdots \end{bmatrix}}_{S \mathbf{k}^{\text{temp}}(S \boldsymbol{\theta}, S \hat{\boldsymbol{\theta}}_{t-1})}, \quad (3.46)$$

where  $S \mathbf{v}_t^-$  models the uncertainty propagated from the ego-motion and surface parameters with

$$S \mathbf{v}_t^- \sim \mathcal{N}(\mathbf{0}, \Sigma_{h_t^- h_t^-}). \quad (3.47)$$

The additional Gaussian noise term

$$S \mathbf{r}_t^- \sim \mathcal{N}(\mathbf{0}, S \sigma_r^2 I_Q) \quad (3.48)$$

represents the model error caused by the reparametrization of the spline surface, i.e. caused by the displacement of the control points with respect to the fixed world system. In our experiments we use  $S \sigma_r = 0.5$  cm. The term further guarantees the joint covariance matrix to be regular.

The joint covariance matrix describing both effects and therefore the precision of  $S \mathbf{v}_t^{\text{temp}}$  is denoted by

$$\Sigma_{h_t^- h_t^-} = \Sigma_{h_t^- h_t^-}^- + S \sigma_r^2 I_Q. \quad (3.49)$$

### 3.4.1.2 Temporal Prior for the Street Boundary Parameters

The prediction of the street boundary is done by a similar sampling procedure. To avoid occluded samples considering the topology assumption, samples are generated by searching for the first intersection of the old spline with each  $y$ -column of the current DEM grid. One can think of the  $y$ -columns as lines  $\mathbf{l}_i$ , with  $i = 1, \dots, I$ , which connect the origin and

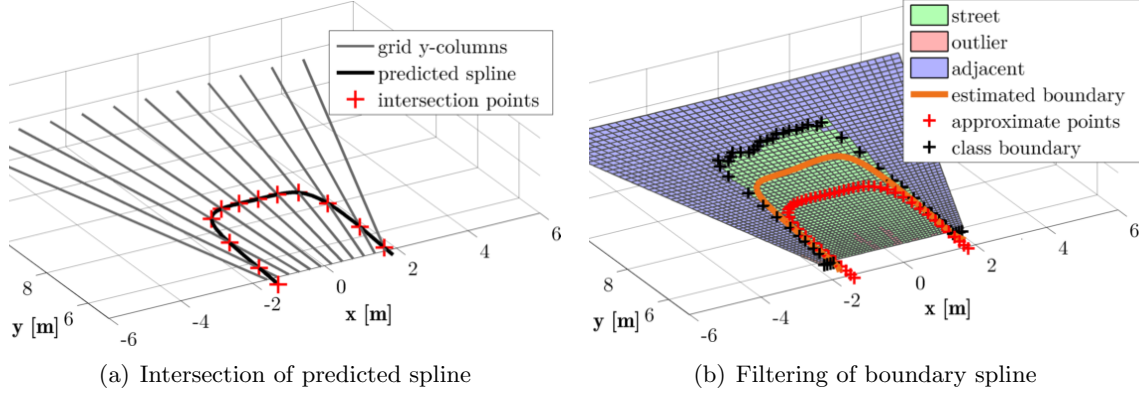


FIGURE 3.18: Figure (a) illustrates the intersection of the predicted boundary function with the  $y$ -columns of the DEM. Note that the number of columns is reduced to improve the visibility. Figure (b) depicts the different entities of boundary spline estimation. The approximate points  $\mathbf{x}_{i,t}^-$  derived from intersecting the predicted spline with the  $y$ -columns are marked by red crosses. The class boundaries achieved from logistic regression  $\mathbf{x}_{i,t}^+$ , as described in Section 3.5.2.2, are marked by black crosses. The final estimated spline is drawn as line in orange color. The scenario is the same as in Figure 3.4(a) but shown in an early stage, where the spline still propagates from its initial position.

the respective point defined by the horizontal coordinates of the first grid row  $[x_{i1}, y_{i1}]^T$ , as illustrated in Figure 3.18(a). Considering the homogeneous representation of a 2d line introduced in (2.3), we derive the line representations

$$\mathbf{l}_i = \begin{bmatrix} -y_{i1} \\ x_{i1} \\ 0 \end{bmatrix}. \quad (3.50)$$

In order to compute the intersection points, the boundary spline is transformed to the current elevation map system by applying the ego-motion matrix  ${}^t\mathbf{M}_{t-1}(\mathbf{u}_t)$  to the previously estimated spline control points  ${}^B\hat{\mathbf{x}}_{m,t-1}$ . This yields the predicted control points

$${}^B\mathbf{x}_{m,t}^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} {}^t\mathbf{M}_{t-1}(\mathbf{u}_t) \begin{bmatrix} {}^B\hat{\mathbf{x}}_{m,t-1} \\ 0 \\ 1 \end{bmatrix}. \quad (3.51)$$

Note that the points are projected to the horizontal plane since the boundary spline is defined as horizontal delimiter of the drivable region, which renders the vertical coordinate useless. The single control points are collected in the vector  ${}^B\boldsymbol{\theta}_t^-$  by concatenation. The precision of the predicted control points  ${}^B\Sigma_{\theta_t^-}$  follows via error propagation from the precision of the ego-motion parameters  $\Sigma_{u_t u_t}$  and the precision of the unpredicted control points  ${}^B\Sigma_{\hat{\theta}_{t-1} \hat{\theta}_{t-1}}$ . The respective derivation is particularized in Section C.2.

The  $i$ th sample point is computed as the intersection point of the  $i$ th grid column and the predicted spline as described in Algorithm 2 yielding

$$\mathbf{x}_{i,t}^- = {}^B\mathbf{f}(\beta_{i,t}^-, {}^B\boldsymbol{\theta}_t^-) = \sum_{m=1}^M \mathcal{B}_{m,3}(\beta_{i,t}^-) {}^B\mathbf{x}_{m,t}^-. \quad (3.53)$$

An example comparing intersected points and final estimated spline is given in Figure 3.18(b).

```

foreach Grid column  $i = 1, \dots, I$  do
  Initialize set of candidate sample points  $\mathcal{S}_i = \emptyset$ ;
  foreach Spline section  $s = 1, \dots, B_n$  do
    if  $\mathbf{l}_i = [-y_{i1}, x_{i1}, 0]^T$  intersects the convex hull of the control points of section  $s$ 
    then
      Compute all real valued roots of the third order polynomial


$$0 = \begin{bmatrix} -y_{i1} & x_{i1} & 0 \end{bmatrix} \begin{bmatrix} \sum_{m=s}^{s+3} \mathcal{B}_{m,3}(\beta) B \mathbf{x}_{m,t}^- \\ 1 \end{bmatrix} \quad (3.52)$$


      with respect to  $\beta$  and discard solutions with parameter values beyond the
      current spline section, i.e.  $\beta \notin [s-1, s]$  ;
      Compute corresponding intersection candidate points for all valid roots
      via (3.53) and append them to  $\mathcal{S}_i$  ;
    end
  end
  if  $\mathcal{S}_i \neq \emptyset$  then
    Choose the actual sample point  $\mathbf{x}_{i,t}^-$  to be the point in  $\mathcal{S}_i$  which is closest to the
    camera. The corresponding spline parameter is stored in  $\beta_{i,t}^-$  ;
  else
    Mark sample point  $i$  as invalid.
  end
end

```

**Algorithm 2:** Algorithm for the computation of spline samples by intersection of the predicted spline and the  $y$ -columns of the DEM grid. Samples marked as invalid are excluded from the temporal prior.

Similar to the surface prediction step, we obtain the temporal component of the prior (3.38) via the linear Gaussian model

$$2I \left\{ \underbrace{\begin{bmatrix} \vdots \\ \mathbf{x}_{i,t}^- \\ \vdots \end{bmatrix}}_{B \mathbf{z}_t^{\text{temp}}} + \underbrace{B \mathbf{v}_t^- + B \mathbf{r}_t^-}_{B \mathbf{v}_t^{\text{temp}}} = \underbrace{\begin{bmatrix} \vdots \\ B \mathbf{f}(\beta_i, B \boldsymbol{\theta}) \\ \vdots \end{bmatrix}}_{B \mathbf{k}^{\text{temp}}(B \boldsymbol{\theta}, B \hat{\boldsymbol{\theta}}_{t-1})} \right\}, \quad (3.54)$$

with  $i = 1, \dots, I$ . Samples marked as invalid are excluded by setting their respective entries to zero.

The residuals  $B \mathbf{v}_t^{\text{temp}}$  consist of two terms. The first term  $B \mathbf{v}_t^-$  represents the sample uncertainty propagated from the predicted control points  $B \hat{\boldsymbol{\theta}}_{t-1}$ . The second term  $B \mathbf{r}_t^-$  covers the model errors caused by the reparametrization of the predicted spline as well as the violation of model assumptions, e.g. in case of non-static obstacles. The following paragraphs introduce both terms in detail.

**Sample Uncertainty Propagated from the Predicted Control Points:** The residuals

$$B \mathbf{v}_t^- \sim \mathcal{N}(\mathbf{0}, J_{\mathbf{g}} B \Sigma_{\theta_t^- \theta_t^-} J_{\mathbf{g}}^T) \quad (3.55)$$



model the effect of the uncertainty of the predicted control points on the intersected samples. The exact Jacobian for the intersection step (3.53)

$$J_{\mathbf{g}} = \left. \frac{\partial {}^B \mathbf{f}(\beta^-, {}^B \boldsymbol{\theta}^-)}{\partial {}^B \boldsymbol{\theta}^-} \right|_{\beta^- = \beta_{i,t}^-, {}^B \boldsymbol{\theta}^- = {}^B \boldsymbol{\theta}_t^-}, \quad (3.56)$$

becomes quite complex since perturbations of the predicted control points  ${}^B \boldsymbol{\theta}_t^-$  affect the solution for the intersection parameter  $\beta_{i,t}^-$  determined via Algorithm 2. Thus, instead of considering the intersection of the lines  $\mathbf{l}_i$  with the spline itself we approximate the entries of the Jacobian considering the intersection of the line  $\mathbf{l}_i$  with the local spline tangent. By local spline tangent we mean the tangent locally evaluated at the determined intersection point  $\mathbf{x}_{i,t}^-$ . The tangent is described by a 2d line in homogeneous representation

$$\mathbf{g}_{i,t} = \begin{bmatrix} G \mathbf{x}_{i,t}^{-'} \\ -(\mathbf{x}_{i,t}^-)^\top G \mathbf{x}_{i,t}^{-'} \end{bmatrix}, \quad (3.57)$$

with the local derivative

$$\mathbf{x}_{i,t}^{-'} = \frac{\partial \mathbf{x}_{i,t}^-}{\partial \beta_{i,t}^-} = \sum_{m=1}^M \mathcal{B}'_{m,3}(\beta_{i,t}^-) {}^B \mathbf{x}_{m,t}^-, \quad (3.58)$$

and the matrix

$$G = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (3.59)$$

Employing the calculus of projective geometry, the intersection point of two 2d lines given in homogeneous representation follows from the cross product. Thus, the intersection points in homogeneous coordinates read as

$$\mathbf{x}_{i,t}^- = S(\mathbf{l}_i) \mathbf{g}_{i,t} \quad (3.60)$$

$$= S \left( \begin{bmatrix} -y_{i1} \\ x_{i1} \\ 0 \end{bmatrix} \right) \mathbf{g}_{i,t} \quad (3.61)$$

using the skew symmetric matrix  $S$  to represent the cross product as well as the definition of the lines representing the  $y$ -columns (3.50). The relation to Euclidean coordinates is given by dividing the Euclidean part by the homogeneous coordinate

$$\mathbf{x}_{i,t}^- = \frac{1}{[\mathbf{x}_{i,t}^-]_3} \begin{bmatrix} [\mathbf{x}_{i,t}^-]_1 \\ [\mathbf{x}_{i,t}^-]_2 \end{bmatrix}. \quad (3.62)$$

Linear error propagation via (3.57), (3.61) and (3.62) yields

$$\Sigma_{\mathbf{x}_t^- \mathbf{x}_t^-} = J_{\mathbf{g}} {}^B \Sigma_{\boldsymbol{\theta}_t^- \boldsymbol{\theta}_t^-} J_{\mathbf{g}}^\top, \quad (3.63)$$

with the Jacobian

$$J_{\mathbf{g}} \approx \begin{bmatrix} \vdots \\ \frac{1}{(\mathbf{x}_{i,t}^-)_h} [l_2 \mid -\mathbf{x}_{i,t}^-] S([-y_{i1}, x_{i1}, 0]^\top) \frac{\partial \mathbf{g}}{\partial ({}^B \mathbf{x}_{m,t}^-)} \\ \vdots \end{bmatrix} \quad (3.64)$$

and the partial derivatives of  $\mathbf{g}$  given by the  $(3 \times 2M)$ -matrix

$$\frac{\partial \mathbf{g}}{\partial \left( {}^B \mathbf{x}_{m,t}^- \right)} = \begin{bmatrix} \cdots & G \mathcal{B}'_{m,3} \left( \beta_{i,t}^- \right) & \cdots \\ \cdots & -(\mathbf{x}_{i,t}^-)^\top G \mathcal{B}'_{m,3} \left( \beta_{i,t}^- \right) - (\mathbf{x}_{i,t}^-)^\top G \mathcal{B}_{m,3} \left( \beta_{i,t}^- \right) & \cdots \end{bmatrix}. \quad (3.65)$$

We refer to [Meidow et al., 2009] for a detailed description of the error propagation for straight line intersections and Euclidean normalization.

**Model Errors:** The second residual term of the Gaussian model (3.54)

$${}^B \mathbf{r}_t^- \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \ddots & & \\ & \left( {}^B \sigma_{r_{i,t}}^{(\nu+1)} \right)^2 I_2 & \\ & & \ddots \end{bmatrix} \right) \quad (3.66)$$

represents deviations from the system model caused by the sampling process and reparametrization. We further use this term to handle violations of the model assumptions with respect to the temporal filtering. This means, the influence of the filtering for spline regions which are still expanding from the initialization point or which model non-static obstacles shall be reduced. The assignment of the respective noise values for  ${}^B \sigma_{r_{i,t}}^{(\nu+1)}$  depends on the assessment of the local validity of the model and is described in Section 3.5.3.1 when discussing the local evaluation of the estimated model.

The joint covariance matrix describing both effects  ${}^B \mathbf{v}_t^-$  and  $\mathbf{r}_t^-$  and therefore the precision of  ${}^B \mathbf{v}_t^{\text{temp}}$  is denoted by

$$\left( \Sigma_{x_t^- x_t^-} \right)^{(\nu+1)} = \Sigma_{x_t^- x_t^-}^- + \begin{bmatrix} \ddots & & \\ & \left( {}^B \sigma_{r_{i,t}}^{(\nu+1)} \right)^2 I_2 & \\ & & \ddots \end{bmatrix}. \quad (3.67)$$

### 3.4.2 Initial Values for the Model Parameters

The iterative process of the EM-algorithm requires initial values  $\hat{\boldsymbol{\theta}}_t^{(0)}$  to perform the E-step in the very first iteration of every acquisition time step. The following both sections describe the computation of this initial values for the street surface parameters (Section 3.4.2.1) and the street boundary parameters (Section 3.4.2.2) based on the prior information provided in the previous section. Finally, Section 3.4.2.3 describes the initialization for the very first acquisition time step.

#### 3.4.2.1 Initial Values for the Street Surface

When discussing the estimation of the street surface parameters in Section 3.5.2.1 we will find that the estimation task is embodied by a linear least squares estimation problem and, thus, we do not require approximate values for the estimation of the initial parameter values. However, considering the optimization task (3.6) the estimation requires knowledge about the latent variables  $\mathbf{l}$  in particular about their posterior distribution  $P \left( \mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^- \right)$ . We use the symbol  $\hat{\boldsymbol{\theta}}_t^-$  to denote the predicted model knowledge. Hence, given initial values for the posterior probabilities, we obtain the initial parameter values in the same way as for any of the subsequent iterations.

In order to get meaningful initial posteriors, we exploit the intersected positions on the grid columns (3.53) provided by Algorithm 2. These positions allow for the determination of an initial labeling  $\mathbf{l}_t^-$  of the grid cells by simply checking on which side of the intersected position a particular grid point lies. All cells on the side facing the ego-vehicle are assigned to *street*, while all cells beyond the predicted boundary spline are assigned to *adjacent*. Treating this information as certain, we simply define the initial posterior probabilities by

$$P(\mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^-) = \begin{cases} 1, & \text{if } \mathbf{l} = \mathbf{l}_t^- \\ 0, & \text{otherwise} \end{cases} \quad (3.68)$$

and estimate the initial surface parameters  $\hat{\boldsymbol{\theta}}_t^{(0)}$  in the way described in Section 3.5.2.1.

### 3.4.2.2 Initial Values for the Street Boundary

In contrast to the street surface, we will find that the estimation of the street boundary parameters results in a non-linear optimization task due to the non-linearity of the spatial prior (3.39). We make use of the linear model (3.54) to obtain approximate values for the non linear estimation of the parameters in the very first EM iteration of the current time step.

Since not all grid columns intersect the old spline at a valid parameter value and distance, as described above, there may be insufficient observations for a proper estimation of the control point coordinates, especially in the most outer spline sections. Thus, we introduce a linear regularization term inducing low curvature assumption. This is done by claiming the local second derivatives of the  $x$  and  $y$  component of the spline function  ${}^B\mathbf{f} = [{}^Bf_x, {}^Bf_y]^\top$  to vanish at the  $I$  parameter values  $\beta_i$  aligned to the longitudinal DEM grid columns (see Section 3.3.3.1)

$${}^Bf_x''(\beta_i, {}^B\hat{\boldsymbol{\theta}}) = \sum_{m=1}^M \mathcal{B}_{m,3}''(\beta_i) {}^Bx_m = 0 \quad (3.69)$$

$${}^Bf_y''(\beta_i, {}^B\hat{\boldsymbol{\theta}}) = \sum_{m=1}^M \mathcal{B}_{m,3}''(\beta_i) {}^By_m = 0. \quad (3.70)$$

The shorthand notations  ${}^Bf_x''$  and  ${}^Bf_y''$  describe the respective second derivatives with respect to the spline parameter  $\beta$ . These terms factorize into a linear combination of the second derivatives of the spline basis functions  $\mathcal{B}_{m,3}''$  and the respective components of the spline control points  ${}^B\mathbf{x}_m = [{}^Bx_m, {}^By_m]^\top$ , as described in (2.108).

In summary, the approximate values for the boundary spline control points, denoted by  ${}^B\hat{\boldsymbol{\theta}}_t^{(0)} = [{}^B\hat{\mathbf{x}}_{1,t}^{(0)}, \dots, {}^B\hat{\mathbf{x}}_{M,t}^{(0)}]^\top$ , are estimated in a weighted least squares sense via the linear model (3.54) including the prior information (3.69) and (3.70)

$$\left\{ \begin{array}{c} 2I \\ \\ \\ \\ 2I \end{array} \right\} \left\{ \begin{array}{c} \vdots \\ \mathbf{x}_{i,t}^- \\ \vdots \\ \vdots \\ 0 \\ 0 \\ \vdots \end{array} \right\} + \begin{bmatrix} {}^B\mathbf{v}_t^{\text{temp}} \\ {}^B\mathbf{v}_{\kappa_t}^- \end{bmatrix} = \begin{bmatrix} \vdots \\ {}^B\mathbf{f}(\beta_i, {}^B\boldsymbol{\theta}) \\ \vdots \\ \vdots \\ {}^Bf_x''(\beta_i, {}^B\boldsymbol{\theta}) \\ {}^Bf_y''(\beta_i, {}^B\boldsymbol{\theta}) \\ \vdots \end{bmatrix}. \quad (3.71)$$

Substitution of the definition of  ${}^B\mathbf{f}$  given by (3.28) together with (3.69) and (3.70) reveals the linearity of the model

$$\begin{Bmatrix} 2I \\ 2I \end{Bmatrix} \left\{ \begin{bmatrix} \vdots \\ \mathbf{x}_{i,t}^- \\ \vdots \\ 0 \\ 0 \\ \vdots \end{bmatrix} + \begin{bmatrix} {}^B\mathbf{v}_t^{\text{temp}} \\ {}^B\mathbf{v}_{\kappa_t}^- \end{bmatrix} \right\} = \begin{bmatrix} \vdots \\ \mathcal{B}_{1,3}(\beta_i) & 0 & \cdots & \mathcal{B}_{M,3}(\beta_i) & 0 \\ 0 & \mathcal{B}_{1,3}(\beta_i) & \cdots & 0 & \mathcal{B}_{M,3}(\beta_i) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{B}_{1,3}''(\beta_i) & 0 & \cdots & \mathcal{B}_{M,3}''(\beta_i) & 0 \\ 0 & \mathcal{B}_{1,3}''(\beta_i) & \cdots & 0 & \mathcal{B}_{M,3}''(\beta_i) \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} {}^B\boldsymbol{\theta}. \quad (3.72)$$

The influence of the prior is governed by the curvature weighting parameter  ${}^B\lambda$  defined in Section 3.3.4.2 yielding the stochastic model

$$\begin{bmatrix} {}^B\mathbf{v}_t^{\text{temp}} \\ {}^B\mathbf{v}_{\kappa_t}^- \end{bmatrix} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} (\Sigma_{x_t^- x_t^-})^{(0)} & 0 \\ 0 & \frac{1}{{}^B\lambda} I_{2I} \end{bmatrix} \right). \quad (3.73)$$

### 3.4.2.3 Model Initialization

For the very first frame,  ${}^B\hat{\boldsymbol{\theta}}^{(0)}$  is assigned by predefined values which generate a spline that encircles a small region in front of the car, as depicted in Figure 3.19. We assume this region to cover solely street to make an initial estimate of the street surface  ${}^S\hat{\boldsymbol{\theta}}^{(0)}$  in the presented way possible.

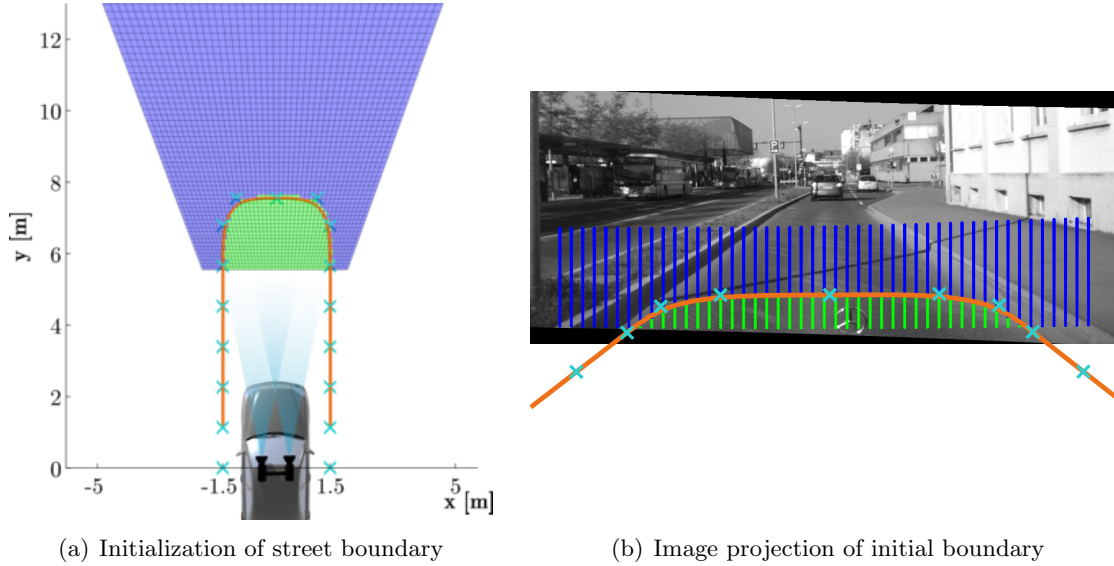


FIGURE 3.19: Initialization of the street boundary spline (orange line) plotted in bird's-eye view (a) and projection to the left input image (b). The control points are marked by light blue crosses. The colors of the grid cells denote the respective initial guess for the best labeling, whereas green means *street* blue means *adjacent*.

### 3.5 Model estimation via EM-Algorithm

This section describes the iterative procedure based on the EM-algorithm which is employed for the estimation of the model parameters via the optimization function

$$\hat{\boldsymbol{\theta}}_t^{(\nu+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[ - \sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln \wp(\mathbf{h}_t, \mathbf{l} \mid \boldsymbol{\theta}) - \ln \wp(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}_{t-1}) \right], \quad (3.74)$$

previously introduced by (3.6). The definition and actual composition of the prior term  $\wp(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}_{t-1}) := \wp({}^S\boldsymbol{\theta} \mid {}^S\hat{\boldsymbol{\theta}}_{t-1})\wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1})$  was already presented by Section 3.3.4 and Section 3.4.1.

In the following, we define the remaining terms of the optimization function and give the detailed description of the single components of the optimization procedure. This includes the description of the estimation step (processing step (IV)) and the specific definition of the posterior probabilities  $P(\mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})$  in Section 3.5.1 as well as the description of the minimization step (processing step (V)) and the specific definition of the complete-data likelihood  $\wp(\mathbf{h}_t, \mathbf{l} \mid \boldsymbol{\theta})$  in Section 3.5.2.

The total procedure is summarized in Algorithm 3. For further details on the general principles of the EM-algorithm, we refer to Section 2.6. The iterative procedure is initialized with the approximate parameter set  $\hat{\boldsymbol{\theta}}_t^{(0)}$  which is derived from the prediction step described in Section 3.4.2. Figure 3.20 provides an example depicting the intermediate results after estimation and maximization step for subsequent iterations in order to illustrate the general idea of the iterative procedure.

As a final step for each iteration of the EM-algorithm, a self diagnostic check is performed (processing step (VI)). This includes the evaluation of the current model with respect to termination criteria in order to check if another iteration is necessary. Further, the validity of the current model with respect to the model assumptions is validated for two reasons:

- To determine an appropriate stochastic model for temporal filtering of the boundary spline embodied by the choice of the noise value  $B\sigma_{r_{i,t}}^{(\nu+1)}$  for the residual term (3.66).
- To trigger a possibly necessary reinitialization of the total system causing a the model to be recomputed from scratch in the next frame.

The criteria and single parts of this processing step are discussed in Section 3.5.3

#### 3.5.1 Expectation Step (Posterior Distribution of the Latent Variables)

The expectation step embodies the evaluation of the posterior probabilities  $P(\mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})$  for every possible assignment of the latent variables  $\mathbf{l} \in \mathcal{L}$ . We define this probability by means of a Conditional Random Field (see Section 2.4.1.2) aligned to the DEM's grid structure and assume the posterior to factorize into a product of unary and binary terms

$$P(\mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \sim \prod_{(ij) \in \mathcal{D}} \Psi(l_{ij} \mid h_{ij,t}, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \prod_{(ij, i'j') \in \mathcal{N}_4} \Phi(l_{ij}, l_{i'j'} \mid h_{ij,t}, h_{i'j',t}). \quad (3.79)$$

The symbol  $\mathcal{N}_4$  denotes the set of index tuples  $(ij, i'j')$  defining all neighbors out of the 4-neighborhood of the grid. The conditional independence properties of the CRF are illustrated by the respective graphical model shown in Figure 3.21.

Start with the initial setting for the parameters  $\hat{\theta}_t^{(0)}$  determined in the prediction step;  
**repeat**

**Estimation step**

Evaluate posterior probabilities for all possible labelings  $\mathbf{l} \in \mathcal{L}$  ;  
 This is approximated by the product of the local marginals

$$P(\mathbf{l} \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \sim \prod_{(ij) \in \mathcal{D}} P(l_{ij} \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \quad (3.75)$$

computed via the Sum-Product algorithm, as described in Section 3.5.1.

**Minimization step**

Estimate the best parameters for the environment model

$$\hat{\theta}_t^{(\nu+1)} = \underset{\theta}{\operatorname{argmin}} \left[ - \sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l} \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln \wp(\mathbf{h}_t, \mathbf{l} \mid \theta) - \ln \wp(\theta \mid \hat{\theta}_{t-1}) \right]; \quad (3.76)$$

As described in Section 3.5.2, this is equivalent to independently estimate the best parameters for the street surface and the street boundary respectively:

- Estimate  ${}^S\hat{\theta}_t^{(\nu+1)} =$

$$\underset{{}^S\theta}{\operatorname{argmin}} \left[ - \sum_{(ij) \in \mathcal{D}} \sum_{l_{ij} \in C^h} P(l_{ij} \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln \wp(h_{ij,t} \mid l_{ij}, {}^S\theta) - \ln \wp({}^S\theta \mid {}^S\hat{\theta}_{t-1}) \right]; \quad (3.77)$$

- Estimate  ${}^B\hat{\theta}_t^{(\nu+1)} =$

$$\underset{{}^B\theta}{\operatorname{argmin}} \left[ - \sum_{(ij) \in \mathcal{D}} \sum_{l_{ij} \in C^h} P(l_{ij} \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln P(l_{ij} \mid {}^B\theta) - \ln \wp({}^B\theta \mid {}^B\hat{\theta}_{t-1}) \right];$$

**until** *Convergence criterion is fulfilled* ;

(3.78)

**Algorithm 3:** EM-algorithm for environment model estimation. Note that we address a minimization task and, thus, refer to the second step as minimization step rather than as maximization step.

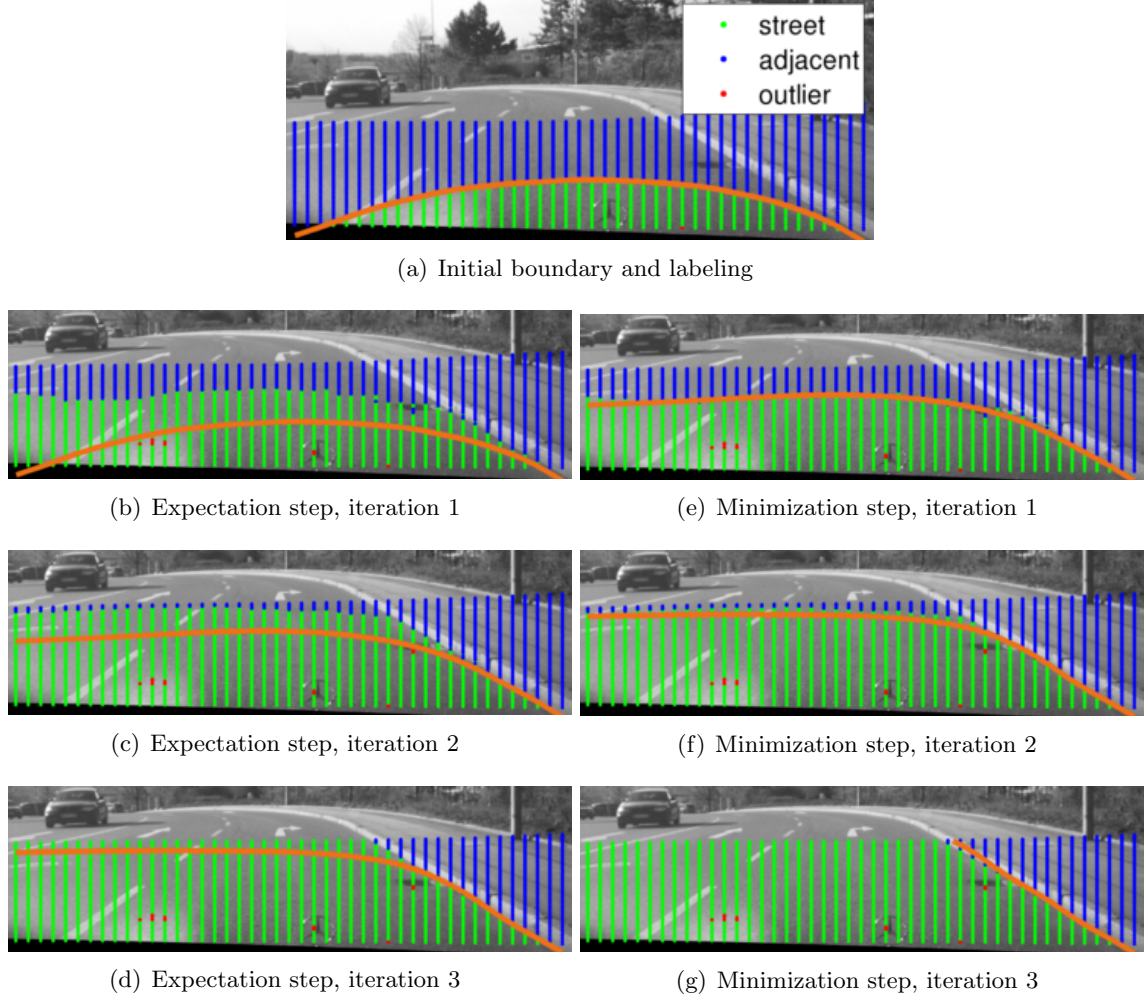


FIGURE 3.20: Example demonstrating the principle of the presented EM algorithm. The algorithm starts with an initial guess about the class decisions and model parameters, as plotted in the top figure. The initial guess is retrieved from the last frame's result via the prediction step (Section 3.4.2) or from the initialization step (Section 3.4.2.3). The estimation step (Section 3.5.1) evaluates the posterior probability of all possible assignments for the latent variables  $\mathbf{l}$  based on the relation of the currently estimated model parameters and observed elevation values. The respective most probable assignment is depicted in the figures of the left column. The minimization step (Section 3.5.2) comprises the estimation of the model parameters, namely the street surface and boundary spline control points, based on the current posterior probabilities. The respective estimated street boundaries are plotted as orange lines.

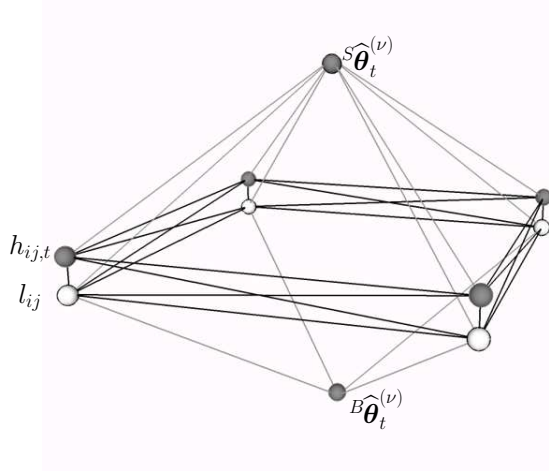


FIGURE 3.21: Graphical model describing the conditional independence properties of (3.79) in the estimation step. Dark shaded nodes are considered as observed, while bright shaded nodes are unobserved. This means, the incomplete dataset  $\mathbf{h}_t$  as well as the model parameters  $\hat{\theta}_t^{(\nu)}$  estimated in the previous iteration are observed, while only the latent variables  $\mathbf{l}$  are unobserved. For simplification the model is depicted for a  $2 \times 2$  subset of DEM grid cells. Note, since the model parameters are observed, the latent variables are conditional independent of the old model parameters estimated in the previous frame  $\theta_{t-1}$  due to the global Markov property (see Section 2.4.1.1). Thus, we omit the nodes for the old parameters in comparison to the graphical model of the complete system presented in Figure 3.3.

**Unary Terms:** The unary terms represent the influence of the observation likelihoods governed by the mixture distribution (3.21) on the assignment decision and read as

$$\Psi(l_{ij} | h_{ij,t}, \hat{\theta}_t^{(\nu)}) = P(l_{ij} | B\hat{\theta}_t^{(\nu)}) \wp(h_{ij,t} | l_{ij}, S\hat{\theta}_t^{(\nu)}). \quad (3.80)$$

For clarification, depending on the assignment of  $l_{ij}$ ,  $P(l_{ij} | B\hat{\theta}_t^{(\nu)})$  is evaluated via (3.29), (3.30) or (3.31), while  $\wp(h_{ij,t} | l_{ij}, S\hat{\theta}_t^{(\nu)})$  is evaluated via (3.24), (3.25), or (3.26).

This yields the following properties induced by the unary terms for the individual class assignment of a cell  $(ij) \in \mathcal{D}$ :

- (a) **The smaller the vertical distance of the cell's height measurement  $h_{ij,t}$  to the currently estimated street surface  $S_f$ , the more likely is the cell assigned to the class *street*.**

This property is encoded by the likelihood of the class *street*

$$\wp(h_{ij,t} | l_{ij} = \text{street}, S\hat{\theta}_t^{(\nu)}) = G(h_{ij,t} | S_f(\mathbf{x}_{ij}, S\hat{\theta}_t^{(\nu)}), S\sigma_{h_{ij,t}}^2) \quad (3.81)$$

defined in (3.24) and becomes explicit when comparing to the likelihood functions of the remaining classes in Figure 3.16.

- (b) **The position of the current boundary spline estimate  $Bf_y(\beta_i, B\hat{\theta}_t^{(\nu)})$  and the local slope  $\hat{w}_{i,t}^{(\nu)}$  of the sigmoid function (3.33) defines the area of region competition between *street* and *adjacent*.**



From (3.31) to (3.33) we obtain the spatial prior of the class *adjacent*

$$P\left(l_{ij} = adjacent \mid B\hat{\theta}_t^{(\nu)}\right) = \frac{1}{1 + \exp\left(-\hat{w}_{i,t}^{(\nu)}\left(y_{ij} - Bf_y\left(\beta_i, B\hat{\theta}_t^{(\nu)}\right)\right)\right)} \quad (3.82)$$

$$P\left(l_{ij} \neq adjacent \mid B\hat{\theta}_t^{(\nu)}\right) = 1 - \frac{1}{1 + \exp\left(-\hat{w}_{i,t}^{(\nu)}\left(y_{ij} - Bf_y\left(\beta_i, B\hat{\theta}_t^{(\nu)}\right)\right)\right)} \quad (3.83)$$

modeled by the sigmoid function depicted in Figure 3.17. In the vicinity of the sigmoid's inflection point  $Bf_y$  the influence of (3.82) and (3.83) on the unary terms (3.80) is small. This allows for a region competition of the classes *street* and *outlier* versus *adjacent* based on the likelihood functions (3.24), (3.25) and (3.26). On the contrary, with increasing distance to the inflection point the spatial prior terms (3.82) and (3.83) converge to the value 1 or 0, respectively. This suppresses the influence of the likelihood terms and, thus, restricts the area of region competition. The magnitude of the restriction is governed by the slope parameter of the sigmoid  $\hat{w}_{i,t}^{(\nu)}$ .

Note, the values of the slope parameters result from the previous minimization step. Their computation is described in Section 3.5.2.2 in the context of the estimation of the boundary parameters via logistic regression. In the initial iteration, they are defined by the fixed value  $\hat{w}_{i,t}^{(0)} := 2$ .

- (c) **Single cells in front of the current border with height measurement deviating from the current street surface are most likely assigned to the class *outlier*.**

As described in (b), the spatial prior equally favors the classes *street* and *outlier* on the near side of the boundary spline, while suppressing class *adjacent*. Therefore, the classes *street* and *outlier* compete based on their likelihood functions (3.24) and (3.25) which are given by Gaussian distributions centered at the same mean and governed by their respective standard deviations  $^S\sigma_{h_{ij,t}}$  and  $^O\sigma_{h_{ij,t}}$ . By definition of  $^O\sigma_{h_{ij,t}}$ , the class *outlier* becomes more likely than class *street* in case the distance of the local height measurement  $h_{ij,t}$  to the local height of the estimated street surface  $Sf\left(\mathbf{x}_{ij}, S\hat{\theta}_t^{(\nu)}\right)$  exceeds three times the standard deviation for height measurements of the class *street*

$$\left|h_{ij,t} - Sf\left(\mathbf{x}_{ij}, S\hat{\theta}_t^{(\nu)}\right)\right| > 3 \ ^S\sigma_{h_{ij,t}}. \quad (3.84)$$

Briefly speaking, the relation of both likelihoods is

$$\wp\left(h_{ij,t} \mid l_{ij} = outlier, S\hat{\theta}_t^{(\nu)}\right) \begin{cases} > \wp\left(h_{ij,t} \mid l_{ij} = street, S\hat{\theta}_t^{(\nu)}\right), & \text{if (3.84) is true,} \\ \leq \wp\left(h_{ij,t} \mid l_{ij} = street, S\hat{\theta}_t^{(\nu)}\right), & \text{if (3.84) is false.} \end{cases} \quad (3.85)$$

Figure 3.16 illustrates this relation showing the likelihood of the class *outlier* to exceed the likelihood of the class *street* for values of  $h$  beyond the interval  $\left[Sf\left(\mathbf{x}_{ij}, S\hat{\theta}_t^{(\nu)}\right) \pm 3 \ ^S\sigma_{h_{ij,t}}\right]$ .

