Design, Simulation, and Implementation of a Vision-Based Vehicle-Following System

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Fakultät für Physik
der Eberhard-Karls-Universität zu Tübingen

vorgelegt von

Stefan K. Gehrig

aus Sindelfingen

2000

Stefan Gehrig Design, Simulation, and Implementation of a Vehicle-Following System

Design, Simulation, and Implementation of a Vision-Based Vehicle-Following System

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
der Fakultät für Physik
der Eberhard-Karls-Universität zu Tübingen

vorgelegt von

Stefan K. Gehrig

aus Sindelfingen

2000

Tag der mündlichen Prüfung:	31. Juli 2000
Dekan:	Prof. Dr. K. Werner
1. Berichterstatter:	Prof. Dr. H. Ruder
2. Berichterstatter:	Prof. Dr. D. Wharam

Acknowledgments

I would like to thank my advisor, Dr. Fridtjof Stein, for ample guidance, valuable advice and productive discussions during my years at DaimlerChrysler. Special thanks to my PhD advisor at the University of Tübingen, Prof. Hanns Ruder, who helped me a lot with fruitful discussions. Furthermore, he supported me in managing the administrative issues. Additionally, I express my gratitude to my departmental supervisor at DaimlerChrysler, Stefan Hahn, who initiated this project. Also, thanks to Prof. David Wharam and Prof. Heinrich Bülthoff for volunteering to be on my PhD committee.

There were many more people contributing to this thesis. Thanks to the whole FT3/AB and FT3/AA team for supplying a great platform to write my thesis on. Especially I would like to thank my two master students, Karl-Heinz Penteker and Birgit Rieger, for working with me and coding up pieces of the presented work. Many more contributed but are too numerous to list.

Credits also to Joel Brown, Dr. Thomas Brunner and Damica Zemault for proof-reading my thesis.

Last, but not least, I am indebted to my parents for supporting my academic education. Special thanks also to Angela Fischer-Brunkow, who supported me on this project at all times.

Contents

List of Figures

List of Tables

Deutsche Zusammenfassung der Dissertation

"Entwurf, Simulation und Implementierung eines Fahrzeugfolgesystems mittels Stereobildverarbeitung"

0.1 Einführung

Das Auto der Zukunft wird zunehmend intelligenter. Systeme wie das ABS sind heute bereits selbstverständlich. Das elektronische Stabilitätsprogramm ESP ist ein weiteres Indiz für diesen Trend. Der adaptive Tempomat, der die eigene Geschwindigkeit an die Geschwindigkeit des vorausfahrenden Fahrzeugs anpaßt, bestätigt den Trend zu mehr Intelligenz im Auto. Als weitere Stufe werden Assistenzsysteme für das Auto auf den Markt kommen, die den Fahrer immer mehr von seinen Fahraufgaben entlasten. Solche Systeme arbeiten unter anderem mit Kamerasensorik.

Die Dissertation beschäftigt sich mit dem Bildverstehen von Straßenszenen, die auf Sensordaten eines Stereo-Kamerasystems basieren. Die Hauptanwendung zielt auf ein autonomes Fahren im Stop-and-go-Verkehr ab. Im Rahmen der Dissertation wird eine Situationsanalyse der Straßenszene für ein Fahrzeugfolge-Szenario vorgenommen, um Daten für die Längs- und Querführung des Fahrzeugs zu gewinnen.

Die vorgestellten Algorithmen nutzen die Sensordaten des Stereo-Kamerasystems, können aber auch für andere Sensoren adaptiert werden, die Entfernungsinformation liefern.

0.2 Das Fahrzeugfolgesystem

0.2.1 Grundprinzip

Die Fahraufgabe im Stop-and-go-Verkehr besteht hauptsächlich aus dem Folgen eines vorausfahrenden Fahrzeugs. Diese Aufgabe läßt sich durch Vermessung des Abstands und des seitlichen Versatzes des vorausfahrenden Fahrzeugs automatisieren. Dieses

Kopplungsprinzip wird auch elektronische Deichsel genannt. Das Verhalten des autonomen Fahrzeugs ist dem eines Anhängers sehr ähnlich. Das Ansteuern von Gas und Bremse (Längsführung) wird dabei so vorgenommen, daß ein Sicherheitsabstand zum vorausfahrenden Fahrzeug eingehalten wird. Das Lenkrad (Querführung) wird so angesteuert, daß man immer auf die aktuelle Position des vorausfahrenden Fahrzeugs zufährt. Im folgenden wird das vorausfahrende Fahrzeug Führungsfahrzeug genannt, da es das autonome Fahrzeug führt.

0.2.2 Der Kamerasensor

Als Sensor für die Umgebungserfassung sind zwei Kameras am Rückspiegel im Wageninneren montiert, die nach vorne gerichtet die Verkehrsszene erfassen. Bei den parallel angeordneten, kalibrierten Kameras werden signifikante Punkte im linken Bild mit dem rechten Bild korreliert und so korrespondierende Punkte gefunden. Deren Entfernung kann durch Triangulation bestimmt werden. Ergebnis dieser Korrespondenzsuche ist eine 3D-Punktwolke. Nimmt man weiterhin eine konstante Orientierung zwischen Kamera und Straße an, kann man zu allen bestimmten 3D-Punkten die Position relativ zur Straße bestimmen, wobei alle Punkte über der Straße Objekten zugeordnet werden. Durch Zusammenfassung mehrerer beieinander liegender 3D-Punkte zu einem Cluster können Roh-Objekte gebildet werden. Durch Verfolgen und Wiederfinden in darauf folgenden Bildpaaren entstehen Objekte.

Das Objekt, das bei gleichbleibendem Lenkradeinschlag als erstes in der Spur des eigenen Fahrzeugs ist, stellt ein potentielles Führungsfahrzeug dar. Der Fahrer muß das System explizit einschalten, bevor das Fahrzeug dem Objekt automatisch folgt. Der automatische Fahrmodus wird beendet, wenn das Führungsfahrzeug aus dem Erfassungsbereich verschwindet. Aus Sicherheitsgründen kann man jederzeit das System abschalten und manuell weiterfahren.

0.2.3 Regelung und Fahrzeugaufbau

Die von der Bildverarbeitung ermittelten Meßdaten des vorausfahrenden Fahrzeugs werden über Regelalgorithmen zu Stellgrößen für die Aktuatoren umgewandelt, die Gas, Bremse und Lenkrad elektronisch ansteuern.

Die Bildverarbeitungs- und Regelalgorithmen laufen auf regulären PCs, die im Kofferraum des Versuchsträgers untergebracht sind und Schnittstellen zu den Aktuatoren haben. Die Bilddaten werden über einen Framegrabber in den PC eingelesen.

0.3 Analyse des Systems

0.3.1 Probleme beim Standard-Fahrzeugfolgen

Das Deichselprinzip hat im regulären Verkehr folgende Nachteile:

- Beim oben beschriebenen Fahrzeugfolgen weicht das autonome Fahrzeug von der Trajektorie des vorausfahrenden Fahrzeugs ab, wenn man in eine Kurve einlenkt.

Die Kurve wird geschnitten. Für beliebige Trajektorien des vorausfahrenden Fahrzeugs treten meistens Abweichungen auf. Diese nehmen mit zunehmendem Abstand zum vorausfahrenden Fahrzeug zu. Der Mindestabstand zum vorausfahrenden Fahrzeug ist jedoch gesetzlich geregelt und begrenzt die Genauigkeit dieser Methode.

- Sobald die vorhandene Sensorik im Auto keine Rundumsicht gewährleistet, kann das Fahrzeug nicht vollautonom fahren. Die begrenzte Umgebungserfassung führt in scharfen Kurven dazu, daß der Erfassungsbereich verlassen wird und das Führungsfahrzeug "verloren" geht. Auch potentielle Hindernisse liegen oft außerhalb des Erfassungsbereichs.

- Beim Folgen der Trajektorie eines Führungsfahrzeugs im Sicherheitsabstand können jederzeit andere Verkehrsteilnehmer in die geplante Trajektorie eindringen. Dies kann bei Systemen, die gegenüber solchen Hindernissen blind sind, zu Unfällen führen.