Note, neglecting the remaining components of the unary and binary terms, this property implicitly represents a 3-sigma outlier test, i.e. a cell is classified as *outlier* if the local height measurement exceeds the 3-sigma interval around the local height of the estimated street surface.

**Binary Terms:** The binary terms  $\Phi$  introduce prior knowledge on the neighborhood relations regarding the cell assignments. These prior assumptions are:

- The segmentation is assumed to be smooth and class transitions are preferred at height discontinuities.
- The class *outlier* is assumed to occur occasionally making two *outlier* cells unlikely to be neighbors.

The first assumption is modeled by favoring the assignment of the same labels via a data dependent Potts model. Neighboring cells are the more likely to belong to the same class the smaller the weighted difference of their height observations

$$\Phi(l_{ij} = l_{i'j'} \neq \text{outlier} \mid h_{ij,t}, h_{i'j',t}) = 1 \quad (3.86)$$

$$\Phi(l_{ij} = l_{i'j'} = \text{outlier} \mid h_{ij,t}, h_{i'j',t}) = 0.1 \quad (3.87)$$

$$\Phi(l_{ij} \neq l_{i'j'} \mid h_{ij,t}, h_{i'j',t}) = 1 - \exp\left(-\frac{1}{2} \frac{(h_{ij,t} - h_{i'j',t})^2}{\sigma_{h_{ij,t}}^2 + \sigma_{h_{i'j',t}}^2}\right). \quad (3.88)$$

The second assumption is modeled by a significantly smaller value in (3.87) compared to (3.86). Figure 3.22 depicts examples for the binary terms.

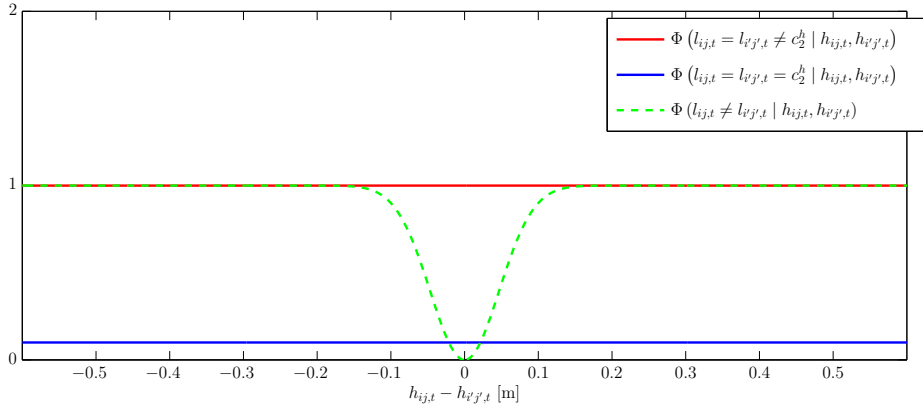


FIGURE 3.22: Example for the binary potentials  $\Phi$ . Class changes are punished if the height difference of neighboring cells is small. The punishment nearly vanishes at discontinuities exceeding 10 cm in height. Further, cells assigned to the class *outlier* are unlikely to be neighbors.

**Inference** As the exact evaluation for each possible assignment is not feasible considering the background of a real-time application, we approximate the posterior by the product of the marginals of the single cell assignments

$$P(\mathbf{l} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \approx \prod_{(ij) \in \mathcal{D}} P(l_{ij} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}). \quad (3.89)$$

Thus, we can make use of the simplified formulation of the EM algorithm described in Section 2.6.2.

To estimate the marginals, we employ Sum-Product algorithm (Loopy Belief Propagation), as described for example in [MacKay, 2003, pp. 334 to 340].

An alternative to the Sum-Product algorithm would be to solve for the most probable assignment, e.g. via Graph Cuts or Max-Product algorithm, and approximate the posterior

distribution via local marginalization in a similar way. However, the decision for Sum-Product algorithm is based on the fact that this algorithm can be implemented efficiently exploiting the potential for parallel processing of the multi-core processor architecture of today's computers which allows for the desired real time performance.

Note, that although the presented approach yields an approximative solution, our experiments show a considerable improvement compared to a cell-wise classification where neighborhood relations are neglected (see Section 4.3.4.3).

### 3.5.2 Minimization Step (Estimation of the Model Parameters)

The minimization step (processing step (V)) embodies the actual estimation of the model parameters. With the prior term and the posterior probabilities defined, the only term missing for the evaluation of the optimization function (3.74) is the complete-data likelihood  $\wp(\mathbf{h}_t, \mathbf{l} \mid \boldsymbol{\theta})$ .

The complete-data likelihood represents the validity of the environment model defined by the mixture distribution (3.21) for each cell of the DEM. More precisely, the likelihood is defined in terms of the spatial class priors  $P(l_{ij} \mid {}^B\boldsymbol{\theta})$  embodied by (3.29) to (3.31) and the local likelihoods of the class assignments with respect to the height measurements (3.24) to (3.26) and reads as

$$\wp(\mathbf{h}_t, \mathbf{l} \mid \boldsymbol{\theta}) := \prod_{(ij) \in \mathcal{D}} \wp(h_{ij,t}, l_{ij} \mid {}^B\boldsymbol{\theta}, {}^S\boldsymbol{\theta}) \quad (3.90)$$

$$:= \prod_{(ij) \in \mathcal{D}} P(l_{ij} \mid {}^B\boldsymbol{\theta}) \wp(h_{ij,t} \mid l_{ij}, {}^S\boldsymbol{\theta}). \quad (3.91)$$

Substituting the factorization of the joint probability (3.90) and the approximation of the posterior of the latent variables (3.89) into the optimization function (3.74) yields

$$\hat{\boldsymbol{\theta}}_t^{(\nu+1)} \approx \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[ - \sum_{\mathbf{l} \in \mathcal{L}} \prod_{(ij) \in \mathcal{D}} P(l_{ij} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \sum_{(ij) \in \mathcal{D}} \ln \wp(h_{ij,t}, l_{ij} \mid \boldsymbol{\theta}) - \ln \wp(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}_{t-1}) \right]. \quad (3.92)$$

Employing the simplified formulation of the maximization step given in Section 2.6.2, the estimation task simplifies to

$$\hat{\boldsymbol{\theta}}_t^{(\nu+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[ - \sum_{(ij) \in \mathcal{D}} \sum_{l_{ij} \in \mathcal{C}^h} P(l_{ij} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln \wp(h_{ij,t}, l_{ij} \mid \boldsymbol{\theta}) - \ln \wp(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}_{t-1}) \right]. \quad (3.93)$$

Finally, substituting the factorization (3.91) and the assumption of independent priors  $\wp(\boldsymbol{\theta} \mid \hat{\boldsymbol{\theta}}_{t-1}) := \wp({}^S\boldsymbol{\theta} \mid {}^S\hat{\boldsymbol{\theta}}_{t-1})\wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1})$ , the total estimation task decomposes into independent estimation tasks for the street surface parameters

$${}^S\hat{\boldsymbol{\theta}}_t^{(\nu+1)} = \underset{{}^S\boldsymbol{\theta}}{\operatorname{argmin}} \left[ - \sum_{(ij) \in \mathcal{D}} \sum_{l_{ij} \in \mathcal{C}^h} P(l_{ij} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln \wp(h_{ij,t} \mid l_{ij}, {}^S\boldsymbol{\theta}) - \ln \wp({}^S\boldsymbol{\theta} \mid {}^S\hat{\boldsymbol{\theta}}_{t-1}) \right] \quad (3.94)$$

and the street boundary parameters

$${}^B\hat{\boldsymbol{\theta}}_t^{(\nu+1)} = \underset{{}^B\boldsymbol{\theta}}{\operatorname{argmin}} \left[ - \sum_{(ij) \in \mathcal{D}} \sum_{l_{ij} \in \mathcal{C}^h} P(l_{ij} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln P(l_{ij} \mid {}^B\boldsymbol{\theta}) - \ln \wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1}) \right]. \quad (3.95)$$

Keep in mind that we already observed this decomposition property in Section 3.1 in the discussion of the conditional independence properties of the complete system defined by the graphical model depicted in Figure 3.3(c).

The following paragraphs give a detailed description of both estimation tasks, which both appear in the manner of a MAP estimation task in a Gauss-Markov model (see Section 2.3.2.2). The paragraphs include the explicit formulation of the functional and stochastic models as well as prior assumptions and matrix dimensions.

### 3.5.2.1 Estimation of the Street Surface Parameters

The MAP estimation of the street surface parameters for the current iteration  $\nu + 1$  embodies the determination of the minimum argument of (3.94). Remind that  $\wp(h_{ij,t} | l_{ij} = \textit{adjacent}, {}^S\theta)$  is defined by the uniform distribution (3.26) and, thus, yields a constant value for the considered height interval  $\hat{h} = [-1.5, 1.5]$  m. Therefore, the term is independent of the unknown parameters  ${}^S\theta$  and can be ignored in the estimation yielding the simplified estimation task

$$\begin{aligned} \hat{{}^S\theta}_t^{(\nu+1)} = \operatorname{argmin}_{{}^S\theta} & \left[ - \sum_{(ij) \in \mathcal{D}} P(l_{ij} = \textit{street} | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln \wp(h_{ij,t} | l_{ij} = \textit{street}, {}^S\theta) \right. \\ & + P(l_{ij} = \textit{outlier} | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln \wp(h_{ij,t} | l_{ij} = \textit{outlier}, {}^S\theta) \\ & \left. - \ln \wp({}^S\theta | {}^S\hat{\theta}_{t-1}) \right]. \end{aligned} \quad (3.96)$$

Put simply, the surface parameters are estimated from the height measurements of all cells located in the drivable free-space region (represented by *street* and *outlier*) weighted by the probability that they belong to this region. This matches the key idea for the estimation of the street surface parameters proposed in Section 3.1.

For illustration purpose, Figure 3.23 depicts the dependency structure of the involved variables.

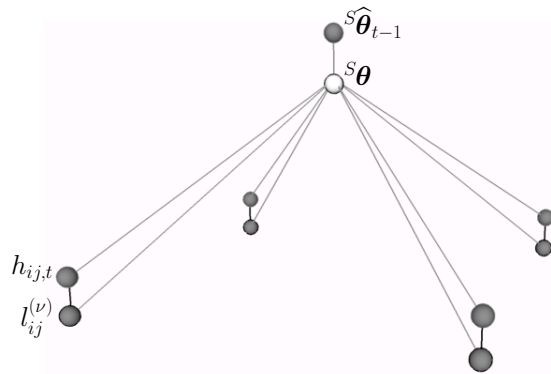


FIGURE 3.23: Graphical model depicting the conditional independence properties of the variables involved in the estimation of the surface parameters. As in the graphical representation of the complete system presented in Figure 3.3, only a  $2 \times 2$  subset of DEM grid cells is depicted and dark shaded nodes are considered as observed, while bright shaded nodes are unobserved. Note, since both the latent variables and the height measurements are observed, the surface parameters are conditional independent of the boundary parameters due to the global Markov property (see Section 2.4.1.1). Thus, the boundary parameters are omitted in the graph since they do not participate in the estimation of the surface parameters.

Since the remaining distributions in 3.96, i.e. the distribution of the height observations and the distribution of the fictitious observations forming the prior term 3.35, are given by normal distributions, we formulate the MAP-estimation task in terms of a Gauss-Markov model similar to (2.45).

The likelihood term is defined by the respective likelihood terms of the surface model (3.24) and (3.25) reweighted by the posterior probabilities  $P(l_{ij,t} | \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})$ . The prior terms model the assumptions of spatial and temporal smoothness, as described in Section 3.3.4.2 and Section 3.4.1.1. In combination, we obtain the linear functional model

$$\begin{bmatrix} \vdots \\ h_{ij,t} \\ \vdots \\ h_{ij,t} \\ \vdots \\ s\mathbf{z}_t^{\text{temp}} \\ s\mathbf{z}^{\text{spät}} \end{bmatrix} + s\mathbf{v}_t^{(\nu+1)} = \begin{bmatrix} \vdots \\ s f([x_{ij}, y_{ij}]^\top, s\boldsymbol{\theta}) \\ \vdots \\ s f([x_{ij}, y_{ij}]^\top, s\boldsymbol{\theta}) \\ \vdots \\ s\mathbf{k}^{\text{temp}}(s\boldsymbol{\theta}, s\hat{\boldsymbol{\theta}}_{t-1}) \\ s\mathbf{k}^{\text{spät}}(s\boldsymbol{\theta}) \end{bmatrix}. \quad (3.97)$$

More precisely, substitution of the respective definitions, i.e. (3.22) for  $s f$ , (3.36) for the spatial prior, and (3.46) for the temporal prior yields

$$\begin{array}{c} \left. \begin{array}{c} IJ \\ \\ IJ \\ \\ Q \\ \\ 3Q \end{array} \right\} \begin{bmatrix} \vdots \\ h_{ij,t} \\ \vdots \\ h_{ij,t} \\ \vdots \\ h_{q,t}^- \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} + s\mathbf{v}_t^{(\nu+1)} = \underbrace{\begin{bmatrix} \vdots \\ \dots \mathcal{B}_{n,2}(x_{ij}, y_{ij}) \dots \\ \vdots \\ \dots \mathcal{B}_{n,2}(x_{ij}, y_{ij}) \dots \\ \vdots \\ \dots \mathcal{B}_{n,2}(x_{q,t}^-, y_{q,t}^-) \dots \\ \vdots \\ \dots \frac{1}{\sqrt{2}} \mathcal{B}_{n,2}^{(2,0)}(x_q, y_q) \dots \\ \dots \mathcal{B}_{n,2}^{(1,1)}(x_q, y_q) \dots \\ \dots \frac{1}{\sqrt{2}} \mathcal{B}_{n,2}^{(0,2)}(x_q, y_q) \dots \\ \vdots \end{bmatrix}}_{s\mathbf{A}} \underbrace{\begin{bmatrix} \vdots \\ s h_n \\ \vdots \end{bmatrix}}_{s\boldsymbol{\theta}}, \quad (3.98)$$

$\underbrace{\begin{bmatrix} \vdots \\ h_{ij,t} \\ \vdots \\ h_{ij,t} \\ \vdots \\ h_{q,t}^- \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}}_{s\mathbf{z}_t}$

with the  $(2IJ + 4Q \times N)$  design matrix  $s\mathbf{A}$  whose entries are given by second order B-spline basis functions  $\mathcal{B}_{n,2}$  and their derivatives.

The stochastic model  $S\mathbf{v}_t^{(\nu+1)} \sim S\Sigma_{z_t z_t}^{(\nu+1)}$  is governed by the covariance matrix

$$S\Sigma_{z_t z_t}^{(\nu+1)} = \begin{bmatrix} \ddots & & & & \\ & \underbrace{\frac{S\sigma_{h_{ij,t}}^2}{P(l_{ij}=street|\mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})}}_{IJ \times IJ} & & & \\ & & \ddots & & \\ & & & \underbrace{\frac{O\sigma_{h_{ij,t}}^2}{P(l_{ij}=outlier|\mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})}}_{IJ \times IJ} & \\ & & & & \underbrace{\Sigma_{h_t^- h_t^-}}_{Q \times Q} \\ & & & & & \underbrace{\frac{1}{S_\lambda} I_{3Q}}_{3Q \times 3Q} \end{bmatrix}, \quad (3.99)$$

where unprinted elements equal zero. The variances of the blocks dedicated to the height observations correspond to the different variances of the elevation models for *street* and *outlier* measurements defined in Section 3.3.2.1. The blocks dedicated to the fictitious observations of the prior terms are adopted from the respective definitions (3.49) and (3.37). Note that the posterior probabilities  $P(l_{ij} | \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})$  appear in the denominator since the covariance matrix represents the inverse of the weight matrix.

The best estimate of the current surface parameters  $S\hat{\boldsymbol{\theta}}_t^{(\nu+1)}$  and the corresponding precision  $\left(S\Sigma_{\hat{\boldsymbol{\theta}}_t \hat{\boldsymbol{\theta}}_t}\right)^{(\nu+1)}$  are obtained by the calculus described in Section 2.3.2.2. An example for an estimated surface comprising the participating 3d point entities is illustrated in Figure 3.24.

### 3.5.2.2 Estimation of the Street Boundary Parameters

This section explains the estimation of the boundary parameters via the optimization function (3.95) as well as the estimation of the slope parameters  $\hat{\mathbf{w}}^{(\nu+1)}$  required for the evaluation of the unary terms in the subsequent expectation step. For a better understanding of the estimation procedure, let us first discuss the key idea in an illustrative manner before we consider the mathematical derivation.

By definition, the street boundary spline represents the horizontal delimiter of the drivable free-space, i.e. the separation of the class *street* (including *outlier*) and the class *adjacent*. Put simply, the street boundary represents the decision boundary between the class *adjacent* and the remaining classes. Thus, given a certain class assignment of the DEM cells  $\mathbf{l}$ , the key idea is to estimate the decision boundary based on the model introduced by the sigmoid functions  $P(l_{ij} = adjacent | {}^B\boldsymbol{\theta}) = g_i({}^B\boldsymbol{\theta}, y_{ij})$  in Section 3.3.3.2. The decision boundary along each grid column (i.e. the 0.5-level-set of the respective sigmoid) is defined by the position of the boundary spline evaluated at the respective spline parameter value  $\beta_i$ . This condition

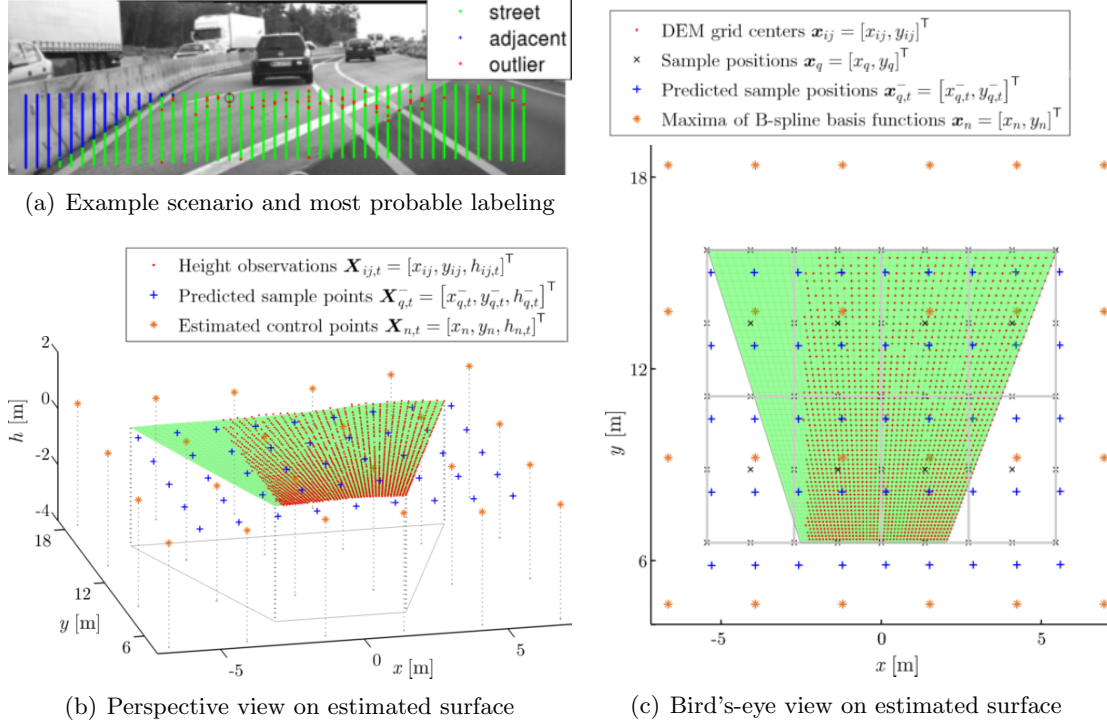


FIGURE 3.24: Example of an estimated street surface. The example scenario is depicted in (a) including the image projection of the DEM grid points. The colors encode the determined most probable labeling. The remaining figures depict the geometric entities participating in the estimation of the street surface described by the functional model (3.98) as well as the estimated street surface itself (green surface). Figure (c) shows a bird's-eye view with the horizontal positions of the entities, while Figure (b) additionally shows their vertical component, if available. The entities are given by: 1. The DEM grid points containing the current height observations  $\mathbf{X}_{ij,t} = [x_{ij}, y_{ij}, h_{ij,t}]^T$  plotted as red dots. Note that only those points are plotted, which participate in the estimation process, i.e. those that are classified as *street* or *outlier*. 2. The horizontal sample positions  $\mathbf{x}_q = [x_q, y_q]^T$  used to apply the curvature constraint and to predict the surface information for the next time step. The sample points are marked by black crosses and omitted in the perspective view since they lack a vertical component. 3. The predicted sample points from the last time step  $\mathbf{X}_{q,t}^- = [x_{q,t}^-, y_{q,t}^-, h_{q,t}^-]^T$  are plotted as blue crosses. Note that their horizontal position corresponds to  $\mathbf{x}_q$  in the previous time step transformed to the current system via ego-motion information. 4. The estimated control points  ${}^S h_{n,t}$  of the spline surface. The maximum position  $\mathbf{x}_n$  of the respective B-spline basis function (see Figure 2.6(a)) is used to form the horizontal component of the respective plotted 3d point  $\mathbf{X}_{n,t}$ . These points are marked by orange stars. Additionally, the spline section borders are drawn as gray lines in the bird's-eye view.

yields a lateral connection of the grid columns and embodies the functional relation of the latent variables and the street boundary parameters. Figure 3.17 visualizes the interaction of the boundary spline and the sigmoid functions.

Note that given  $\mathbf{l}$ , the estimation of the street boundary parameters is independent of the street surface parameters and the current height observations, as indicated by the graphical model presented in Figure 3.25.

In next paragraph, the optimization function (3.95) is reformulated in the manner of a logistic regression task in order to demonstrate the applicability of the just presented idea. The subsequent paragraphs present two strategies for the actual estimation of the boundary and slope parameters. Both strategies are based on the calculus of logistic regression presented in Section 2.5 in combination with the least squares procedure explained in Section 2.3.2.2.

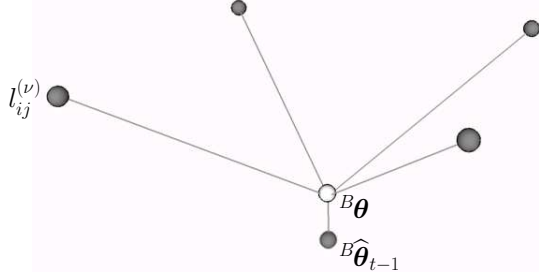


FIGURE 3.25: Graphical model depicting the conditional independence properties of the variables involved in the estimation of the boundary parameters. As in the graphical representation of the complete system presented in Figure 3.3, only a  $2 \times 2$  subset of DEM grid cells is depicted and dark shaded nodes are considered as observed, while bright shaded nodes are unobserved. Note, in case the latent variables are observed, the boundary parameters are conditional independent of the height measurements and the surface parameters due to the global Markov property (see Section 2.4.1.1). Thus, these entities do not participate in the estimation of the boundary parameters and are omitted in the graph.

### Transformation of the Optimization Task into an Instance of Logistic Regression:

The MAP estimation of the street boundary parameters for the current iteration  $\nu + 1$  embodies the determination of the minimum argument of (3.95). From the definition of the probabilities  $P(l_{ij} | {}^B\theta)$  given by (3.29), (3.30) and (3.31) follows the identity

$$P(l_{ij} = street | {}^B\theta) = P(l_{ij} = outlier | {}^B\theta) = \frac{1 - P(l_{ij} = adjacent | {}^B\theta)}{2} \quad (3.100)$$

$$= \frac{P(l_{ij} \neq adjacent | {}^B\theta)}{2}. \quad (3.101)$$

With this identity, we can reformulate the optimization task (3.95) expanding the sum over the possible class assignments

$$\begin{aligned} {}^B\hat{\theta}_t^{(\nu+1)} = \underset{{}^B\theta}{\operatorname{argmin}} \left[ - \sum_{(ij) \in \mathcal{D}} P(l_{ij} = street | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln \frac{P(l_{ij} \neq adjacent | {}^B\theta)}{2} \right. \\ + P(l_{ij} = outlier | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln \frac{P(l_{ij} \neq adjacent | {}^B\theta)}{2} \\ + P(l_{ij} = adjacent | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) \ln P(l_{ij} = adjacent | {}^B\theta) \\ \left. - \ln \wp({}^B\theta | {}^B\hat{\theta}_{t-1}) \right]. \quad (3.102) \end{aligned}$$

Finally, exploiting the fact that the posterior probabilities sum to 1 with respect to the classes, i.e.

$$P(l_{ij} = street | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) + P(l_{ij} = outlier | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}) = P(l_{ij} \neq adjacent | \mathbf{h}_t, \hat{\theta}_t^{(\nu)}), \quad (3.103)$$



allows to further simplify the formulation by summarizing the first both terms of the sum. We obtain

$$\begin{aligned} {}^B\hat{\boldsymbol{\theta}}_t^{(\nu+1)} = \operatorname{argmin}_{{}^B\boldsymbol{\theta}} & \left[ - \sum_{(ij) \in \mathcal{D}} P(l_{ij} = \textit{adjacent} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln P(l_{ij} = \textit{adjacent} \mid {}^B\boldsymbol{\theta}) \right. \\ & \quad \left. + P(l_{ij} \neq \textit{adjacent} \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \ln P(l_{ij} \neq \textit{adjacent} \mid {}^B\boldsymbol{\theta}) \right. \\ & \quad \left. - \ln \wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1}) \right]. \end{aligned} \quad (3.104)$$

Note that the factor  $1/2$  vanishes since the logarithm yields a constant additive term which is independent of  ${}^B\boldsymbol{\theta}$ .

As defined by (3.31) to (3.33), the probability  $P(l_{ij} = \textit{adjacent} \mid {}^B\boldsymbol{\theta})$  is given by the local evaluation of the sigmoid function

$$g_i({}^B\boldsymbol{\theta}, y_{ij}) = \frac{1}{1 + \exp(-w_i(y_{ij} - {}^Bf_y(\beta_i, {}^B\boldsymbol{\theta})))}. \quad (3.105)$$

Thus, (3.104) roughly corresponds to a logistic regression task with non deterministic targets (see Section 2.5.4) for each longitudinal grid column, with targets defined by

$$\mathbf{t}_{ij} = \begin{cases} 1, & \text{if } l_{ij} = \textit{adjacent} \\ 0, & \text{if } l_{ij} \neq \textit{adjacent} \end{cases}. \quad (3.106)$$

The only differences to the task presented in Section 2.5.4 are given by the additional prior information  $\wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1})$  and the explicit representation of the decision boundary in  $g_i({}^B\boldsymbol{\theta}, y_{ij})$  via  ${}^Bf_y$  in comparison to the logistic regression model (2.59).

**Estimation Strategies:** We present two approaches for the estimation of the boundary spline which both employ logistic regression for the estimation of the decision boundary and the slope parameters. For detailed information about the principles of logistic regression, we refer to Section 2.5.

The first approach decomposes the task in three successive steps with the first both of them illustrated in Figure 3.26. In the first step, the decision boundary is computed for each DEM  $y$ -column independently using logistic regression. This yields one observation of the form

$$\left(\mathbf{x}_{i,t}^+\right)^{(\nu+1)} + {}^B\mathbf{v}_{i,t}^{(\nu+1)} = {}^B\mathbf{f}\left(\beta_i, {}^B\boldsymbol{\theta}_t^{(\nu+1)}\right) \quad (3.107)$$

for every  $y$ -column  $i$ . In the second step, this observation model is combined with the prior assumptions  $\wp({}^B\boldsymbol{\theta} \mid {}^B\hat{\boldsymbol{\theta}}_{t-1})$  to perform MAP estimation of the unknown parameters in a weighted least squares sense. The final step re-estimates the slope parameters  $\hat{\mathbf{w}}^{(\nu+1)}$  in order to adapt them to the new decision boundary induced by the estimated boundary spline. The effect of this adaption is illustrated in Figure 3.27.

As an alternative to this three-step approach we present an one-step approach which demonstrates how to reformulate the logistic regression task and directly integrate it into the MAP estimation.

However, we prefer the three-step approach over the one-step approach due to several reasons:

- We observed a poor convergence behavior of the one-step approach especially in the case of imprecise approximate values. We assume this behavior to be caused by the high-grade non-linearity of the functional model.
- It requires a high amount of additional regularization effort in the one-step approach to keep the estimated boundary inside the range of the grid boundaries  $[y_{\min}, y_{\max}]$ .
- The explicit formulation of the decision boundary via the observations  $\mathbf{x}_{i,t}^+$  in the three-step approach allows for consideration of the variation of the decision boundary over successive EM iterations, as described in (3.111).
- The reduced complexity of the equation system as well as the better convergence behavior of the three-step approach result in reduced computational effort.

Thus, although approximative, we deem the three-step approach to be more suitable for practical usage. Nevertheless, we also present the one-step approach for the sake of completeness but will not consider it in the remainder of this thesis.

The following paragraphs describe both approaches in detail beginning with the three-step approach.

### Three-Step Approach for the Estimation of the Boundary Spline:

**Step 1, Boundary Sample Observations from Logistic Regression:** As described above, the first step of the approach determines a set of samples  $\left(\mathbf{x}_{i,t}^+\right)^{(\nu+1)}$  which represent the most likely separation between the class set  $\{street, outlier\}$  and the class *adjacent* along the longitudinal columns of the DEM grid, as depicted in Figure 3.26(a). The samples are computed for each grid column independently via logistic regression using the procedure presented in Section 2.5. For this purpose, we introduce target values  $\mathbf{t}_{ij} = 0$  for all cells assigned to *street* or *outlier* and  $\mathbf{t}_{ij} = 1$  for those assigned to *adjacent*. The  $y$ -coordinate of the cell centers act as respective feature positions with uncertainties  $\sigma_{y_{ij}}$  embodied by the discretization uncertainties (3.14).

As result, we obtain the distance  $\left(y_{i,t}^+\right)^{(\nu+1)}$  of the decision boundary for each longitudinal column of the DEM  $i = 1 \dots I$ .

The corresponding  $x$  coordinates follow from exploiting the relation (3.7) of the cell coordinates. The corresponding image row coordinate  $u_i$  and the principle distance  $c$  and read as

$$\left(x_{i,t}^+\right)^{(\nu+1)} = \frac{u_i}{c} \left(y_{i,t}^+\right)^{(\nu+1)}. \quad (3.108)$$

In combination we achieve the 2d-samples

$$\left(\mathbf{x}_{i,t}^+\right)^{(\nu+1)} = \begin{bmatrix} \frac{u_i}{c} \left(y_{i,t}^+\right)^{(\nu+1)} \\ \left(y_{i,t}^+\right)^{(\nu+1)} \end{bmatrix}. \quad (3.109)$$

Let us now consider the stochastic properties of the 2d-samples. The lateral precision of each sample is governed by the local lateral discretization uncertainty of the DEM grid. Considering the constant cell width  $d_u$  with respect to the image projection of the DEM

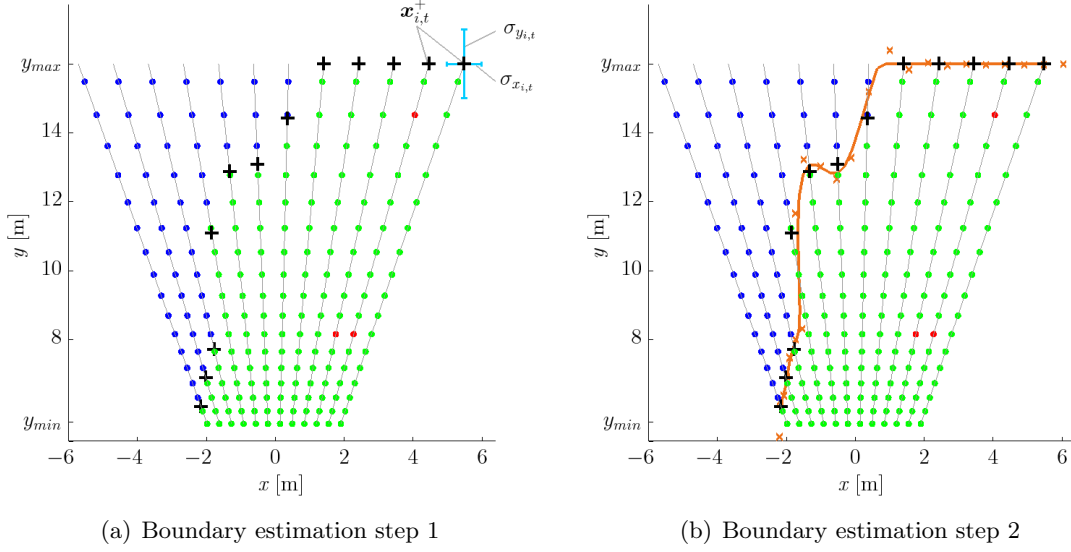


FIGURE 3.26: Principle procedure of the proposed two-step approach for the estimation of the boundary spline. The utilized scenario is illustrated in Figure 3.28(b). Figure (a) depicts the procedure for the first step. For each  $y$ -column  $i$  (gray lines) a single sample position  $\mathbf{x}_{i,t}^+$  (black pluses) is computed via logistic regression in order to define a separation along the  $y$ -column between cells assigned to *street* (green dots) or *outlier* (red dots) and those assigned to *adjacent* (blue dots). By means of the right most sample, the derived lateral and longitudinal uncertainties  $\sigma_{x_{i,t}}$  and  $\sigma_{y_{i,t}}$  are shown schematically. Figure (b) illustrates the second step. The computed set of sample positions is used to estimate the control points  ${}^B\boldsymbol{\theta}_t$  governing the boundary spline. The estimated control points are plotted by orange crosses. The estimated boundary spline is drawn in orange color. Note that only a sparse subset of  $y$ -columns and DEM cell centers is plotted due to visualization issues.

grid, as depicted in Figure 3.7(a), the discretization uncertainty is given by  $\sigma_u = \frac{d_u}{\sqrt{12}}$ . Error propagation via (3.109) results in the lateral precision of the samples

$$\left(\sigma_{x_{i,t}^+}^{(\nu+1)}\right)^2 = \left(\frac{\left(y_{i,t}^+\right)^{(\nu+1)}}{c}\right)^2 \sigma_u^2. \quad (3.110)$$

We define the precision in longitudinal direction by means of two terms

$$\left(\sigma_{y_{i,t}^+}^{(\nu+1)}\right)^2 = \left(\sigma_{y_{i,t}}^{(\nu+1)}\right)^2 + \frac{1}{\nu+1} \sum_{v=1}^{\nu+1} \left(\left(y_{i,t}^+\right)^{(v)} - \left(y_{i,t}^+\right)^{(\nu+1)}\right)^2. \quad (3.111)$$

The first term models the precision obtained from logistic regression, as described in (2.82). We add the second term to consider the variation of the samples over the EM iterations. This causes observations which strongly vary over the iterations, such as spline regions still propagating from the initialization, to obtain a high uncertainty value.

Samples exceeding the longitudinal grid boundaries  $[y_{\min}, y_{\max}]$  are set to  $y_{\min}$  and  $y_{\max}$ , respectively, and are assigned with an fictitious uncertainty value of 1 m for the longitudinal component.

In summary, linear error propagation yields the covariance matrix of the 2d samples

$$\left(\Sigma_{x_{i,t}^+, x_{i,t}^+}\right)^{(\nu+1)} = \begin{bmatrix} \frac{\left(y_{i,t}^+\right)^{(\nu+1)}}{c} & \frac{u_i}{c} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{d_u^2}{12} & 0 \\ 0 & \left(\sigma_{y_{i,t}^+}^{(\nu+1)}\right)^2 \end{bmatrix} \begin{bmatrix} \frac{\left(y_{i,t}^+\right)^{(\nu+1)}}{c} & \frac{u_i}{c} \\ 0 & 1 \end{bmatrix}^T, \quad (3.112)$$

with the Jacobians derived from (3.109).



The parameters  $B\hat{\theta}_t^{(\nu+1)}$  are estimated in a weighted least squares sense employing the Gauss-Newton algorithm due to the non linearity of the problem, as described in Section 2.3.3. Approximate values are given by the result of the previous EM iteration result  $B\hat{\theta}_t^{(\nu)}$  or, in the very first iteration of each time step, obtained from the prediction step described in Section 3.4.2.2. The respective  $(4I + A \times 2I)$  design matrix for iteration  $\nu$  which embodies the Jacobian of (3.113) reads as

$$B\mathbf{A}^{(\nu)} = \begin{bmatrix} \vdots & & & \vdots & & \\ & \mathcal{B}_{1,3}(\beta_i) & 0 & & \mathcal{B}_{M,3}(\beta_i) & 0 \\ & 0 & \mathcal{B}_{1,3}(\beta_i) & \cdots & 0 & \mathcal{B}_{M,3}(\beta_i) \\ & \vdots & & & \vdots & \\ & \vdots & & & \vdots & \\ & \mathcal{B}_{1,3}(\beta_i) & 0 & & \mathcal{B}_{M,3}(\beta_i) & 0 \\ & 0 & \mathcal{B}_{1,3}(\beta_i) & \cdots & 0 & \mathcal{B}_{M,3}(\beta_i) \\ & \vdots & & & \vdots & \\ & \vdots & & & \vdots & \\ \frac{\partial \kappa_{\beta_a}}{\partial Bx_1} \Big|_{B\theta = B\hat{\theta}^{(\nu)}} & \frac{\partial \kappa_{\beta_a}}{\partial By_1} \Big|_{B\theta = B\hat{\theta}^{(\nu)}} & \cdots & \frac{\partial \kappa_{\beta_a}}{\partial Bx_M} \Big|_{B\theta = B\hat{\theta}^{(\nu)}} & \frac{\partial \kappa_{\beta_a}}{\partial By_M} \Big|_{B\theta = B\hat{\theta}^{(\nu)}} \\ & \vdots & & & \vdots & \end{bmatrix}. \quad (3.115)$$

The entries embody the partial derivatives of the boundary spline function (3.28) and the partial derivatives of the curvature given by (2.110) and (2.111).

**Step 3, Adaption of the Slope Parameters:** The final step embodies the estimation of the sigmoid slope parameters  $\hat{w}_t^{(\nu+1)}$ . The slope parameters can be seen as control parameters for an adaptive step size modulation for the iterative update of the boundary spline. Considering the definition of the unary terms in Section 3.5.1, a lower slope parameter causes a lower influence of the spatial prior term in the subsequent estimation step of the EM-algorithm, i.e. the area of region competition becomes larger. This in turn allows for a larger shift of the boundary spline in the subsequent minimization step. Put simply, the quality of the decision boundary governs the step size via the slope parameters.

Let us first discuss why we refrain from taking the slope parameters derived from the logistic regression step (step 1). The adjusted boundary spline, which by definition represents the decision boundary between the class *adjacent* and the remaining classes, in general differs from the samples  $(\mathbf{x}_{i,t}^+)^{(\nu+1)}$  derived in step 1. This means in particular that the decision boundary along the grid columns, which is embodied by the local evaluation of the spline  $B\mathbf{f}(\beta_i, B\hat{\theta}_t^{(\nu+1)})$ , is shifted in comparison to the decision boundary obtained from the logistic regression step. Since the slope of the logistic function is correlated to the quality of the class separation given by the decision boundary, as illustrated in Figure 3.27, the slope parameters estimated from the logistic regression step may be inappropriate.

The procedure for the estimation of the local slope parameter  $\hat{w}_{i,t}^{(\nu+1)}$  corresponds to the minimization of the original logistic regression cost function (2.94) keeping the decision boundary  $y_0 = B\mathbf{f}(\beta_i, B\hat{\theta}_t^{(\nu+1)})$  fixed. This means, with the definition of the decision boundary (2.61), we can reformulate the sigmoid function (2.59) in terms of the decision

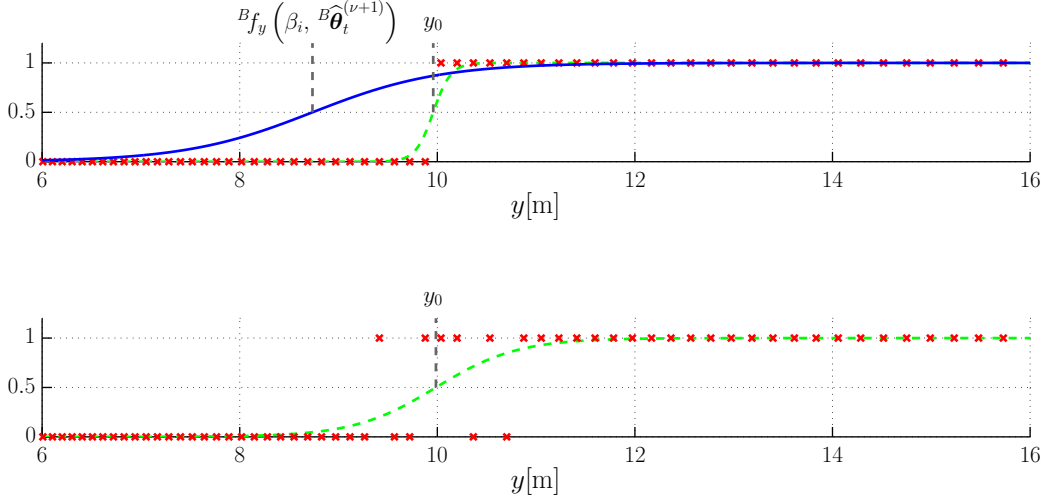


FIGURE 3.27: Example plots describing the meaning of the slope of the sigmoid functions which represent the posterior probability for the assignment of class *adjacent*. The functions are estimated from the targets depicted by red crosses. The vertical axis describes the value of the targets  $t_{i,j}$  and the sigmoid functions, respectively. The two functions plotted by green dashed lines result from the logistic regression approach presented in Section 2.5, while the blue line results from the adaption of the slope of the sigmoid function to the shifted decision boundary  $Bf_y(\beta_i, B\hat{\theta}_t^{(\nu+1)})$ . Comparison of the functions reveals that the slope depends on how well the decision boundary (0.5-level-set) separates the classes. In the upper figure the classes are perfectly separable favoring a very steep slope that is solely damped by the regularization term (see Section 2.5.3.3). The lower figure demonstrates the obviously lower slope in case of non-smooth distribution of the targets. The blue line in the upper figure depicts the adapted slope after shifting the decision boundary away from the locally optimal position. The distinct difference of the slope motivates the additional effort of the adaption step (step 3). Note that we used fixed targets for visualization issues. Actually, the information about the targets is given by the probabilities for both possible assignment represented by  $P(l_{ij} = adjacent \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)})$  and  $P(l_{ij} \neq adjacent \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)})$  rather than by fixed values (see Section 2.5.4). However, the argumentation does not change.

boundary

$$\bar{g}_{i,t}(w_i, y) = \frac{1}{1 + \exp\left(-w_i \left(y - Bf\left(\beta_i, B\hat{\theta}_t^{(\nu+1)}\right)\right)\right)}, \quad (3.116)$$

with  $w_i$  representing the only free parameter. Note that this corresponds to the definition of  $g_i(B\theta, y)$  in (3.33).

With the shortcut  $\bar{g}_{ij,t} := \bar{g}_{i,t}(w_i, y_{ij})$  the adapted cost function reads as

$$\bar{C}_t(w_i) = \sum_{j=1}^J -P\left(l_{ij} \neq adjacent \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}\right) \ln(1 - \bar{g}_{ij,t}) - P\left(l_{ij} = adjacent \mid \mathbf{h}_t, \hat{\theta}_t^{(\nu)}\right) \ln \bar{g}_{ij,t}. \quad (3.117)$$

Estimation of the slope parameters as minimum argument of the cost function

$$\hat{w}_{i,t}^{(\nu+1)} = \underset{w_i}{\operatorname{argmin}} \bar{C}_t(w_i) \quad (3.118)$$

results in an iterative procedure which is quite similar to the one presented in Section 2.5.3.1 and derived analogously. The respective Jacobian and Hessian for the iterative update

scheme (2.67) simplify to a first and second derivative of  $\bar{C}_t$  with respect to the slope parameter and read as

$$J_{\bar{C}_t}(w_i) = \sum_{j=1}^J \left( \bar{g}_{ij,t} - P \left( l_{ij} = adjacent \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)} \right) \right) y_{ij} \quad (3.119)$$

and

$$H_{\bar{C}_t}(w_i) = \sum_{j=1}^J \bar{g}_{ij,t} (1 - \bar{g}_{ij,t}) y_{ij}^2. \quad (3.120)$$

Finally, including the regularization procedure presented in Section 2.5.3.3, the Newton update step follows as

$$\hat{w}_{i,t}^{(v+1)} = \hat{w}_{i,t}^{(v)} - \frac{J_{\bar{C}_t}(\hat{w}_{i,t}^{(v)}) + \lambda \hat{w}_{i,t}^{(v)}}{H_{\bar{C}_t}(\hat{w}_{i,t}^{(v)}) + \lambda}, \quad (3.121)$$

with the iteration counter  $v$  and a small regularization parameter  $\lambda = 10^{-6}$ .

**One-Step Approach for the Estimation of the Boundary Spline:** As an alternative to the three-step approach, we could combine the regression and spline estimation steps in a single adjustment based on the reformulated logistic model (3.116) treating  ${}^B\boldsymbol{\theta}$  as free parameters as well. This means, instead of using an intermediate representation given by spline samples as in the three-step approach, we can directly estimate the unknown parameters, i.e. the control points of the boundary spline  ${}^B\hat{\boldsymbol{\theta}}_t^{(\nu+1)}$  and the slope parameters  $\hat{\mathbf{w}}_t^{(\nu+1)}$ , from the logistic model

$$\bar{g}(w_i, {}^B\boldsymbol{\theta}, y) = \frac{1}{1 + \exp(-w_i (y - {}^B f_y(\beta_i, {}^B\boldsymbol{\theta})))}. \quad (3.122)$$

and the information  $P(l_{ij} = adjacent \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)})$  about the targets

$$\mathbf{t}_{ij} = \begin{cases} 1, & \text{if } l_{ij} = adjacent \\ 0, & \text{if } l_{ij} \neq adjacent \end{cases}. \quad (3.123)$$

Considering the interpretation of the logistic regression task as iterative re-weighted least squares problem defined by the functional model (2.80), we achieve  $J$  observations per longitudinal grid column  $i = 1, \dots, I$  of the form

$$P(l_{ij} = adjacent \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) + v_{\mathbf{t}_{ij}} = \frac{1}{1 + \exp(-w_i (y_{ij} - {}^B f_y(\beta_i, {}^B\boldsymbol{\theta})))} \quad (3.124)$$

yielding information about the slope parameters and the position of the boundary spline in longitudinal direction. The stochastic model  $v_{\mathbf{t}_{ij}} \sim \mathcal{N}\left(0, \bar{\sigma}_{\mathbf{t}_{ij}}^{(v)^2}\right)$  is governed by the uncertainties (2.88), which need to be recomputed for each iteration  $v$  of the non-linear estimation procedure.

For the position of the boundary spline in lateral direction, let us consider the parameterization of the boundary spline along the horizontal axis of the image ( $u$ -axis) presented in

Section 3.3.3.1. By definition, the image projection of the local evaluation of the estimated boundary spline at the spline parameter values  $\beta_i$  needs to be incident with the corresponding image column  $u_i$ . Exploiting the relation (3.7) of the horizontal coordinates in the elevation map system to the image row coordinate  $u_i$  and the principle distance  $c$ , we obtain  $I$  observations of the form

$$u_i + v_{u_i} = c \frac{Bf_x(\beta_i, B\theta)}{Bf_y(\beta_i, B\theta)} \quad (3.125)$$

for the lateral direction. The stochastic model  $v_{u_i} \sim \mathcal{N}(0, \sigma_u^2)$  describes the lateral discretization uncertainty of the DEM grid with respect to its image projection. Considering the constant cell width  $d_u$ , the discretization uncertainty is given by  $\sigma_u = \frac{d_u}{\sqrt{12}}$ .

The complete functional model follows from adding the temporal and spatial prior terms (3.54) and (3.39) as well as regularization terms for the slope parameters  $w_i$  and reads as

$$\begin{array}{c} I \\ \left\{ \right. \\ IJ \\ \left\{ \right. \\ 2I \\ \left\{ \right. \\ A \\ \left\{ \right. \\ I \\ \left\{ \right. \end{array} \begin{bmatrix} \vdots \\ u_i \\ \vdots \\ P(l_{ij} = adjacent \mid \mathbf{h}_t, \hat{\boldsymbol{\theta}}_t^{(\nu)}) \\ \vdots \\ \mathbf{x}_{i,t}^- \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} + B\mathbf{v} = \begin{bmatrix} \vdots \\ c \frac{Bf_x(\beta_i, B\theta)}{Bf_y(\beta_i, B\theta)} \\ \vdots \\ \frac{1}{1 + \exp(-w_i(y_{ij} - Bf_y(\beta_i, B\theta)))} \\ \vdots \\ B\mathbf{f}(\beta_i, B\theta) \\ \vdots \\ \kappa_{\beta_a}(B\theta) \\ \vdots \\ w_i \\ \vdots \end{bmatrix} . \quad (3.126)$$



The respective stochastic model is given by

$$B\mathbf{v} \sim \mathcal{N} \left( \mathbf{0}, \begin{bmatrix} \underbrace{\frac{d_u^2}{12} I_I}_{I \times I} & & & & \\ & \ddots & & & \\ & & \underbrace{\bar{\sigma}_{t_j}^{(v)2}}_{IJ \times IJ} & & \\ & & & \underbrace{\left( \Sigma_{x_t^- x_t^-} \right)^{(\nu+1)}}_{2I \times 2I} & \\ & & & & \underbrace{\frac{1}{B_\lambda} I_A}_{A \times A} \\ & & & & & \underbrace{\frac{1}{\lambda} I_I}_{I \times I} \end{bmatrix} \right). \quad (3.127)$$

The parameters of the boundary spline  $B\hat{\boldsymbol{\theta}}_t^{(\nu+1)}$  and the slope parameters  $\hat{\mathbf{w}}_t^{(\nu+1)}$  are estimated by the iterative procedure based on the Gauss-Newton algorithm presented in Section 2.3.3. Note that the  $\bar{\sigma}_{t_j}^{(v)2}$  are recomputed in every iteration  $v$ .

Since we will not consider this estimation model in the remainder of this thesis, we refrain from providing additional information about the linearization of the non-linear model.

### 3.5.3 Evaluation of the Estimated Model

As a final step in each iteration (processing step (VI)), we evaluate the estimated model for self-diagnostic purpose pursuing two goals. First, we evaluate the local validity of the estimated boundary model at the sample positions  $\beta_i$  in order to determine the appropriate model for temporal filtering and to provide a local confidence statement. Second, termination criteria are evaluated examining whether the iterative process has converged or the estimated model is degenerated and, thus, to decide whether to accept or decline the overall model.

#### 3.5.3.1 Local Evaluation of the Boundary Spline

To obtain a statement on the local model validity, we consider the local configuration of the three sample points provided by the estimated spline  $B\mathbf{f}(\beta_i, B\hat{\boldsymbol{\theta}}_t^{(\nu+1)})$ , the predicted spline samples  $\mathbf{x}_{i,t}^-$  and current decision boundary  $(\mathbf{x}_{i,t}^+)^{(\nu+1)}$ . More precisely, we compare the distances

$$d^+ = \left| B\mathbf{f}(\beta_i, B\hat{\boldsymbol{\theta}}_t^{(\nu+1)}) - (\mathbf{x}_{i,t}^+)^{(\nu+1)} \right|_2 \quad (3.128)$$

and

$$d^- = \left| B\mathbf{f}(\beta_i, B\hat{\boldsymbol{\theta}}_t^{(\nu+1)}) - \mathbf{x}_{i,t}^- \right|_2 \quad (3.129)$$

to the three empirically defined thresholds  $\tau_1^+ < \tau_2^+$  and  $\tau^-$  and distinguish three cases:



FIGURE 3.28: Two examples illustrating the three cases defined for local evaluation of the estimated boundary spline. Spline regions representing static obstacles are painted in orange color, those representing non stationary obstacles or segments which slightly violate the model assumptions are drawn in yellow color. Invalid parts are drawn in green color. Figure (a) shows an example for a moving obstacle. Note that the right side of the car is classified as stationary since the car moves forward and, thus, its right side boundary remains constant. Figure (b) shows an example of a gap between two obstacle which is bridged by the boundary spline implying a local violation of the model assumptions.

- (a) The distance between all three sample points is small, i.e.  $d^+ < \tau_1^+$  and  $d^- < \tau^-$ . We assume the local boundary is valid and describes a stationary object.
- (b) The model is slightly violated or the boundary is not static, i.e.  $\tau_1^+ < d^+ < \tau_2^+$  or  $(d^+ < \tau_1^+) \wedge (d^- < \tau^-)$ . This case represents moving obstacles or situations with small measurement errors or insufficient flexibility of the model.
- (c) The model is significantly violated, i.e.  $d^+ > \tau_2^+$ . This is the case for severe violations of the model assumptions or if spline regions have not yet propagated to the actual obstacle. Further, this case includes spline samples beyond the longitudinal grid boundaries  ${}^B f_y \left( \beta_i, {}^B \hat{\theta}_t^{(\nu+1)} \right) \notin [y_{\min}, y_{\max}]$ .

Figure 3.28 shows two examples illustrating the presented cases. An example for a moving obstacle is given in 3.28(a). Figure 3.28(b) shows a bridged gap between two obstacles. In our experiments, we use the thresholds  $\tau_1^+ = .1$  m,  $\tau_2^+ = .4$  m and  $\tau^- = .1$  m. In the first iteration of each time step, the case decision is adopted from the respective nearest sample point of the previous time step.

The case decision affects the local assignment of the appropriate model for temporal filtering via the additive noise term  ${}^B \mathbf{r}_t$  representing the local deviation from the model, as defined in (3.66). We assume the local deviation for samples classified to case (a) to be small and, thus, assign a small value of  ${}^B \sigma_{r_{i,t}}^{(\nu+1)} = 0.02$  m to the respective samples. For those classified to the remaining cases (b) and (c), we assign a larger value of  ${}^B \sigma_{r_{i,t}}^{(\nu+1)} = 0.4$  m.

Although the presented evaluation method turned out to be suitable for most situations, it purely relies on a heuristic making the development of a more sophisticated approach a worthwhile goal for future work.

### 3.5.3.2 Termination Criteria

The iterative process of the EM-Algorithm is stopped if one of the following three termination criteria is fulfilled:

- A maximum number of iterations  $\nu_{\max}$  is reached.
- No label was changed in the last classification step.
- The estimated model is degenerated.

In the last case, the system is reinitialized and the model is computed from scratch. We call the model degenerated in case the percentage of *street* labels falls beyond a certain threshold  $\tau_{\text{street}}$  or the percentage of *outlier* labels exceeds a certain threshold  $\tau_{\text{outlier}}$ . In our experiments, we choose  $\tau_{\text{street}} = 20\%$  and  $\tau_{\text{outlier}} = 10\%$  of the total number of cells as well as a maximum number of  $\nu_{\text{max}} = 3$  iterations.



## Chapter 4

# Experiments

The evaluation of the presented approach is divided in two sections. Section 4.2 describes a set of experiments using a simulated testing environment to evaluate the performance and limits of the approach with respect to measurement noise and outliers under controlled and ideal conditions. Synthetically generated disparity maps are systematically disturbed in order to analyze the effect on the deviation of estimated and true free-space boundary.

Section 4.3 evaluates the performance and limits of the approach and selected key components under real-world conditions. In addition to qualitative studies, the evaluation comprises experiments on a benchmark dataset which provides manually annotated reference results. These experiments yield quantitative statements about the approach’s applicability in real-world traffic scenarios.

In Section 3, a set of model parameters defining the configuration and interaction of the approach’s components was introduced. The parameters control the structure of the DEM, the flexibility of the street surface and boundary model, as well as their estimation procedures, and the self evaluation and termination criteria.

Table 4.1 summarizes our proposal for the assignment of all relevant parameters which represents the default parameter set for all following experiments unless stated otherwise. Several parameter values were already motivated in the sections they were introduced, while the remainder was empirically determined. For convenience, the table provides references to the respective section or equation the parameters were initially introduced.

TABLE 4.1: Default parameter set: The table summarizes the proposed configuration of the approach’s model parameters and thresholds. Beside the parameter value, the section or equation which defines or introduces the respective parameter is given by a reference. Parameters which were not labeled by a symbol so far are marked with ‘-’ in the symbol column.

Description	Symbol	Value	Defined in
<i>DEM computation:</i>			
# cells in longitudinal direction	$I$	45	Section 3.2.2.1
# cells in lateral direction	$J$	47	Section 3.2.2.1
# cells in vertical direction	$\mathcal{V}$	85	Section 3.2.2.1
lateral cell spacing in image columns	$d_u$	20 pel	Section 3.2.2.1
vertical cell spacing in image rows	$d_v$	3 pel	Section 3.2.2.1
longitudinal grid limit far	$y_{\min}$	16 m	Section 3.2.2.1
longitudinal grid limit near	$y_{\max}$	5.5 m	Section 3.2.2.1
uncertainty of disparity measurements	$\sigma_d$	0.5 pel	Section 3.2.2.3
<i>Street surface estimation:</i>			
# lateral sections	-	4	Section 3.3.2.1
# longitudinal sections	-	2	Section 3.3.2.1
# control points in ${}^S\theta$	$N$	24	Section 3.3.2.1
spatial smoothness weight	${}^S\lambda$	20	Section 3.3.4.1
system noise	${}^S\sigma_r$	0.5 cm	(3.48)
<i>Boundary spline estimation:</i>			
# spline sections	$B_n$	18	Section 3.3.3.1
# control points in ${}^B\theta$	$M$	21	Section 3.3.3
spatial smoothness weight	${}^B\lambda$	0.5	Section 3.3.4.2
system noise	${}^B\sigma_{r_{i,t}}^{(\nu+1)}$	0.02 m/0.4 m	(3.66)
<i>Self evaluation:</i>			
model violation threshold	$\tau_1^+$	0.1 m	Section 3.5.3.1
severe model violation threshold	$\tau_2^+$	0.4 m	Section 3.5.3.1
prediction threshold	$\tau^-$	0.1 m	Section 3.5.3.1
<i>Termination criteria:</i>			
max. number of EM iterations	$\nu_{\max}$	3	Section 3.5.3.2
min. percentage of street labels	$\tau_{\text{street}}$	20 %	Section 3.5.3.2
max. percentage of outlier labels	$\tau_{\text{outlier}}$	10 %	Section 3.5.3.2

## 4.1 Evaluation Criteria

As defined Section 3.1, the overall objective of the presented approach is to determine and model the drivable free-space in front of the vehicle. Beside visual examination, the performance of the approach with respect to this objective is evaluated on both synthetic and real world data by comparing the estimated result with ground truth knowledge. While in the synthetic experiments the ground truth is known from the simulation model, it is obtained from manual annotation for the real world experiments. The comparison of estimated result and ground truth data is done by means of two measures. The first of them rates the ac-

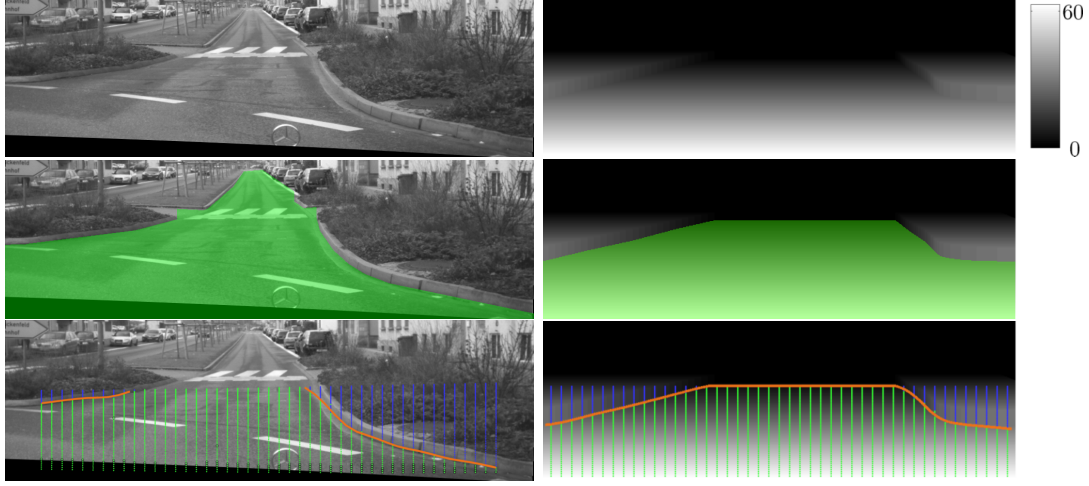


FIGURE 4.1: Examples for the image projection of the free-space obtained from a real world scenario (left column) and a synthetic scenario (right column). The free-space defined by the ground truth is depicted in the middle row and estimation result in the bottom row. The upper row depicts the original image for the real world scenario and the artificial disparity image for the synthetic scenario.

curacy of the estimated free-space region with respect to its image projection. The second measure evaluates the spatial accuracy of the estimated free-space boundary with respect to the horizontal space. The following sections describe both measures in detail.

#### 4.1.1 Accuracy Assessment of the Image Projection of the Free-Space:

The first approach rates the similarity of the annotation and the image projection of the estimated free-space, where the latter means the set of all pixels encircled by the image projection of the estimated boundary spline. To compute the image projection of a boundary sample, we assign the local elevation of the true street surface (in case of synthetic scenarios) or estimated street surface (in case of real-world scenarios) as height coordinate for the respective sample position. Figure 4.1 depicts examples for estimated results and the respective ground truth annotations for both a real world and a synthetic scenario.

We present the comparison results typically by confusion matrices holding the rates of pixels correctly classified to *free-space* or *non-free-space* as well as the rates of incorrect assignments. We use an alternative representation of the confusion matrices by aligning the entries in a single table row, as described in Table 4.2, to neatly arrange multiple results in a single table allowing for more convenient comparison.

TABLE 4.2: Employed pattern for the confusion matrices comparing pixel assignments of the estimated result (Est.) and the annotated ground truth (GT) information. Referring to pixels assigned to *free-space* in the annotated image as *free-space* pixels and to those assigned to *non-free-space* as *non-free-space* pixels, the entries hold the percentage of *non-free-space*-pixels correctly assigned to *non-free-space* by the estimated result (*a*) and those incorrectly assigned to *free-space* (*b*). Analogously, they hold the percentage of *free-space*-pixels assigned to *non-free-space* (*c*) and *free-space* (*d*).

	GT	<i>non-free-space</i>		<i>free-space</i>	
		<i>non-free-space</i>	<i>free-space</i>	<i>non-free-space</i>	<i>free-space</i>
Description of experiment	Est.	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>

For interpretation, a high value of *b* induces that the free-space boundary estimated in

the respective experiment tends to overshoot the obstacles. A high value of  $c$  induces that the free-space boundary was estimated to close to the camera, i.e. the algorithm erroneously detected some non-existent obstacles.

#### 4.1.2 Evaluation of the Spatial Accuracy of the Free-Space Boundary:

The second evaluation approach assesses the spatial accuracy of the estimated free-space boundary. The measure is based on the shortest distance of the estimated free-space boundary to the ground truth boundary. For each frame we evaluate the measure for a set of sample positions. The samples are drawn equidistant from the estimated spline with respect to the spline parameter such that we obtain one horizontal sample point per image column. We refer to this samples points as *ground truth samples*.

Remember that the data basis provided for the free-space estimation is limited to the horizontal extent of the DEM grid which causes the estimation result to be cropped by the longitudinal grid boundaries  $y_{\min}$  and  $y_{\max}$ , as described in Section 3.5.2.2. In case the true free-space boundary lies beyond the DEM grid borders, we consider the estimated boundary to be correct if it lies on (or beyond) the grid borders. Thus, we crop the polygon formed by the ground truth reference samples by the grid boundaries  $y_{\min}$  and  $y_{\max}$  as well. As accuracy measure for a certain sample position on the estimated spline, we consider the shortest Euclidean distance to the polygon formed by the cropped ground truth samples. Figure 4.3 gives a visual explanation of the ground truth computation.

Note, an alternative would be to measure the error along the projection ray. However, this would cause disproportionate error measures especially at lateral obstacle borders, as depicted in Figure 4.2.

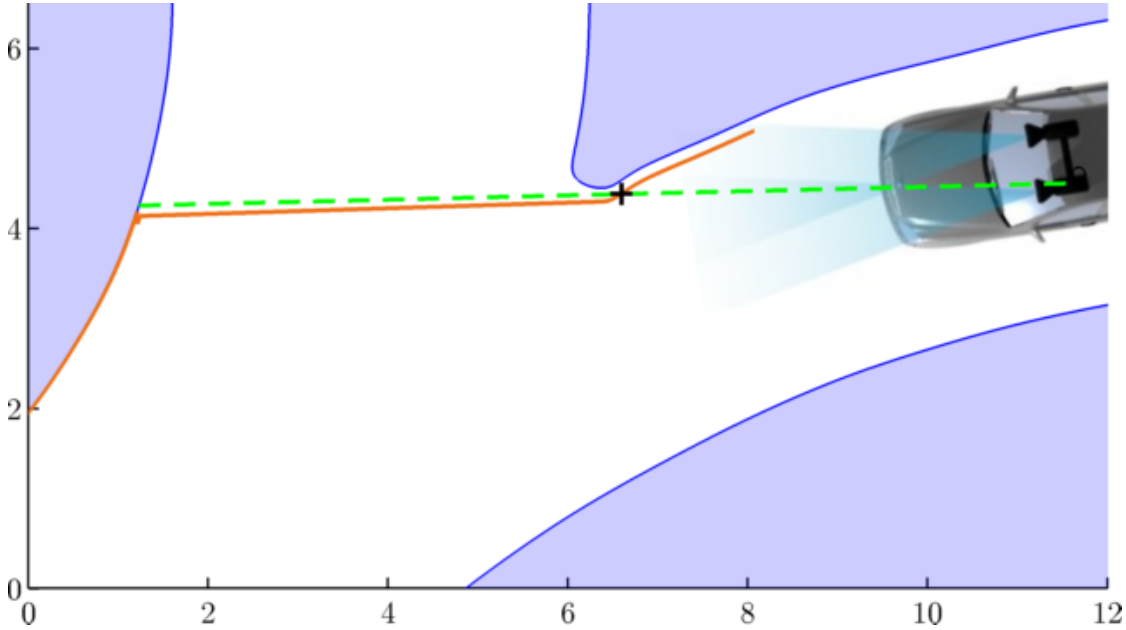


FIGURE 4.2: Example motivating the choice of the distance measure for evaluation. The blue lines mark the true road borders which separate the free-space (white) from non-free-space (bright blue). The estimated free-space boundary is given by the orange line which is approximately 15 cm away from the ground truth considering the shortest distance. Assume we aim to evaluate the error of the estimated spline sample marked by the black cross. In case we would measure the distance to the ground truth along the projection ray (green dashed line), we would achieve a very large error which is visually out of proportion to the actual accuracy of the reconstruction. The annotation of the coordinate axes is given in [m].

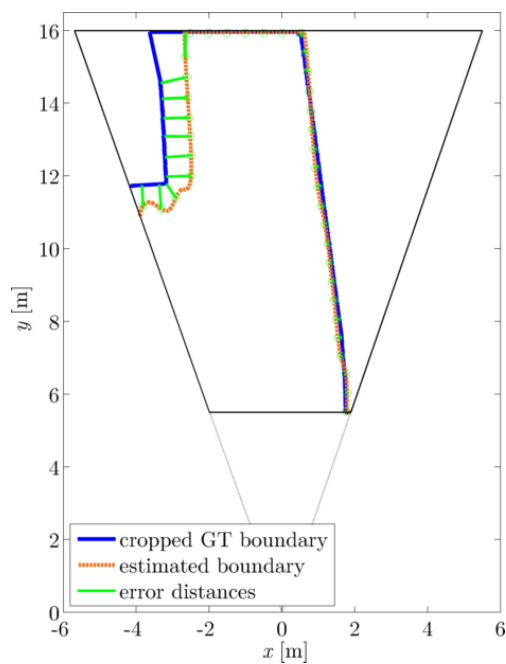




(a) Annotated free-space and image projection of the free-space GT boundary



(b) Image projection of the cropped free-space GT boundary and estimated free-space boundary



(c) Comparison of GT and estimated boundary

FIGURE 4.3: Example for the ground truth (GT) data used to evaluate the accuracy of the estimated free-space boundary. Figure (a) shows the annotated ground truth as green carpet and the resultant image projection of the GT boundary as blue line. The GT boundary after cropping by the DEM grid borders is indicated as blue dashed line. Figure (b) additionally depicts the image projection of the estimated free-space boundary, which shows considerable errors caused by erroneous stereo measurements in the region of the vehicle's shadow. Figure (c) presents the scene from a bird's-eye view and illustrates the comparison of ground truth and estimated free-space boundary depicting a sample set of the error distances by green lines.

## 4.2 Experiments on Synthetic Data

This section discusses a set of experiments in a simulated testing environment. The simulation allows for evaluation under controlled conditions and enables the comparison to ideal reference results, which we refer to as *ground truth*.

The following aspects are analyzed and discussed:

- (a) Effect of increasing noise in the disparity measurements on the precision of the estimated free-space boundary with respect to the ground truth (outer precision);
- (b) Effect of increasing noise in the disparity measurements on the stability of the estimated free-space boundary (inner precision);
- (c) Effect of increasing outlier percentage in the disparity measurements on the estimated free-space boundary;

The following section describes the simulation setup and introduces the synthetic benchmark dataset. Subsequently, the aspects (a)-(c) are analyzed in the experiments presented in Section 4.2.2-Section 4.2.4. Section 4.2.5 provides a final discussion of the achieved results and insights.

### 4.2.1 Setup of Synthetic Sequences

The synthetic benchmark database consists of a set of simulated disparity map sequences. In each sequence, a virtual camera is moved along a predefined trajectory through a synthetic scenario with a constant speed of 0.5 m per frame. The virtual camera is defined as pinhole camera governed by the parameters presented in Table 4.3. The trajectory is represented by a quadratic B-spline in order to allow for a smooth motion.

TABLE 4.3: Parameters of the virtual camera. The horizontal viewing direction is defined by the local tangent of the trajectory. The lateral and longitudinal axes of the local camera system are parallel to the ground plane causing the rotation angles to equal zero. The remaining parameters are designed to match those of the camera system utilized for the real world experiments, as described in Section 4.3.1.

Parameter	Value
principal distance	1250 pel
# image columns	1024
# image rows	440
height above ground plane	1.2 m
stereo-baseline	0.3 m
yaw angle	0°
pitch angle	0°
roll angle	0°

The scenarios are motivated by concept drawings of design templates for roadways and road boundaries presented in [American Association of State Highway and Transportation Officials, 2001]. The templates model different types of four-way-intersections, T-intersections, roundabouts and highway-ramps including medians and small traffic isles. Figure 4.4 depicts the total dataset of design templates.

Horizontal obstacle boundaries are represented by planar sample points with a sampling distance of 5 cm. For determination of the precise position, e.g. for intersection with optical rays, the boundary is locally refined by cubic interpolation.

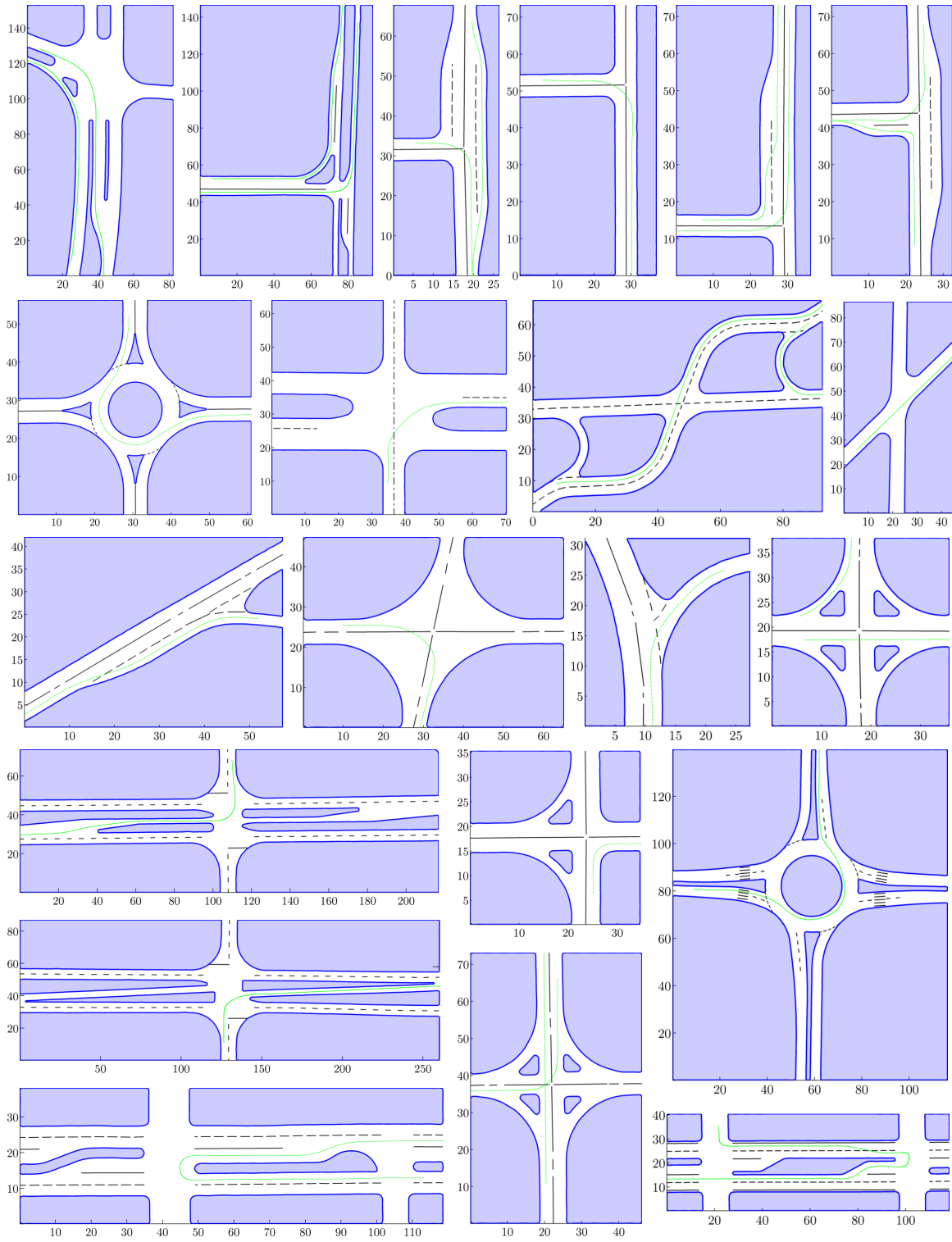


FIGURE 4.4: Design templates of the scenarios included in the synthetic benchmark database. Street regions are plotted in white color, while non-street regions are plotted in bright blue. The street boundaries are drawn by blue lines. The green line represents the trajectory of the virtual camera. Note that there exist multiple trajectories in some scenarios. The annotation of the coordinate axes is given in [m] to demonstrate the scale.

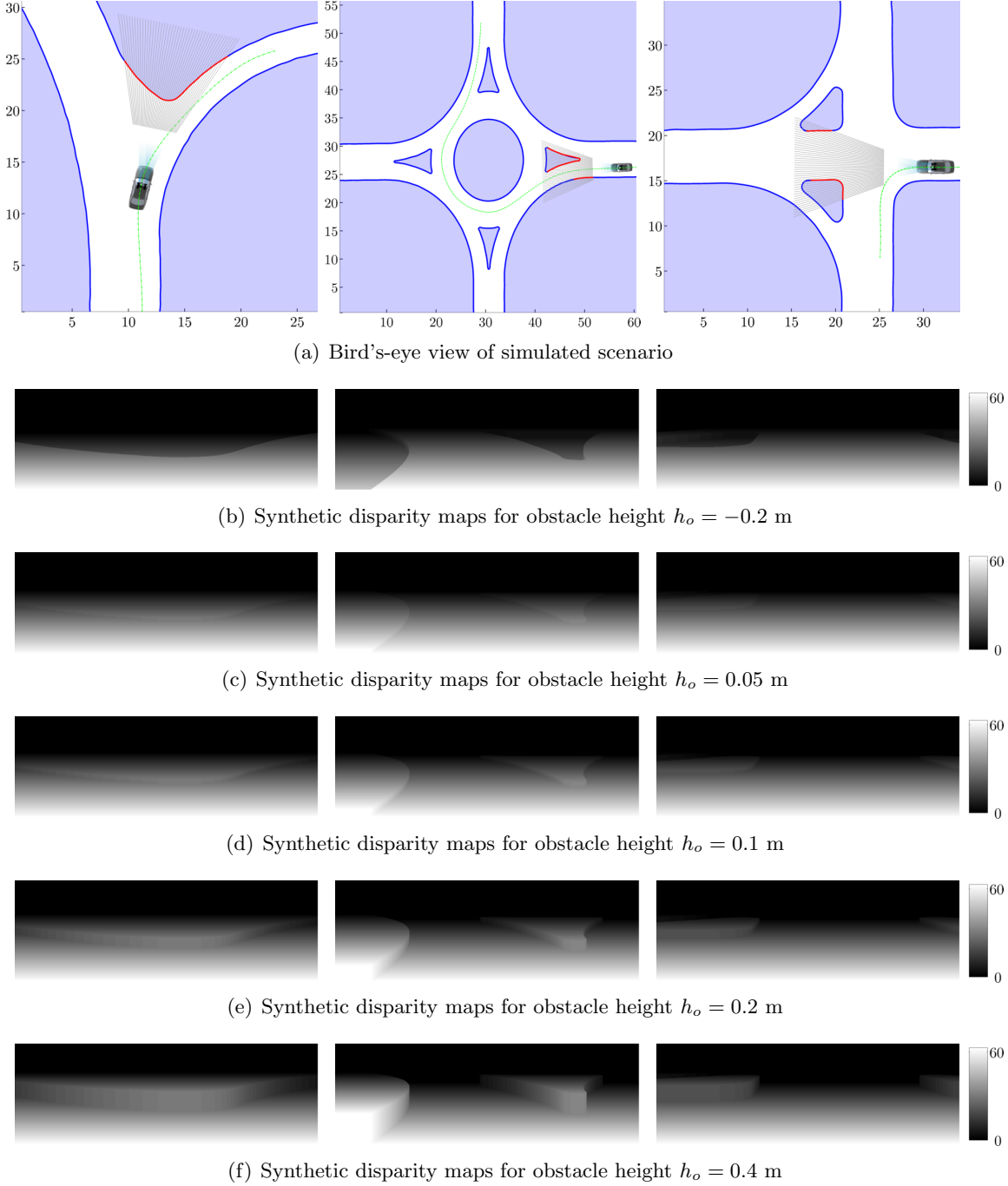


FIGURE 4.5: Examples for scenarios included in the synthetic benchmark database. The first row (a) depicts the respective design template. Street regions are plotted in white color, while non-street regions are plotted in bright blue. The street boundaries are drawn by blue lines. The green line represents the trajectory of the virtual camera whose current position is marked by the car model. The longitudinal grid columns of the DEM, plotted by a set of gray lines, indicate the field of view. Street boundary parts visible from the current camera position and in the range of the DEM are highlighted by red lines. The annotation of the coordinate axes is given in [m] to demonstrate the scale. The remaining rows (b)-(f) illustrate the synthetic disparity maps. The color-encodings defined by the respective colorbars are given in [pel].

Disparity maps are computed for each position of the trajectory via ray tracing. The elevation of street regions is constantly set to 0 m, while the elevation of obstacle regions is set to a constant height  $h_o$ . Through each single pixel of the virtual camera the optical ray is casted into the 3d space using the pinhole camera concept. The distance of the closest intersection point with either the street surface, a horizontal obstacle surface, or a vertical obstacle border defines the respective pixel's disparity value. Examples for synthetic disparity maps are given in Figure 4.5.