0.3.2 Lösungsansätze

Oben beschriebene Probleme können durch neu entworfene Algorithmen behoben werden. Im einzelnen wurden folgende Algorithmen konzipiert:

- Das Kurvenschneiden kann vermieden werden (siehe Abschnitt 0.4.1), wenn der abgefahrene Pfad des vorausfahrenden Fahrzeugs rekonstruiert wird. Dazu ist auch eine Eigenpositionsbestimmung notwendig (siehe Abschnitt 0.4.2).

- Bei Verlassen des Erfassungsbereichs ist es möglich, die Trajektorie oder die Position von Hindernissen in einem unbewegten Koordinatensystem zu extrapolieren (siehe Abschnitt 0.4.3).

- Durch Berücksichtigung aller Hindernisse in der Szene wird die Trajektorie des vorausfahrenden Fahrzeugs modifiziert. Dabei wird die Trajektorie als elastisches Band behandelt, das von Hindernissen abgestoßen wird und eine gewisse Steifigkeit besitzt, um glatte Trajektorien zu erzeugen (siehe Abschnitt 0.4.4).

0.4 Erweiterungen des Systems

0.4.1 Pfadbasiertes Fahrzeugfolgen - der CUT-Algorithmus

Durch Ausnutzung der Historie der 3D-Messungen des vorausfahrenden Fahrzeugs kann man die Trajektorie des Führungsfahrzeugs rekonstruieren. Dazu muß zuerst eine Eigenpositionsbestimmung in einem unbewegten Koordinatensystem erfolgen.

Eine schnelle Eigenpositionsbestimmung kann durch Aufintegration der Daten des Geschwindigkeitssensors und des Lenkwinkelsensors über die Zeit erfolgen. Unter der Annahme eines stationären Lenkverhaltens, einer flachen Fahrbahn und vernachlässigbaren Seitenwinds kann man über ein Fahrzeugmodell die Orientierung und Position des Fahrzeugs im unbewegten Koordinatensystem bestimmen.

Mit dieser Information kann man zu jedem Zeitpunkt die Positionsdaten des vorausfahrenden Fahrzeugs im unbewegten Koordinatensystem bestimmen. Durch Rücktransformation ins Fahrzeug-Koordinatensystem steht die gesamte Trajektorie zur Verfügung.

Von dieser Trajektorie wird ein Punkt in bestimmter Entfernung ausgewählt, auf den mit dem Deichselprinzip zugefahren wird. Je nach Vorausschau wird dabei weiterhin die Kurve geschnitten, aber wenn die Vorausschau die halbe Entfernung zum Führungsfahrzeug ist, reduziert sich das Kurvenschneiden bereits auf ein Viertel. Mit diesem Ansatz wird der Trajektorie des Führungsfahrzeugs gefolgt anstatt direkt dem Führungsfahrzeug. Dieses Verhalten liefert den Namen des Algorithmus: CUT - **C**ontrol **U**sing **T**rajectory.

0.4.2 Eigenpositionsbestimmung

Die oben beschriebene Eigenpositionsbestimmung durch Integration der Eigenbewegung hat mehrere Fehlerquellen:

- Die Annahme eines stationären Lenkverhaltens, einer flachen Fahrbahn und vernachlässigbaren Seitenwinds treffen offensichtlich nicht immer zu.
- Bei niedrigen Geschwindigkeiten liefert der Geschwindigkeitssensor falsche Werte (Diskretisierung der Raddrehzahlmessung).
- Der Lenkwinkelsensor liefert baubedingt Offsets des Lenkwinkels, die aufintegriert zu erheblichen Positionsfehlern führen.

Durch Ausnutzung von Bilddaten kann man diese Eigenpositionsbestimmung unterstützen. Dazu werden vertikale Landmarken und gerade Linien benutzt, soweit sie im Kamerabild gefunden werden können. Vertikale Landmarken sind Leitpfosten, Verkehrsampeln, Bäume oder Verkehrszeichen, die ähnliche Signaturen im Entfernungsbild haben. Diese werden als stationär angenommen, und über Verfolgung der Landmarken über mehrere Bilder hinweg wird die Eigenbewegung geschätzt. Dasselbe Vorgehen kann unter Beobachtung von geraden Linien entlang der Straße gewählt werden, wozu sich beispielsweise oft Spurmarkierungen eignen. Die Bewegungsschätzungen mit Bilddaten und Bewegungssensoren werden über ein Kalman-Filter fusioniert, das die Fehler der Schätzung unter der Annahme von Gaußschem Rauschen minimiert. Das Kalman-Filter schätzt Eigenposition und Ausrichtung des Fahrzeugs im ruhenden Koordinatensystem. Lenkwinkel und Geschwindigkeit gehen als Parameter in das Systemmodell ein. Die Positionen stationärer Landmarken bzw. die Geradenparameter der Spurmarkierungen werden als Messungen behandelt und tragen zur Korrektur der prädizierten Position bei.

0.4.3 Kartographierung

Wenn man eine erfolgreiche Eigenpositionsbestimmung vorgenommen hat, kann man alle 3D-Punkte in das unbewegte Koordinatensystem eintragen. Als nächsten Schritt

nimmt man auf dieser Basis eine Clusterung vor, bei der auch 3D-Punkte aus vergangenen Zeitpunkten berücksichtigt werden können. Damit kann eine Extrapolation stationärer Hindernisse vorgenommen werden.

Bei bewegten Hindernissen müssen die alten 3D-Punkte möglichst schnell gelöscht werden, da sonst diese Objekte einen "Schweif" von vergangenen 3D-Punkten hinter sich herziehen.

0.4.4 Kollisionsvermeidung

Bisher wurde zur Führung des Fahrzeugs nur der Pfad des Führungsfahrzeugs genutzt. Das reicht nicht aus in Situationen, bei denen ein Hindernis in den Pfad zwischen Führungsfahrzeug und eigenem Fahrzeug hineinragt. Um alle Hindernisse in der Umgebung zu berücksichtigen, wird der Pfad des Führungsfahrzeugs modifiziert, indem er als elastisches Band behandelt wird. Dabei werden Anfangs- und Endpunkt festgehalten. Hindernisse in der Umgebung des elastischen Bands üben abstoßende Kräfte auf das Band aus. Diese Kräfte werden durch Modellierung der Hindernisse als Potentialberge und durch Bildung des Gradienten entlang des Potentials berechnet. Auch Fahrbahnspuren werden als virtuelle Hindernisse modelliert. Innerhalb des Bandes, das als Teilchenkette mit Federn dazwischen modelliert wird, sorgen die Federkräfte für einen glatten Verlauf des Bandes. Eine Restriktionskraft stellt sicher, daß die Teilchen sich nicht entlang des elastischen Bandes bewegen. Die Federkräfte bekommen eine Ruhelage, die sicherstellt, daß in Abwesenheit von Hindernissen weiterhin dem Pfad des Führungsfahrzeugs gefolgt wird.

Diese Kräfte werden iterativ berechnet, bis sich ein Kräftegleichgewicht einstellt. Das Versuchsfahrzeug fährt den daraus resultierenden Pfad ab. Mit diesem Ansatz kann auch ein autonomes Fahren ohne Führungsfahrzeug implementiert werden, indem man den Zielpunkt des elastischen Bandes dem Hindernispotentialfeld frei aussetzt. Falls sich Hindernisse unausweichlich in den Weg zum Führungsfahrzeug stellen, wird durch Bremsen eine Kollision vermieden. Danach muß der Fahrer wieder das System aktivieren, um in den automatischen Fahrmodus zu gelangen.

Das natürliche Fahrverhalten eines typischen Fahrers dient als Anhaltspunkt für die Wahl der Hindernispotentialform. Diese ist sowohl von der Position als auch von der Geschwindigkeit des Hindernisses bezüglich des eigenen Fahrzeugs abhängig.

0.5 Simulation

Um die Wirksamkeit der im letzten Abschnitt vorgestellten Algorithmen zu überprüfen, wurden Simulationen durchgeführt. Die Bildverarbeitung, die Fahrzeugsensoren oder das Fahrzeug selbst können getrennt simuliert werden. Des weiteren können simulierte Hindernisse im Versuchsfahrzeug genutzt werden, um die Algorithmen unabhängig von der Bildverarbeitung zu testen.

0.6 Ergebnisse

0.6.1 Ergebnisse für das Trajektorienfolgen

Beim Einlenken in eine Kurve wurde das Kurvenschneiden bei hochdynamischen Fahrmanövern von $1.2m$ auf $0.5m$ reduziert. Bei zahlreichen Testfahrten konnten Verbesserungen des Fahrzeugfolgeverhaltens in allen Situationen erreicht werden.

Die Präzision des Fahrzeugfolgens ist durch die Meßgenauigkeit der Bildverarbeitung und durch die Trägheit des Lenkaktuators begrenzt. Des weiteren sind Tiefpaßfilter implementiert, die die Präzision des Systems zugunsten des Fahrkomforts reduzieren. Durch Einsatz eines Tiefpaßfilters werden plötzliche Änderungen des Lenkwinkels gefiltert.

0.6.2 Ergebnisse der Eigenpositionsbestimmung

Insbesondere bei Langsamfahrten wurden, im Vergleich zu der puren Integration der Bewegungsparameter, erhebliche Verbesserungen in der Eigenpositionsbestimmung erreicht. In Simulationen wurde eine Eigenpositionsbestimmung auf $15cm$ genau vorgenommen bei einer Fahrstrecke von $50m$ und unter Annahme von stark verrauschten Sensordaten. Hierbei wurden sechs Landmarken beobachtet. Die Genauigkeit der Positionsbestimmung steigt mit der Anzahl der beobachteten Landmarken.

0.6.3 Ergebnisse der Kollisionsvermeidung

Viele dynamische Szenarien wurden nachgestellt, um den Algorithmus sicher zu testen. In allen Fällen wurde dem Hindernis erfolgreich ausgewichen, selbst wenn Abweichungen von der Trajektorie des Führungsfahrzeugs von bis zu $1m$ notwendig waren. In Szenen ohne zusätzliche Hindernisse folgt das System weiterhin der Trajektorie des Führungsfahrzeugs.

Alle Algorithmen wurden erfolgreich im Versuchsfahrzeug erprobt und laufen in Echtzeit. Trotz des iterativen Verfahrens zur Kollisionsvermeidung liegen die Rechenzeiten pro Zyklus selbst im ungünstigsten Fall unter $50ms$.

0.7 Schlußfolgerungen

Mit oben beschriebenen Algorithmen kann die Präzision beim Fahrzeugfolgen erheblich erhöht werden. Darüber hinaus gewinnt das System an Sicherheit, da auf alle erkannten Hindernisse reagiert und gegebenenfalls ausgewichen wird. Durch einen begrenzten Erfassungsbereich und durch unumfahrbare Hindernisse ist unter Umständen eine manuelle Übernahme notwendig. Solche Situationen erkennt das System selbst und warnt den Fahrer.

Abstract

The motor vehicle of the future is equipped with an increasing number of intelligent systems. Anti-lock brakes (ABS) are considered standard today. The electronic stability program (ESP) system underlines the tendency towards more intelligence in the car. Further steps include driver assistance systems that facilitate driving by taking over parts of the driving task. For that purpose, the environment around the vehicle is perceived with appropriate sensors, such as cameras.

This dissertation focuses on the image analysis of traffic scenes using a stereo camera system as the main sensor. Its main application is autonomous driving in stop-and-go traffic. This thesis considers situation assessment, planning, and decision. In order to extract symbolic data from the camera data flow, it is necessary to develop some additional algorithms in addition to the well established image processing algorithms. The resulting data are used by control algorithms to send commands to the actuators for longitudinal and lateral vehicle guidance.

The thesis is divided into three parts. First, an algorithm is developed that allows the autonomous vehicle to exactly follow the path of a preceding vehicle. For this purpose, the paths of the autonomous vehicle and of the preceding vehicle are reconstructed in an inertial reference frame. Second, for the temporal integration of the acquired traffic scene data, a global map of the vehicle's environment is built. This map is the basis for the planning and decision stages that ultimately generate the data for longitudinal and lateral control. This necessitates a self-localization in an inertial reference frame. A Kalman filter fusing motion parameters of the vehicle and measurements of stationary objects from the cameras is employed. Third, the control commands which are issued to the actuators must guarantee a collision-free path. This collision avoidance capability is obtained using the notion of an elastic band for the path between the autonomous and the preceding vehicle. The elastic band is deformed by obstacles in its vicinity that exert forces on the band and consequently deform it. As a result, either a collision-free path is obtained or, in case of infeasible situations, a warning to the driver is issued who must take over the control of the vehicle manually.

In this dissertation, design, simulation, and implementation of the above mentioned algorithms are presented as well as results obtained under regular traffic conditions with a research vehicle.

Chapter 1
Introduction

This chapter gives an introduction to the dissertation. Important terms used throughout the thesis are defined.

1.1 Motivation

For a lot of people, driving is fun. However, it is also a tiring and dangerous task at times. Although driving is considered simple and boring most of the time, drivers have to be attentive all the time. They must always be prepared to react in a timely manner to changes in the environment.

Doing the right thing at the right time while driving is not always easy. Humans have limitations or even deficiencies when driving. For example, humans have a short attention span. One becomes inattentive when performing a monotonous task. Fortunately, this deficiency rarely leads to an accident. Car manufacturers have built counter measures to human driving deficiencies for decades. One deficiency of human driving is the skill to brake and to keep the car steerable at the same time. The anti-lock brake system (ABS) introduced in cars in the eighties assists the driver while braking to keep the car steerable at all times. It is standard equipment in today's cars. Further investigations showed that drivers press the brake pedal too moderately when braking in an emergency. The Brake Assistant introduced in 1997 assists the driver when performing an emergency brake to obtain maximum deceleration. Frequently, drivers enter a curve with too much speed and start skidding. The Electronic Stability Program (ESP) introduced in 1997 assists drivers to prevent skidding in curves by braking. All these systems only perceive the ego-vehicle. The adaptive cruise control (ACC) introduced in 1998 is a system that perceives the environment in front of the car and determines the distance to vehicles. It adjusts the speed accordingly.

Above systems are examples for **driver assistance system**s which are systems that assist the driver while driving. This is usually achieved by on-board sensors that perceive the ego-vehicle state and/or the vehicle's surroundings.

The systems introduced above show the two main driving forces for innovations in the automotive field. First, driving is made safer by ABS and ESP by assisting the driver in critical situations. Second, a convenience system such as an adaptive cruise control makes driving more comfortable and assists the driver in certain standard situations. A third key factor for innovations is the competition in the automotive field.

The need for more safety on the roads is obvious when looking at the following numbers: According to the "Bundesanstalt für Straßenwesen", 8758 people were killed in traffic accidents in Germany in 1996 [Straube et al. 97]. In the United States, 41967 people were killed in traffic accidents in 1997. In Japan, the situation is similar. There, 9640 fatalities in traffic accidents occurred in 1997 [oC98]. Normalized by the number of inhabitants, all three countries have a similar traffic fatality rate per inhabitant. About one half of these incidents could have been prevented by issuing a warning of pertinent hazards in time. These are accidents due to delays in detection of hazards. Warnings of pertinent hazards can be provided by driver assistance systems that perceive their environment. Another 27% of the incidents could have been prevented with driver assistance systems that control the vehicle by braking or steering [oC98, Iwasaki 99]. These are accidents due to mistakes in operation and judgment. Note that these numbers assume perfect detection capabilities of the driver assistance system.

Another problem is the constantly increasing number of motorized traffic participants. The number of registered vehicles in Germany increased from 17 million in 1970 to 36 million in 1990 and 40 million in 1997. This is leading to dramatically increased traffic density and congestion, since the length of the road network increased by less than 3% in the last 20 years. The system introduced in this thesis is especially suited to assist the driving task in a traffic congestion.

These numbers motivate research to increase safety, traffic flow, and convenience by introducing so-called **intelligent vehicles**, which represent the core of this dissertation. In this context, **intelligence** is defined as the capability to perceive, remember, adapt, and manipulate.

1.2 Intelligent Vehicles

1.2.1 Introduction

Intelligent vehicles are equipped with sensors and computing devices that allow the perception of the ego-state and the environment. From this input, intelligent vehicles decide on appropriate action [Williamson 98b]. For example a steer-by-wire vehicle senses the current vehicle state and is able to prevent skidding within physical limits. Other systems need to sense the environment of the vehicle. An increasing intelligence requires an increasing level of autonomy in the vehicle. Hence, many intelligent vehicles are labeled autonomous vehicles. Looking in related transportation fields, aviation or navy, autonomous components were introduced some years ago. An autopilot is standard equipment in passenger planes today.