The synthetic dataset consists of 29 scenes with a total of 4831 frames.

To assess the performance of the approach with respect to different obstacle heights, the disparity maps are computed for five different values of the constant obstacle elevation  $h_o \in \{0.05 \text{ m}, 0.1 \text{ m}, 0.2 \text{ m}, 0.4 \text{ m}, -0.2 \text{ m}\}$  scaling the total number of sequences and frames to 145 and 24155, respectively. The values are chosen to represent obstacles distinctive in height (0.4 m) as well as flat obstacles in the vicinity of the average height of curbs (0.2 m) and below (0.05 m and 0.1 m). Further, a negative obstacle elevation (-0.2 m) is considered to evaluate the ability of the approach to deal with street boundaries embodied by negative elevation discontinuities, such as roadside ditches.

#### 4.2.2 Evaluation of the outer precision at increasing noise

This experiment assesses the effect of increasing measurement noise on the outer precision of the estimated free-space boundary. By 'outer precision', we mean the accuracy of the estimated result with respect to the ground truth, while measurement noise denotes the uncertainty of the disparity values obtained from stereo matching.

We expect the precision of the result to diminish with increasing noise until the algorithm reaches its limit. Regarding the different obstacle heights we expect 0.05 m to be too low to allow for a proper reconstruction result. Especially in larger distances, the elevation difference to the street surface might be too small to enable the algorithm to separate non-street measurements from street measurement causing the boundary to overshoot obstacles. We expect 0.1 m, however, to be large enough to allow for a sufficient reconstruction in the considered range interval of 6 to 16 m to the camera. For the larger values of 0.2 m and 0.4 m, we expect similar results which might be slightly more accurate than for 0.1 m but show a similar behavior with respect to the increasing noise.

**Experimental Setup:** In order to determine the effect of increasing measurement noise the algorithm is applied on the synthetic dataset in five runs. In each run the synthetic disparity measurements are disturbed by Gaussian noise with different standard deviations. The standard deviations for the single runs are given by  $\sigma_d \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$  pel. The parameter settings of the algorithm correspond to the values presented in Table 4.1.

Note, in order to prevent the evaluation statistics to be biased by initialization errors, the first five images of every sequence are discarded from evaluation.

#### Discussion of the Assessment of the Image Projection of the Detected Free-Space:

The precision of the estimated result is evaluated with respect to the criteria presented in Section 4.1. The accuracy assessment of the image projection of the detected free-space is presented in Table 4.4. The table is subdivided in five blocks, one block for each obstacle height. Each block consists of five rows which hold the entries of the confusion matrices obtained for the run with the respective indicated measurement noise.

For an obstacle height of 0.05 m, the algorithm performs surprisingly well in case of zero noise. However, the algorithm reaches its limits for higher noise values and shows a strong

TABLE 4.4: Accuracy assessment of the image projection of the detected free-space. Each row holds the entries of a confusion matrix describing the classification result obtained on the synthetic dataset. The table is separated in five blocks, each of which holds the results obtained for a different value for the obstacle height under the indicated noise levels. A detailed description of the pattern describing the meaning of the row entries is given in Table 4.2. The total ratio of *free-space* pixels in the synthetic database is 64.0% yielding 36.0% for the total ratio of *non-free-space* pixels.

		GT	<i>non-free-space</i>		<i>free-space</i>	
		Est.	<i>non-free-space</i>	<i>free-space</i>	<i>non-free-space</i>	<i>free-space</i>
obstacle height	disparity noise $\sigma_d$					
0.05 m	0.00 pel		97.7	2.3	0.3	99.7
	0.25 pel		67.1	32.9	0.3	99.7
	0.50 pel		33.4	66.6	0.2	99.8
	0.75 pel		26.6	73.4	0.2	99.8
	1.00 pel		53.3	47.7	5.4	94.6
0.10 m	0.00 pel		97.1	2.9	0.6	99.4
	0.25 pel		97.2	2.8	0.6	99.4
	0.50 pel		97.5	2.5	0.8	99.2
	0.75 pel		99.5	0.5	1.1	98.9
	1.00 pel		99.7	0.3	4.7	95.4
0.20 m	0.00 pel		98.4	1.6	0.6	99.4
	0.25 pel		98.5	1.5	0.7	99.3
	0.50 pel		98.9	1.1	0.9	99.1
	0.75 pel		99.5	0.5	2.4	98.6
	1.00 pel		99.7	0.3	6.2	93.8
0.40 m	0.00 pel		99.4	0.6	0.7	99.3
	0.25 pel		99.4	0.6	0.9	99.1
	0.50 pel		99.6	0.4	1.1	98.9
	0.75 pel		99.8	0.2	2.7	97.3
	1.00 pel		99.9	0.1	6.5	93.5
-0.20 m	0.00 pel		92.3	7.7	0.1	99.9
	0.25 pel		92.2	7.8	0.1	99.9
	0.50 pel		92.5	7.5	0.1	99.9
	0.75 pel		93.5	6.5	0.3	99.7
	1.00 pel		97.7	2.3	5.4	94.6

tendency to overshoot obstacles which is reflected in the high amount of misclassified *non-free-space* pixels. The tendency is confirmed by visual examination of the results.

Comparison of the second, third and fourth block of Table 4.4 reveals that the results obtained for the obstacle heights 0.1 m, 0.2 m and 0.4 m show a related behavior. The respective results for the noise levels up to  $\sigma_d = 0.5$  pel appear to be very similar within each block showing low misclassification ratios. For  $\sigma_d = 0.75$  pel all three blocks present increased ratios for misclassified free-space pixels. This effect occurs significantly stronger in the results obtained for a noise value of  $\sigma_d = 1$  pel. Comparison among the different obstacle heights reveals that the algorithm performs slightly better for larger obstacles regarding noise levels up to  $\sigma_d = 0.5$  pel. Surprisingly, for the noise level  $\sigma_d = 0.75$  pel and more extreme for  $\sigma_d = 1$  pel, the ratio of misclassified *free-space* pixels rises for increasing obstacle height.

We assume that this behavior is caused by the scattering effect already mentioned in Section 3.2.2.2 (see also Figure 3.12). The effect is depicted in Figure 4.6 for a longitudinal slice corresponding to a single image column. Consideration of the region in front of the obstacle reveals that the scattering of the points originating from the vertical surface of the obstacle border induces a large elevation error when vertically comparing the point position to the ground truth. Thus, the obstacle appears closer than it really is causing the algorithm to estimate the free-space boundary to close to the camera, i.e. to classify *free-space* regions as *non-free-space-regions*.

Compared to the larger obstacle heights, the vertical surface for the height of 0.1 m depicted in Figure 4.6(a) is small and causes fewer erroneous measurements which are significantly closer to the street surface. This enables the probabilistic model of the DEM computation to deal with this errors more easily than in case of larger obstacle heights and, thus, to cause fewer misclassified *free-space* pixels.

Naturally, the effect intensifies with increasing disparity noise. Visual examination reveals that especially for  $\sigma_d = 1$  pel even the scattering of the points originating from the street surface mimics faked obstacles in distant regions.

Finally, the results for the negative obstacle height  $-0.2$  m summarized in the last block of Table 4.4 reveal an obvious tendency of overshooting for noise values  $\sigma_d \leq 0.75$  pel. A probable explanation of this behavior relates to the fact that the region behind the true street boundary is occluded and, thus, the elevation in this region is unobserved. Consider the utilized method for the estimation of the DEM presented in Section 3.2.2.2 and the example depicted in Figure 4.7. The estimated elevations for the grid cells in the region directly behind the true street boundary lie in the vicinity of the projection ray to the street boundary (green dotted line). Thus, the transition from the street elevation level to the non-street elevation level in the DEM heights is smooth. The small alteration of the elevation of the cells close behind the true boundary makes them likely to be classified as *street* and, thus, encourages the estimated street boundary to overshoot.

For  $\sigma_d = 1$  pel the strong noise mimics obstacles in the remote areas of the DEM grid and causes too close estimates as described for the remaining obstacle height values.

**Discussion of the Spatial Accuracy of the Free-Space Boundary:** Let us now consider the spatial accuracy of the free-space boundary. Figure 4.8 presents the statistics of the computed error distances plotted against the distance to the camera. More precisely, the complete set of errors is arranged in successive 1 m wide bins along the longitudinal axis. For each bin, the respective mean error is plotted. For comparison issues the local horizontal cell spacing of the DEM is plotted as blue dashed line. Furthermore, Figure 4.9 presents the cumulative distribution of the error for each distance bin in order to provide an overview of the distribution of the errors.

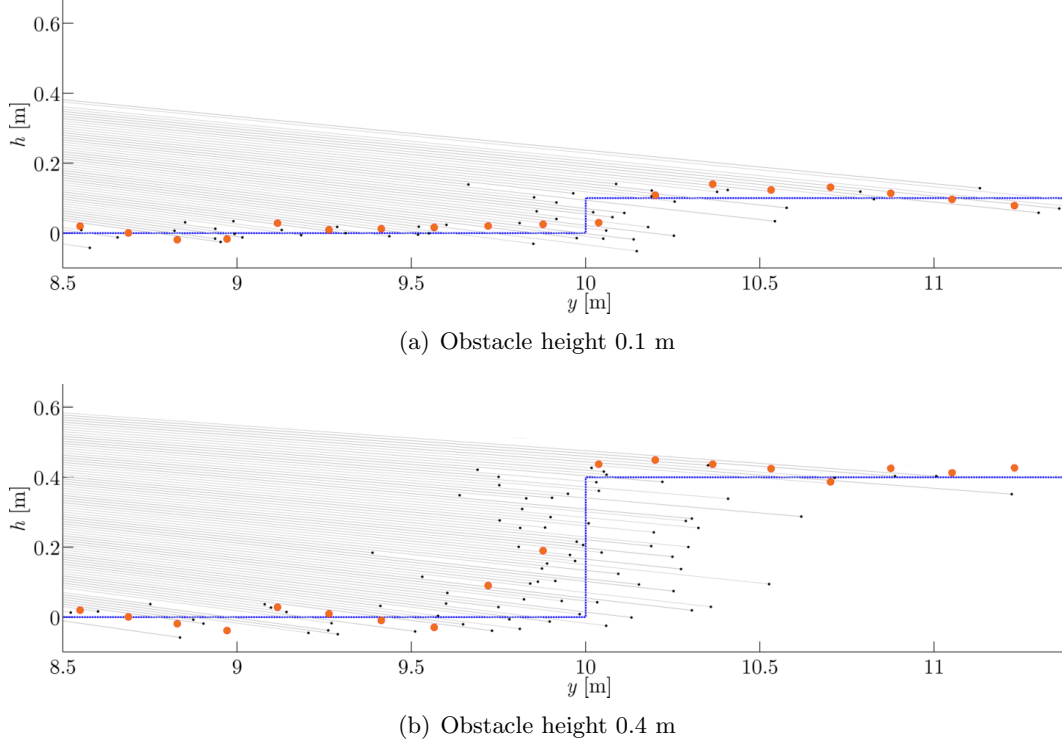


FIGURE 4.6: Side view on a synthetically generated obstacle of 0.1 m height (a) and 0.4 m height (b) demonstrating the scattering effect. The axes represent the longitudinal coordinate of the elevation map system  $y$  and the height coordinate  $h$ . The dotted blue line describes the ground truth of the surface, i.e. viewing from left to right the line describes the street surface at height level zero passing over into the vertical obstacle border and finally into the obstacle's top surface. The black dots mark the points triangulated from a single column of the synthetic disparity image with the thin gray lines indicating the projection rays. The estimated heights of the respective longitudinal column of the elevation map are plotted as orange dots. The disparity image was disturbed by Gaussian noise with standard deviation  $\sigma_d = 0.75$  pel resulting in strong displacements of the ground truth points along their projection rays. Consideration of the region in front of the obstacle around  $y \in [9.5 \text{ m}, 10 \text{ m}]$  reveals that the scattering of the points originating from the vertical surface induce a large elevation error when vertically comparing the point position to the ground truth. While the probabilistic model appears to be able to robustly handle the scattering for the small vertical surface at obstacle height 0.1 m, the extensive scattering at larger obstacle heights appears to be more challenging. In some cases this leads to massed erroneous elevation measurements in front of the obstacle causing the algorithm to estimate the free-space boundary to close to the camera, i.e. causing *free-space* regions to be classified as *non-free-space-regions*.



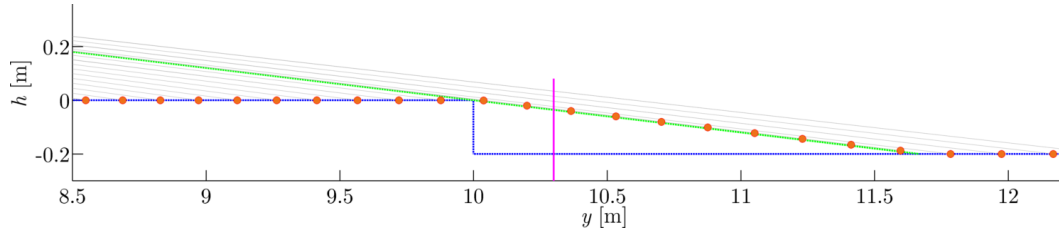


FIGURE 4.7: Side view on a synthetically generated obstacle of negative height ( $-0.2$  m). The axes represent the longitudinal coordinate of the elevation map system  $y$  and the height coordinate  $h$ . The dotted blue line describes the ground truth of the surface. The orange dots mark the estimated heights of the elevation map computed from the synthetic disparity image with the thin gray lines indicating the corresponding projection rays. Note, that the step at  $y = 10$  m occludes the ground truth surface up to approximately  $y = 11.6$  m. Considering the method for the elevation estimation presented in Section 3.2.2.2, the voxels below the projection ray touching the upper edge of the boundary (green dotted line) are dominated by *occluded* votes, while the voxels above the ray are dominated by *pass-through* votes. Thus, the estimated elevations in this area are in line with the projection ray. The vertical magenta line at  $y = 10.3$  m indicates the estimated distance of the street boundary. The estimated boundary distinctly overshoots the true boundary caused by the smoothed transition from the street surface to the surface of the non-street region in the computed elevation map heights. Note, one might wonder why the algorithm detects a boundary anyway since the street surface model is flexible enough to fit the smooth transition. However, the majority of the elevation measurements in the transition region have no influence on the surface estimation since DEM cells without any *hit* votes in the respective voxel column are excluded from the street surface estimation (see Section 3.2.2.2).

The statistics confirm the observed tendencies presented in the previous paragraph. Considering obstacle heights of  $0.05$  m, the results show mean errors of about half the local cell spacing in case no noise is added which significantly rise with increasing disparity noise reflecting the strong tendency of overshooting.

The results for obstacle heights of  $0.1$  m to  $0.4$  m reveal a more robust behavior with mean errors in the range of half the local cell spacing up to noise values of  $\sigma_d = 0.5$  pel. This implies the ability for a reasonable reconstruction of the free-space boundary within the tolerance of the cell spacing. The respective statistics in Figure 4.9 show that more than 97% of the errors are smaller than  $0.2$  m. While the mean errors for  $\sigma_d = 0.75$  pel are nearly twice the size but still in the range of the local cell spacing, the algorithm obviously reaches its limits for  $\sigma_d = 1$  pel with around 30% of the errors exceeding a value of  $0.5$  m. As in the previous paragraph, comparison of the statistics obtained for the different obstacle heights reveals that for the upper noise levels the errors slightly increase for larger obstacle heights. We assume that this reflects the increasing influence of the scattering effect depicted in Figure 4.6.

Finally, for negative obstacle heights the statistics present a very similar behavior for noise values between  $\sigma_d = 0$  pel and  $\sigma_d = 0.75$  pel. The mean errors lie in the range of the local cell spacing and, thus, distinctly exceed the errors obtained for the positive obstacle heights over  $0.05$  m considering noise levels up to  $\sigma_d = 0.5$  pel. This reflects the tendency of overshooting discussed in the previous paragraph (see Figure 4.7). Analogous to the remaining obstacle heights, the errors significantly rise for  $\sigma_d = 1$  pel.

Note, the drop of the mean errors in Figure 4.8 for the last bin may seem counterintuitive. However, keep in mind the way the errors are computed in case the ground truth boundary lies beyond the DEM grid borders. The algorithm has no information about the world beyond the DEM grid borders. Thus, we accept the estimated boundary to be correct if it lies on or beyond the grid borders since in this case it correctly tells us that there is no obstacle in the considered region of interest. In this case, error distances are computed as the shortest distance to the cropped ground truth boundary, as described in Section 4.1.2, causing the errors of estimates close to the grid border to drop.

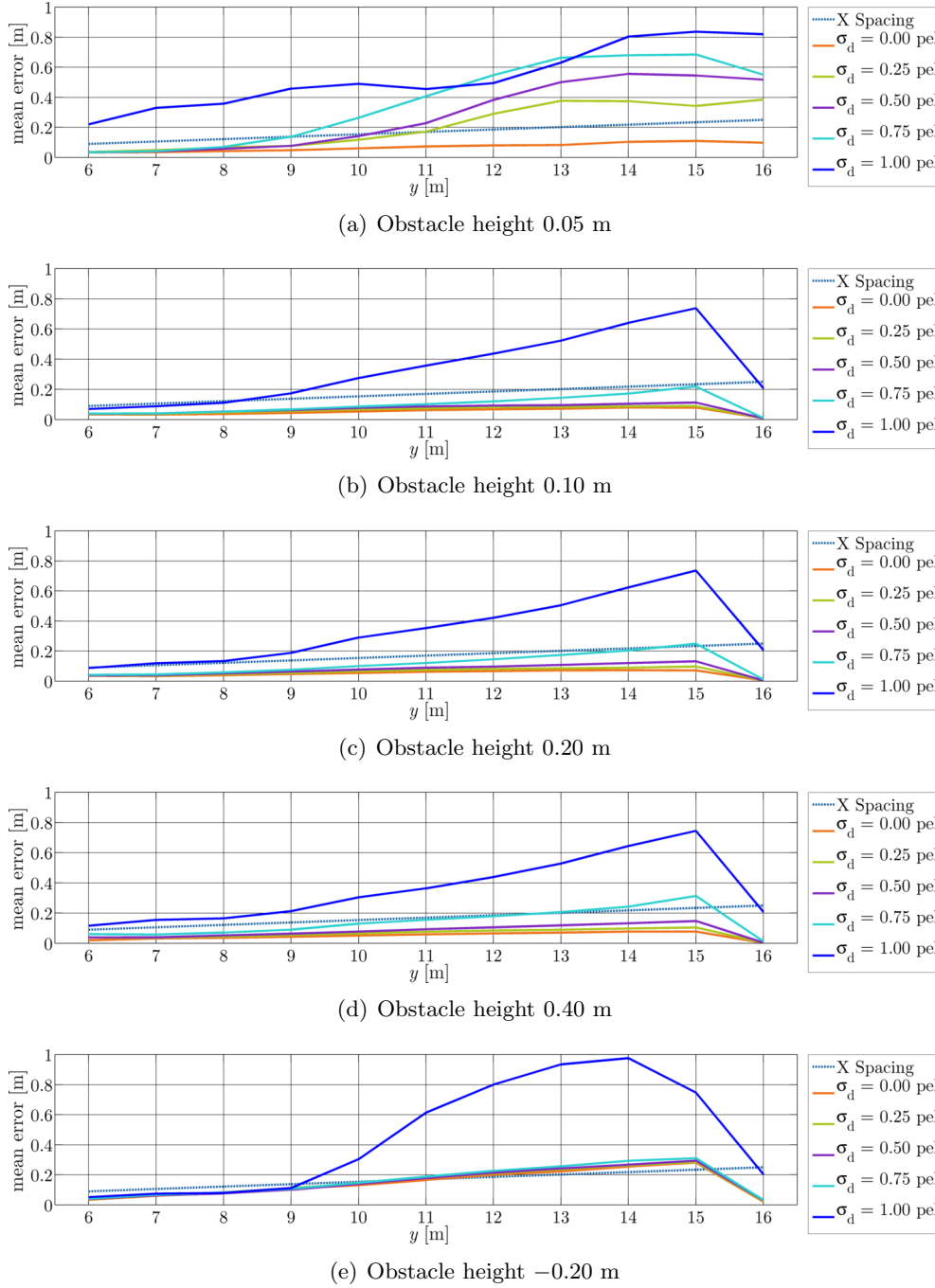


FIGURE 4.8: Mean errors obtained on the synthetic dataset for the evaluation of the outer precision plotted against the distance to the camera. The error measure is given by the distance of the estimated boundaries to the respective ground truth, as described in Section 4.1.2. The mean errors were evaluated in successive 1 m intervals with respect to the camera distance. The blue dashed reference line describes the local cell spacing of the DEM. Each plot collects the results for the indicated obstacle height composed of the mean errors obtained for the considered disparity noise levels  $\sigma_d$ . The results reveal that the algorithm reaches its limits at latest at noise level of  $\sigma_d = 1$  pel for all obstacle height values. An obstacle height of 0.05 m appears to be too low to allow for a proper reconstruction except in case of zero noise. The errors in the lowest figure show that the algorithm is able to handle obstacles of negative height, albeit at an higher error rate in comparison to the positive obstacle heights due to its overshooting tendency (see Figure 4.7). Note that the errors drop in the last bin since the ground truth is cropped to the DEM grid borders, as explained in Section 4.1.2.

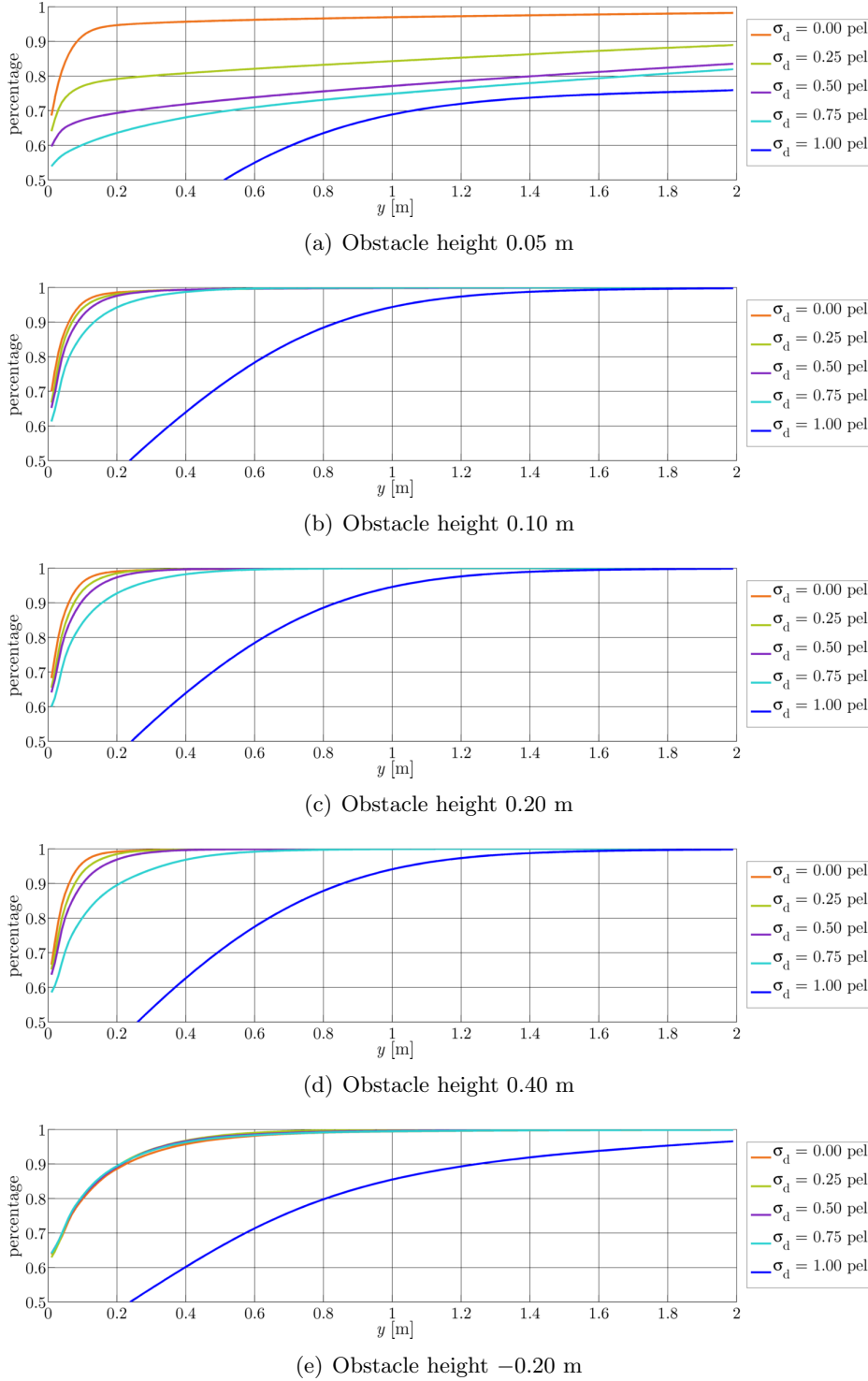


FIGURE 4.9: Cumulative distributions of the errors obtained on the synthetic dataset for the evaluation of the outer precision. The group structure of the single figures is adopted from Figure 4.8. Note that the labeling of the ordinate begins at 0.5 to improve the resolution. It is obvious that the best results are obtained for the obstacle heights between 0.1 m and 0.4 m and for noise levels  $\sigma_d \leq 0.5$  pel, where more than 97% of the errors are smaller than 0.2 m. Comparison of the single results among each other reveals the same relative behavior as observed for the mean error. The high percentage of large errors for  $\sigma_d = 1$  pel as well as for all noise levels  $\sigma_d > 0$  pel at an obstacle height of 0.05 m confirms that the ability of the approach to properly model the free-space boundary in this cases is limited.

### 4.2.3 Evaluation of the inner precision

This experiment evaluates the effect of increasing measurement noise on the inner precision of the estimated free-space boundary. By ‘inner precision’, we understand the variation of the estimated result with respect to repeated runs on the same test bed. In each run, the noise which is used to disturb the synthetic disparity images is independently generated based on the considered noise level (Monte Carlo simulation). We expect the inner precision to behave similar to the outer precision regarding the comparison of the combinations of obstacle height and noise level. This means we expect the inner precision to be low for all runs with obstacle height of 0.05 m or noise level  $\sigma_d = 1$  pel.

**Experimental Setup:** The setup largely corresponds to the setup of the previous experiment since we consider the same set of synthetic sequences and obstacle heights. The considered noise levels are given by  $\sigma_d \in \{0.25, 0.5, 0.75, 1.0\}$  pel. The individual experiments for each combination of obstacle height and noise level are repeated  $K = 20$  times, whereas the artificial disparity noise is recomputed for each repetition. To assess the inner precision of the estimated free-space boundary, we employ a procedure similar to the presented in Section 4.1.2. The only difference is embodied by the fact that we do not analyze the shortest distance to the ground truth but compute the shortest distance to an ‘average’ boundary spline. This average spline is derived from the set of results obtained from the repetitions of the considered frame, i.e. its single control point coordinates  ${}^B\bar{\theta}_t$  are averaged from the respective control point coordinates  ${}^B\hat{\theta}_{t,k}$  estimated in the repetitions  $k = 1, \dots, K$  yielding

$${}^B\bar{\theta}_t = \frac{1}{K} \sum_{k=1}^K {}^B\hat{\theta}_{t,k}. \quad (4.1)$$

Figure 4.10 provides an example depicting the estimated free-space boundaries obtained from the same frame in repeated runs as well as the respective average boundary.

**Discussion:** Figure 4.11 presents the statistics of the computed error distances to the average spline. As for the outer precision, the error is plotted against the distance to the camera using the same discretization bins as for Figure 4.8. The cumulative distribution of the errors is depicted in Figure 4.12.

As expected, the statistics reveal that the results for obstacle heights between 0.1 m and 0.4 m for noise levels  $\sigma_d \leq 0.5$  m are extremely stable, i.e. show very low deviation with respect to the average boundary. From the cumulative distribution of the error, we see that approximately 99% of the deviations are smaller than 0.1 m. In this regard, the algorithm performs slightly better with increasing obstacle height. The precision slightly decreases for the noise value  $\sigma_d = 0.75$  pel and significantly drops for  $\sigma_d = 1$  pel.

For an obstacle height of 0.05 m, the statistics show high deviations to the average spline implying the results of the Monte Carlo runs to be mutual inconsistent.

The statistics of the negative obstacle height of  $-0.2$  m show a behavior which is similar to those of the heights  $> 0.05$  m. The deviations to the average boundary are very small for noise levels  $\sigma_d \leq 0.75$  pel, although we observed the tendency of overshooting when evaluating the outer precision in the previous paragraph. Put simply, the algorithm tends to overshoot obstacle borders of negative height but, however, yields consistent results even for input noise levels up to  $\sigma_d = 0.75$  pel.

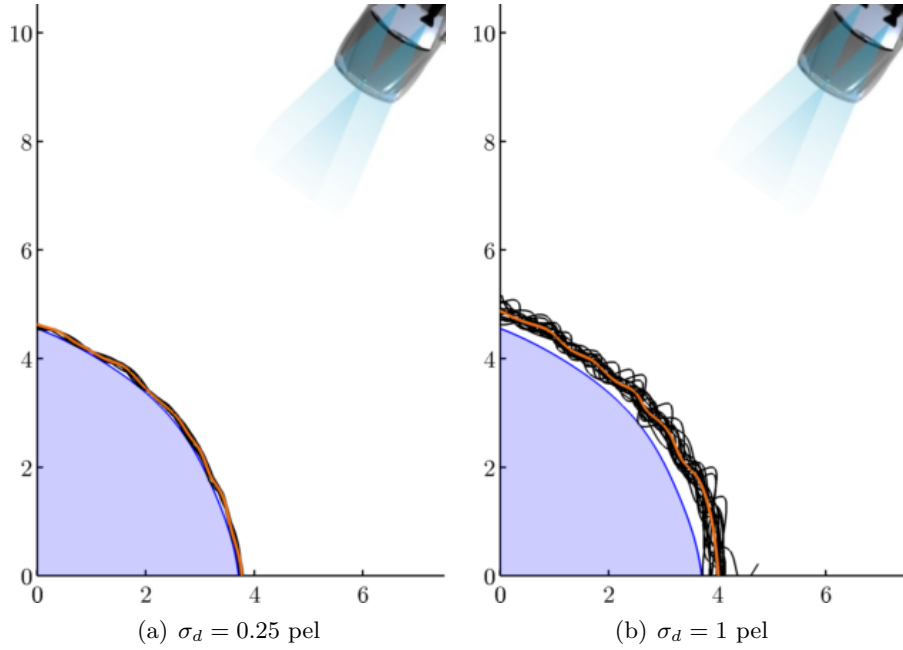


FIGURE 4.10: Examples for an average spline used to compute the inner precision for two noise levels. Street regions are plotted in white color, while non-street regions are plotted in bright blue. The true street boundaries are drawn by blue lines. The annotation of the coordinate axes is given in [m]. The thin black lines represent the estimated free-space boundaries of the  $N = 20$  Monte Carlo runs. The orange line depicts the respective average spline computed via (4.1). Both results are obtained for an obstacle height of 0.2 m.

#### 4.2.4 Evaluation of the influence of outliers

In this experiment, the ability of the approach to deal with outliers (gross errors) in the input data is assessed. This means that the input data is not only disturbed by Gaussian noise but is additionally modified by a certain amount of errors which exceed the  $3\sigma_d$  confidence interval of the noise distribution.

We expect the approach to be able to deal with outliers up to a moderate degree due to the sophisticated approach utilized for DEM computation as well as the explicit modeling of outliers in the expectation step. However, the treatment of outliers is challenging because of the high flexibility of the spline representation employed to model the street surface and in particular the free-space boundary.

**Experimental Setup:** The setup differs from that of the first experiment only in the way the synthetic disparity maps are disturbed. Instead of using Gaussian noise of different levels for the single runs, the single noise values are drawn from mixture distributions. More precisely, a certain percentage  $100\% - \Omega$  of the noise values is drawn from a Gaussian distribution of fixed standard deviation  $\sigma_d = 0.5$  pel, which matches the noise assumption of the algorithm (see Table 4.1). The remaining percentage  $\Omega$  is homogeneously drawn from two uniform distributions representing outliers in the disparity measurements. We define outliers to be noise values exceeding the  $3\sigma_d$  confidence interval of the Gaussian distribution. Thus, the uniform distributions are defined to lie on both sides of the  $3\sigma_d$  confidence interval in the ranges  $[-10\sigma_d, -3\sigma_d]$  and  $[3\sigma_d, 10\sigma_d]$  respectively. The single runs of the experiments utilize different ratios of outliers  $\Omega \in 10\%, 20\%, 30\%$ , i.e. of randomly chosen noise values which are drawn from the uniform distributions. Figure 4.13 illustrates the mixture distributions for all considered outlier ratios. The experiments are evaluated for a fixed obstacle height of 0.2 m.

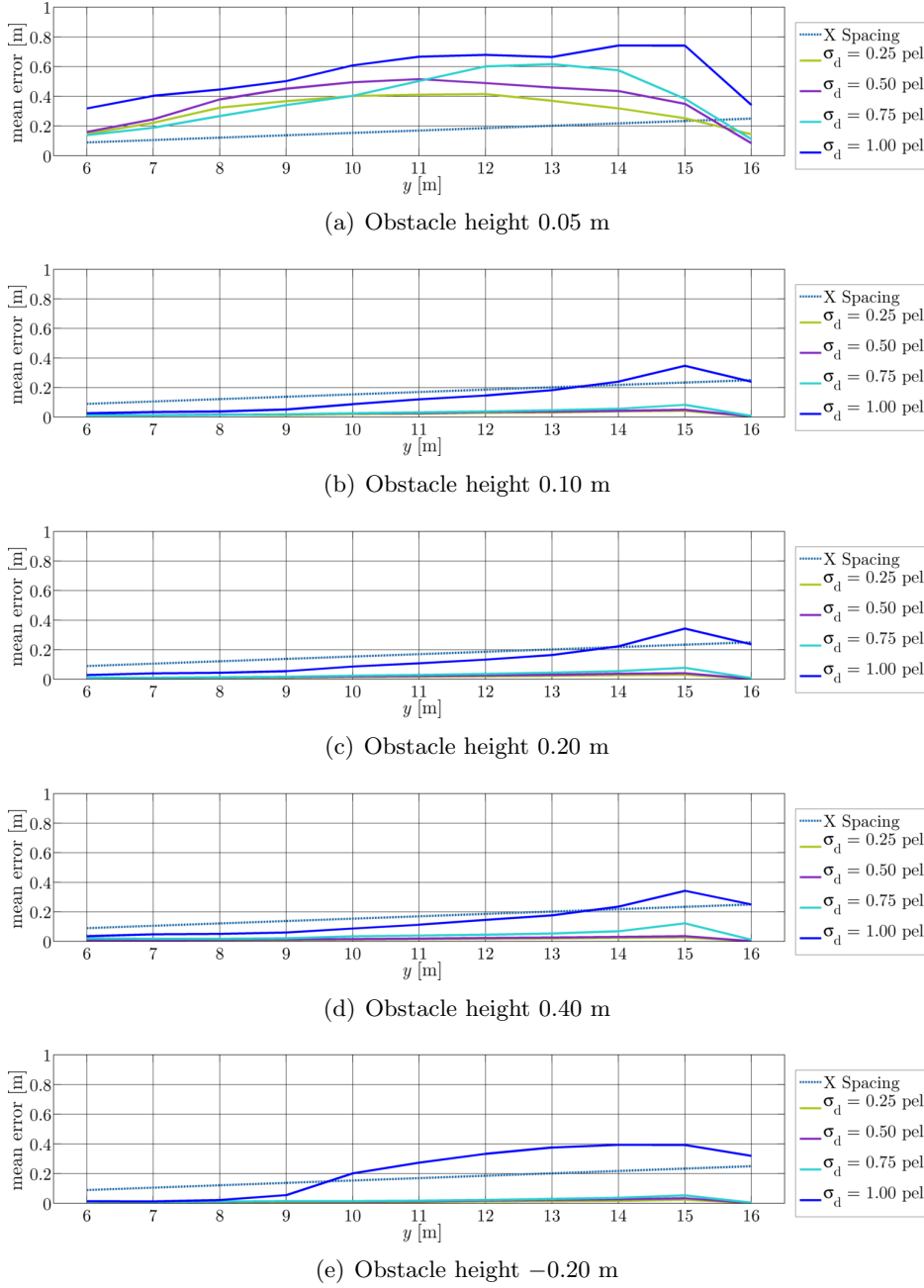


FIGURE 4.11: Mean errors obtained on the synthetic dataset for the evaluation of the inner precision plotted against the distance to the camera. The error measure describes the shortest distance of the estimated spline to the average spline computed from the results of all Monte Carlo runs of the respective frame via (4.1). The mean errors were evaluated in successive 1 m intervals with respect to the camera distance. The blue dashed reference line describes the local cell spacing of the DEM. Each plot collects the result for the indicated obstacle height composed of the mean errors obtained for the considered disparity noise levels  $\sigma_d$ . The results reveal that the algorithm yields consistent results for the Monte Carlo runs of all obstacle heights  $\neq 0.05$  m and noise levels  $\sigma_d < 1$  pel. The results obtained for the obstacle height of 0.05 m appear to be inconsistent. Note that the errors drop in the last bin since by construction the estimated boundary spline will not significantly exceed the DEM grid borders (The samples computed in step 1 of the three-step approach presented in Section 3.5.2.2 will never exceed the DEM grid borders).

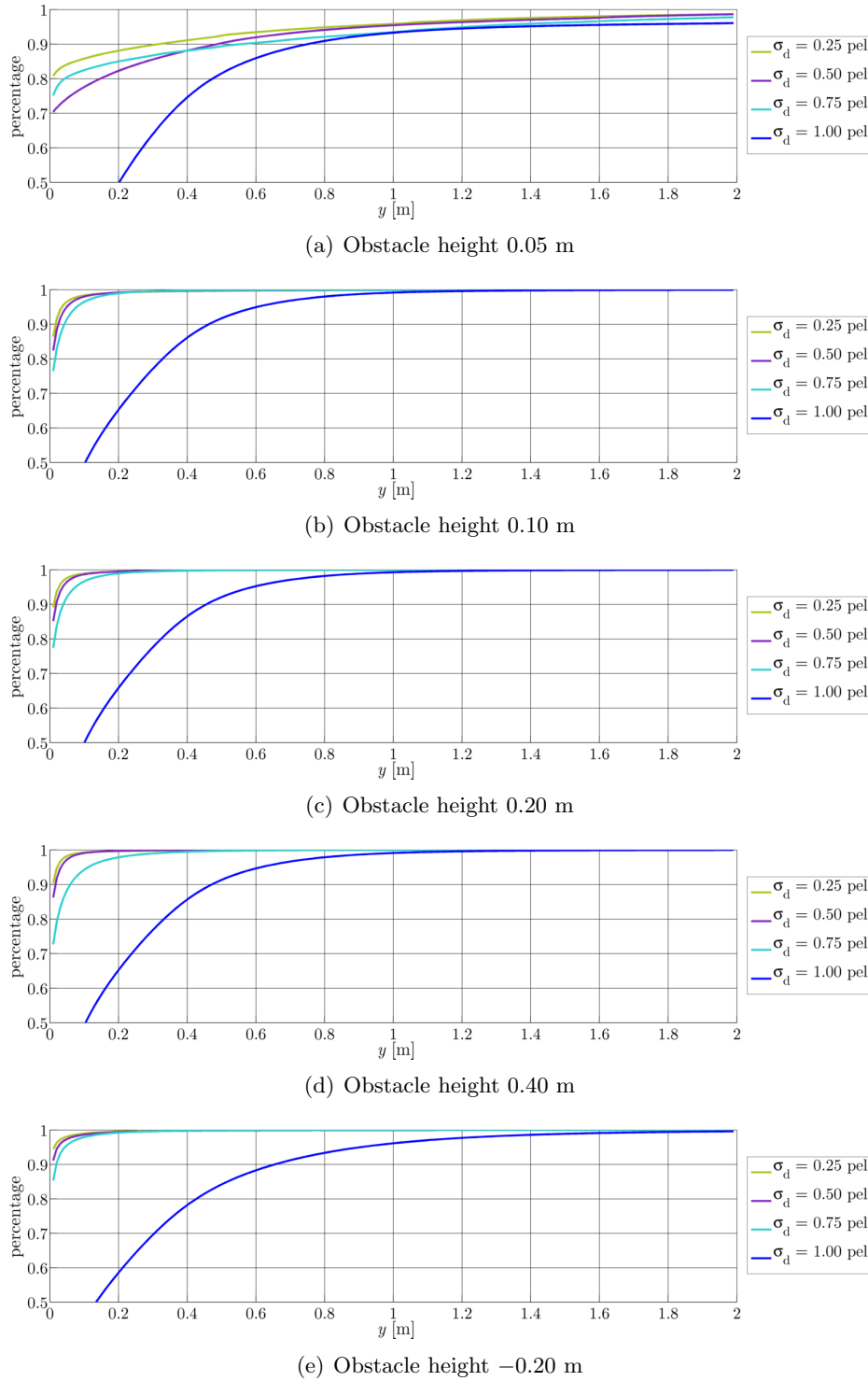


FIGURE 4.12: Cumulative distributions of the errors obtained on the synthetic dataset for the evaluation of the inner precision. The group structure of the single figures corresponds to the structure chosen in Figure 4.11. Note that the labeling of the ordinate begins at 0.5 to improve the resolution. For obstacle heights  $\neq 0.05$  m and for noise levels  $\sigma_d \leq 0.5$  pel, approximately 99% of the errors are smaller than 0.1 m implying a high consistency of the results obtained from the different Monte Carlo runs. Comparison of the single results among each other reveals the same relative behavior as observed for the mean error. For  $\sigma_d = 1$  pel as well as for an obstacle height of 0.05 m, the results reveal a high percentage of large errors implying inconsistent results.