For environment-perceiving sensors, dependability, perception range, and perception quality are extremely important. These requirements are reasons for the delay of automating functions in the car in comparison to factory automation. The outdoor world on roads is much more challenging to perceive than an indoor environment on a factory floor with controlled environmental conditions. Automotive sensors must maintain function under all weather conditions. Compared to the environment on the water and in the air, the traffic density on roads is much higher and the environment is much more complex.

The capabilities of today's sensors are still very limited compared to the human's perception capabilities. Only small parts of the human perception can be replicated. Nevertheless, a driver assistance system might have fewer perception capabilities, but it is always attentive, unlike humans.

Intelligent vehicle research has a long history starting with extensive work in the sixties. Consult [Fenton et al. 91] for an overview of the early work. Most of the work focused on Automated Highway Systems (AHS). AHS are systems enabling automated highway driving. Often, the intelligence is put on the roadway (e.g. inserting magnetic markers in the road) which requires a change of the existing infrastructure. Systems requiring a change of infrastructure are not considered here. A comprehensive analysis of many aspects including the non-technical aspects of intelligent vehicles can be found in [Lund 91].

The first vision-based intelligent vehicle applications were presented in the eighties. Consult [Dickmanns 98] and references therein for a short overview of the early work. Vision-based applications of intelligent vehicles are presented in more detail in the following subsection.

1.2.2 Intelligent Vehicle Applications

Intelligent vehicle applications are becoming increasingly popular. The systems range from safety features to convenience features. To limit this survey, only applications that involve vision-based input are described. The first driver assistance system brought to market that perceives its environment was an intelligent or adaptive cruise control. Other systems such as lane-keeping assistant, lane change assistant, intersection assistant, traffic sign assistant, route guide, traffic jam pilot, platooning system, and auto-pilot are currently being investigated. Such systems are the focus of research at several universities and car manufacturers and are described in the following.

Intelligent Cruise Control (ICC)

On a busy highway the main driving task is to stay in one highway lane and to follow the preceding vehicle at a safe distance while obeying the speed limit. This task requires constant attention, and a system that supports the driver to achieve this task increases safety and convenience. Since the basic task to obey a speed limit is performed by the cruise control, a system that also keeps a safety distance to leading vehicles is called intelligent cruise control.

For this purpose, the environment around the vehicle must be sensed. Such a system takes over part of the driving task on a busy highway: It keeps a certain maximum speed and adjusts the speed when a preceding vehicle is detected. A safety distance to the preceding vehicle on the highway is maintained. The regular cruise control is enhanced with this feature. The first system of this kind was brought to the Japanese market by Mitsubishi in the Mitsubishi Diamante in the mid nineties [Ludmann 95]. A laser radar was used for vehicle detection. A camera supported lane assignment. At that time, the system was too expensive and only moderately reliable. Only the idle engine torque was used for deceleration. Radar is used to detect the preceding vehicle

in the system Distronic introduced on the market by DaimlerChrysler in 1998. Brake and throttle are applied automatically to keep the safety distance.

Subaru [Sub99] introduced a system that uses two cameras to determine the distance to the preceding vehicle [Saneyoshi 96]. A detailed description for such a vision-based ICC system can be found in [Bohrer et al. 95]. There, monocular image processing is used for tracking vehicles at highway speed and a stereo camera system is used for small distances in stop-and-go traffic.

Summarizing, the main idea for intelligent cruise control is to follow another vehicle at a safe distance, also called vehicle following (see also Section 1.3).

Lane Keeping Assistant

On most roads, lane markings facilitate navigation along the road. Lane-guided highway-cruising, which is equivalent to driving along lane markings, is another simple driving task that can be automated.

Lane markings must be detected to perform this task. This is a well-researched domain (see e.g. [Dickmanns 98] and references therein). Being able to robustly detect the highway lanes, several applications are conceivable. A lane departure warner issues an audible or visual warning when the vehicle departs from its lane. Alternatively, a vehicle equipped with electronic steering can be autonomously guided along the highway lane [Franke et al. 94]. The systems takes over only part of the driving to keep the driver in the loop, i.e. alert. The necessary control algorithms to navigate the vehicle along the lanes are described e.g. in [Mayr 95].

Lane Change Assistant

Performing a lane change on a highway is potentially dangerous. Vehicles behind and on the sides must be accounted for before performing a lane change. The support for monitoring and detecting vehicles in the blind spot is considered especially helpful [Yoshioka et al. 99]. A warning is issued when a vehicle closes up fast in the blind spot. In Germany, due to the large relative velocities on highways, detection of vehicles far away is necessary [Knoeppel et al. 99].

Intersection Assistant

Intersections have a high risk potential for collisions. So applications that recognize intersections and vehicles closing in from the side can prevent collisions by warning the driver. [Mori et al. 95] presents a vision-based system that quantifies the risk potential for vehicles at an intersection. The necessary recognition of intersections is investigated in [Gengenbach et al. 98]. Tracking of the objects around the intersection is presented in [Heimes et al. 98]. There, the camera sensors cover a large field of view to perceive the intersection in front of the ego-vehicle.

Traffic Sign Assistant

Traffic signs inform the driver of the current speed limit and other vital information. Occasionally, the driver overlooks traffic signs or forgets what the most recent traffic sign denoted. So while driving on a highway it is beneficial to have a system that recognizes traffic signs and informs the driver of the currently imposed speed limit. Traffic sign recognition systems usually rely on color information and use the significant shapes of traffic signs. See [Ritter 92] for some early work running at near real-time speed. More mature systems running in real-time are described in [Ritter 96], [Estable 96], and [Paclik et al. 99].

Traffic Jam Pilot

Automated stop-and-go driving is the task of a traffic jam pilot. In contrast to the other systems introduced so far, this system operates at small velocities. It is considered a comfort or convenience system. Driving in a traffic jam is a boring driving task. Such a system is able take over the complete driving task, although the driver must still be prepared to take over manually at any time. Early work at very small velocities is presented in [Parent et al. 94]. Technically, it is a similar task to the intelligent cruise control, i.e. driving behind a preceding vehicle at a certain safety distance. Only the velocity range is different (see also Section 1.3). The preceding vehicle is subsequently called **leader vehicle** since it leads the ego-vehicle.

Platooning System

On highways, driving in convoys would be very useful for companies operating large truck fleets. Having a fleet of vehicles driving one behind the other is also called platooning. Platoon applications permit a convoy of guided vehicles to operate autonomously. Only the first vehicle must be driven manually.

An early vision-based system designed for platooning is presented in [Daviet et al. 96]. Platooning uses the same basic technology as a single vehicle-following system but asymptotic stability must be considered for platoons. Asymptotic stability is achieved when the control algorithms operate in such a way that no matter how many vehicles are connected all vehicles are controlled in a stable fashion. Commercial truck platoons are the most likely application of platooning techniques.

Route Guide - Navigation Systems with Digital Maps

Today, navigation systems provide digital maps of the current vehicle environment and are able to guide the driver to a specified location. With these maps and accurate positioning, more advanced applications are conceivable. With very accurate digital maps, even controlling a car becomes feasible [Scrase 99]. To account for unforseen obstacles an obstacle detection system is necessary in addition. The fusion of digital maps with local sensors such as radar data is necessary to achieve a reasonable safety level. Research on such fusion schemes is presented e.g. in [Jocoy et al. 99]. A first

spin-off of such activities could be a system that warns the driver of upcoming curves by using only digital map data.

Many other application systems can be envisioned by using global positioning and digital map data. These systems are beyond the scope of this thesis.

Auto-Pilot

Autonomous driving systems would be able to completely take over the driving task. Fully automated highway cruising in regular traffic is far from reality today. This is considered a longterm objective. A panoramic view around the car is necessary. Nevertheless, feasibility studies show the capabilities and potential of such systems (see e.g. [Ulmer 94]). Important recognition tasks to approach complete autonomy are presented in [Franke et al. 98]. These recognition tasks include the recognition of vehicles, cyclists, pedestrians, miscellaneous obstacles, traffic lights, and traffic signs.

1.2.3 Sensors for Intelligent Vehicles

A crucial part of an intelligent vehicle is its sensor system. It must be able to take over parts of the recognition tasks of the human eyes and constitutes the basis for the intelligent vehicle's behavior. Although we focus on optical sensor input here, the following list of sensors covers the most popular sensors for intelligent vehicle applications and is provided for completeness. The sensors to determine the ego-vehicle state such as velocity are not covered here. Only one hardware element out of this list, the camera, is a passive sensor, i.e. relies solely on reflected, not self-emitted, radiation signals. The other sensors, active sensors such as radar or laser range finder, measure distance by measuring the travel time of the signal from emission to reception, which is proportional to the distance. Envisioning a world of intelligent vehicles, interference among similar active sensors might become a problem [Williamson 98b]. Another important criterion for intelligent vehicles is the susceptibility to weather conditions. This short survey utilizes insights from [Masaki 98].