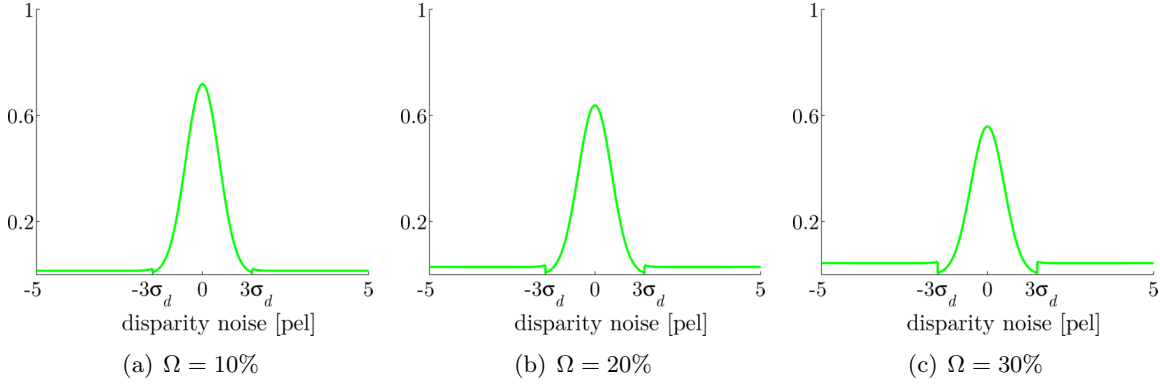


FIGURE 4.13: Mixture distributions utilized to generate the disparity noise including different percentages of outliers  $\Omega \in 10\%, 20\%, 30\%$ . The distributions are composed of a normal distribution  $\mathcal{N}(0, \sigma_d^2)$  governed by a standard deviation of  $\sigma_d = 0.5$  pel and of a uniform distribution divided evenly on both sides of the  $3\sigma_d$  confidence interval of the Gaussian distribution. Put simply, the uniform distribution is defined over the range  $[-10\sigma, 10\sigma]$  with the interval  $[-3\sigma, 3\sigma]$  carved out.

**Discussion:** Table 4.5 presents the accuracy assessment of the image projection of the detected free-space. The single rows hold the entries of the confusion matrices obtained for the different outlier ratios. For comparison, the first row holds the results obtained for the same noise level of  $\sigma_d = 0.5$  pel but without any outliers attached.

Figure 4.14 summarizes the statistics obtained for the spatial precision of the estimated free-space boundary with respect to the ground truth. Figure 4.14(a) presents the mean errors plotted against the distance to the camera, while Figure 4.14(b) depicts the cumulative distribution of the error.

Comparison of the first two rows in Table 4.5 and the respective statistics in Figure 4.14 reveals that the presence  $\Omega = 10\%$  outliers does not considerably affect the achieved precision. The effect for twice the value  $\Omega = 20\%$  is more apparent resulting in nearly twice the mean error for distances over 12 m. The third row in Table 4.5 shows an increased tendency to underestimate the distance to the obstacle, which we again assume to be caused by the scattering effect discussed for the first experiment. By comparison, the precision observed for  $\Omega = 20\%$  nearly matches the results achieved without outliers in the first experiment for the noise level  $\sigma_d = 0.75$  pel.

Finally, a ratio of  $\Omega = 30\%$  shows a significantly drop in the achieved precision implying that the algorithm reaches its limits. Visual examination reveals that the strong scattering of the points triangulated from the disparity image even those originating from the street surface lead to massed erroneous elevation measurements mimicking faked obstacles. For illustration purpose, Figure 4.15 depicts an example for a DEM computed under an outlier ratio of  $\Omega = 30\%$ .

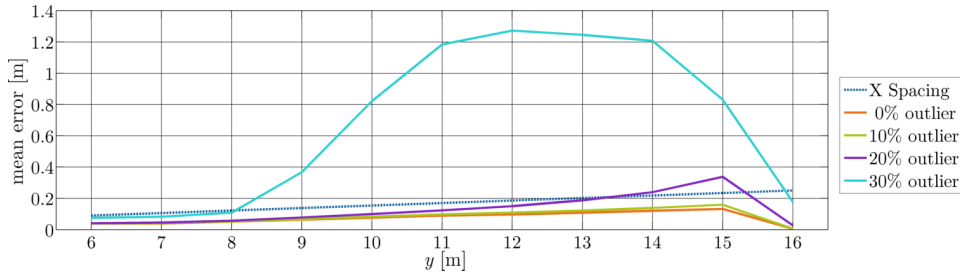
#### 4.2.5 Discussion of the Results on Synthetic Data

Summarized, the results of the single experiments reveal that for obstacle heights  $\geq 0.1$  m and for noise levels  $\sigma_d \leq 0.5$  pel the algorithm performs well, i.e. the mean error lies in the order of magnitude of half the local cell spacing of the DEM grid and the result of multiple Monte Carlo runs appears to be consistent. Note that  $\sigma_d \leq 0.5$  pel matches the noise assumption of the algorithm (see Table 4.1). The performance decreases for  $\sigma_d = 0.75$  pel

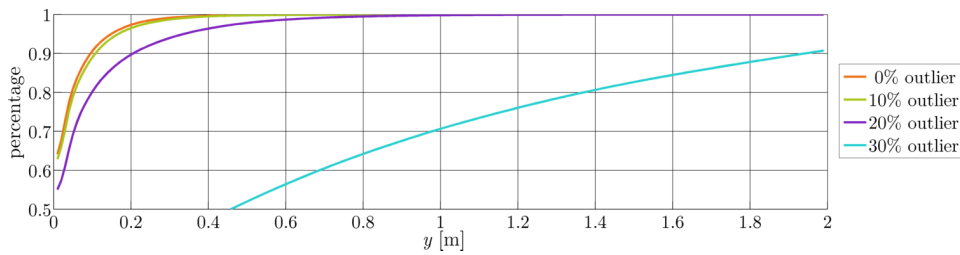


TABLE 4.5: Accuracy assessment of the image projection of the detected free-space. Each row holds the entries of a confusion matrix describing the classification result obtained on the synthetic dataset under different outlier ratios. A detailed description of the pattern describing the meaning of the row entries is given in Table 4.2. The total ratio of *free-space* pixels in the synthetic database is 64.0% yielding 36.0% for the total ratio of *non-free-space* pixels.

	GT	<i>non-free-space</i>		<i>free-space</i>	
	Est.	<i>non-free-space</i>	<i>free-space</i>	<i>non-free-space</i>	<i>free-space</i>
outlier percentage					
0%		98.9	1.1	0.9	99.1
10%		99.0	1.0	1.2	98.8
20%		99.4	0.6	2.3	97.7
30%		99.9	0.1	15.2	84.8



(a) Mean error



(b) Cumulative Error

FIGURE 4.14: Error statistics obtained for the different outlier ratios  $\Omega \in 10\%, 20\%, 30\%$  with an obstacle height of 0.20 m and a noise level of  $\sigma_d = 0.5$  pel. The statistics describe the mean error to the ground truth plotted against the distance to the camera (a) and the cumulative error distribution (b). For comparison, the results obtained without outliers are plotted as well. The results reveal that adding 10% outlier does not significantly affect result. For 20% outlier ratio the error increases noticeably, while under 30% the algorithm reaches its limits.

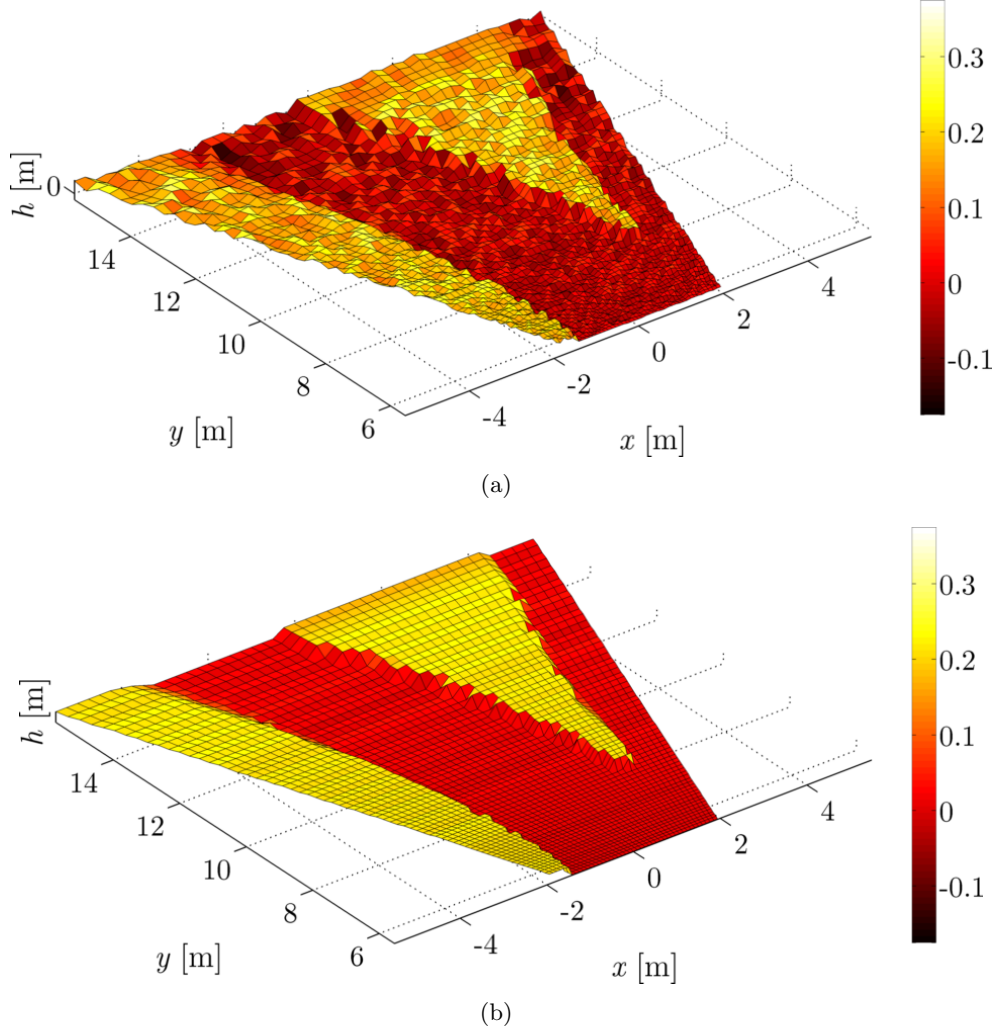


FIGURE 4.15: Example for an DEM computed for the scenario depicted in the first column of Figure 4.5. The obstacle height of the simulated scenario amounts 0.20 m. In (a) the synthetic disparity measurements are perturbed with noise level of  $\sigma_d = 0.5$  pel and an outlier ratio of  $\Omega = 30\%$ . Figure (b) shows the result obtained without any perturbation for comparison. The colors denote the local elevation whereas the colorbar scale is given in [m]. Especially in the rear area of the DEM grid, one can observe massed erroneous elevation measurements which are high enough to mimic faked obstacles.

yielding approximately twice the value for the mean error. This is, however, still in the order of magnitude of the local cell spacing. For larger noise levels ( $\sigma_d = 1$  pel) and smaller obstacles (0.05 m) the algorithm obviously reaches its limits.

Furthermore, the results show that the approach is able to deal with obstacles of negative height. In this case, the estimated free-space boundary shows a stable tendency to slightly overshoot.

Regarding gross errors in the input data, the approach demonstrates the ability to nearly compensate outliers up to a ratio of 10% and to yield reasonable results for ratios up to 20%.

Note that this insights originate from experiments on synthetic data which was generated based on simplified assumptions. Thus, the insights will hold for real-world data only to a limited extend but, however, provide an idea of the achievable performance.

### 4.3 Experiments on Real-World Data

This section presents experiments based on stereo image sequences acquired in real-world traffic scenarios. By means of the experiments we analyze the following aspects:

- (a) Applicability of the approach in real-world traffic scenarios.
- (b) Influence of selected model components on the performance of the overall approach.
- (c) Transferability of the insights achieved on synthetic data.
- (d) Limitations of the approach.

First of all we describe the setup of the research vehicle serving as sensor platform for the data acquisition in Section 4.3.1. Section 4.3.2 describes the benchmark dataset utilized for evaluation of the presented approach. The experiments evaluating the performance of the approach on real-world data are described and discussed in three sections. Section 4.3.3 evaluates the performance using the default parameter set. Section 4.3.4 evaluates the performance of single object components and the effect of certain model modifications. Section 4.3.5 assesses the frequency of the erroneous detection of non-existing obstacles (alias *ghost objects*). Section 4.3.6 summarizes the insights obtained from the experiments with respect to the aspects stated above.

Note, experiments on real-world data are vital for approaches targeting real-world applications in order to evaluate the actual applicability of the approach. However, due to the lack of unbiased and exact ground truth information, we understand the statements obtained from real-world experiments to be rather of qualitative than of quantitative nature.

#### 4.3.1 Data Acquisition and Setup of the Research Vehicle

The datasets forming the basis for the real-world experiments were recorded by a Mercedes-Benz research vehicle, as depicted in Figure 4.16. The vehicle is designed for testing and evaluation of innovative camera based driver assistance systems.

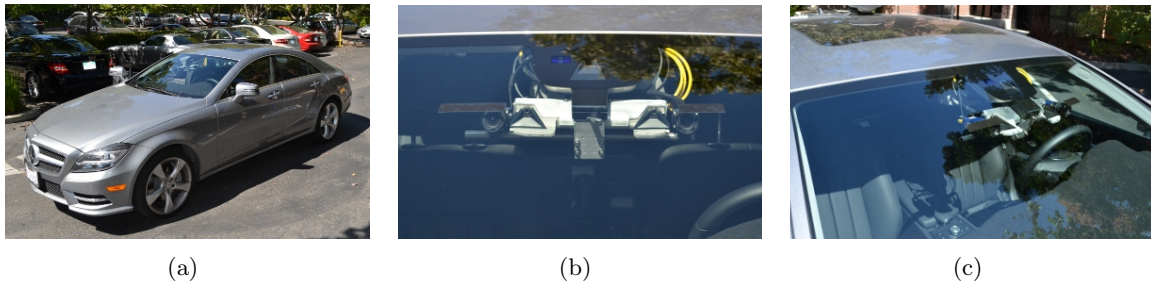


FIGURE 4.16: (a) Research vehicle providing the sensor platform for data acquisition. The stereo camera system depicted in (b) and (c) is mounted behind the windshield.

A stereo camera rig with a baseline of  $\approx 0.3$  m and aperture angle of approximately  $44^\circ$  is mounted behind the windshield pointing in driving direction along the vehicle's longitudinal axis. The system provides stereo image sequences with a frequency of 25 frames per second. The recorded 12 bit gray-scale images have a resolution of  $1024 \times 440$  pel and are rectified to the normal case. For each stereo image pair, dense disparity maps were computed via SGM in real-time on a FPGA. Section 2.2 provides detailed information regarding rectification and disparity map computation.

Additionally, full 3d ego-motion information is provided by the image based approach of [Badino et al., 2007], as described in Section 3.2.3.

The approach presented in this thesis is evaluated offline since multiple components are implemented in Matlab and, thus, are not real-time capable so far. See Section 4.3.6.1 for a discussion of computation times.

In the reminder of this chapter we use the term *sequence* to address the entirety of the data acquired simultaneously with a recorded image stream, i.e. disparity maps, camera calibration parameters and ego-motion information.

### 4.3.2 Benchmark Dataset

For the sake of evaluation, a set of sequences was combined to a *benchmark dataset* and assigned with manually annotated ground truth information.

#### 4.3.2.1 Dataset Description

The dataset comprises 11 sequences recorded in urban and suburban environment with a total number of 8325 frames. The sequences are chosen to cover multiple scenarios comprising intersections, traffic isles and roundabouts under the presence of various moving and static objects, such as parking and driving cars and pedestrians crossing the street. It further contains a short sequence of a highway construction site.

Every 25th frame was manually annotated by an expert, i.e. the image is pixel-wise labeled with *free-space* or *non-free-space*. Figure 4.17 and Figure 4.18 depict sample images taken from the dataset as well as example annotations and estimated results achieved from the presented approach using the default parameter set.

In the following we use the term ‘*free-space* pixels’ to address pixels assigned to *free-space* in the annotated image, and the term ‘*non-free-space* pixels’ to refer to those assigned to *non-free-space*.

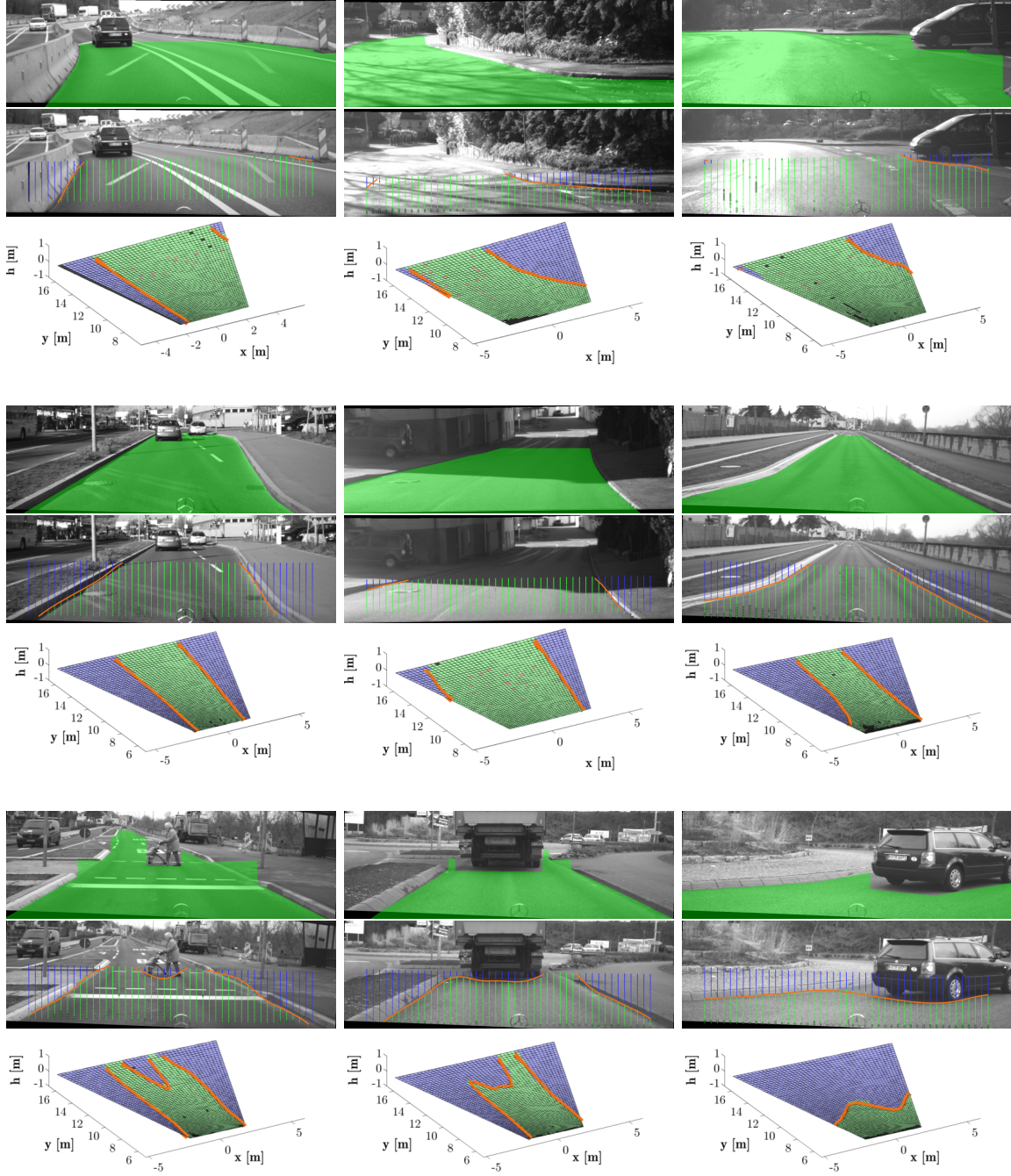
#### 4.3.2.2 Evaluation Methods

Similar to the synthetic experiments, we compare the annotation with the respective estimated result by means of the two evaluation measures presented in Section 4.1. While the ground truth for the assessment of the image projection of the free-space is directly given by the annotation, some precomputation is necessary in order to obtain ground truth information for the spatial evaluation.

The idea is to project the annotated boundary from the image onto the estimated street surface. More precisely, the term *annotated boundary* denotes the set which holds the lowest non-free-space pixel for each column of the annotated image. We intersect the optical rays through these pixels with the estimated street surface using the known intrinsic camera calibration parameters. This provides the desired ground truth samples of the free-space boundary in the horizontal space which line up to the ground truth boundary.

### 4.3.3 Benchmark Evaluation Using the Proposed Model and Parameters

In this experiment, we evaluate the performance of the presented approach on the benchmark dataset with respect to the validity of the estimated free-space region and the accuracy of the estimated free-space boundary. For that purpose, the evaluation measures presented in Section 4.1 are computed for each frame of the database. The results are obtained using the default parameter set presented in Table 4.1.





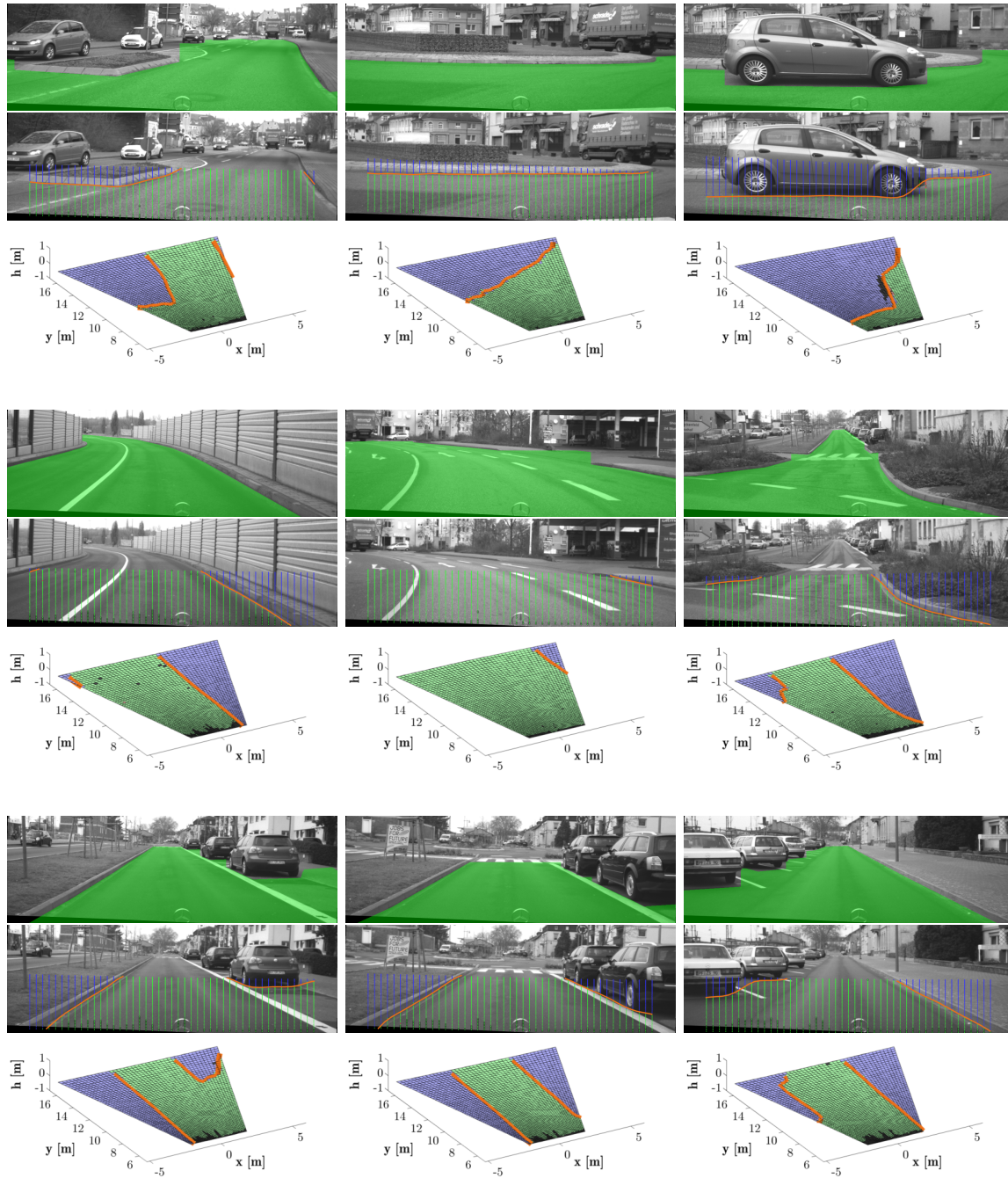


FIGURE 4.18: Three example scenarios of the benchmark dataset represented by a vertical triple of images, respectively. The first row depicts the ground truth annotation marked by a green carpet. The remaining rows illustrate the reconstruction and classification result in 3d (third row) and its image projection (second row). Color meanings for these rows are equivalent to those presented in Figure 3.14(d), i.e. green color marks cells classified as *street*, blue color marks cells classified as *adjacent* and red color marks cells classified as *outlier*. The spline representing the free-space boundary is plotted in orange color. Additionally, we mark the grid cells tagged as invalid due to insufficient data density in the DEM estimation with black color (part II).

As observed in the experiments on synthetic data, we expect the accuracy to drop with increasing distance to the sensor, proportional to the DEM's cell spacing. For the absolute value, we expect the accuracy to be clearly lower than in the synthetic experiments due to the high complexity of real-world scenarios causing severe violations of the environment and measurement model assumptions.

#### **Discussion of the Assessment of the Image Projection of the Detected Free-Space:**

The accuracy assessment of the image projection of the detected free-space computed from all frames of the benchmark dataset is presented in the first line of Table 4.6, which we will use as reference for the experiments in Section 4.3.4.

The respective table entries present high and balanced class-wise accuracies, i.e. 98.7% for *non-free-space* and 96.9% for *free-space*. Considering the missclassifications, slightly more *free-space* pixels were incorrectly assigned to *non-free-space* than vice versa. This trend applies for almost all results of the following experiments. We assume the spatial constraining model of the boundary spline to be a potential cause for this effect. Yielding robustness on the one hand it causes the boundary spline to bridge over narrow gaps between neighboring obstacles causing a intermediate free-space region to be misclassified. Examples for this bridging effect appear in the scenarios illustrated in the bottom row in Figure 4.17, where gaps between the car/pedestrian and the sidewalk cannot be exactly modeled by the spline. A detailed discussion of this effect is given in Section 4.3.6.4.

**Discussion of the Spatial Accuracy of the Free-Space Boundary:** Let us now consider the spatial accuracy of the free-space boundary. Figure 4.19(a) presents the statistics of the computed error distances plotted against the distance to the camera. More precisely, the complete set of errors is arranged in successive 1 m wide bins along the longitudinal axis. The plotted lines connect the values of the respective mean error as well as the 0.75, 0.9 and 0.95-percentile value of each bin. For comparison issues, the local cell spacing of the DEM is plotted as blue dashed line. Note that the error drops for the last bin since we crop the ground truth to the DEM grid. This effect was already discussed in the third paragraph of Section 4.2.2.

Figure 4.20(a) presents the local cumulative distribution of the error for each distance bin in more detail. The topmost row underlines the just mentioned statistical effect induced by samples on the DEM grid borders. The statistics show that 75% of the errors do not significantly exceed the mean error, whereas a little less than 90% of the errors are smaller than the duplicated mean value. We consider around 5% percent of the errors in distances between 12 m and 15 m as significant since they exceed a value of 1 m. We will present several effects causing the significant errors when discussing the limitations of the approach in Section 4.3.6.4.

Figure 4.20(b) presents the cumulative distribution plotted against the lateral position on the spline. The plot reveals that the errors appear to be smaller at the center of the spline than at the outsides. Comparison to Figure 4.20(a) shows that the correlation of the error to the lateral position is significantly smaller than to the longitudinal position.

When discussing the results obtained under several model modifications in Section 4.3.4, we will use the mean error as well as the cumulative error distribution plotted in Figure 4.19(b) as reference values.

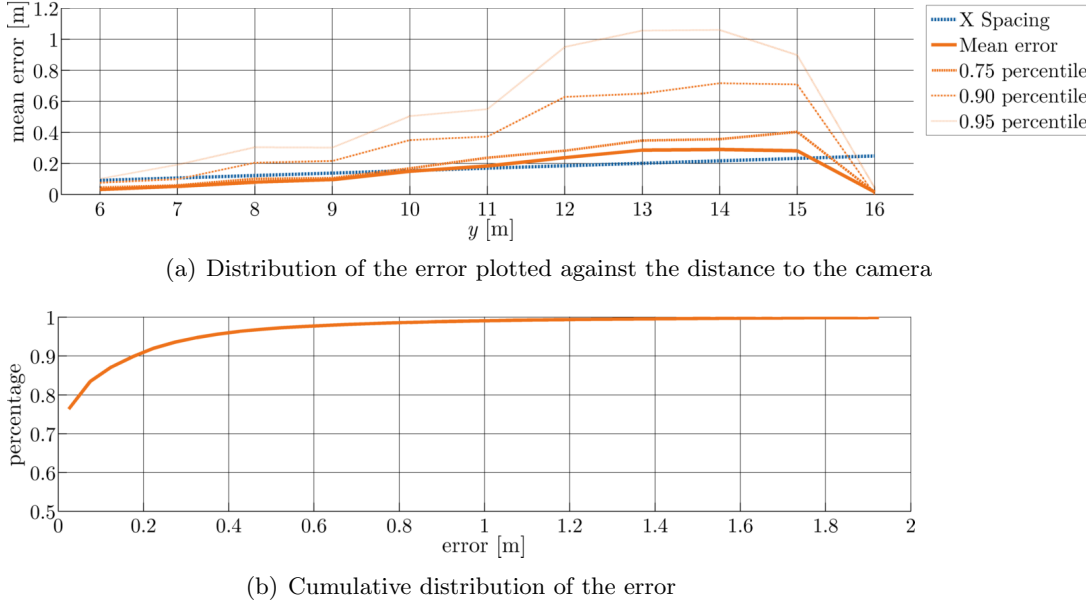


FIGURE 4.19: Error statistics obtained on the benchmark dataset using the default parameter set. Figure (a) illustrates the distribution of the error of the estimated boundary in relation to the distance to the camera. The error statistics are given by the mean error as well as the 0.75, 0.9 and 0.95-percentiles which were evaluated in successive 1 m intervals with respect to the camera distance. The blue dashed reference line describes the local cell spacing of the DEM. Figure (b) depicts the cumulative distribution of the error. Note that the labeling of the ordinate begins at 0.5 to improve the resolution. The mean error lies in the order of magnitude of the horizontal cell spacing. Around 75% of the errors do not significantly exceed the mean value and around 90% are smaller than the duplicated mean value. Significant errors appear especially in the region between 12 m to 15 m.

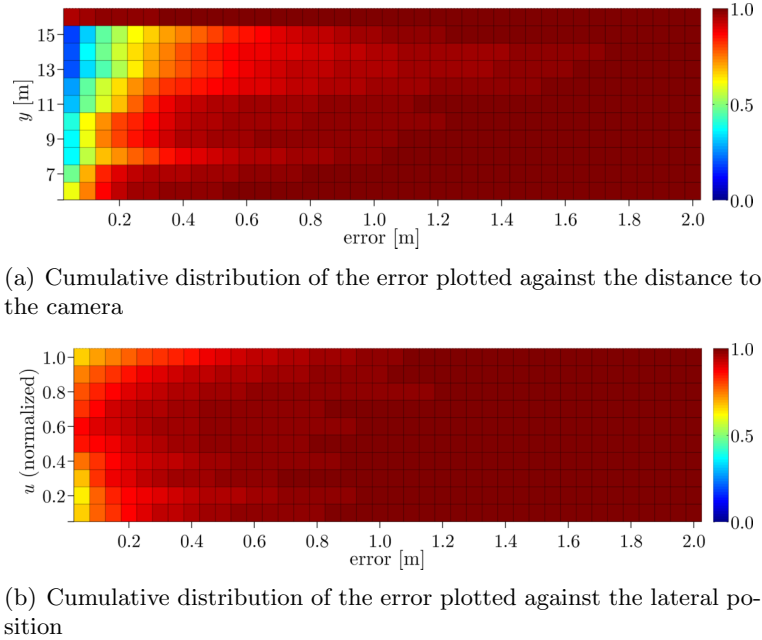


FIGURE 4.20: Detailed examination of the error distribution with respect to the relative position of the sensor. Figure (a) illustrates the cumulative distribution of the error evaluated in successive 1 m intervals on the  $y$ -axis (same intervals as used in Figure 4.19(b)). Figure (b) plots the cumulative distribution of the error with respect to the boundary spline parameter and, thus, to the lateral image axis  $u$ . To simplify the notation the values of  $u$  are normalized to the interval  $[0, 1]$ . The magnitude of the error grows as expected with increasing distance and outwardly regarding the lateral direction.



#### 4.3.4 Evaluation of Selected Parameter and Model Modifications

Beside the evaluation using the default model parameters, we repeat the last experiment with several changes of the presented model and parameters in order to analyze their impact on the estimated result. This comprises the evaluation using the following modifications which we refer to as *sub-experiments*:

- Section 4.3.4.1: Changes of the number of maximum iterations in the EM-algorithm.
- Section 4.3.4.2: Simplification of the street surface model.
- Section 4.3.4.3: Simplification of the DEM classification step by simplifying binary and unary terms.
- Section 4.3.4.4: Omission of the temporal priors for the estimation of street surface and free-space boundary parameters.
- Section 4.3.4.5: Omission of the spatial prior for the estimation of free-space boundary parameters.
- Section 4.3.4.6: Omission of the local self evaluation step presented in Section 3.5.3.1.
- Section 4.3.4.7: Simplified computation of the DEM.

We expect all modifications to cause a drop of the classification accuracy as well as of the spatial accuracy of the boundary spline.

The following sections describe the respective changes of the model and the model parameters in detail. They also discuss the results with respect to the evaluation criteria presented in Section 4.3.2.2. To allow for a more convenient comparison, Table 4.6 summarizes the confusion matrices which represent the accuracy assessment for each single sub-experiment. Each confusion matrix is arranged in a single row using the pattern presented in Table 4.2.

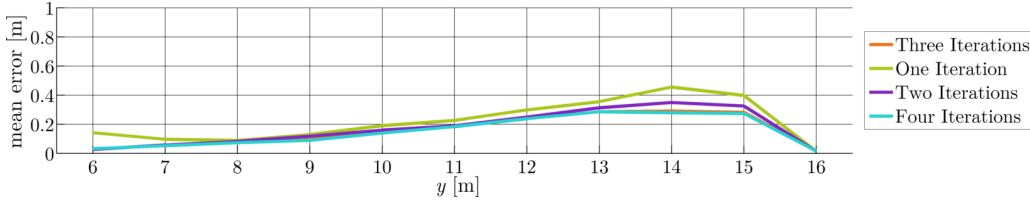
Analogously, Figure 4.21 compares the spatial accuracy of the estimated boundary spline for each sub-experiment based on the mean error plotted against the distance to the camera. Further statistics are presented by means of the cumulative distribution of the error in Figure 4.22. In each plot, the reference values obtained using the default parameter set are drawn in orange color.

##### 4.3.4.1 Evaluation of the Influence of the Maximum Number of EM Iterations

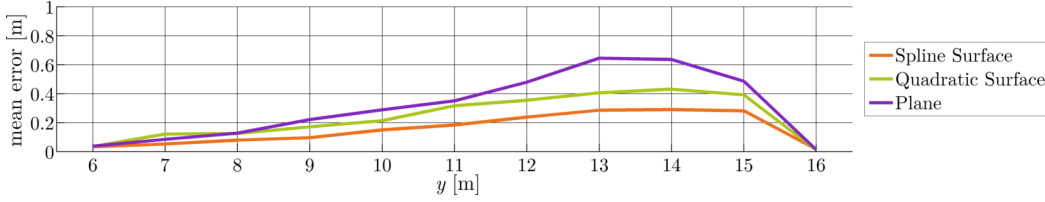
In this sub-experiment, we analyze the progression of the estimated result with increasing number of iterations used for the EM-algorithm. In the default parameter set, the method is terminated after a maximum of 3 iterations. For comparison, the experiment was repeated using a maximum number of 1, 2 and 4 iterations.