Millimeter-wave Radar

A modulated radar signal is sent out and the incoming reflections are sensed. The emitted signals are typically pulse-coded. Doppler shifts are measured via Fast Fourier Transform which yields information on the relative velocity of the measured object. The distance is measured via time of flight methods. Radar sensors are widely used in aviation to determine distances and velocities.

The reflection signals work well for most human-made materials. However, the reflected signal becomes very weak in the presence of highly fractal surfaces such as a fur coat or sand. Obstacles made out of these materials might not be detected. Radar works well at day and night. Rain and fog do not deteriorate the signal significantly whereas heavy snowfall causes problems.

One drawback is the limited total opening angle achievable at one time. Even a combination of radar beam signals provides only limited angular resolution.

To get a reasonable opening angle and several signals, the radar beam is usually scanned electronically. Scanning must be performed very quickly to avoid skewed range measurements. Alternatively, having several radar beams measuring simultaneously becomes prohibitively expensive.

Ultrasonic/Sonar Sensors

Ultrasonic or sonar sensors send out ultrasonic signals and sense the impulse response. Sonar systems are widely used in underwater vehicles. The signal range is limited to a few meters and the most common intelligent vehicle application is a parking assist system. In addition, the measurement rate is limited due to time of flight delays. The speed of sound is basis for the time of flight measurements. Moreover, imaging is not possible due to the low angular resolution of the sensor. Another problem are winds that "blow" the signal away.

Laser Radar

Laser range finders (laser radars) use time of flight measurements of modulated laser pulses to determine depth. A scanner system can build a range image, allowing post processing of the obtained range image. Laser scanners typically operate in the infrared wavelength domain. Due to intensity limitations to prevent eye damage, the range is somewhat limited ($< 20m$). In addition, these signals are susceptible to rain and fog since the intensity decreases quickly. A scanning device can be avoided by building an array of lasers but this is still very expensive for a reasonable imaging array.

Infrared Pattern Systems

Infrared pattern systems send out an infrared pattern with a projector and sense the projected patterns with infrared cameras. The distance for every part of the pattern is computed via triangulation, where the source and the camera are at different positions. Pattern correspondence for triangulation is established via image processing. Current systems deliver distance measurements up to $20m$, limited by the emitter signal which has to be low in intensity to be considered safe for the human eye. Near infrared cameras operate on regular CCD camera hardware with an infrared filter.

This method of depth determination has mostly been used in indoor environments.

Camera

Cameras, i.e. sensors exploiting vision cues, are exceptional to this list since they operate passively. The same sensor principle is applied to signals in the near infrared wavelength domain by attaching an appropriate filter to the camera. These systems always produce intensity grayscale or color images that do not deliver direct Euclidian measurements. Pattern recognition algorithms have to be used (see Section 2.1).

Another exception of vision sensors is the ability to detect lane markings, whereas the other sensor types measure distances which are not discriminative for lane markings.

Cameras capture visual information. The image acquisition proceeds roughly as follows: Photons reflected from the viewed scene enter the camera and go through a lens. These photons are converted to electrons on a photo-sensitive layer in the imaging plane, where the electric charges are read out and yield intensities.

The main camera types used to date are CCD[1] and CMOS[2] cameras. Typically both types deliver analog read out signals that have to be converted with a frame grabber to digital values. One intrinsic problem of CCD cameras is blooming, a phenomenon occurring when too many photons hit one CCD pixel in a short time. The conversion from photons to electrons yield too much charge which "leaks" into adjacent pixels and produces artifacts along the readout direction of the CCD chip.

Digital cameras are becoming increasingly popular since a frame grabber is no longer necessary, but other impediments such as the lack of accurate synchronization among several cameras delay their employment.

When using two cameras, 3D information is obtained by correspondence analysis and triangulation of the corresponding points. A stereo camera system delivers 3D information covering a solid angle depending on the field of view. Radar and laser scanners typically deliver 3D information for a scan line in one plane. In this work we use two CCD cameras to build a system that performs vehicle following at small velocities for automated stop-and-go driving.

1.3 Vehicle-Following Systems

The principle of following a leader vehicle has already been applied several times in the above mentioned applications. These applications include

- intelligent highway cruising,
- automated stop-and-go driving, and
- platooning.

The system described in this thesis performs automated stop-and-go driving. Vehicle following is considered the main task for such a traffic situation. We define **vehicle following** as the procedure of following a leader vehicle with the position of the leader vehicle being the main guidance. When an infeasible or dangerous situation is detected, the system has to recognize it and has to return control to the driver.

Extending the stop-and-go scenario, vehicle following at a safe distance always provides a safe driving mechanism. The main idea is simple: Follow the leader vehicle that just passed through that path and therefore was collision-free at that time. The notion is that the leader vehicle "creates" a collision-free path, which the intelligent vehicle follows. Even when the leader vehicle gets involved in a collision the following vehicle can still brake in time. So we extend our vehicle-following system beyond the stop-and-go velocity range up to the urban velocity range.

[1] Charged Coupled Devices

[2] Complementary MOS transistor technology, low power consumption

1.4 Limitations of Previously Proposed Solutions

The focus of this thesis is on vehicle following as required to perform autonomous stop-and-go driving. Several research institutes and universities have developed vehicle-following systems (see e.g. [Franke et al. 95, Daviet et al. 96, Lorei et al. 99]). Platooning includes the same capability but is usually applied to more than two vehicles following each other. None of the platooning applications known to the author consider the case of non-automated vehicles interfering with the platoon in traffic besides the leader vehicle. Moreover, most of the platooning systems require external infrastructure (e.g. [Guldner et al. 98]). Other traffic participants except for the leader vehicle are basically not detected and consequently, a collision with these traffic participants cannot be excluded. Hence, these applications are not suitable for regular traffic - they typically need a dedicated lane.

[Weber et al. 95] and [McLauchlan et al. 97] describe a vision-based system that performs distance-keeping following a leader vehicle and lane-keeping. Again, no other traffic participants besides the leader vehicle are considered. Similarly in [Franke et al. 98], the main focus of recognition is on the leader vehicle and no other traffic participants are used to determine the path.

If the distance between the leader vehicle and the intelligent vehicle is very small, this collision-avoidance component is not relevant. However, for vehicle following systems in regular traffic a minimum safety distance applies just as it does for manual drivers. Hence, a component ensuring a collision-free path at all times is deemed necessary. This collision-free path must be maintained at all times, even when it becomes necessary to depart from the leader vehicle's path. These situations occur when obstacles intruded the leader vehicle's path after the leader vehicle has passed through. When no collision-free path is found, a braking maneuver is performed.

Moreover, in almost all previously cited vision-based vehicle-following systems, only the current position of the leader vehicle is used for vehicle following and a circular arc is interpolated in between which can lead to significant deviations in arbitrarily shaped paths. Again, this becomes more significant with increasing distance between the vehicles.

In addition, only the current scene is analyzed and objects beyond the sensor range are not considered. This is helpful for systems with a limited field of view to extend their field of view.

1.5 Objectives of the Proposed System

The vision-based vehicle-following system described in this thesis overcomes the above mentioned limitations. The algorithms leading to the performance increase are the main part of the thesis and are described in Chapters 4 through 7. Within these improvements, a novel ego-position estimation algorithm is presented.

An extremely important criterion for algorithm selection is their suitability for real-time on regular off-the-shelf PCs. For our system, no specialized hardware equipment is necessary. In addition, this thesis avoids restrictions imposed on previous vehicle-

following systems. Specifically we perform vehicle following

- without dedicated leader vehicle, i.e. any vehicle can be followed,
- without dedicated lanes, i.e. anywhere,
- without external infrastructure, e.g. GPS, digital maps, magnetic markers, transponders, or vehicle-to-vehicle communication,
- with only local on-board sensors, and
- considering all obstacles in the traffic scene.

This strategy makes our system independent of external sources and allows the same performance everywhere, e.g. also in tunnels, where GPS signals cannot be received. The system has been tested extensively in urban traffic environments, where the variety of obstacles is extremely challenging.

1.6 Contributions of this Thesis

In this section, the main contributions of this thesis are described. First, a novel algorithm for following the exact path of the leader vehicle is presented. Second, an algorithm for ego-position estimation is described, which exploits previously detected features and is hence very fast. This precise ego-position estimation allows the creation of a global map in an inertial reference frame. One benefit of this map is an increased field of view compared to the current local field of view in the ego-centered reference frame. Third, a general path-based approach to collision avoidance is presented. This approach uses global knowledge of the whole path of the ego-vehicle, avoids local minima problems for feasible paths, and allows the fusion of vehicle-following and lane-following behavior. In addition, other behaviors such as lane-keeping at constant speed can be achieved.

Note that all algorithms presented here are tested with a vision-based autonomous vehicle. However, none of the algorithms make explicit use of vision-data; they rely on the extracted 3D information from a stereo-camera system. Any other range sensor delivering 3D data can be used to implement the algorithms. For an overview of alternative sensors, see Section 1.2.3.

The restriction of performing vehicle-following at small velocities is not essential for the proposed algorithms. Only the kinematic constraints such as imposing a minimum turning radius have to be extended to higher velocities. These resulting dynamic constraints such as limiting the occurring lateral accelerations have to be considered.

1.7 Thesis Overview

The remainder of this thesis is devoted to vision-based vehicle following. In Chapter 2 related work in this field is described, with a focus on real-time image processing and

intelligent vehicles More specific related work of the developed algorithms precede the relevant chapters.

Chapter 3 gives a description of the vehicle-following system upon which this thesis is based. The description covers the hardware, the software environment, image processing algorithms, the controller algorithms, the actuators, and the simulation environment.

A novel algorithm for path-based vehicle-following is presented in Chapter 4. In contrast to previous algorithms, the history of the leader vehicle is taken into account.

In Chapters 5 and 6, algorithms are presented, that allow a representation of the environment in an inertial reference frame. The former chapter focuses on ego-position estimation whereas the latter describes the actual mapping algorithm.

Chapter 7 constitutes the main part of this thesis describing collision avoidance strategies for vision-based vehicle-following systems. Besides reasoning for and describing of the chosen elastic band approach, details on the potential field choice and on human driving behavior are also included.

Results on the overall performance of the system are presented in the result sections of Chapters 4 through 7. Comparisons to similar systems found in the literature are made when available.

The final chapter includes a summary of the presented work, some discussion, and conclusions. The chapter closes by pointing out future work.

Why did we organize the thesis that way? In order to globalize vehicle following a view of the scene in an inertial reference frame is necessary. This enables us to perform path-based vehicle following (Chapter 4). However, determining the ego-position by only integrating the motion parameters introduces errors. So an ego-position estimation system using vision information is used (Chapter 5). Previously detected obstacles that are beyond the current field of view must be kept in a global map to consider them for path planning (Chapter 6). Finally, a dynamic collision avoidance component is necessary to avoid obstacles that interfered with the leader vehicle path since the leader vehicle has passed through that point (Chapter 7). To perform this task, an initial path to follow must be provided (Chapter 4) and the own position in a global reference frame must be known (Chapter 5).

Chapter 2
Related Work

This chapter gives an overview of related work in the fields of image processing and intelligent vehicles. Work specifically related to the algorithms presented in this thesis can be found in the "related work" sections of the respective chapters.

2.1 Image Processing

This section gives an introduction to related work in image processing. After a brief introduction to image processing and vision theory, a list of related work is provided. Due to the huge amount of literature, this list is limited to work relevant to real-time image processing for automotive applications.

2.1.1 Introduction

Image processing algorithms can be categorized with respect to a number of features. Image processing steps such as image rectification or image restoration are considered low-level image processing. The next step, detecting structure or objects in an image is called pattern analysis or pattern recognition. Specifically for visual signals, it is also called computer vision or machine vision. Algorithms using more abstract representations of the image, are classified as either image or pattern understanding. These distinctions are fuzzy throughout the literature.

Image processing started to grow with computer speed. An early and good book on pattern recognition is [Duda et al. 73].

A general introduction to image processing and image understanding can be found in [Shirai 87]. [Jähne 91] presents a more mathematical treatment of image processing. A detailed description of morphological methods of image processing is presented in [Zamperoni 91]. Most methods described here can be found in these books.

With increasing computing power, analysis of image sequences becomes more feasible. [Faugeras 93] covers image sequence analysis and presents a very good and rigorously mathematical introduction to three-dimensional computer vision.

2.1.2 Human and Machine Vision Theory

Human vision was compared to computer vision by Marr in the late 70's. [Marr et al. 80] describes a theory of edge detection where human edge detection is identified

as multi-scale edge detection. Edge detection via Laplacian of Gaussian, a contrast enhancing second derivative filter, delivers the so-called primal sketch, a 2 1/2 D representation of the scene. This edge detection scheme has a similar structure to a physiological model of simple cells.

A computational theory of human stereo vision is presented in [Marr et al. 79], which exhibits many similarities to the currently used stereo algorithms in the automotive field. However, algorithms on resolving ambiguities of multiple potential matches are hard to incorporate in real-time stereo algorithms.

2.1.3 Image Processing Algorithms

The list below covers computer vision algorithms relevant to intelligent vehicle applications. The following discrimination criteria are used:

1. Number of cameras: One, two or more cameras can be used.

2. Feature basis: Image processing algorithms are categorized based upon pixels, features (e.g. edges, variances or textures) or areas/regions here. Frequency information as opposed to spatial information is also used to generate features. However, these applications are not included here.

3. Number of dimensions: Static 2D or 3D image analysis have different applications. Time represents an additional dimension when performing image sequence analysis.

4. Frequency of the sensed signal: Image processing algorithms can operate on all kinds of image types including images from the visible spectrum, infrared images, and radar images.

5. Resolution of the sensed signal: Here, one distinction is between binary images and grayscale images. Another one concerns the frequency resolution. Grayscale images yield intensities without color resolution, color images maintain the color information by applying spectral filters.

6. Processing speed: Underlying control algorithms for intelligent vehicles require response times faster or as fast as the human reaction time. Otherwise, prompt response in a dynamic environment cannot be guaranteed. This leads to desired processing times of the sensor inputs of less than $100ms$. In less dynamic environments, such as on planetary explorations e.g. on Mars, cycle times may be larger. Also, information systems not prompting for immediate response (e.g. traffic sign recognition systems) may consume more time. All intelligent vehicle applications must run in real-time, i.e. must have a guaranteed maximum response time depending on the application.

7. Implementation: Implementation can be done in hardware which allows massively parallel and fast computation. Hardware dedicated to a specific task must be designed. Alternatively, the implementation can be done in software which allows more flexibility.

8. Application domain: Image processing algorithms are tailored to all kinds of applications, i.e. structured objects of all shapes can be detected. Automotive applications represent only a small field.

9. Active versus Passive Vision: In [Schiehlen et al. 94], a vision system is presented that controls the pan and tilt of the camera depending on the image input. In this way an active or purposive vision is achieved. The camera focuses on the part of the image that seems interesting. Without active camera control the same paradigm is applied by only analyzing a certain region of interest in the image depending on the current image input. Passive vision devotes the same attention level to every part of the image independent of the current input.