##### Discussion of the Assessment of the Image Projection of the Detected Free-Space:

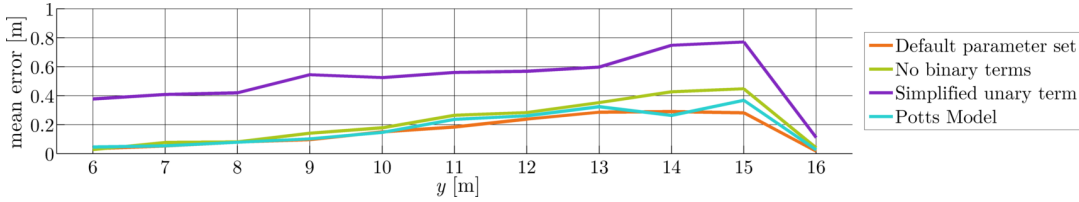
The classification results presented in the second block of Table 4.6 show only minor changes for the accuracy of *non-free-space* pixels but a continuous improvement for *free-space* pixels with increasing iteration number. This seems reasonable due to the fact that the vehicle is moving forward. In each frame the boundary spline needs to propagate from its predicted position into the newly observed free-space region until it reaches an obstacle. The distance which the spline may propagate is limited by the weighting factors  $\alpha_{ij}$  defined in (3.33) which are iteratively updated in the EM-algorithm. Limiting the iterations may prevent the spline



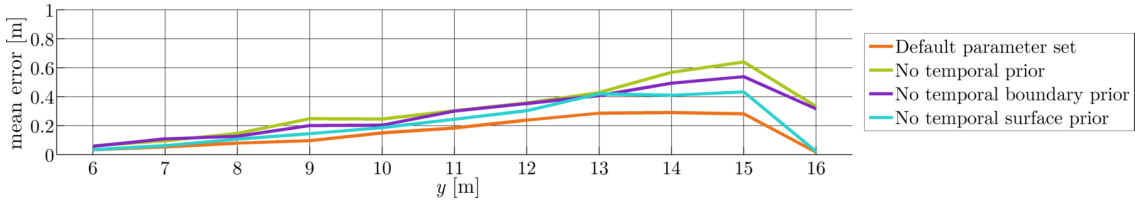
(a) Evaluation of the influence of the number of EM iterations



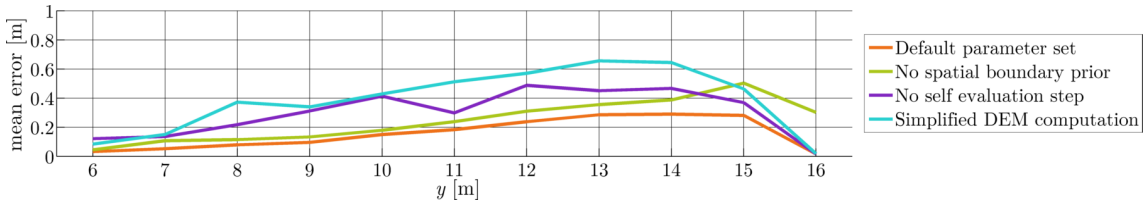
(b) Evaluation of the usage of simplified street surface models



(c) Evaluation of changes of the CRF's unary and binary terms



(d) Evaluation of the omission of the temporal priors



(e) Evaluation of several model simplifications

FIGURE 4.21: Mean errors obtained on the benchmark dataset using modified model parameters plotted against the distance to the camera. The means were evaluated using the same successive intervals as for Figure 4.19(a). The plots are divided in four groups: Figure (a) illustrates the variation of the error with respect to the number of iterations used for the EM-algorithm. The influence of alternative surface models is depicted in Figure (b). Figure (c) describes the impact of changes of the CRF's unary and binary terms, while Figure (d) shows the effects of the omission of temporal prior terms. Figure (e) compares the results obtained from omission of the spatial boundary prior, the omission of the self evaluation step and the simplified computation of the DEM. In each figure the orange line describes the reference error obtained using the default parameter set. The default parameter set achieves the lowest error for all experiments. Further comparison reveals that the sophisticated approach for DEM computation, the usage of a flexible street surface model and the usage of the full unary term appear to be most crucial.

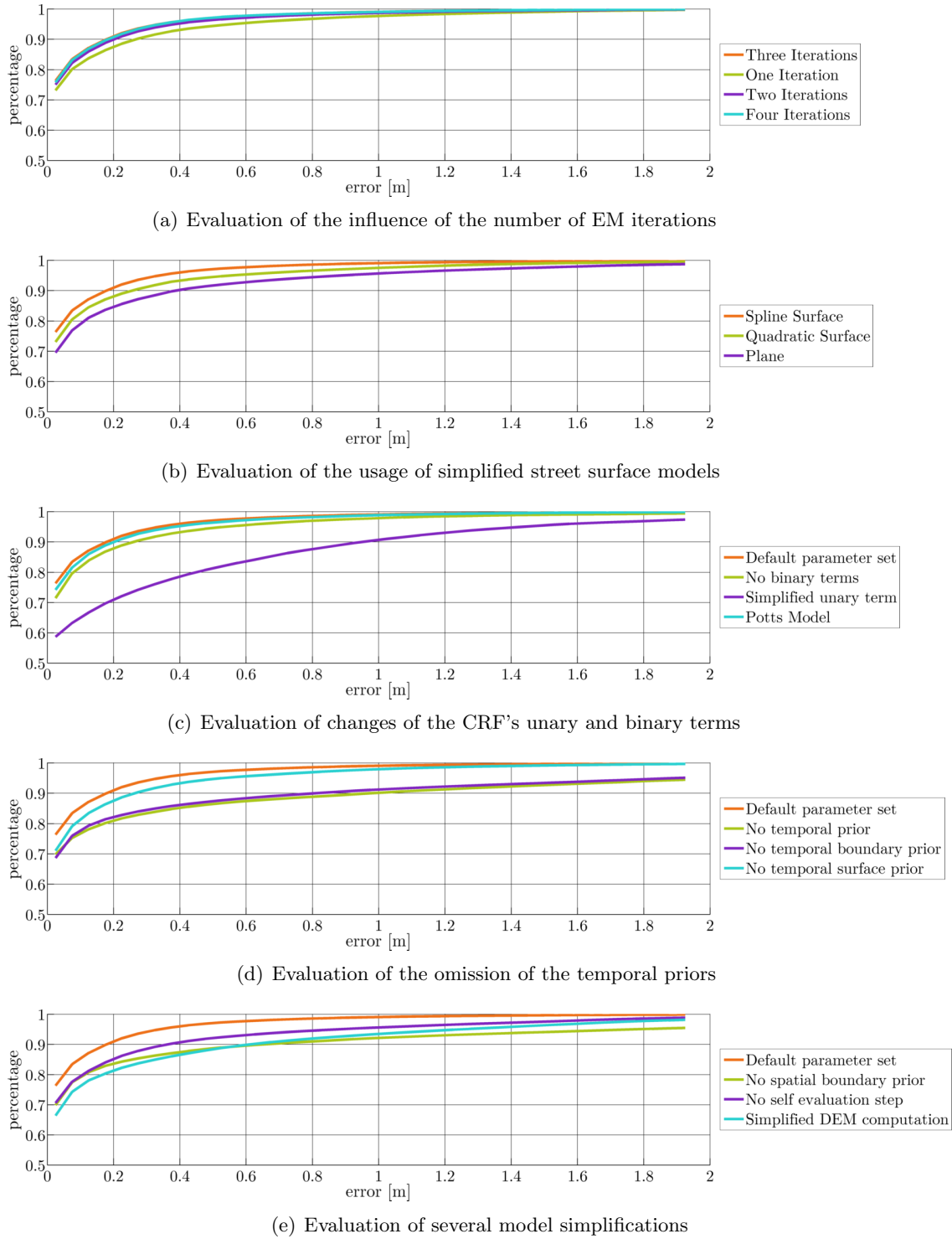


FIGURE 4.22: Cumulative distributions of the errors obtained on the benchmark dataset using modified model parameters. The group structure of the single figures corresponds to the structure chosen in Figure 4.21. Again, the orange lines describe the values obtained using the default parameter set, also depicted in Figure 4.19(b). Note that the labeling of the ordinate begins at 0.5 to improve the resolution. The most crucial drop of precision appears for the simplified unary term which causes a significant increase of errors in the range of 0 m to 1.5 m. Further, the omission of the spatial and temporal boundary priors show a vast effect indicated by the relatively slow ascend of the respective cumulative distributions and by the fact that over 5% of the errors exceed a value of 1.9 m.

TABLE 4.6: Accuracy assessment of the image projection of the detected free-space. Each row holds the entries of a confusion matrix describing the classification result obtained on the benchmark dataset. The results are obtained under the default parameter set (first row), varying iteration numbers of the EM-Algorithm (second block), alternative street surface models (third block), alternative CRF terms (fourth block), omission of temporal priors (fifth block) or other modifications (sixth block). A detailed description of the pattern describing the meaning of the row entries is given in Table 4.2. The total ratio of *free-space* pixels in the annotated database is 43.6% yielding 56.4% for the total ratio of *non-free-space* pixels.

	GT	<i>non-free-space</i>		<i>free-space</i>	
	Est.	<i>non-free-space</i>	<i>free-space</i>	<i>non-free-space</i>	<i>free-space</i>
default parameters		98.7	1.3	3.1	96.9
1 EM iteration		98.8	1.2	5.2	94.8
2 EM iteration		98.7	1.3	3.5	96.5
4 EM iteration		98.6	1.4	2.9	97.1
quadratic surface		98.9	1.1	6.1	93.9
plane		99.0	1.0	7.3	92.7
no binaries terms		97.9	2.1	3.5	96.6
simplified unary terms		91.9	8.1	8.4	91.6
Potts model		98.7	1.6	3.5	96.5
no temp. prior		89.8	10.2	2.1	97.9
no temp. boundary prior		90.9	9.1	1.8	98.2
no temp. surface prior		98.5	1.5	4.3	95.7
no spatial boundary prior		92.2	7.8	1.9	98.1
no self evaluation step		98.7	1.3	6.9	93.1
simplified DEM comp.		99.0	1.0	9.7	90.3

to propagate right up to the newly observed obstacles and, thus, causes *free-space* pixels to be incorrectly classified as *non-free-space*.

The fact that the accuracy of *non-free-space* pixels does not drop in the same order of magnitude implies that the spline does not tend to overshoot the obstacle when propagating. This was confirmed from visual evaluation of the result sequences.

**Discussion of the Spatial Accuracy of the Free-Space Boundary:** The analysis of the error statistics given in Figure 4.21(a) and Figure 4.22(a) shows notable improvements from iteration 1 to iteration 2, marginal improvements from iteration 2 to iteration 3 and nearly no improvement from iteration 3 to iteration 4. This confirms the choice of a maximum of 3 iterations in the default parameter set.

#### 4.3.4.2 Evaluation of the Usage of Simplified Street Surface Models

Unlike the majority of free-space detection methods, the presented approach uses a spline based street surface model in order to allow for the representation of complex scenarios with varying surface curvatures. Figure 4.23 depicts a set of example scenarios motivating the usage of a complex surface model.

For comparison, error statistics were computed on the benchmark dataset using a quadratic surface as well as a plane as street surface model. For clarification, by the term



FIGURE 4.23: Example for street surfaces with high and multiple curvatures recorded in suburban regions in Redwood City, California. Note, the images are chosen for illustration issues and are not part of the benchmark database.

‘quadratic surface’, we understand the height model

$$h_{ij} = a_0x_{ij}^2 + a_1y_{ij}^2 + a_2x_{ij}y_{ij} + a_3x_{ij} + a_4y_{ij} + a_5 \quad (4.2)$$

with six free parameters  $a_0, \dots, a_5$ . A plane yields the height model

$$h_{ij} = b_0x_{ij} + b_1y_{ij} + b_2 \quad (4.3)$$

with three free parameters  $b_0, \dots, b_2$ .

The classification result shows a significant decrease for the class accuracy of *free-space* for both alternative surface models. This trend is confirmed by the statistics of the boundary spline errors presented in Figure 4.21(b) and Figure 4.22(b). Using a plane as street surface model results in a mean error which is nearly twice the error achieved with the spline model. Figure 4.24 gives some examples of erroneous estimations caused by the low flexibility of the plane model typically appearing at intersections or in case of roof-shaped street surfaces.

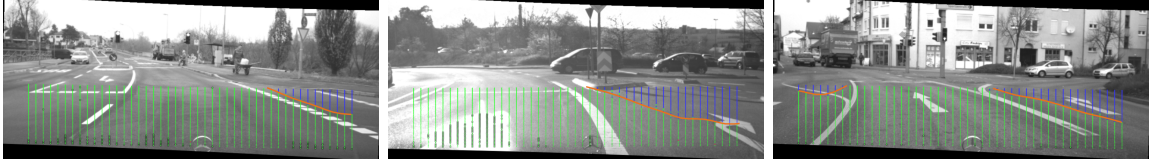


FIGURE 4.24: Example results showing estimation errors using a plane as street surface model. The low flexibility of the model prevents it from correctly modeling street lane adjacent surfaces with different slope, such as neighboring lanes in case of a roof-shaped street surface or crossing lanes at an intersection. Color meanings are equivalent to those presented in Figure 4.17.

The quadratic surface visually appears to be sufficient in the majority of scenarios but fails in complex situations such as in the highway construction site example presented in Figure 4.25. Further, it also fails in some situations similar to those presented in Figure 4.24 when a quadratic surface is insufficient to adequately model two adjacent planes with different slope. In those cases, the spline surface benefits from its ability to model multiple curvatures.

#### 4.3.4.3 Evaluation of Changes of the CRF’s Unary and Binary Terms

In the following, we analyze the influence of the unary and binary terms of the CRF governing the classification step, as described in Section 3.5.1. Three modifications were tested in different runs on the benchmark dataset:

- (I) Simplification of the unary term: Considering the definition of the unary term (3.80), we omit the spatial prior  $P(l_{ij,t} | B\hat{\theta}_t^{(\nu)})$ . Thus, the area of region competition between *street* and *adjacent* is no longer restricted.

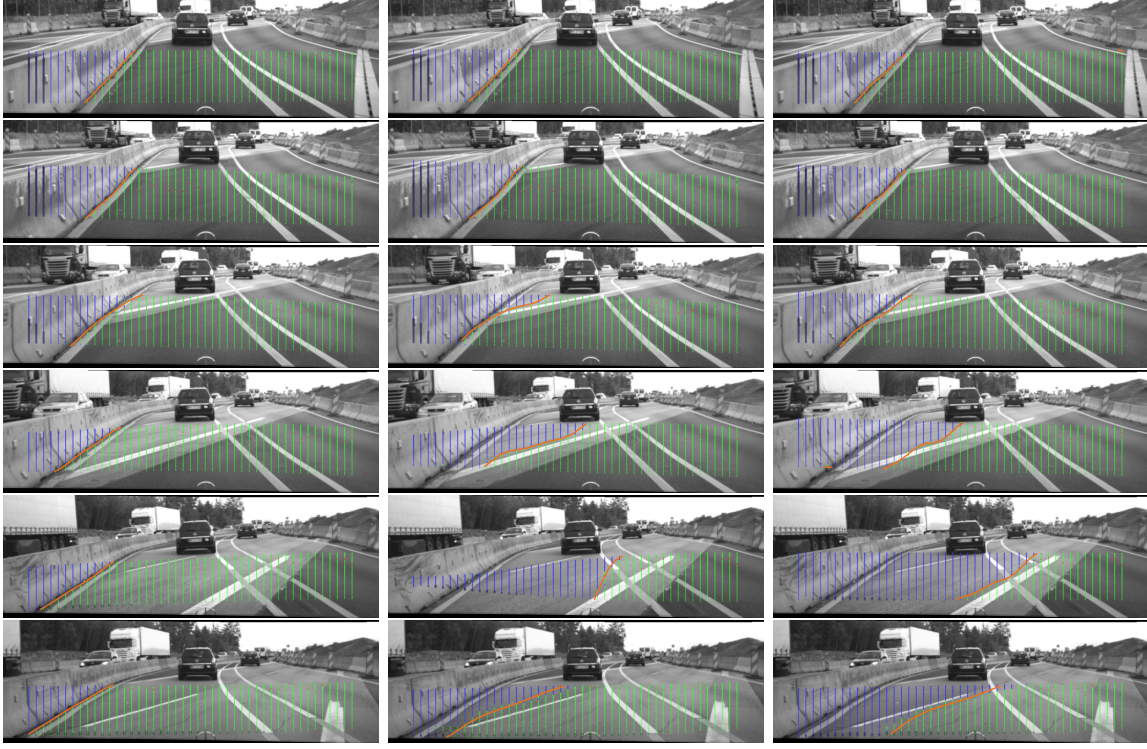


FIGURE 4.25: Comparison of different street surface models based on a highway construction site scenario. The lane crosses the median strip causing complex curvature characteristics for the street surface. The left column shows the estimated result using the proposed spline surface. The center column illustrates the result obtained using a quadratic surface, while the results in the right column were computed using a plane. To allow for a compact representation only every fifth image is plotted. Color meanings are equivalent to those presented in Figure 4.17.

- (II) Omission of the binary term: In this case, the classification is performed for each cell independently based on the unary term.
- (III) Simplification of the binary term: Instead of data dependent binary terms as defined in (3.86)-(3.88), we use a data independent Potts model. Neighboring cells are assumed to belong to the same class with probability  $1 - \epsilon$ , while  $\epsilon$  defines the probability to belong to different classes. We chose  $\epsilon = 0.01$  for this experiment.

The accuracy assessment is presented in the fourth block of Table 4.6. Figure 4.21(c) and Figure 4.22(c) illustrate the evaluation of the spatial accuracy.

**Discussion of the Results Obtained from Simplification of the Unary Term:** Both statistics show a significant drop of the accuracy for the simplified unary term (I). This demonstrates the importance of the spatial prior induced by the sigmoid function (3.33). The term constrains the alteration of the boundary spline by limiting the region competition in the classification step to a narrow band around the currently estimated boundary. The omission of the term changes the topology assumptions and allows for *street* cells on the far side of the current boundary as well as for *adjacent* cells on the near side. This significantly complicates the task to detect the correct free-space boundary via logistic regression.

**Discussion of the Results Obtained from Omission of the Binary Term:** The importance of the unary spatial prior term also reflects in the evaluation of the omission



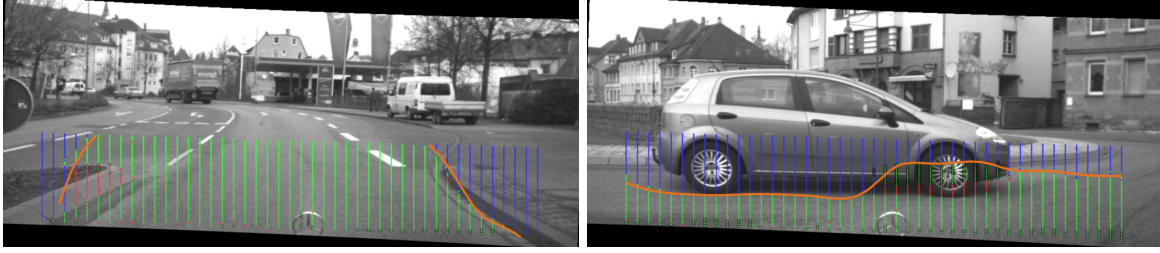


FIGURE 4.26: Example for erroneousness estimates when omitting the binary term. Cells assigned to the class *outlier* tend to cluster and form a rival hypothesis to the correct assignment of *adjacent*. This allows the boundary spline to overshoot obstacle borders. Color meanings are equivalent to those presented in Figure 4.17.

of the binary term (II). Comparing the accuracy of the boundary to the one estimated by means of the default parameter set, we observe a decrease especially in distances over 10 m. However, the accuracy does not drop as significantly as in (I). We assume this to be caused by strong unary terms. The main influence of the binary term is limited to the narrow band around the currently estimated spline since the unary spatial prior term already claims homogeneous class distributions in regions which are distant to the boundary. Omission of the binary term primarily impairs the classification result in this limited region and, thus, the overall impairment of the estimation result is limited.

Further, the binary term suppresses clusters of cells to be assigned to *outlier* (3.87). Without the binary term, *outlier*-clusters may form a rival hypothesis to the correct assignment of *adjacent* allowing the boundary spline to overshoot obstacle borders. Figure 4.26 provides an example for this effect.

**Discussion of the Results Obtained by Means of the Potts Model:** The error statistics obtained for the Potts model show that the improvement of the data dependent binaries used in the reference experiment is small but, however, apparent. Visual examination of the result sequences reveals high similarity between the estimated boundaries of both approaches, also in terms of the limitations.

However, in the majority of cases the data dependent binary terms yield a more exact estimation of the boundary. We assume that this behavior is caused by the fact that the model explicitly prefers positions at elevation discontinuities. These, in general, are indicators for the presence of an obstacle border especially for objects which are distinctive in height. This means, although simple height thresholding is insufficient particularly when detecting obstacles like curbs from extremely noisy data, the consideration of height discontinuities as feature is nevertheless useful.

#### 4.3.4.4 Evaluation of the Impact of the Temporal Prior Assumptions

The purpose of this sub-experiment is to assess the stabilizing effect of the temporal prior assumptions introduced in Section 3.3.4 on the estimated result. Therefore, we abandon the temporal prior terms in both the estimation of the street surface (3.98) as well as the estimation of the free-space boundary (3.113). To evaluate the impact of the individual priors, we rerun the experiment and omit only one of the priors respectively.

The entries of the confusion matrices printed in the fifth block of Table 4.6 show a significant decrease of the classification accuracy of *non-free-space* pixels stating over 10% to be incorrectly assigned to *free-space* in case both priors are omitted. The reduced spatial accuracy (see Figure 4.21(e) and Figure 4.22(e)) confirms this result.

Comparison of the results obtained from omission of the individual priors reveals that omission of the surface prior causes fewer miss-classifications in total than omission of the boundary prior. Further, while omission of the surface prior causes mainly *free-space* pixels to be miss-classified, omission of the boundary prior mainly affects the assignment of *non-free-space* pixels, which pretty much conforms to the result obtained from omission of both terms.

In the following, we discuss the insights obtained from visual examination in detail.

**Visual Examination of the Impact of the Temporal Boundary Prior:** As expected, omission of the temporal boundary prior causes a considerable drop of the temporal smoothness of the estimated free-space boundary, i.e. the boundary spline seems shaky and volatile. Additionally, the boundary spline tends to overshoot low obstacles, which explains the high amount of incorrectly assigned *non-free-space* pixels.

We explain this behavior in the following way. Without the restraining influence of the prior, the spline may immediately react to erroneous height estimations which blur the elevation discontinuity of the obstacle border and, thus, overshoot the obstacle borders. In case the elevation behind the obstacle matches the currently estimated street surface, e.g. in case of small traffic isles (Figure 4.17, bottom row, left) or thin median strips (Figure 4.17, center row, left), the spline may stay in the wrong position in the subsequent frames, while the obstacle's height measurements are classified as outliers. This effect also significantly intensifies the cropping of obstacle edges which we describe in more detail when discussing the challenges and limitations in Section 4.3.6.4.

**Visual Examination of the Impact of the Temporal Surface Prior:** Visual examination with respect to the omission of the temporal surface prior reveals that the overall result appears very similar to the reference result. The major difference is given by the effect of occasional occurring stereo errors. Since the restraining influence of the prior is gone, the estimated surface reacts stronger to erroneous height measurements. As a consequence, the estimated surface locally significantly differs from the actual street surface causing the local free-space region to be classified as *non-free-space*.

In case both priors are omitted, the overshooting effect dominates causing the lower percentage of miss-classified *free-space* pixels.

#### 4.3.4.5 Evaluation of the Impact of the Spatial Prior Assumptions

This section discusses the impact of the spatial boundary prior for the estimation of the free-space boundary. Similar to the temporal prior in the preceding sub-experiment, the spatial boundary prior was omitted. The respective confusion matrix entries in Table 4.6 show a considerable drop in the class accuracies which is only slightly smaller than the one observed from omission of the temporal prior. The error statistics of the estimated boundary in Figure 4.21(e) indicate that the accuracy drop gets more significant with increasing distance to the camera.

As expected, visual examination reveals a considerable reduction of the boundary spline's smoothness. The higher flexibility intensifies the spline's reaction to erroneous elevation measurements and, thus, reduces the robustness. Particular adverse effects are observable at the border of flat obstacles in the rearmost part of the DEM, where the higher flexibility enables the spline to penetrate and finally overshoot flat obstacles.

As a positive aspect, the effects caused by insufficient flexibility, such as bridging effects and cropping of obstacle edges, are considerably reduced. However, the impact of the positive



aspects is all too small to compensate the negative effects.

Future work may adapt the spatial filtering to the complexity of the current free-space boundary using map information or a similar approach as introduced for adaption of the temporal filtering in Section 3.5.3.1.

Note, the influence of the spatial prior for the estimation of the street surface could not be evaluated in the same way since the prior is mandatory to guarantee a robust estimation even in case the number of valid cells assigned to the class *street* is small.

#### 4.3.4.6 Evaluation of the Impact of the Local Self-Evaluation Step

The sub-experiment presented in Section 4.3.4.4 has proven the temporal boundary prior to be crucial for a proper and temporally stable estimation of the boundary splines. Of course, in case the assumption of static obstacle borders is violated, e.g. in case of moving obstacles or if the spline propagates into newly observed regions, the prior term may be rather contra-productive than helpful. Thus, we introduced the heuristic self-evaluation approach presented in Section 3.5.3.1 in order to detect local violations of the model assumptions along the estimated spline.

In this sub-experiment we abandon the self-evaluation approach from the total workflow to evaluate its impact on the final result. The uncertainty of the additive noise term  ${}^B\mathbf{r}_t$  representing the local deviation from the model, as defined in (3.66), is constantly set to  ${}^B\sigma_{r_{i,t}}^{(\nu+1)} = 0.02 \text{ m}$ .

The second-last row of Table 4.6 states a high rate of incorrect assignments of *free-space* pixels, which additionally causes a considerably lower accuracy of the free-space boundary, as illustrated in Figure 4.21(e) and Figure 4.22(e).

The visual examination of the result sequences shows that the low value of  ${}^B\sigma_{r_{i,t}}^{(\nu+1)}$ , although vital for proper temporal filtering at static obstacles, distinctly restricts the propagation speed of the boundary spline. In many cases, the spline is not able to reach and properly adhere to the border of newly observed obstacles within the first frame of their appearance. The effect increases at higher driving speeds and explains the high amount of incorrectly assigned *free-space* pixels.

Figure 4.27 depicts an example showing a moving obstacle. The comparison of the results obtained with self-evaluation (left column) and without self-evaluation (right column) reveals that the online adaption of  ${}^B\sigma_{r_{i,t}}^{(\nu+1)}$  is crucial to adequately react to local violations of the assumption of static obstacle borders. The higher value of  ${}^B\sigma_{r_{i,t}}^{(\nu+1)}$  in the spline regions drawn in yellow and green color enables the estimated spline to adequately adhere to the cars front which occludes the free-space boundary estimated in the last frame. Further, it allows the spline to faster convert back to the background obstacle (roundabout) when the car has passed. The respective results in the right column show a significantly slower reaction when omitting the self-evaluation step.

#### 4.3.4.7 Evaluation of the Usage of a Naive DEM Computation Approach

The presented algorithm for the computation of the DEM heights, see Section 3.2.2, is quite complex compared to previously used methods. Although we already mentioned the advantages of the utilized approach, as illustrated in Figure 3.9, this sub-experiment yields a comparison to a naive but faster method. In the alternative method, the maximum height of all triangulated points assigned to a cell is taken to represent the cell's elevation value  $h_{ij}$  similar to the approach proposed by [Oniga et al., 2007a]

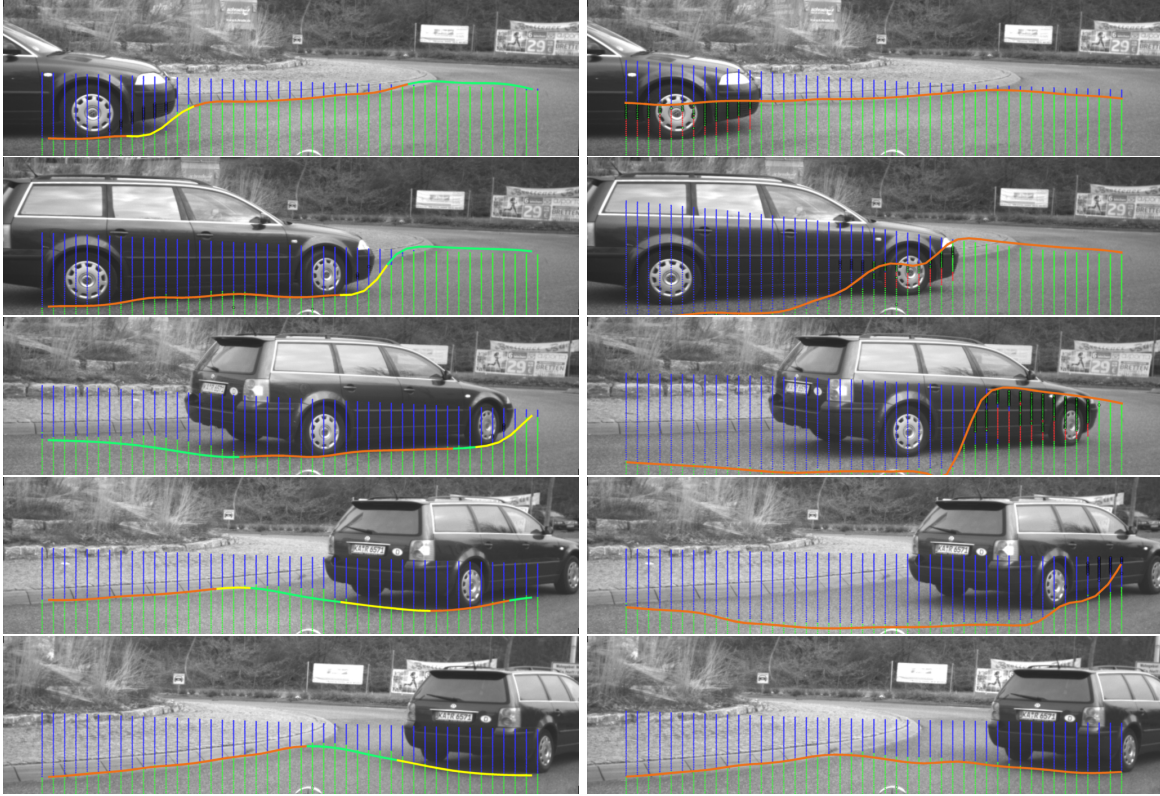


FIGURE 4.27: Example scenario incorporating a moving obstacle. The images compare the estimation result obtained with self-evaluation step (left column) and without self-evaluation step (right column). Color meanings are equivalent to those presented in Figure 4.17. For clarification, we used the color coding proposed in Figure 3.28 to annotate the assignment of the spline sections obtained by the self-evaluation step. Spline regions representing static obstacles are painted in orange color, those representing non stationary obstacles or segments which slightly violate the model assumptions are drawn in yellow color. Invalid parts are drawn in green color.

Results show a significantly lower percentage of correctly assigned *free-space* pixels compared to the reference result. In numbers, nearly 10% of all *free-space* pixels were incorrectly assigned to *non-free-space*. The error statistics in Figure 4.21(e) show that the mean error is two to three times the value obtained in the reference experiment.

There are two factors explaining the high error rates. The first factor is given by the significantly higher amount of erroneousness elevation measurements causing improper estimates of the street surface spline as well as incorrect assignments in the DEM classification step.

For the second factor, consider the scattering effects of the triangulated stereo measurements described in Figure 4.6. Points scattered from an object towards the camera position overrule the maximum in closer cells on the same optical ray. This causes the boundary to be estimated too short with respect to the sensor position.

To effectively employ a DEM for the detection of low obstacles which is computed by the naive method a very constraining model for the obstacle boundaries and street surface needs to be used to cope with the high amount of outliers including occasional scattering effects. Unfortunately, this would significantly limit the ability of the approach to model complex obstacle boundaries.

### 4.3.5 Falsely Detected Objects (*Ghost Objects*)

In addition to erroneous deviations to actual obstacle borders, the estimated free-space boundary may hallucinate completely new obstacles. We refer to such erroneous estimations which are visually not assignable to an actual obstacle as *ghost objects*. Figure 4.30 gives two examples for large sized *ghost objects*.

This section discusses the reasons as well as the frequency of occurrence of *ghost objects*. The evaluation is based on the estimation results obtained on the benchmark dataset using the default parameters.

To simplify the discussion, we divided the detected *ghost objects* in four groups specifying their kind and reason of occurrence. Table 4.7 summarizes the amounts of detections per group as well as the total amount of *ghost objects*. A total of 124 *ghost objects* was detected.

TABLE 4.7: Amounts and types of *ghost objects* detected in the benchmark dataset using the default parameter set.

type of <i>ghost object</i>	amount
minor <i>ghost objects</i>	87
<i>ghost objects</i> from insufficient street model	7
<i>ghost objects</i> from erroneous stereo measurements (illumination)	7
<i>ghost objects</i> from erroneous stereo measurements (road markings)	18
unspecified	5
$\Sigma$	124
$\Sigma$ (without minor <i>ghost objects</i> )	37

We classified 87 of them as ‘minor *ghost objects*’, i.e. *ghost objects* of small size which are only visible in a single frame, as illustrated in Figure 4.28. This kind of *ghost objects* is relatively easy to eliminate by claiming temporal consistency and a minimum extent for detected obstacles. Neglecting the minor *ghost objects*, the total number reduces to 37. We

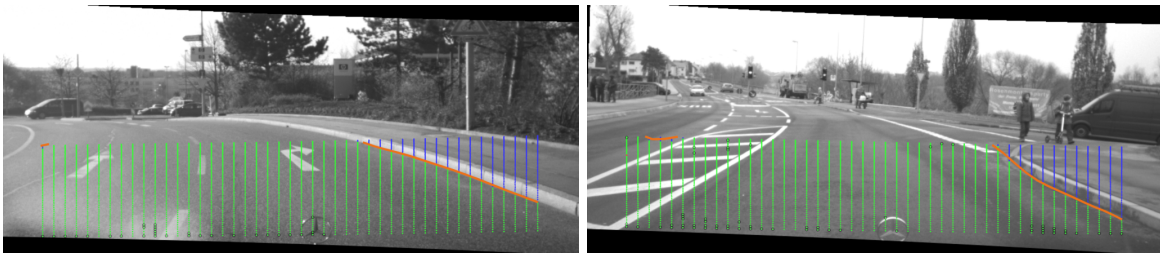


FIGURE 4.28: Two examples for minor *ghost objects* visible in the upper left corner of the DEM’s image projection. This type of *ghost objects* occasionally occur and disappear in the successive frame. Color meanings are equivalent to those presented in Figure 4.17.

assigned most of them to the remaining groups which are described and discussed in the remainder of this section. Five *ghost objects* occur due to individual reasons and are marked as *unspecified* in Table 4.7.

***Ghost Objects from Insufficient Street Model:*** Although the utilized street model is designed to be very flexible, it may be insufficient in special situations. A common scenario causing the *ghost objects* of this group is depicted in Figure 4.29. The edge between two adjacent lane surfaces of a roof-shaped street is located in the outermost corner of the DEM grid.

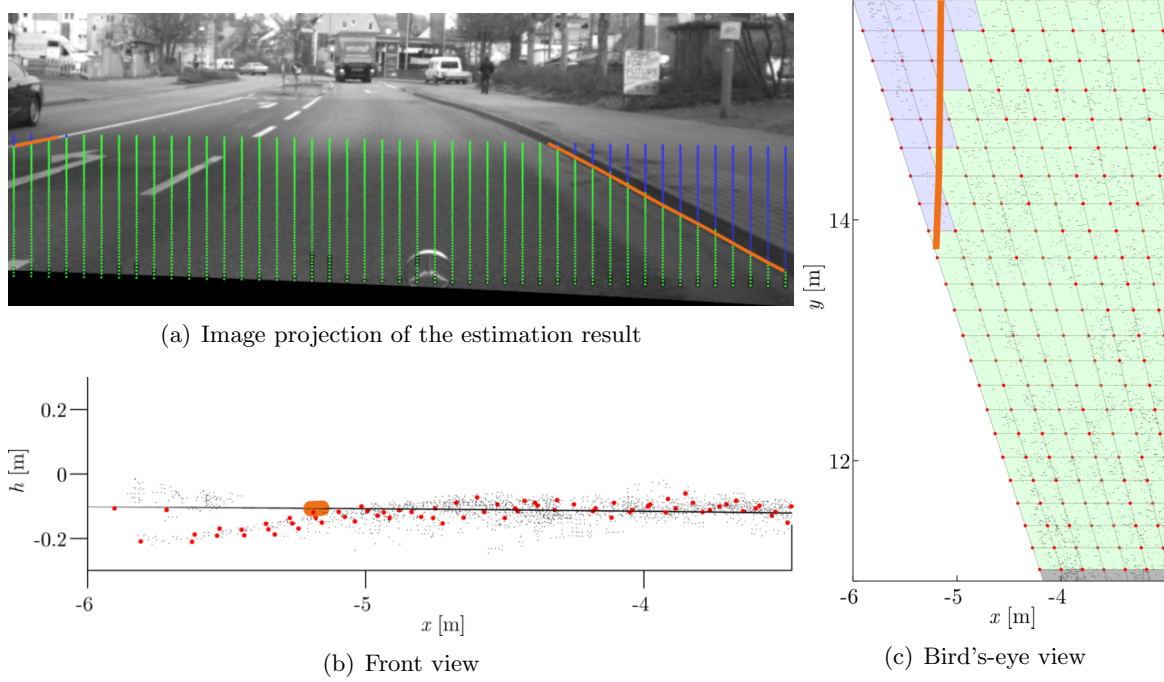


FIGURE 4.29: *Ghost object* caused by insufficient flexibility of the street surface model. The *ghost object* covers the outermost left corner of the DEM grid. Figure (b) depicts a front view of the respective spline section of the estimated street surface. The triangulated point cloud achieved from SGM stereo computation is plotted by black dots. The estimated elevations for the cells of the DEM are plotted by red dots. The orange line marks the boundary spline. In (c), the spline section is depicted from bird's-eye view. Additional color meanings are equivalent to those presented in Figure 4.17.

Figure 4.29(c) shows a bird's-eye view of the respective spline section of the street surface. Although evidently incorrect when comparing to Figure 4.29(a), a small region in the outermost left edge of the DEM grid is labeled as *adjacent* and, thus, separated by the estimated boundary spline. The front view presented in Figure 4.29(b) reveals that, due to the roof shape of the street surface, the measured elevation values of the cells in this region follow a different surface slope than the remaining height measurements. Caused by their relatively low number (less than 10% of the total number of cells in the spline section), their influence with respect to the street surface estimation is too small to compete with the estimate claimed by the remaining observations and the spatial smoothness prior. Additionally, the incorrect estimate is supported by some erroneous height measurements in the corner region.

We observed this kind of *ghost object* 7 times in the benchmark dataset results. The presented example embodies the occurrence with the largest spatial and temporal extend. It appeared for 86 successive frames, which means all the way down to the small traffic isle visible in the background in Figure 4.29(a).

**Ghost Objects from Erroneous Stereo Measurements:** The introduction of stabilizing spatial and temporal priors as well as the explicit modeling of outliers in the classification step allows for accurate results even under the presence of a moderate number of occasional occurring outlier measurements (see Section 4.2.4 for the influence of outliers). However, the quality of the estimated result eventually depends on the reliability of the elevation measurements and, thus, on the reliability of the stereo measurements since we use cameras for data acquisition. Large-area stereo errors, in particular if they are of persistent nature, have the same appearance with respect to the DEM's elevation measurements as real obstacles.



We observed two major groups of persistent large-area errors in stereo measurements obtained from the utilized SGM implementation.

***Ghost Objects* from Erroneous Stereo Measurements Caused by Illumination:**

Extreme illumination conditions such as complex shadow patterns and extreme glare effects on the street or on the windshield cause large-area stereo errors. Under constant driving direction, the illumination effect and therefore the location of the measurement errors may stay persistent over a certain frame period.

Figure 4.30 depicts two examples of illumination effects causing the detection of *ghost objects*. The second row shows side views of the local elevation measurements (red dots) as well as triangulated points obtained from SGM stereo matching (black dots). The plots reveal a significant scattering of the triangulated points towards areas below the street surface which causes a high amount of erroneousness elevation measurements of the respective DEM cells. Consequently, the algorithm classifies these cells as *adjacent* and incorrectly delimits the estimated free-space in a too close position.