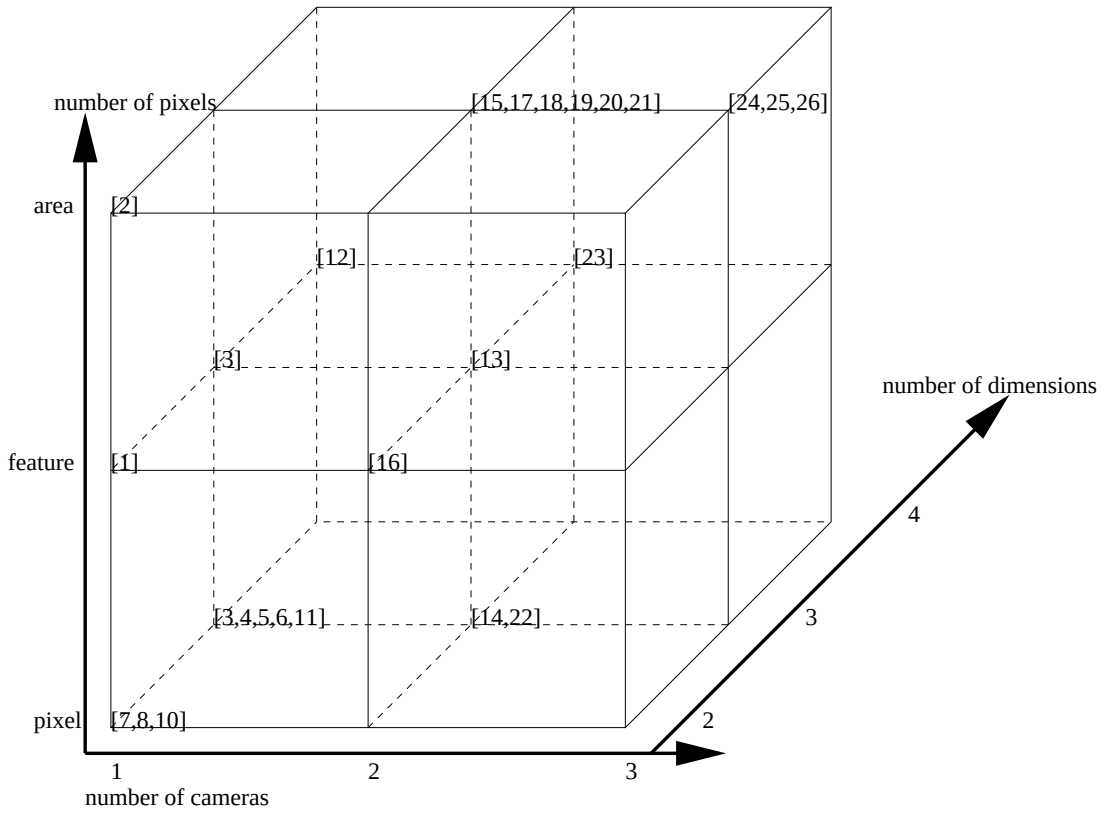

Figure 2.1: "Image cube" of the image processing algorithms listed here. Note that the relative majority of the algorithms operate with area-based stereo vision algorithms in three dimensions.

For object detection on roadways, color image analysis does not contribute much additional information compared to grayscale image analysis. Considering adverse weather conditions, such as night, rain, fog, and snow, color information is the first piece of information to deteriorate. Color blind drivers are just as good drivers as others when it comes to object detection although traffic light and traffic sign recognition might be impeded.

Binary images are more relevant in other domains (e.g. character recognition, track recognition in particle detectors [Dao et al. 99, Marks et al. 99, Gehrig 97]) than in

our application domain and are not covered here.

The processing speed is not a good means of discrimination since it depends on the hardware. The same holds true for algorithmic implementation: Running an algorithm in hard- or software is mostly a matter of cost and computation time.

The application domain is noted in the list below, but a division in application domains is not considered meaningful, since a lot of the algorithms are generic to the type of object they detect.

This section can still be divided in several ways. Therefore a cube to distinguish the respective features is deemed appropriate (see Figure 2.1). The axes of the cube are the number of cameras, the number of pixels for feature extraction (pixel, feature, area), and the number of dimensions (2 to 4 dimensions, where time represents one dimension).

The reference numbers in the cube refer to the numbers in the enumerated section below. The index numbers are sorted by the number of cameras.

Looking at the image processing cube (see Figure 2.1), it is striking that the majority of references is found in the "two-camera plane" performing area-based image analysis. This is due to the fact that our system uses a stereo camera system as well. Stereo vision needs no prior model knowledge to extract depth from images and is thus suitable for the task of detecting and measuring objects in wildly varying environments.

Approaches that rely heavily on complex models are not covered in this survey since they tend to become unreliable in complex traffic scenarios. Extreme variations of shapes of cars and other traffic participants occur. Moreover, strongly varying illumination conditions must be accounted for. Most algorithms presented here have weak model assumptions.

Other approaches not adequately covered are learning systems such as neural networks (see e.g. [Pomerleau 92]). The performance of such systems relies heavily on numerous and well selected samples. Since these systems are mostly concerned with classification or recognition rather than detection, they are not covered here. We are mostly concerned with **object detection**, i.e. finding objects of any kind, whereas **object recognition** also specifies the type of object. One example of a trainable detection system can be found in [Papageorgio et al. 99].

One more technique to extract depth from images not covered here is depth from shading. This technique exploiting knowledge of the illumination model is not suitable for real-time applications due to its complexity. In addition, sensor fusion schemes are omitted here (see e.g. [Langer et al. 96]).

An interesting general introduction to object recognition is given in [Liter et al. 96], where parallels to biological systems are emphasized. The use of models versus features is discussed.

2.1.4 Image Processing Using One Camera

In the following list, the most popular image processing methods are presented, including snakes, optical flow, motion stereo, and various stereo correlation methods.

1. The concept of snakes to segment non-rigid regions in an image is introduced in [Kass et al. 88]. Extensions can be found in [Blake et al. 92]. In that approach,

a Physics model of an elastic band is used and it is exposed to forces exerted by edge features of the segment boundaries. This yields a smooth segmented region. Since this is an iterative process, it is not straightforward to implement this algorithm in real-time. The mathematical model of such an elastic band is used in Chapter 7 for collision avoidance.

2. [Yezzi et al. 99] describes a region-based segmentation algorithm. It is based on the snake algorithm described by Terzopolous in [Kass et al. 88]. In contrast to the original approach, no feature extraction is performed to find the segment boundaries. All pixels in the image contribute to the snake shape (global evolution). There, the number of regions to be segmented has to be known.

3. [Matthies et al. 89] tries to estimate depth from image sequences using stereo motion. Stereo motion is equivalent to stereo vision provided that the relative camera position is known and provided that the observed scene is static. The recursive depth estimation is performed by an extended Kalman filter. Two methods of computing the depth, pixel-based and feature-based, are presented, and the pixel-based approach proves to be more robust and accurate. This approach is not applicable to dynamic environments since it assumes static objects.

4. Object detection based on color cluster flow in color images is presented in [Heisele et al. 95] and [Heisele 98]. It shows good performance even in occluded scenes but motion cues are needed. Non-moving objects in a scene remain undetected.

5. Object detection based on optical flow in image sequences is presented in [Enkelmann et al. 94] . One important drawback affecting all optical flow approaches is the dependence on motion cues.

6. Early work on optical flow can be found in [Bülthoff et al. 89]. Based on this work, [Mallot et al. 91] introduces inverse perspective mapping for object detection and simplification of optical flow computation. Inverse perspective mapping means transforming the 2D image into the 3D world assuming everything to be in one plane, i.e. the ground. Elevated objects appear distorted and optical flow computation enables object detection. This idea was taken over by several vision-based intelligent vehicle systems (e.g. [Broggi 95]). The "flat road assumption" is an assumption underlying most stereo vision systems presented here.

7. A reduced optical flow method looking at only one line in the camera, called time-slice analysis, is presented in [Heichel 95]. This method is applied to blind-spot monitoring. Complex object detection is not possible since only one spatial dimension is accounted for, but the algorithm speed is a great advantage.

8. Since cameras project 3D information onto 2D, depth information is lost. To recover depth, it is also possible to use the change of focus of the camera and determine depth through blurring [Krotkov 89]. This approach proves to be time-consuming and not very robust, but yields depth perception without stereo or additional models.

9. A lot of effort is also devoted transferring the time-consuming low-level vision algorithms into hardware (VLSI chips) to allow very fast computation (see e.g. [Wyatt et al. 92, Masaki et al. 94]). Another example for parallelizing simple algorithms in hardware is the GOLD system [Broggi 95]. There, the inverse perspective mapping [Mallot et al. 91] is implemented in hardware. Toyota has developed a real-time vision system processing several low-level algorithms in hardware (covering optical flow, stereo vision, and template matching) and allowing high-level algorithms in software to use these results [Ninomiya et al. 95].

10. The perspective effect in road scenes can be compensated for by sub-sampling [Graefe et al. 96]. The near region is sub-sampled (see Figure 5.1 for a typical traffic scene), so that all distance regions have the same number of pixels in horizontal direction. That yields a constant number of pixels per lane (constant lane width assumed). This procedure yields roughly constant object detection performance for a large range of distances.

11. Motion stereo is used in [Zhang et al. 97] to perform object detection with three different algorithms. Two of these algorithms only give qualitative information about the presence/absence of objects using one camera while the third one obtains the 3D location of objects above the ground assuming a flat ground similar to [Franke et al. 96] using a stereo camera system. While the first two algorithms are appealing because of their speed, they lack metric information which is vital for our application.