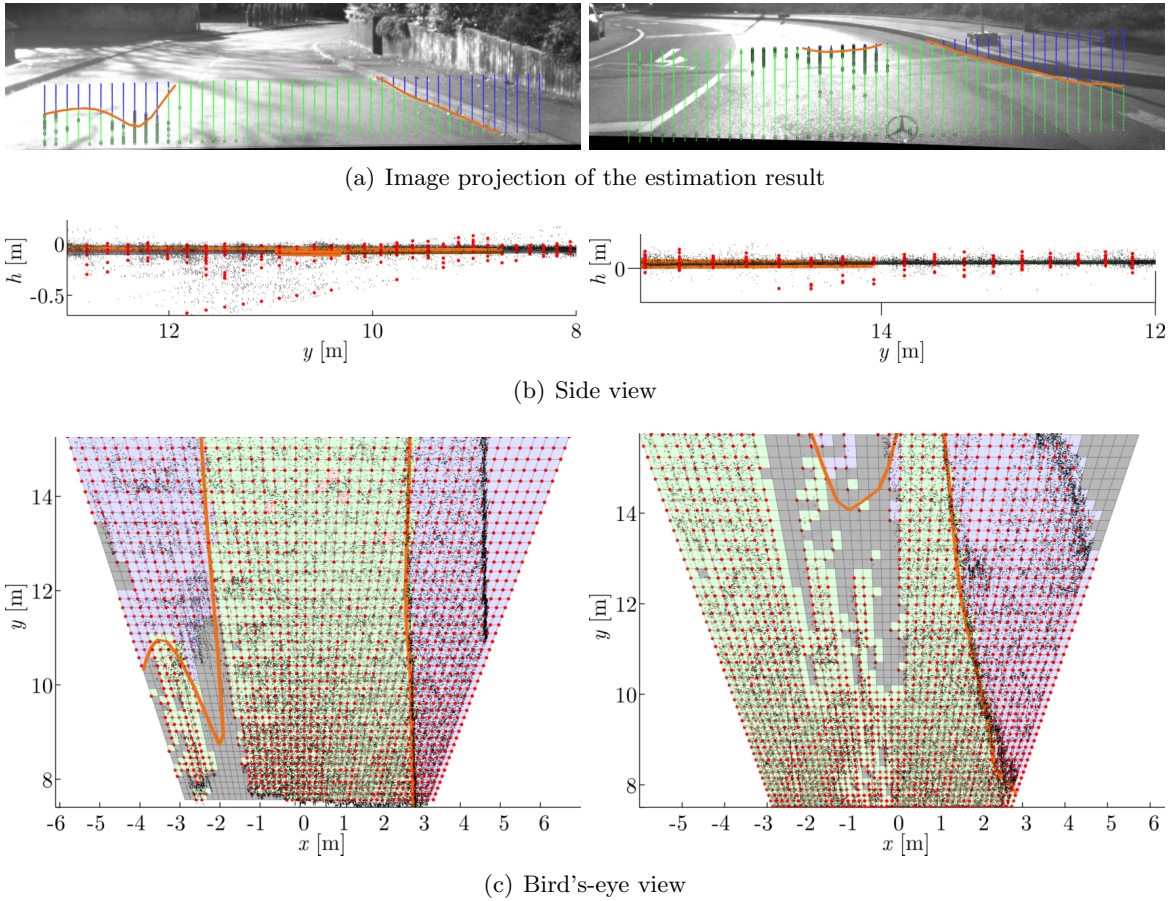


FIGURE 4.30: Two examples illustrating *ghost objects* which are caused by illumination effects. In (a) both example images show a significant glare effect on the road in front of the car. The respective figures in (b) show a side view of the 3d point cloud measurements and estimated cell elevations in a local region around the position of the *ghost objects*. The triangulated point cloud achieved from SGM stereo computation is marked by black dots. The estimated elevations for the cells of the DEM are plotted by red dots. The respective bird's eye views are plotted in (c). Additional color meanings are equivalent to those presented in Figure 4.17.

The black colored regions in the respective bird's-eye views in Figure 4.30(c) show a significant reduction of the number of valid measurements in the regions of the glare effects.

This sparseness of measurements may be used in future work in order to detect regions of unreliable input data.

**Ghost Objects from Erroneous Stereo Measurements Caused by Road Markings:** The second group of large-area errors in the estimated stereo correspondences occasionally occur in the area of street markings, as depicted in Figure 4.31. Figure 4.31(b) shows front views of the respective scenarios which demonstrate the effect on the triangulated point cloud and cell elevations. Comparing the positions of the *ghost objects* and real objects in the front view reveals similar obstacle structures in the local triangulated point clouds. Thus, the usage of additional information, e.g. obtained from the appearance in the image, seems inevitable for the elimination of this kind of *ghost objects* in future work.

The demonstrated effect is considerably stronger when observing almost horizontal structures, which is a known issue for common stereo matching algorithms. We refrain from a more detailed consideration of the reasons for this effect since the evaluation of stereo matching algorithms is not in the focus of this thesis.

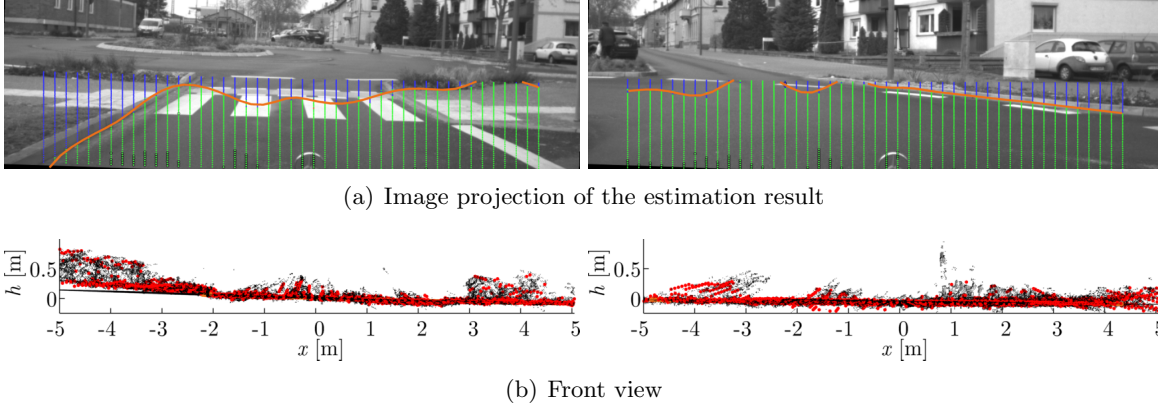


FIGURE 4.31: Examples for *ghost objects* occasionally occurring in the vicinity of road markings. Color meanings are equivalent to those presented in Figure 4.17. The figures in (b) present front views of the respective triangulated point cloud (black dots) and DEM cell elevations (red dots). Comparison to the overlying images shows significant amounts of erroneous point cloud measurements in the vicinity of the *ghost objects*.

### 4.3.6 Discussion of Real-World Results

In the following we present a summary of insights obtained from the experiments on real-world data with respect to the four aspects stated in the very beginning of Section 4.3.

#### 4.3.6.1 Applicability of the Approach in Real-World Traffic Scenarios

**Performance of the approach:** The overall results indicate the ability of the approach to deal with the high complexity and variability of real-world traffic scenarios and to accurately estimate the boundary of the drivable free-space as well as the street surface in the majority of cases.

The approach was tested on challenging scenarios showing complex characteristics of the actual free-space boundary. The scenarios included intersections, driveways, roundabouts, small traffic isles, as well as moving objects such as cars and pedestrians crossing the street. The height of the well detected obstacles reached from several meters (vehicles, walls) to

less than 10 cm (curbs), where the latter dominated in the scenarios. This shows the high flexibility and generality of the presented approach.

With regard to the accuracy of the estimated free-space boundary the experiments revealed that the error of the estimated boundary is in average about the size of the local cell spacing of the DEM.

The approach reaches its limits under the presence of persistent, large-area stereo errors, as described in Section 4.3.5, since they cause an extremely similar appearance in the elevation observations as actual obstacles. Further challenges caused by model limitations are presented in Section 4.3.6.4. These limitations reflect in the outlier statistics of the estimation result. They cause occasional large-scale deviations of the estimated spline to the actual free-space boundary, especially observable in larger distances between 12 m and 15 m to the camera. Of course, the influence of the individual limitations on the applicability of the approach needs to be eventually evaluated in terms of the particular requirements of the respective higher-level applications.

**Computation times:** The major drawback preventing from online testing in the research vehicle is given by the computation times. As stated above, the current implementation is partially written in Matlab which significantly increases the computation times compared to a pure C++ implementation.

The current computation time on a PC equipped with an Intel Core 2 Extreme ( $4 \times 2.53$  GHz) processor and 4 GB RAM amounts to approximately 0.5 s per frame excluding image acquisition and stereo computation. This significantly exceeds the available time for online processing at 25 fps which amounts to 40 ms deducting approximately 10 ms for transfer of the stereo result from the FPGA and 3 ms for the computation of the ego-motion. A detailed analysis of the computation times of the individual submodules would not be meaningful for the current implementation since they highly depend on the respective proportion of C++ and Matlab code.

However, we are convinced that a pure C++ implementation will fulfill real-time requirements. The conviction is caused by the earlier implementation of a similar approach presented in [Siegemund et al., 2010] and [Siegemund et al., 2011]. This approach was designed to detect and reconstruct curbs on the roadside. It is based on a less dynamic environment model but yet shows a high degree of similarity in the overall concept and the majority of the single processing steps. The computation time of the current C++ implementation of this approach amounts to less than 10 ms per frame and, thus, indicates a very probable real-time capability of the approach presented in this thesis. Further, the major components of the approach bear a high degree of parallelization capability, such as the computation of the DEM, the inference in the E-step via the sum-product algorithm and the simultaneous processing of the independent logistic regression tasks in the M-step.

#### 4.3.6.2 Influence of Single Model Components on the Performance of the Overall Approach

The analysis of the model modifications in Section 4.3.4 demonstrates that all considered submodules yield a positive update to the accuracy of the overall approach.

The comparison of alternative, less sophisticated street surface models demonstrates the benefit of the spline model's flexibility. The approach can handle scenarios with complex shaped street surfaces, such as different slopes of the adjacent lanes in intersection scenarios, roads of distinctive roof shape as well as the crossing of a median strip in a highway construction site.

The most significant impact was observed from omitting the spatial prior in the unary term of the CRF. This indicates that the restriction of the area of region competition in the classification step has a stabilizing effect on the estimation result.

The apparent impact of the binary terms is limited to the area of region competition governed by the unary prior term and, thus, less significant. However, the binary terms facilitate uncluttered estimation results in the proximity of obstacle borders improving the accuracy of the estimated boundary spline. Comparison to a Potts model reveals that the effect of the data dependent formulation acts rather as vernier adjustment. Nevertheless, quantitative and visual results prove a small but positive impact on the result's accuracy.

The analysis of the prior assumptions introduced to stabilize the estimation of the model, i.e. street surface and boundary spline, demonstrates temporal filtering to be crucial for accurate estimation results. The omission of the temporal boundary prior causes the boundary spline to be shaky and volatile when observed over successive frames and encourages the spline to overshoot flat obstacles. The omission of the surface prior appeared to be less crucial but yet causes the estimation to be more sensitive to erroneousness elevation data.

Analysis of the spatial filtering of the boundary reveals similar drops of the estimation accuracy in case the spatial prior is omitted. The higher flexibility of the spline intensifies the effects of erroneous elevation measurements especially in larger distances causing a significant reduction of the robustness.

The self evaluation step presented in Section 3.5.3 appeared to be highly valuable. The adaption of the temporal filtering of the boundary spline to the specific local conditions allows the spline to propagate into newly revealed free-space regions as well as to handle moving obstacles. On the other hand, in case of static objects, a less dynamic model can be used to increase the robustness.

Finally, the utilized approach for the computation of the DEM's elevation measurements which considers the path of the optical ray as well as occlusion properties shows a significant update in comparison to a naive approach. It appears to be highly valuable when dealing with extremely noisy data.

#### 4.3.6.3 Transferability of the Insights Achieved on Synthetic Data

Let us now compare the results obtained under real-world conditions to those achieved for the synthetic dataset in Section 4.2. In both cases the mean error behaves proportional to the local cell spacing of the DEM grid. More precisely, the behavior of the mean error with respect to the distance to the camera roughly corresponds to the statistics obtained for a noise level of  $\sigma_d = 0.75$  pel on the synthetic dataset.

Consideration of the respective cumulative error reveals that the results obtained for the experiments on the synthetic data appear to be optimistic in comparison to those achieved under real-world conditions which reveal a higher amount of large scale errors. This is due to the fact that the general conditions for real-world scenarios are more challenging than the synthetic scenarios which are generated based on simplified assumptions. For instance, the complexity of the free-space boundary and the curvature characteristics of the street surface is much higher than the in the simulation. Moving obstacles such as cars or bicyclists additionally violate the model assumptions. Further, illumination effects may perturb the stereo matching results in nearly arbitrary manner.

Regarding the robustness of the approach with respect to outliers, the insights obtained in the discussion of *ghost objects* from erroneousness stereo measurements in Section 4.3.5 reveal the ability of the approach to handle a moderate number of outliers also under real-world conditions. However, the discussion reveals further that the performance is rather bound to



the configuration of the outliers than to their sheer amount, i.e. massed stereo errors and those which are persistent over time appear more challenging than a possibly larger amount scattered over the whole image.

#### 4.3.6.4 Challenges and Limits

The following paragraphs point out and discuss challenging scenarios and limitations of the presented approach.

**Detection Range:** An obvious limitation is given by the restricted detection range. In the presented experiments a maximum range of 16 m was used. This is due to the fact that in larger distances small obstacles, such as curbs, vanish in the noise of the triangulated point cloud. Thus, the range off applications for higher-level modules based on the estimated free-space or surface model is limited to urban or suburban scenarios with moderate driving speeds. Certainly, the approach could be adapted to detect larger objects, such as cars or guardrails, in greater distances. However, this would change the actual scope and is not discussed in this thesis.

**Insufficient Flexibility of the Boundary Spline:** Although more sophisticated than representations based on lines, single polynomial functions, or splines parameterized along the viewing axis, the flexibility of the presented boundary model is limited, all the more since spatial smoothness is explicitly claimed as prior assumption. The model was designed to yield a compromise between flexibility and robustness. Thus, special scenarios appear to be challenging or even intractable since the actual boundary is too complex or severely violates the model assumptions.

In the following, we discuss typical challenging scenarios and associated effects.

**Narrow Objects:** The major challenge is given by very narrow objects, such as the poles of traffic signs and street lights, as depicted in the left image of Figure 4.32(a). Due to the thin shape, the obstacle causes only the elevations of cells in a single longitudinal grid column to differ from the street surface. Since the spline is parameterized along the horizontal axis of the image, its ability to model strictly longitudinal objects is limited and causes incorrect assignments in the lateral neighborhood of the obstacle, as depicted in Figure 4.32(b). Thus, the overall costs of treating the elevation measurements as outlier are

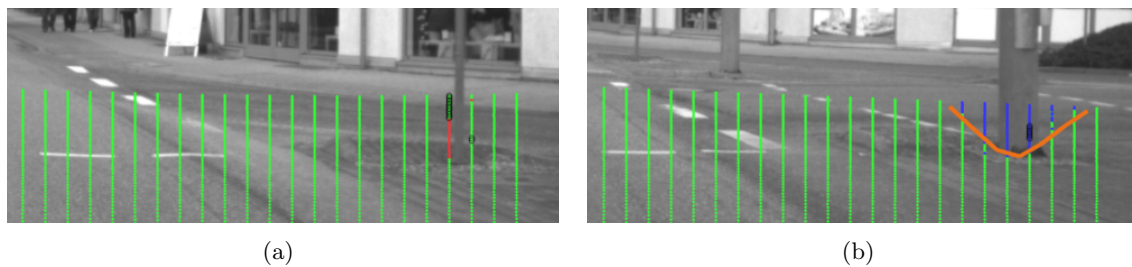


FIGURE 4.32: Two examples for pole-like obstacles. Figure (a) shows a very thin obstacle (traffic sign pole) which is erroneously classified as outlier. Figure (b) depicts a wider pole-like object which is correctly classified as non-free-space by the boundary spline. However, the spline is not flexible enough to correctly reconstruct the vertical borders causing a too wide estimation with increasing distance. Color meanings are equivalent to those presented in Figure 4.17.

smaller than those of modeling the spline around the obstacle.

A possible solution to overcome this limitation would be the combination of the approach with a method specialized for the detection tall obstacles that is able to detect this type of objects, such as the Stixel world presented in [Badino et al., 2009].

However, the occurrence of pole-like obstacles on the driving lane which not separated by a curb is rather rare.

**Bridging Effects:** Another effect caused by insufficient flexibility of the spline model is given by what we call ‘bridging effect’. The free-space between neighboring obstacles is incorrectly classified as non-free-space since the boundary spline ‘bridges’ the gap between the obstacles in case it is too narrow. Of course, the intensity of the effect rises when using a high weight for the spatial filtering for the boundary spline.

Figure 4.33 illustrates examples of the bridging effect. Figure 4.33(a) shows the bridging between a moving obstacle (vehicle) and the newly revealed background obstacle (round-about), i.e. the object that was occluded by the vehicle some frames ago. The preceding frames of the sequence are depicted in Figure 4.27. The bridging of the gap between the obstacles is supported by both the spatial filtering prior as well as the spatial smoothness prior. The latter retards the propagation of the spline into the newly revealed free-space region.

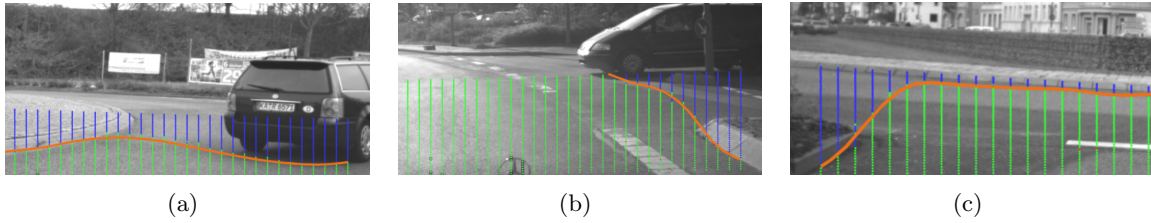


FIGURE 4.33: Three examples demonstrating the bridging effect, i.e. the bridging of gaps between neighboring obstacles by the boundary spline. Figure (a) shows the bridging effect in a scenario including a moving obstacle. The remaining figures depict bridging effects which may occur when the free-space boundary migrates from an occluding obstacle to the occluded obstacle. Color meanings are equivalent to those presented in Figure 4.17.

A more common occurrence of the bridging effect is depicted in Figure 4.33(b) and Figure 4.33(c). The bridging occurs when the free-space boundary migrates from the edge of an occluding obstacle to the occluded obstacle. Due to the spline parametrization and spatial smoothness assumption, the spline is not able to properly model the abrupt discontinuity. Instead, a smoothed result is obtained which bridges the region in between which actually represents free-space.

**Cropping of Obstacle Edges:** The last effect also occurs when the boundary spline reaches the edge of an obstacle. This effect restricts to scenarios where the obstacle is flat and reveals the street surface beyond, i.e. the elevation of the region behind the obstacle matches the street surface, as illustrated in Figure 4.34. This violates the topology assumption which states that there exists only a single transition from free-space to non-free-space in viewing direction.

During the classification step, the spatial unary prior is not strong enough to prevent some of the cells behind the obstacle edge to be classified as *street*. Consequently, the boundary spline tries to encompass this cells and, thus, crops the obstacle edge.

**Illumination and Stereo Artifacts:** The presented approach entirely relies on elevation information obtained from an arbitrary depth map sensor. In our experiments a stereo camera



FIGURE 4.34: Three examples demonstrating the effect of cropping of obstacle edges. The obstacle is flat and reveals the street surface beyond. The boundary spline tries to include the area behind the obstacle into the estimated free-space region causing the obstacle edge to be cropped.

system was employed for data acquisition. Thus, the quality of the estimation result suffers from the same issues as the utilized stereo algorithm. The explicit treatment of outliers in the environment model as well as the sophisticated computation of the DEM allows for a compensation of erroneous stereo measurements up to a certain degree.

However, the consequences of large-area stereo errors caused by illumination issues, as discussed in detail in Section 4.3.5, clearly limit the applicability of the approach in special scenarios, such as strong rain, wet reflective road surfaces, or night scenes without auxiliary equipment. To cope with those challenging scenarios an additional appropriate sensor would be useful.

**Extensive Occlusion of the Street Surface:** Let us now discuss how extensive occlusions of the street surface affect the estimation results obtained on the benchmark dataset.

Section 3.5.3.2 describes the detection and treatment of so called ‘degenerated models’. This term describes estimation results where the percentage of DEM cells assigned to *outlier* exceeds the threshold  $\tau_{\text{outlier}}$  or the percentage of DEM cells classified as *street* falls below the threshold  $\tau_{\text{street}}$ .

The experiment on the benchmark dataset using the default parameter set reveals two occurrences of a degenerated models. Both are caused by extensive occlusions of the street surface by other vehicles. Figure 4.35 shows the respective scenarios by means of the last valid estimation result and the first estimation result after successful reinitialization. In between, the algorithm continuously classifies the result as degenerated stating correctly that there is no reliable free-space detected.



FIGURE 4.35: Two scenarios of the benchmark dataset which caused degenerated models. In both situations, the vehicle occludes extensive parts of the street surface such that the number of DEM cells classified as *street* falls below the threshold  $\tau_{\text{street}}$ . The upper row shows the respective frames providing the last estimation result before the degenerated model was detected. The bottom row shows the respective first frame in which the system was successfully reinitialized, i.e. the model was estimated from scratch without detecting a degenerated model. Color meanings are equivalent to those presented in Figure 4.17.



## Chapter 5

# Conclusion and Outlook

**Conclusion:** In this thesis, we presented and analyzed a novel approach for the detection and reconstruction of street surfaces and boundaries from depth map sequences. The system addresses the scope of driver assistance systems which denies the usage of multiple expensive sensors and motivates the acquisition of the depth maps via a single vehicle mounted stereo camera system. However, the approach is generic in such a way that other sensors providing synchronously captured depth maps, such as time-of-flight cameras, would also be suitable. The focus of the approach lies on the explicit modeling of the street boundary (alias free-space boundary) which represents the first geometric delimiter of the free drivable space in front of the vehicle. This includes even low obstacles, such as curbs. By explicit modeling we mean that the free-space boundary is continuously represented by a parameterized curve in the horizontal plane.

The major challenges lie in the high geometric variability of the obstacles and their possibly low height occurrence compared to the measurement noise of stereo vision depth maps. The flexible models based on B-splines that are used to represent the free-space boundary and the street surface enable the approach to model the visible horizontal silhouette of nearly arbitrary obstacles even in case of undulating street surfaces. Prior assumptions about the spatial and temporal rigidity of the boundary are involved in the estimation process to permit a stable and temporally consistent result despite the high flexibility of the model.

In a precomputation step, a DEM is computed from the depth information acquired from stereo matching in order to model the elevation characteristics in front of the vehicle. For the computation of the discrete elevation entries of the DEM, a sophisticated approach is used which traces the optical ray's path and assesses occlusion and free-space information rather than solely the triangulated point position. This proved crucial to deal with the measurement noise and outliers in the data.

An iterative approach is performed to estimate the parameters of the spline model from the height information stored in the DEM. The first step of each iteration utilizes a CRF to derive a probabilistic statement for each cell of the DEM's grid of whether the cell belongs to the drivable street region or the non-drivable region. Additionally, outliers are explicitly modeled. In the second step, the parameters of the street surface are estimated from the elevation measurements whereas the influence of the single observations is weighted by the probability of the respective cell to belong to the street region. Further, the free-space boundary is estimated as most probable separation of drivable and non-drivable regions with respect to the viewing direction. Therefore, in the manner of logistic regression a classification model is fitted into the probabilistic assignment information of the DEM cells. Put simply, for each longitudinal column of the DEM grid, a logistic function is estimated which continuously describes the local assignment probabilities. The free-space boundary is then estimated from the inflection

points of the logistic functions which represent the local decision boundary between drivable and non-drivable region. The logistic regression model is in turn considered as spatial prior in the unary terms of the CRF in the subsequent iteration in order to restrict the area of region competition and, thus, to prevent the boundary from overshooting obstacles. In a postprocessing step for each iteration, a self-diagnostic routine is performed which assesses the local validity of the estimated boundary model in order to adapt the temporal filtering or to trigger a reinitialization if necessary. This proved to be essential to deal with moving obstacles and to allow the boundary to propagate into previously occluded free-space regions.

Experiments on both synthetic and real-world scenarios revealed that the system is able to accurately detect and model the free-space boundary in the considered distance range of up to 16 m to the camera. This includes even low obstacles down to a height of 0.1 m as well as obstacles of negative height. The system is able to deal with the high measurement noise and with, theoretically, up to 20% outliers in the data. The experiments further prove the applicability of the approach under real-world conditions and demonstrate the impact of the approach's single components on the result. Regarding the components, the sophisticated approach for DEM computation, the usage of a flexible street surface model and the consideration of the logistic regression model in the unary term of the CRF appear to be most crucial. To the knowledge of the author, there is recently now other vision based system yielding comparable performance at a similar spectrum of allowed obstacle shapes and heights as well as curvature characteristics of the street surface.

However, the system reaches its limits for large-area stereo errors, especially if they appear persistent over a couple of successive frames which causes the erroneous detection of faked obstacles (*ghost objects*). Furthermore, although the approach utilizes a very flexible model to represent the free-space boundary, it proved insufficient for very narrow obstacles, such as poles, and tends to bridge gaps between obstacles standing close together.

## Outlook:

**Real-Time Capable Implementation:** The online application of the system in a test vehicle essentially requires the assessment of its real-time capability based on a full C++ implementation (or based on an alternative programming language of similar efficiency). However, as discussed in Section 4.3.6.1 we are optimistic that an appropriate implementation will match the respective computation time requirements considering the low computation time of a C++ implementation of a predecessor version and the parallelization capabilities of the approach.

**Consideration of Appearance Information:** In the presented experiments we utilize a stereo camera system to monitor the vehicle's environment. So far, only the geometric information derived from stereo matching was utilized to solve the addressed task. Additional appearance based information, such as color, texture and line structures should be considered in future versions. The fusion of geometric and appearance information grants new opportunities to overcome some limitations of the system and, in particular, to significantly extend the detection range. Similar to [Michalke et al., 2010], a possible strategy could be to learn the appearance of street and obstacles in close range based on the results of the presented system and to trace the street boundary in more distant regions.

**Probabilistic Model for Self Evaluation:** The presented procedure for the self evaluation of the estimated result (processing step (VI)) which is employed to detect and treat

model violations purely relies on a heuristic. The development of a more sophisticated approach is a worthwhile goal for future work all the more because the experiments have demonstrated that this step is crucial to deal with real-world conditions especially under the presence moving obstacles. Similar to the classification of the voxels for the elevation estimation in Section 3.2.2.2, a linear chain CRF could be used to determine the states of the local boundary segments, whereas the possible states are given by the three cases presented in Section 3.5.3.1 indicating if the segment represents a stationary object, or slightly or even significantly violates the model. The respective potential functions could be learned from a manual annotated test set.

**Alternative Environment Models:** The utilized environment model based on B-spline representations for both street surface and boundary was chosen to yield a compromise of flexibility and robustness. However, alternative models could be more suitable in special situations and, in particular, if special scenarios are addressed. In [Siegemund et al., 2010] and [Siegemund et al., 2011], a less flexible model is utilized where the street boundary is defined to laterally separate the drivable region (classes *street* and *outlier*) from the non-drivable region (class *adjacent*) along a third order polynomial with respect to the vehicle’s longitudinal axis (see Figure 5.1). This model is designed to represent curbs and proved to be very robust but limits the approach to obstacles being collateral to the vehicles driving corridor.

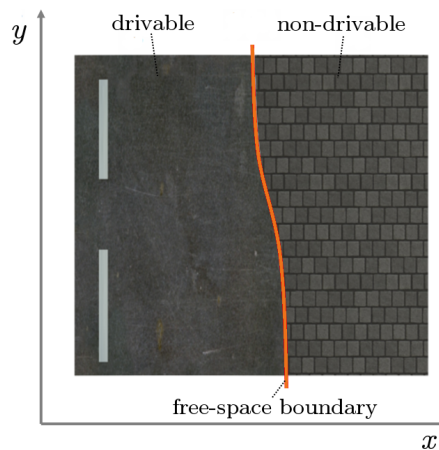
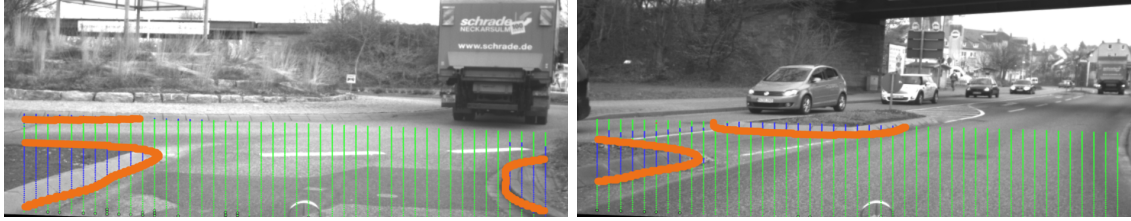


FIGURE 5.1: Schematic illustration of the less flexible model employed in [Siegemund et al., 2010] and [Siegemund et al., 2011]. This model was designed to represent curbs being collateral to the vehicles driving corridor. The free-space boundary which is plotted by an orange line is defined to laterally separate the drivable region from the non-drivable region along a third order polynomial with respect to the vehicle’s longitudinal axis.

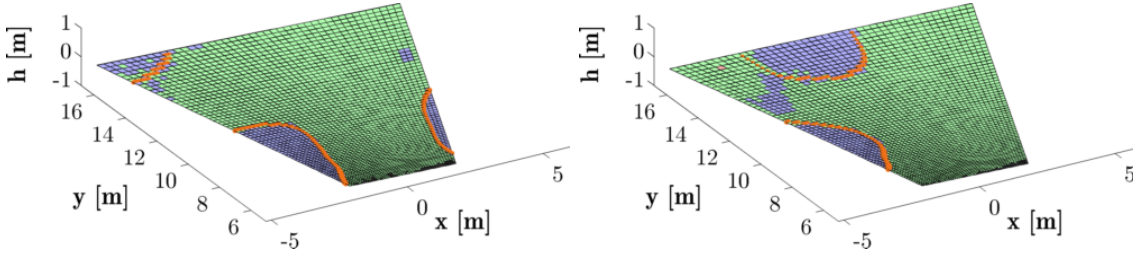
Similarly, one can think of a even more flexible model than the one presented in this thesis. This might help to overcome limitations in the shape and topology of obstacles the system is able to model. Remind that the logistic regression model which is used to represent the local a posterior probabilities of the class *adjacent* implies a topology assumption that allows only for a single transition from the drivable region to the non-drivable region in viewing direction. This assumption proved to stabilize the result but prevents the system to detect drivable regions behind the first obstacle which might be worthwhile for instance in intersection scenarios to model the crossing lane. In a preliminary experiment, we replace the logistic regression model with an alternative classifier. We use the implementation of [Roscher et al., 2012] based on the theory of Import Vector Machines (IVM) introduced by [Zhu and Hastie, 2005], i.e. a probabilistic kernel based classifier. As for the logistic regression model,



the feature domain of the classifier is given by the horizontal space. The free-space boundary is embodied by the decision boundary between the class *adjacent* and the remaining classes. Preliminary results on intersection scenarios are depicted in Figure 5.2. Due to the kernel based representation, the classifier does, in theory, not imply any topology assumption which enables the system to model multiple transitions between drivable and non-drivable regions in viewing direction. However, some work needs to be done to stabilize this extreme flexible model over time and to reduce the additional computational effort which denies a real-time capable solution so far.



(a) Image projection



(b) Perspective view

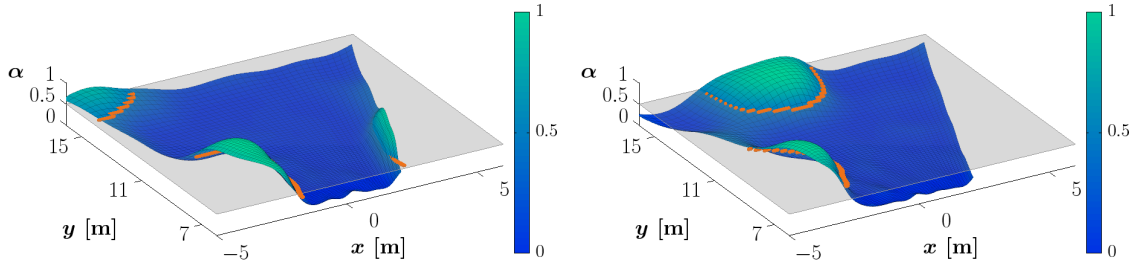
(c) Local a posteriori probabilities of class *adjacent*

FIGURE 5.2: Two examples (one per column) demonstrating preliminary results for the usage of an alternative environment model based on IVM. Figure (a) depicts the image projection of the result. Figure (b) provides a perspective view on the estimated street surface. Color meanings are equivalent to those presented in Figure 4.17. Figure (c) illustrates the local a posteriori probabilities of class *adjacent* obtained from the IVM classifier, which are used as weighting factors  $\alpha$  in the mixture distribution (3.21). For comparison, examples for the a posterior probabilities obtained for the logistic regression model were presented in Figure 3.17. Note that this alternative model in theory does not entail any topology assumption. In particular it is able to model multiple transitions between drivable and non-drivable regions in viewing direction.



## Appendix A

# Construction of the Digital Elevation Map

In this section we describe how to compute the cell center coordinates of a grid that satisfies the requirements defined in Section 3.2.2.1, i.e. the cell centers are regular in the image columns and the cell spacings in both dimensions coincide  $d_{x_j} = d_{y_j}$ , as illustrated in Figure 3.6(a). Further, the assignment of depth measurements to the respective cells is demonstrated.

**Computation of Cell Center Coordinates:** The grid structure is completely defined by the following parameters:

- The distance  $y_{\min} = y_{i1} - \frac{d_{y_1}}{2}, i = 1, \dots, I$ , of the grid to the origin in longitudinal direction.
- The image column coordinate of the left grid boundary  $u_{\min}$ .
- The spacing in the image columns  $d_u$ .
- The principle distance  $c$ .
- The number of cells in lateral and longitudinal direction, i.e.  $I$  and  $J$ .

By definition, all cell centers of the same longitudinal grid column  $i$  lie on the same horizontal projection ray, i.e. project to the same image column  $u_i := u_{\min} + \frac{1}{2}d_u + (i-1)d_u$ . Assuming the local cell spacings  $d_{y_j}$  to be known, the cell center coordinates can be computed by

$$y_{ij} = y_{\min} + \sum_{j'=1}^{j-1} d_{y_{j'}} + \frac{d_{y_j}}{2} \quad (\text{A.1})$$

$$x_{ij} = y_{ij} \frac{u_i}{c}. \quad (\text{A.2})$$

Since on the other hand the cell spacings depend on the  $y_{ij}$  coordinates

$$d_{y_j} = d_{x_j} = x_{ij} - x_{i-1j} = \frac{d_u}{c} y_{ij}, \quad (\text{A.3})$$

we can only derive a direct formulation for the spacings  $d_{y_1}$  of the first grid row. To obtain the remainder of the cell spacings, let us consider a recursive formulation of (A.1)

$$y_{ij} = y_{ij-1} + \frac{d_{y_{j-1}}}{2} + \frac{d_{y_j}}{2}. \quad (\text{A.4})$$

Substituting (A.3) for  $y_{ij}$  and  $y_{ij-1}$  yields a recursive formulation for the cell spacings

$$d_{y_j} = d_{y_{j-1}} + \frac{d_u}{c} \left( \frac{d_{y_{j-1}}}{2} + \frac{d_{y_j}}{2} \right) \quad (\text{A.5})$$

$$= a d_{y_{j-1}} = a^{j-1} d_{y_1}, \quad (\text{A.6})$$

with the shortcut

$$a := \frac{1 + \frac{d_u}{2c}}{1 - \frac{d_u}{2c}}. \quad (\text{A.7})$$

In a similar way we derive an explicit formulation for the spacing  $d_{y_1}$  from  $y_1 = y_{\min} + \frac{d_{y_1}}{2}$  by substituting (A.3) for  $y_1$  yielding

$$d_{y_1} = \frac{1}{\frac{c}{d_u} - \frac{1}{2}} y_{\min}. \quad (\text{A.8})$$

Finally, substituting (A.6) and (A.8) in (A.1) results in

$$y_{ij} = y_{\min} + d_{y_1} \sum_{j'=1}^{j-1} a^{j'-1} + d_{y_1} \frac{a^{j-1}}{2} \quad (\text{A.9})$$

$$= y_{\min} + d_{y_1} \frac{a^{j-1} - 1}{a - 1} + d_{y_1} \frac{a^{j-1}}{2}, \quad (\text{A.10})$$

where we used the property of a partial sum of a geometric series

$$\sum_{j=0}^{J-1} a^j = \sum_{j=1}^J a^{j-1} = \frac{a^J - 1}{a - 1}. \quad (\text{A.11})$$

**Cell Assignment of Depth Measurements:** The assignment of a certain depth measurements  $y$  in image column  $u$  to the respective lateral index  $i$  is straight forward

$$i = \left\lfloor \frac{u - u_{\min}}{d_u} + 1 \right\rfloor. \quad (\text{A.12})$$

The assignment to the longitudinal cell index  $j$  can either be done by online evaluating the inverse function of (A.10) omitting the last term

$$j = \left\lceil \log_a \left( \frac{y - y_{\min}}{d_{y_1} \left( \frac{1}{a-1} \right)} + 1 \right) + 1 \right\rceil, \quad (\text{A.13})$$

or more efficient by creating a lookup table of high resolution.

## Appendix B

# Treatment of Sample Position Uncertainty for the Estimation of the Street Surface Spline

In Section 3.3.2.1 we discussed the advantages of treating the sample positions  $\mathbf{x}_{ij} = [x_{ij}, y_{ij}]^T$  as certain when estimating the street surface from the DEM's height observations  $h(x_{ij}, y_{ij})$ . However, the sample positions are defined by the horizontal cell centers which act as representatives of the whole cell and are therefore uncertain due to the discretization of the DEM grid. The functional model for the estimation of the control point heights  $h_n$  from the observations  $\mathbf{X} = [\dots, x_{ij}, y_{ij}, h_{ij}, \dots]^T$  reads as

$$\begin{bmatrix} \vdots \\ h(x_{ij}, y_{ij}) + v_{h_{ij}} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \sum_{n=1}^N \mathcal{B}_{n,2}(x_{ij} + v_{x_{ij}}, y_{ij} + v_{y_{ij}}) h_n \\ \vdots \end{bmatrix}, \quad (\text{B.1})$$

with the sample residuals  $[\dots, v_{x_{ij}}, v_{y_{ij}}, v_{h_{ij}}, \dots]^T \sim \mathcal{N}(\mathbf{0}, \Sigma_{XX})$ . The covariance matrix is given by the diagonal matrix  $\Sigma_{XX} = \text{Diag}([\dots, \sigma_{x_{ij}}^2, \sigma_{y_{ij}}^2, \sigma_{h_{ij}}^2, \dots]^T)$  holding the horizontal and vertical precision of the DEM measurements defined in (3.13), (3.14) and (3.15). Since (B.1) cannot be converted into an explicit function with respect to the observations, the estimation of the parameters would require an expensive iterative procedure involving multiple recomputations of the spline weights.

This section describes an approximative approach for the treatment of horizontal uncertainties of the sample positions in order to allow for an efficient estimation in a linear model resulting in a significant reduction in computational effort.

In order to obtain a function that is explicit with respect to the observations, we replace the right sides of (B.1) by the first-order approximation

$$\begin{aligned} & \sum_{n=1}^N \mathcal{B}_{n,2}(x_{ij} + v_{x_{ij}}, y_{ij} + v_{y_{ij}}) h_n \\ & \approx \sum_{n=1}^N \mathcal{B}_{n,2}(x_{ij}, y_{ij}) h_n + v_{x_{ij}} \frac{\partial}{\partial x} h(x_{ij}, y_{ij}) + v_{y_{ij}} \frac{\partial}{\partial y} h(x_{ij}, y_{ij}). \end{aligned} \quad (\text{B.2})$$

The local derivatives embodying the last both terms are approximated from the surface estimated the previous frame. Let the respective horizontal positions of the sample positions

with respect to the coordinate system of the previous frame be denoted by  $[x_{ij}^{\text{old}}, y_{ij}^{\text{old}}]^\top$  and the respective estimated control points by  $\hat{h}_n^{\text{old}}$ . Using the simplified notation for the derivatives of B-spline basis functions introduced in (2.117), the derivatives are given by

$$\frac{\partial}{\partial x} h(x_{ij}, y_{ij}) = \sum_{n=1}^N \mathcal{B}_{n,2}^{(1,0)}(x_{ij}^{\text{old}}, y_{ij}^{\text{old}}) \hat{h}_n^{\text{old}} \quad (\text{B.3})$$

$$\frac{\partial}{\partial y} h(x_{ij}, y_{ij}) = \sum_{n=1}^N \mathcal{B}_{n,2}^{(0,1)}(x_{ij}^{\text{old}}, y_{ij}^{\text{old}}) \hat{h}_n^{\text{old}}. \quad (\text{B.4})$$

Substituting the approximation (B.2) in the initial model (B.1) yields the approximated linear model

$$\begin{bmatrix} \vdots \\ h(x_{ij}, y_{ij}) \\ \vdots \end{bmatrix} + \underbrace{\begin{bmatrix} \vdots \\ v_{h_{ij}} - v_{x_{ij}} \frac{\partial}{\partial x} h(x_{ij}, y_{ij}) - v_{y_{ij}} \frac{\partial}{\partial y} h(x_{ij}, y_{ij}) \\ \vdots \end{bmatrix}}_{\mathbf{v}} = \sum_{n=1}^N \mathcal{B}_{n,2}(x, y) h_n, \quad (\text{B.5})$$

with  $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Sigma_{hh})$  and the sample position treated as certain. The precision of  $\mathbf{v}$  is given by the diagonal matrix

$$\Sigma_{hh} = J_h \Sigma_{XX} J_h^\top \quad (\text{B.6})$$

holding the height precision of the single samples  $S\sigma_{h_{ij}}^2$  as diagonal elements. The Jacobian  $J_h$  is given by the block diagonal matrix

$$J_h = \begin{bmatrix} \ddots & & & \\ & -\frac{\partial}{\partial x} h(x_{ij}, y_{ij}) & -\frac{\partial}{\partial y} h(x_{ij}, y_{ij}) & 1 \\ & & & \\ & & \ddots & \end{bmatrix}. \quad (\text{B.7})$$

## Appendix C

# Error propagation for the Prediction Step

The prediction step discussed in Section 3.4 involves the transformation of the previously estimated environment model into the current elevation map system. This comprises in detail the transformation of a set of samples of the previously estimated surface (Section 3.4.1.1) as well as the transformation of the control points of the previously estimated boundary (Section 3.4.1.2). In the following, we describe the respective error propagation based on the precisions  $\Sigma_{u_t u_t}$ ,  ${}^S\Sigma_{\hat{\theta}_{t-1}\hat{\theta}_{t-1}}$  and  ${}^B\Sigma_{\hat{\theta}_{t-1}\hat{\theta}_{t-1}}$  of the current ego-motion parameters  $\mathbf{u}_t$  and previously estimated model parameters  ${}^S\hat{\boldsymbol{\theta}}_{t-1}$  and  ${}^B\hat{\boldsymbol{\theta}}_{t-1}$ . Section C.1 describes the derivation of the precision  $\Sigma_{X_t^- X_t^-}^-$  of the predicted 3d surface samples. Section C.2 comprises the quite similar derivation of the precision  ${}^B\Sigma_{\theta_t^- \theta_t^-}$  of the 2d control points predicted from the previously estimated boundary spline.

### C.1 Error propagation for the Prediction of the Street Surface

As defined in Section 3.4.1.1, the previously estimated street surface is sampled at the horizontal positions  $\mathbf{x}_q = [x_q, y_q]^\top$ , with  $q = 1, \dots, Q$ , in the previous elevation map system  $S_{\ell_{t-1}}$ . By means of the definition of the street surface model (3.23) the 3d-surface samples read as

$$\mathbf{X}_{q,t-1} = {}^S\mathcal{X}_q({}^S\hat{\boldsymbol{\theta}}_{t-1}) \quad (\text{C.1})$$

using the assistance function

$${}^S\mathcal{X}_q({}^S\boldsymbol{\theta}) := \begin{bmatrix} x_q \\ y_q \\ {}^Sf(\mathbf{x}_q, {}^S\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \\ [\mathcal{B}_{1,2}(x_q, y_q), \dots, \mathcal{B}_{N,2}(x_q, y_q)] {}^S\boldsymbol{\theta} \end{bmatrix}. \quad (\text{C.2})$$

The transformation into the current elevation map system  $S_{\ell_t}$  based on the ego-motion information introduced in Section 3.2.3 yields the predicted samples

$$\mathbf{X}_{q,t} = {}^S\mathcal{M}_q(\mathbf{u}_t, {}^S\hat{\boldsymbol{\theta}}_{t-1}), \quad (\text{C.3})$$

where the prediction function is defined by

$${}^S\mathcal{M}_q(\mathbf{u}, {}^S\boldsymbol{\theta}) := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} {}^t\mathbf{M}_{t-1}(\mathbf{u}) \begin{bmatrix} {}^S\mathcal{X}_q({}^S\boldsymbol{\theta}) \\ 1 \end{bmatrix}. \quad (\text{C.4})$$

The ego-motion matrix  ${}^t\mathbf{M}_{t-1}(\mathbf{u})$  defines the coordinate transformation from the previous elevation maps system into the current system caused by the vehicles motion. The matrix is governed by the parameter vector  $\mathbf{u}$  which holds the three parameters of rotation  ${}^R\mathbf{u}_t$  and translation  ${}^T\mathbf{u}_t$ . The exact definition is given by (3.19).

Using linear error propagation, the precision of the predicted samples is given by the covariance matrix

$$\Sigma_{X_t^- X_t^-} = {}^S J_u \begin{bmatrix} {}^S \Sigma_{\hat{\theta}_{t-1} \hat{\theta}_{t-1}} & 0 \\ 0 & \Sigma_{u_t u_t} \end{bmatrix} {}^S J_u^\top. \quad (\text{C.5})$$

The Jacobian is defined by the partial derivatives of  ${}^S \mathcal{M}_q$  with respect to  $\mathbf{u}$  and  ${}^S \boldsymbol{\theta}$  evaluated at the currently determined ego-motion parameters  $\mathbf{u}_t$  and previously estimated street surface parameters  ${}^S \hat{\boldsymbol{\theta}}_{t-1}$

$${}^S J_u = \begin{bmatrix} \vdots & \vdots \\ \left. \frac{\partial {}^S \mathcal{M}_q}{\partial \mathbf{u}} \right|_{u=u_t, {}^S \theta = {}^S \hat{\theta}_{t-1}} & \left. \frac{\partial {}^S \mathcal{M}_q}{\partial {}^S \boldsymbol{\theta}} \right|_{u=u_t, {}^S \theta = {}^S \hat{\theta}_{t-1}} \\ \vdots & \vdots \end{bmatrix}. \quad (\text{C.6})$$

To obtain this partial derivatives, we expand the prediction function (C.4) in terms of the rotation part  ${}^R\mathbf{u}$  and translation part  ${}^T\mathbf{u}$  of the ego-motion parameters using definition (3.19)

$${}^S \mathcal{M}_q(\mathbf{u}, {}^S \boldsymbol{\theta}) = R({}^R\mathbf{u}) {}^S \mathcal{X}_q({}^S \boldsymbol{\theta}) + {}^T\mathbf{u}. \quad (\text{C.7})$$

From this, we directly obtain the partial derivatives with respect to the street surface parameters

$$\frac{\partial {}^S \mathcal{M}_q}{\partial {}^S \boldsymbol{\theta}} = \frac{\partial {}^S \mathcal{M}_q}{\partial {}^S \mathcal{X}_q} \frac{\partial {}^S \mathcal{X}_q}{\partial {}^S \boldsymbol{\theta}} = R({}^R\mathbf{u}) \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ \mathcal{B}_{1,2}(x_q, y_q) & \dots & \mathcal{B}_{N,2}(x_q, y_q) \end{bmatrix} \quad (\text{C.8})$$

and with respect to the translation parameters

$$\frac{\partial {}^S \mathcal{M}_q}{\partial {}^T\mathbf{u}} = I_3. \quad (\text{C.9})$$

The partial derivative with respect to the rotation parameters  ${}^R\mathbf{u}$  requires some more effort due to the nonlinearity. The idea is to reformulate the rotation in terms of an approximate rotation  $R({}^R\mathbf{u}^{(0)})$  and a small update  $R(\Delta {}^R\mathbf{u})$

$$R({}^R\mathbf{u}) = R({}^R\mathbf{u}^{(0)}) R(\Delta {}^R\mathbf{u}) \quad (\text{C.10})$$

$$\approx R({}^R\mathbf{u}^{(0)}) (I_3 + S(\Delta {}^R\mathbf{u})), \quad (\text{C.11})$$

whereas the update is approximated by means of the skew symmetric matrix

$$S(\Delta {}^R\mathbf{u}) = \begin{bmatrix} 0 & -[\Delta {}^R\mathbf{u}]_3 & [\Delta {}^R\mathbf{u}]_2 \\ [\Delta {}^R\mathbf{u}]_3 & 0 & -[\Delta {}^R\mathbf{u}]_1 \\ -[\Delta {}^R\mathbf{u}]_2 & [\Delta {}^R\mathbf{u}]_1 & 0 \end{bmatrix}. \quad (\text{C.12})$$

For a detailed discussion of this approximation we refer to [McGlone et al., 2004, chapter 2.1.2.9].

Inserting the approximation into (C.7) yields the approximated prediction function

$$\begin{aligned} {}^S\overline{\mathcal{M}}_q \left( \Delta {}^R\mathbf{u}, {}^T\mathbf{u}^{(0)}, {}^T\mathbf{u}, {}^S\boldsymbol{\theta} \right) \\ = R \left( {}^R\mathbf{u}^{(0)} \right) \left( I_3 + S \left( \Delta {}^R\mathbf{u} \right) \right) {}^S\mathcal{X}_q \left( {}^S\boldsymbol{\theta} \right) + {}^T\mathbf{u} \end{aligned} \quad (\text{C.13})$$

$$= R \left( {}^R\mathbf{u}^{(0)} \right) {}^S\mathcal{X}_q \left( {}^S\boldsymbol{\theta} \right) + R \left( {}^R\mathbf{u}^{(0)} \right) S \left( \Delta {}^R\mathbf{u} \right) {}^S\mathcal{X}_q \left( {}^S\boldsymbol{\theta} \right) + {}^T\mathbf{u} \quad (\text{C.14})$$

$$= R \left( {}^R\mathbf{u}^{(0)} \right) {}^S\mathcal{X}_q \left( {}^S\boldsymbol{\theta} \right) - R \left( {}^R\mathbf{u}^{(0)} \right) S \left( {}^S\mathcal{X}_q \left( {}^S\boldsymbol{\theta} \right) \right) \Delta {}^R\mathbf{u} + {}^T\mathbf{u}, \quad (\text{C.15})$$

whereas the last transformation step follows from the identity  $S(a)b = -S(b)a$ .

From (C.15) the partial derivation with respect to the rotation parameters follows by the approximation

$$\left. \frac{\partial {}^S\mathcal{M}_q \left( \mathbf{u}, {}^S\boldsymbol{\theta} \right)}{\partial {}^R\mathbf{u}} \right|_{R_{\mathbf{u}} = R_{\mathbf{u}}^{(0)}} \approx \left. \frac{\partial {}^S\overline{\mathcal{M}}_q \left( \Delta {}^R\mathbf{u}, {}^T\mathbf{u}^{(0)}, {}^T\mathbf{u}, {}^S\boldsymbol{\theta} \right)}{\partial \Delta {}^R\mathbf{u}} \right|_{\Delta {}^R\mathbf{u} = 0} \quad (\text{C.16})$$

$$= -R \left( {}^R\mathbf{u}^{(0)} \right) S \left( {}^S\mathcal{X}_q \left( {}^S\boldsymbol{\theta} \right) \right) \quad (\text{C.17})$$

Finally, we achieve the required entries of the Jacobian (C.6) by evaluation of the partial derivatives (C.8), (C.9) and (C.17) for  $\mathbf{u} = \mathbf{u}_t$  and  ${}^S\boldsymbol{\theta} = {}^S\hat{\boldsymbol{\theta}}_{t-1}$

$$\left. \frac{\partial {}^S\mathcal{M}_q}{\partial \mathbf{u}} \right|_{\mathbf{u} = \mathbf{u}_t, {}^S\boldsymbol{\theta} = {}^S\hat{\boldsymbol{\theta}}_{t-1}} \approx \begin{bmatrix} -R \left( {}^R\mathbf{u}_t \right) S \left( \mathbf{X}_{q,t-1} \right) & I_3 \end{bmatrix} \quad (\text{C.18})$$

$$\left. \frac{\partial {}^S\mathcal{M}_q}{\partial {}^S\boldsymbol{\theta}} \right|_{\mathbf{u} = \mathbf{u}_t, {}^S\boldsymbol{\theta} = {}^S\hat{\boldsymbol{\theta}}_{t-1}} = R \left( {}^R\mathbf{u}_t \right) \begin{bmatrix} 0 & \dots & 0 \\ 0 & \dots & 0 \\ \mathcal{B}_{1,2}(x_q, y_q) & \dots & \mathcal{B}_{N,2}(x_q, y_q) \end{bmatrix}. \quad (\text{C.19})$$

Note that we substituted the definition of the 3d-sample points  $\mathbf{X}_{q,t-1} = {}^S\mathcal{X}_q({}^S\hat{\boldsymbol{\theta}}_{t-1})$ .

## C.2 Error propagation for the Prediction of the Street Boundary

Let us now consider the error propagation for the prediction of the street boundary described in Section 3.4.1.2. The transformation of the previously estimated boundary into the current system is embodied by the transformation of the previously estimated control points of the boundary spline  ${}^B\hat{\boldsymbol{\theta}}_{t-1}$ . Keep in mind that the parameter vector  ${}^B\boldsymbol{\theta}$  embodies the concatenation of the horizontal coordinates of the single control points (see Section 3.3.3). Thus, we can define a function to access the 3d-coordinates of the  $m$ th control point

$$\begin{bmatrix} {}^B\hat{\mathbf{x}}_{m,t-1} \\ 0 \end{bmatrix} = {}^B\mathcal{X}_m({}^B\hat{\boldsymbol{\theta}}_{t-1}) \quad (\text{C.20})$$

by means of the unit vector  $\mathbf{e}_{2m}$  and  $\mathbf{e}_{2m-1}$

$${}^B\mathcal{X}_m({}^B\boldsymbol{\theta}) := \begin{bmatrix} \mathbf{e}_{2m-1}^\top {}^B\boldsymbol{\theta} \\ \mathbf{e}_{2m}^\top {}^B\boldsymbol{\theta} \\ 0 \end{bmatrix}. \quad (\text{C.21})$$

Similar to the previous section, the coordinates of the control points are transformed to the current system via the prediction function

$${}^B\mathcal{M}_m(\mathbf{u}, {}^B\boldsymbol{\theta}) := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} {}^t\mathbf{M}_{t-1}(\mathbf{u}) \begin{bmatrix} {}^B\mathcal{X}_m({}^B\boldsymbol{\theta}) \\ 1 \end{bmatrix}. \quad (\text{C.22})$$

Note that the points are projected to the horizontal plane since the boundary spline is defined as horizontal delimiter of the drivable region, which the vertical coordinate useless. The precision of the predicted control points follows from linear error propagation

$${}^B\Sigma_{\theta_t^-, \theta_t^-} = {}^BJ_u \begin{bmatrix} {}^B\Sigma_{\hat{\theta}_{t-1}\hat{\theta}_{t-1}} & 0 \\ 0 & \Sigma_{u_t u_t} \end{bmatrix} {}^BJ_u^\top, \quad (\text{C.23})$$

with the Jacobian

$${}^BJ_u = \begin{bmatrix} \vdots & \vdots \\ \frac{\partial {}^B\mathcal{M}_m}{\partial \mathbf{u}} \Big|_{\mathbf{u}_t, {}^B\hat{\theta}_{t-1}} & \frac{\partial {}^B\mathcal{M}_m}{\partial {}^B\boldsymbol{\theta}} \Big|_{\mathbf{u}_t, {}^B\hat{\theta}_{t-1}} \\ \vdots & \vdots \end{bmatrix}. \quad (\text{C.24})$$

As in the previous section, we substitute the definition of the ego-motion matrix (3.19) into the prediction function

$${}^B\mathcal{M}_m(\mathbf{u}, {}^B\boldsymbol{\theta}) := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (R({}^R\mathbf{u}) {}^B\mathcal{X}_m({}^B\boldsymbol{\theta}) + {}^T\mathbf{u}) \quad (\text{C.25})$$

in order to determine the partial derivatives which assemble the Jacobian. The derivatives with respect to the boundary parameters and the translation parameters read as

$$\frac{\partial {}^B\mathcal{M}_m}{\partial {}^T\mathbf{u}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (\text{C.26})$$

$$\frac{\partial {}^B\mathcal{M}_m}{\partial {}^B\boldsymbol{\theta}} = \frac{\partial {}^B\mathcal{M}_m}{\partial {}^B\mathcal{X}_m} \frac{\partial {}^B\mathcal{X}_m}{\partial {}^B\boldsymbol{\theta}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R({}^R\mathbf{u}) \begin{bmatrix} \mathbf{e}_{2m-1}^\top \\ \mathbf{e}_{2m}^\top \\ \mathbf{0}^\top \end{bmatrix}. \quad (\text{C.27})$$

The partial derivative with respect to the rotation parameters is obtained analogously to the procedure presented for the street surface prediction using the approximation (C.11) for the rotation matrix. Thus, with the only difference embodied by the additional projection matrix the derivative reads as

$$\frac{\partial {}^B\mathcal{M}_m}{\partial {}^R\mathbf{u}} \Big|_{R_u = R_u^{(0)}} \approx - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R({}^R\mathbf{u}^{(0)}) S({}^B\mathcal{X}_m({}^B\boldsymbol{\theta})). \quad (\text{C.28})$$

In combination, evaluation with  $\mathbf{u} = \mathbf{u}_t$  and  ${}^B\boldsymbol{\theta} = {}^B\hat{\theta}_{t-1}$  yields the partial derivatives which assemble the Jacobian  ${}^BJ_u$

$$\frac{\partial {}^B\mathcal{M}_m}{\partial \mathbf{u}} \Big|_{u=u_t, {}^B\theta = {}^B\hat{\theta}_{t-1}} \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -R({}^R\mathbf{u}_t) S\left(\begin{bmatrix} {}^B\hat{\mathbf{x}}_{m,t-1} \\ 0 \end{bmatrix}\right) & I_3 \end{bmatrix} \quad (\text{C.29})$$

$$\frac{\partial {}^B\mathcal{M}_m}{\partial {}^B\boldsymbol{\theta}} \Big|_{u=u_t, S\theta = S\hat{\theta}_{t-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R({}^R\mathbf{u}_t) \begin{bmatrix} \mathbf{e}_{2m-1}^\top \\ \mathbf{e}_{2m}^\top \\ \mathbf{0}^\top \end{bmatrix}. \quad (\text{C.30})$$

Note that we used the definition of  ${}^B\mathcal{X}_m$  given by (C.20) to write the derivative in terms of the previously estimated control points  ${}^B\hat{\mathbf{x}}_{m,t-1}$ .



## Appendix D

# Mathematical Derivations

This Section contains mathematical proofs and derivations for statements given earlier within this thesis.

### D.1 Simplified Computation of the Expectation in the EM-Algorithm

**Statement:** Assume a vector of random variables  $\mathbf{l} = [l_1, \dots, l_I]^\top$  to be given which take discrete values  $l_i \in \{c_1, \dots, c_K\}$  and satisfies  $\sum_{k=1}^K P(l_i = c_k) = 1$  and  $P(\mathbf{l}) = \prod_{i=1}^I P(l_i)$ . The expected value of a function  $f(\mathbf{l}) := \sum_{j=1}^I f_j(l_j)$  is then given by

$$E(f(\mathbf{l})) = \sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l}) f(\mathbf{l}) \quad (\text{D.1})$$

$$\stackrel{!}{=} \sum_{i=1}^I \sum_{k=1}^K P(l_i = c_k) f_i(c_k), \quad (\text{D.2})$$

where  $\mathcal{L}$  denotes the set of all possible assignments of  $\mathbf{l}$ .

**Proof:** The proof is done by mathematical induction over the number of elements in  $\mathbf{l}$ :

- Assume  $I = 1$ :

$$\sum_{l_1 \in \mathcal{L}} P(l_1) f(l_1) = \sum_{k=1}^K P(l_1 = c_k) f(c_k) \quad \square \quad (\text{D.3})$$

- Assume  $I > 1$ : We subdivide the total sum over  $\mathcal{L}$  in  $K$  sums over subsets  $\{\mathcal{L} | l_1 = c_k\}$ , where the first respective element of  $\mathbf{l}$  is fixed to  $c_k$  and split off the terms including  $f_1$

to achieve

$$\sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l}) f(\mathbf{l}) = \sum_{\mathbf{l} \in \mathcal{L}} \prod_{i=1}^I P(l_i) \sum_{j=1}^I f_j(l_j) \quad (\text{D.4})$$

$$= \sum_{k=1}^K \sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k\}} P(l_1 = c_k) \prod_{i=2}^I P(l_i) \left( f_1(c_k) + \sum_{j=2}^I f_j(l_j) \right) \quad (\text{D.5})$$

$$= \sum_{k=1}^K P(l_1 = c_k) f_1(c_k) \sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k\}} \prod_{i=2}^I P(l_i) \\ + \sum_{k=1}^K P(l_1 = c_k) \sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k\}} \prod_{i=2}^I P(l_i) \sum_{j=2}^I f_j(l_j). \quad (\text{D.6})$$

We now use the induction hypothesis to rearrange the second term

$$\sum_{k=1}^K P(l_1 = c_k) \sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k\}} \prod_{i=2}^I P(l_i) \sum_{i=2}^I f_i(l_i) \quad (\text{D.7})$$

$$\stackrel{\text{induction}}{=} \sum_{k=1}^K P(l_1 = c_k) \sum_{k'=1}^K \sum_{i=2}^I P(l_i = c_{k'}) f_i(c_{k'}) \quad (\text{D.8})$$

$$= \sum_{k'=1}^K \sum_{i=2}^I P(l_i = c_{k'}) f_i(c_{k'}) \underbrace{\sum_{k=1}^K P(l_1 = c_k)}_1. \quad (\text{D.9})$$

Moreover, from successive factorization follows the identity

$$\sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k\}} \prod_{i=2}^I P(l_i) = \sum_{k'=1}^K P(c_{k'}) \sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k, l_2 = c_{k'}\}} \prod_{i=3}^I P(l_i) \quad (\text{D.10})$$

$$= \sum_{\mathbf{l} \in \{\mathcal{L} | l_1 = c_k, l_2 = c_1\}} \prod_{i=3}^I P(l_i) \underbrace{\sum_{k'=1}^K P(c_{k'})}_1 = \dots = 1, \quad (\text{D.11})$$

where the second equation uses the fact that the product does not include  $l_2$ , i.e. is independent on  $k'$ .

Finally, inserting (D.9) and (D.11) in (D.6) and aggregating the sums yields

$$\sum_{\mathbf{l} \in \mathcal{L}} P(\mathbf{l}) f(\mathbf{l}) = \sum_{k=1}^K P(l_1 = c_k) f_1(c_k) + \sum_{k'=1}^K \sum_{i=2}^I P(l_i = c_{k'}) f_i(c_{k'}) \quad (\text{D.12})$$

$$= \sum_{k=1}^K \sum_{i=1}^I P(l_i = c_k) f_i(c_k). \quad \square \quad (\text{D.13})$$

## D.2 Standard Deviation from Intersection of Gaussians

**Statement:** Assume a Gaussian PDF  $f(x) = G(x | \mu, \sigma^2)$  to be given. Neglecting the trivial solution  $\sigma_c = \sigma$ , the standard deviation of the normal distribution with the same mean whose

density function  $f_c(x) = G(x \mid \mu, \sigma_c^2)$  intersects  $f$  at  $x_1 = \mu - c\sigma$  and  $x_2 = \mu + c\sigma$  is defined by

$$\sigma_c = \sqrt{-\frac{c^2}{\mathcal{L}(-c^2 \exp(-c^2))}} \sigma, \quad (\text{D.14})$$

with  $\mathcal{L}$  being the Lambert W function that satisfies  $\mathcal{L}(x) \exp(\mathcal{L}(x)) = x$ . For more information on the Lambert W function, we refer to [Corless et al., 1996].

**Proof:** The identity (D.14) follows from solving

$$f(\mu + c\sigma) = f_c(\mu + c\sigma) \quad (\text{D.15})$$

for  $\sigma_c$  claiming  $\sigma_c \neq \sigma$ . Using the substitution  $v_c = \exp(-\frac{1}{2}c^2)$ , we get

$$G(\mu + c\sigma \mid \mu, \sigma^2) = G(\mu + c\sigma \mid \mu, \sigma_c^2) \quad (\text{D.16})$$

$$\Leftrightarrow \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}c^2\right) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{1}{2}\frac{(c\sigma)^2}{\sigma_c^2}\right) \quad (\text{D.17})$$

$$\Leftrightarrow v_c = \frac{\sigma}{\sigma_c} \exp\left(-\frac{1}{2}c^2\frac{\sigma^2}{\sigma_c^2}\right). \quad (\text{D.18})$$

Squaring both sides and multiplying  $-c^2$  yields

$$-c^2 v_c^2 = -c^2 \frac{\sigma^2}{\sigma_c^2} \exp\left(-c^2 \frac{\sigma^2}{\sigma_c^2}\right). \quad (\text{D.19})$$

From the property  $\mathcal{L}(x) \exp(\mathcal{L}(x)) = x$  of the Lambert W function, we finally derive

$$\mathcal{L}(-c^2 v_c^2) = -c^2 \frac{\sigma^2}{\sigma_c^2} \quad (\text{D.20})$$

$$\Leftrightarrow \sigma_c = \sqrt{\frac{-1}{\mathcal{L}(-c^2 v_c^2)}} c\sigma \quad (\text{D.21})$$

$$= \sqrt{\frac{-c^2}{\mathcal{L}(-c^2 \exp(-c^2))}} \sigma. \quad \square \quad (\text{D.22})$$



# Bibliography

- W. Alt. *Nichtlineare Optimierung*. Vieweg Verlag, 2002.
- American Association of State Highway and Transportation Officials. *A Policy on Geometric Design of Highways and Streets*. American Association of State Highway and Transportation Officials, 2001.
- R.C. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.
- H. Badino. A robust approach for ego-motion estimation using a mobile stereo platform. In *International Conference on Complex Motion (IWCM)*, pages 198–208, 2007.
- H. Badino, U. Franke, and R. Mester. Free space computation using stochastic occupancy grids and dynamic programming. In *Workshop on Dynamical Vision (ICCV)*, 2007.
- H. Badino, U. Franke, and D. Pfeiffer. The stixel world - a compact medium level representation of the 3d-world. In *Symposium of the German Association for Pattern Recognition (DAGM)*, pages 51–60, 2009.
- A. Barth and U. Franke. Tracking oncoming and turning vehicles at intersections. In *Intelligent Transportation Systems (ITS)*, pages 861–868, 2010.
- H. C. Batagelo and S.-T. Wu. Dynamic scene occlusion culling using a regular grid. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 43–50, 2002.
- R. Benenson, R. Timofte, and L. Van Gool. Stixels estimation without depth map computation. In *Computer Vision Workshops (ICCV)*, pages 2010–2017, 2011.
- M. Bertozzi and A. Broggi. Gold: a parallel real-time stereo vision system for generic obstacle and lane detection. *Transactions on Image Processing (TIP)*, 7:62–81, 1998.
- S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision (IJCV)*, 35:1073–1080, 1996.
- C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- J.-Y. Bouguet. Camera calibration toolbox for matlab, 2007. URL [http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html).
- Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23:1222–1239, 2001.
- A. Broggi and S. Bert. Vision-based road detection in automotive systems: A real-time expectation-driven approach. *Journal of Artificial Intelligence Research (JAIR)*, 3:325–348, 1995.

- A. Broggi, L. Bombini, S. Cattani, P. Cerri, and R. I. Fedriga. Sensing requirements for a 13,000 km intercontinental autonomous drive. In *Intelligent Vehicles Symposium (IV)*, 2010.
- A. Broggi, M. Buzzoni, M. Felisa, and P. Zani. Stereo obstacle detection in challenging environments: the viac experience. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1599–1604, 2011.
- I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Thun und Frankfurt am Main, 2001.
- G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision (ECCV)*, pages 44–57, 2008.
- M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25:993–1008, 2003.
- K.-Y. Byun, B.-S. Kim, H.-K. Kim, J.-E. Shin, and S.-J. Ko. An effective pedestrian detection method for driver assistance system. In *International Conference on Consumer Electronics (ICCE)*, pages 229–230, 2012.
- S. Carlsson and J.-O. Eklundh. Object detection using model based prediction and motion parallax. In *European Conference on Computer Vision (ECCV)*, pages 297–306, 1990.
- P. Cerri and P. Grisleri. Free space detection on highways using time correlation between stabilized sub-pixel precision ipm images. In *International Conference on Robotics and Automation (ICRA)*, pages 2223 – 2228, 2005.
- C. Chen and J. Liu. A reinforced road detection method in complicated environment. In *Global Congress on Intelligent Systems (GCIS)*, pages 65–69, 2010.
- P. Clifford. Markov random fields in statistics. In *Disorder in Physical Systems*, pages 19–32, 1990.
- R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the lambert w function. *Advances in Computational mathematics*, 5:329–359, 1996.
- J. Crisman and C. Thorpe. Scarf: A color vision system that tracks roads and intersections. *Transactions on Robotics and Automation (T-RA)*, 9:49– 58, 1993.
- C. De Boor. On calculating with b-splines. *Journal of Approximation Theory (JAT)*, 6:50–62, 1972.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society (JRSSB)*, 39:1–38, 1977.
- P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34: 743–761, 2012.
- I. Dryanovski, W. Morris, and J. Xiao. Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1553–1559, 2010.

- A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22: 46–57, 1989.
- A. Ess, B. Leibe, K. Schindler, and L. Van Gool. Moving obstacle detection in highly dynamic scenes. In *International Conference on Robotics and Automation (ICRA)*, pages 4451–4458, 2009.
- G. Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann Publishers Inc., 2002.
- U. Franke, S. Gorzig, F. Lindner, D. Mehren, and F. Paetzold. Steps towards an intelligent vision system for driver assistance in urban traffic. In *Conference on Intelligent Transportation System (ITSC)*, pages 601–606, 1997.
- B.J. Frey and D.J.C. MacKay. A revolution: Belief propagation in graphs with cycles. In *Advances in Neural Information Processing Systems (NIPS)*, pages 479–485, 1998.
- S. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In *International Conference on Computer Vision Systems (ICVS)*, pages 134–143, 2009.
- E. L. W. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press, 1981.
- R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, 2003.
- T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2009.
- H. Hattori and A. Maki. Stereo without depth search and metric calibration. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 117–184, 2000.
- H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 807–814, 2005.
- F. Homm, N. Kaempchen, J. Ota, and D. Burschka. Efficient occupancy grid computation on the gpu with lidar and radar for road boundary detection. In *Intelligent Vehicles Symposium (IV)*, pages 1006–1013, 2010.
- H. G. Jung, Y. H. Lee, B. J. Kim, P. J. Yoon, and J. H. Kim. Stereo vision-based forward obstacle detection. *International Journal of Automotive Technology (IJAT)*, 8:493–504, 2007.
- K. Kaliyaperumal, S. Lakshmanan, and K. Kluge. An algorithm for detecting roads and obstacles in radar images. *Transactions on Vehicular Technology (TVT)*, 50:170–182, 2001.
- J. Kang and M. J. Chung. Stereo-vision based free space and obstacle detection with structural and traversability analysis using probabilistic volume polar grid map. In *Conference on Robotics, Automation and Mechatronics (RAM)*, pages 245 –251, 2011.
- B. Kim, J. Son, and K. Sohn. Illumination invariant road detection based on learning method. In *Conference on Intelligent Transportation Systems (ITSC)*, pages 1009 –1014, 2011.

- R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.
- A. Kirchner and C. Ameling. Integrated obstacle and road tracking using a laser scanner. In *Intelligent Vehicles Symposium (IV)*, pages 675–681, 2000.
- K.R. Koch. *Einführung in die Bayes-Statistik*. Springer, 2000.
- K.R. Koch. *Parameterschätzung und Hypothesentests*. Dümmlers Verlag, 1997.
- K. Kohara, N. Suganuma, T. Negishi, and T. Nanri. Obstacle detection based on occupancy grid maps using stereovision system. *International Journal of Intelligent Transportation Systems Research*, 8:85–95, 2010.
- V. Kolmogorov and R. Zabini. What energy functions can be minimized via graph cuts? *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26:147–159, 2004.
- S. Kubota, T. Nakano, and Y. Okamoto. A global optimization algorithm for real-time on-board stereo obstacle detection systems. In *Intelligent Vehicles Symposium (IV)*, pages 7–12, 2007.
- F. Küçükay and J. Bergholz. Driver assistant systems. In *International Conference of Automotive Technologies (ICAT)*, 2004.
- K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision (IJCV)*, 38:199–218, 2000.
- R. Labayrade and D. Aubert. A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision. In *Intelligent Vehicles Symposium (IV)*, pages 31 – 36, 2003.
- R. Labayrade, D. Aubert, and J. P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In *Intelligent Vehicles Symposium (IV)*, pages 646–651, 2002.
- A. Lacaze, K. Murphy, and M. DelGiorno. Autonomous mobility for the demo iii experimental unmanned vehicles. In *Conference on Unmanned Vehicles (AUVSI)*, 2002.
- J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, pages 282–289, 2001.
- N. Lazaros, G. C. Sirakoulis, and A. Gasteratos. Review of stereo vision algorithms: From software to hardware. *International Journal of Optomechatronics*, 2:435–462, 2008.
- D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- R. Manduchi, A. Castano, A. Talukder, and L. Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous Robots*, 18:81–102, 2005.
- C. McGlone, E. Mikhail, and J. Bethel, editors. *Manual of Photogrammetry*. American Society of Photogrammetry and Remote Sensing, 2004.



- G. Van Meerbergen, M. Vergauwen, M. Pollefeys, and L. Van Gool. A hierarchical symmetric stereo algorithm using dynamic programming. *International Journal of Computer Vision (IJCV)*, 47:275–285, 2002.
- J. Meidow, C. Beder, and W. Förstner. Reasoning with uncertain points, straight lines, and straight line segments in 2d. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64: 125–139, 2009.
- T. Michalke, R. Kastner, J. Fritsch, and C. Goerick. A self-adaptive approach for curb-stone/roadside detection based on human-like signal processing and multi-sensor fusion. In *Intelligent Vehicles Symposium (IV)*, 2010.
- D. Murray and J. J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 8:161–171, 2000.
- S. Nedeveschi, R. Danescu, D. Frentiu, T. Marita, F. Oniga, C. Pocol, T. Graf, and R. Schmidt. High accuracy stereovision approach for obstacle detection on non-planar roads. In *International Conference on Intelligent Engineering Systems (INES)*, pages 211–216, 2004.
- S. Nedeveschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S.n Sobol, C. Tomiuc, C. Vancea, M.M. Meinecke, T. Graf, T. B. To, and M. A.Obojski. A sensor for urban driving assistance systems based on dense stereovision. In *Intelligent Vehicles Symposium (IV)*, pages 276–283, 2007.
- S. Nedeveschi, A. Vatavu, and F. Oniga. Forward collision detection based on elevation map from dense stereo. In *Workshop on Planning, Perception and Navigation for Intelligent Vehicles (IROS)*, pages 76–81, 2008.
- S. Nedeveschi, R. Danescu, T. Marita, F. Oniga, Ciprian Pocol, S. Bota, M. M. Meinecke, and M. A. Obojski. Stereovision-based sensor for intersection assistance. In *International Forum on Advanced Microsystems for Automotive Applications (AMAA)*, pages 129–163, 2009.
- M. Nikolova and A. Hero. Segmentation of a road from a vehicle-mounted radar and accuracy of the estimation. In *Intelligent Vehicles Symposium (IV)*, pages 284–289, 2000.
- M. Okutomi and S. Noguchi. Extraction of road region using stereo images. In *International Conference on Pattern Recognition (ICPR)*, pages 853–856, 1998.
- F. Oniga, S. Nedeveschi, Marc M. M., and T. B. To. Road surface and obstacle detection based on elevation maps from dense stereo. In *Intelligent Transportation Systems Conference (ITSC)*, pages 859–865, 2007a.
- F. Oniga, S. Nedeveschi, and M.M. Meinecke. Curb detection based on elevation maps from dense stereo. In *International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 119–125, 2007b.
- K. Onoguchi, N. Takeda, and M. Watanabe. Planar projection stereopsis method for road extraction. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 249–256, 1995.
- N. Papadakis and V. Caselles. Multi-label depth estimation for graph cuts stereo problems. *Journal of Mathematical Imaging and Vision (JMIV)*, 38:70–82, 2010.

- D. Payton. An architecture for reflexive autonomous vehicle control. In *International Conference on Robotics and Automation (ICRA)*, pages 1838–1845, 1986.
- D. Pfeiffer and U. Franke. Towards a global optimal multi-layer stixel representation of dense 3d data. In *British Machine Vision Conference (BMVC)*, 2011.
- D. Pfeiffer, S. Moralez, A. Barth, and U. Franke. Ground truth evaluation of the stixel representation using laser scanners. In *Intelligent Transportation Systems Conference (ITSC)*, 2010.
- L. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
- T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences (SIIMS)*, 3:1122–1145, 2010.
- J. Pohl, M. Sethsson, P. Degerman, and J. Larsson. A semi-automated parallel parking system for passenger cars. *Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 220:53–65, 2006.
- R. Rajamani. *Vehicle Dynamics and Control*. Springer, 2012.
- R. Roscher, W. Förstner, and B. Waske. I2vm: Incremental import vector machines. *Image and Vision Computing*, 30:263–278, 2012.
- H. Satonaka, M. Okuda, S. Hayasaka, T. Endo, Y. Tanaka, and T. Yoshida. Development of parking space detection using an ultrasonic sensor. In *World Congress on Intelligent Transportation Systems and Services*, pages 1–8, 2006.
- U. Scheunert, B. Fardi, N. Mattern, G. Wanielik, and N. Keppeler. Free space determination for parking slots using a 3d pmd sensor. In *Intelligent Vehicles Symposium (IV)*, pages 154–159, 2007.
- T. Shioyama, H. Wu, M. Takebe, and N. Shimaoka. Segmentation and free space detection using gabor filters. In *Scandinavian Conference on Image Analysis (SCIA)*, pages 311–319, 2003.
- J. Siegemund, D. Pfeiffer, U. Franke, and W. Förstner. Curb reconstruction using conditional random fields. In *Intelligent Vehicles Symposium (IV)*, pages 203–210, 2010.
- J. Siegemund, U. Franke, and W. Förstner. A temporal filter approach for detection and reconstruction of curbs and road surfaces based on conditional random fields. In *Intelligent Vehicles Symposium (IV)*, pages 637–642, 2011.
- P. Steingrube, S. Gehrig, and U. Franke. Performance evaluation of stereo algorithms for automotive applications. In *Computer Vision Systems*, pages 285–294. Springer, 2009.
- P. Sturgess, K. Alahari, L. Ladicky, and P.H.S. Torr. Combining appearance and structure from motion features for road scene understanding. In *British Machine Vision Conference (BMVC)*, 2009.
- Z. S. Z. Sun, G. Bebis, and R. Miller. On-road vehicle detection: a review. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28:694–711, 2006.

- Y. Taguchi, B. Wilburn, and C.L. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.
- C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the carnegiemellon navlab. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 10: 362–373, 1988.
- S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 2005.
- R. Turchetto and R. Manduchi. Visual curb localization for autonomous navigation. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1336–1342, 2003.
- C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, MN Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, and J. Zigar. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics (JFR)*, 25:425–466, 2008.
- A. Wedel, H. Badino, C. Rabe, H. Loose, U. Franke, and D. Cremers. B-spline modeling of road surfaces with an application to free-space estimation. *Transactions on Intelligent Transportation Systems (TITS)*, 10:572–583, 2009.
- C. Wellington, A. Courville, and A. Stentz. A generative model of terrain for autonomous navigation in vegetation. *International Journal of Robotics Research (IJRR)*, 25:1287–1304, 2006.
- W.S. Wijesoma, K.R.S. Kodagoda, and A.P. Balasuriya. Road-boundary detection and tracking using ladar sensing. *Transactions on Robotics and Automation (T-RA)*, 20:456–464, 2004.
- W. Zhang. Lidar-based road and road-edge detection. In *Intelligent Vehicles Symposium (IV)*, pages 845–848, 2010.
- J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics (JCGS)*, 14:185–205, 2005.
- C. Zinner, M. Humenberger, K. Ambrosch, and W. Kubinger. An optimized software-based implementation of a census-based stereo matching algorithm. In *International Symposium on Advances in Visual Computing (ISVC)*, pages 216–227, 2008.