12. [Dickmanns et al. 94] presents a 4D approach to image processing problems using extended Kalman filters. This idea has been adopted by numerous researchers and is considered one of the crucial ideas in using image processing for control tasks. Multiple object detection and tracking is possible. Refer to [Dickmanns 98] and references therein for a complete review. Kalman filtering is also performed in our object tracking algorithm.

2.1.5 Image Processing Using Two Cameras

13. [Moravec et al. 99] performs stereo matching with the sum of squared differences correlation criterion. The next processing step builds a scale-tree of the image driven by image features, which allows a stereo matching on multiple resolution scales. The result is a dense depth image that is able to associate depth even with unstructured parts of the image. However, this algorithms takes about $10s$ per image on current Pentium PCs.

14. [Franke et al. 96] introduces a pixel-based bit-operator as the basis for stereo matching. The result is a relatively dense disparity map. With a flat road assumption and histogramming by distance, objects on the road are formed. A leader vehicle is identified as the closest object in front of the ego-vehicle which can be autonomously followed.

15. A powerful object detection algorithm is presented in [Saneyoshi 96], [Saneyoshi 94], and [Saneyoshi et al. 93]. In [Saneyoshi 94] typical problems for stereo approaches such as non-detection of low-textured objects are described. Some high-level algorithmic details are given in [Saneyoshi et al. 93]. Object detection is performed by histogramming the dense depth map. This algorithm relies on specialized hardware for real-time performance.

16. One of the first systems to apply stereo vision to autonomous vehicles can be found in [Bruyelle et al. 92]. The key to speed here is to use only one scan line to perform object detection. Stereo matching is performed by matching edges in the scan line. However, cluttered scenes cannot be analyzed with one scan line.

17. A promising stereo system is introduced in [Weber et al. 95]. Based on that work, a combined lane detection and object detection system is presented in [Taylor et al. 96]. Further work shows object detection with an application to distance keeping to the leader vehicle with the same algorithm [McLauchlan et al. 97] . The key difference to most other stereo systems presented here is the use of "ground plane stereo". Before starting the correlation, all pixels of one image are transformed with the assumption that they lie on a plane on the ground. So all points on the ground have zero disparity which is determined quickly. Protruding points have non-zero disparity and constitute objects.

18. A stereo vision system for off-road applications in an HMM-WV is presented in [Matthies et al. 95]. It is able to detect both objects and holes in the ground. In addition, infrared cameras are used with the same algorithm to allow night vision. Standard stereo procedures such as using multi-resolution image-pyramids, correlation via sum of squared differences and left-right-consistency checks are efficiently performed in specialized digital signal processors.

19. [Konolige 97] introduces a stereo engine enhancing contrast with the Laplacian of Gaussian. The sum of absolute distances is used as the correlation measure. Searching a limited range of disparities yields a simple manageable piece of hardware that delivers a full range image in real-time.

20. Based on [Broggi 95], a lane and object detection system is implemented [Bertozzi et al. 96]. For the object detection, a very similar stereo approach to [Weber et al. 95] is implemented making use of the inverse perspective mapping in the GOLD system.

21. In [Yoda et al. 98] a fast depth extracting stereo algorithm is presented that operates on a coarse level. The coarse depth image helps the system to determine specific regions of interest. These regions are subsequently analyzed with the maximum available resolution. Specialized hardware is used to build the image pyramid necessary for the correlation.

22. In [Bohrer et al. 95], a short-range stereo vision system called "vision bumper" is presented. The disparity between the two images are removed by inverse perspective mapping of both images assuming a flat road. Comparing the two warped

images yield the same image except for regions where the flat road assumption is violated. These areas constitute objects protruding from the ground. No metric information is obtained.

23. [Schiehlen et al. 94] presents a bifocal camera system. Two different focal lengths provide different resolutions. The camera system is attached to a rotating platform where gaze control is performed. Areas of interest are selected in saccadic motions similar to vision systems of vertebrates. No stereo evaluation by triangulation is performed. An experiment with saccadic movements for traffic sign recognition is presented.

2.1.6 Image Processing Using Multiple Cameras

24. [Schenk 98] uses the same algorithm described in this thesis to measure distances to boxes in a robotic environment. He uses three cameras to avoid occlusion problems. Note that only two cameras are evaluated at one time. Hence this system is labeled a multi stereo vision system. The relative error in distance measurement is less than 1%.

25. An obstacle detection system for long range is presented in [Williamson 98a] and [Williamson 98b]. The basic matching problem is solved with the sum of absolute differences (SAD). A discrimination between objects on the ground and protruding objects is achieved by comparing the regular stereo matching result with the ground plane stereo matching result. Protruding objects yield better matches with regular stereo matching, whereas features on the ground match better with ground plane stereo. Moreover, three cameras and a large baseline are used to detect objects far away and to perform multi-baseline stereo matching. This removes accidental correspondences.

26. A remarkable stereo engine that delivers a video-rate dense depth map is presented in [Kanade 95]. It uses up to six cameras and performs multi-baseline stereo matching to avoid ambiguities. The sum of absolute differences (SAD) is used as the basic correlation criterion and the sum over all SADs of all stereo pairs determines the depth for every pixel. A Laplacian of Gaussian filter is applied before correlation to enhance contrast. Due to the heavy computational load of the correlation, a video rate was only achievable by implementing the correlation in hardware.

2.2 Intelligent Vehicles

This section describes related work on intelligent vehicles. Only a list of the most prominent intelligent vehicle systems is provided here. This list is by no means complete and covers only publicly funded projects. Many car manufacturers also conduct research on intelligent vehicles within their company. This thesis is part of such a research effort. A survey on vehicle-following systems is given in Chapter 4.

2.2.1 Europe

PROMETHEUS

A breakthrough in autonomous vehicle research was achieved during the work on the EU sponsored project PROMETHEUS[1]. The final demonstration of all involved parties was held in Paris in 1994. Among other research demonstrators **VITA**[2]-II was presented [Ulmer 94]. It was able to autonomously drive on freeways, to obey speed limits, to perform lane-keeping, and to pass other vehicles on command. The underlying vision-based detection system used 18 cameras that made a panoramic view possible around the vehicle. A transputer system with 80 processors provided the computing power for the image processing. Lane detection, object detection, and traffic sign recognition were successfully implemented in real-time. Also, the symmetry of vehicle rears was exploited in the object detection system [Brauckmann et al. 94].

Another demonstrator within PROMETHEUS was **VaMoRs-P** of the University of Armed Forces in Munich. Four cameras, some on rotating platforms, provided an almost panoramic view [Dickmanns et al. 94]. VaMoRs-P and VITA II shared some algorithms.

French automobile manufacturers cooperated to build the prototype **CAROS**[3] [Raboisson et al. 94] performing obstacle detection with a monocular color camera. In addition, they implemented an intersection assistant [Bellon et al. 94] that is able to detect cars approaching an intersection.

MOTIV

MOTIV[4] is an ongoing EU sponsored research program focusing on methods to enhance traffic throughput and safety. [Lorei et al. 99] and [Fritz 99] describe projects on truck platooning in mixed traffic within this project. Both use vision to detect the leader vehicle with a fixed pattern. Hence this vehicle following procedure only applies to dedicated leader trucks. The vehicle-following principle is called electronic towbar, representing the idea of a truck towing another truck.

2.2.2 USA

California PATH Project

The PATH[5] project was initiated in 1986 under the leadership of the University of California at Berkeley. Various techniques for AHS are researched. An overview of the PATH program is given in [Shladover et al. 91]. Besides other projects there has also been extensive research in vision-based intelligent vehicles [Weber et al. 95, McLauchlan et al. 97]. Regarding the control algorithms a lot of work has been performed in the area of robust lateral control [Guldner et al. 96, Guldner et al. 98].

[1] **Pro**gram**m**e for a **E**uropean Traffic with **H**ighest **E**fficiency and **U**nprecedented **S**afety
[2] **V**ision **T**echnology **A**pplication
[3] **C**ollision **A**voidance by **R**oad extraction and **O**bstacle detection **S**ystem
[4] **Mo**bilität und **T**ransport im **i**ntermodalen **V**erkehr (mobility and transport in mixed traffic)
[5] **P**artners for **A**dvanced **T**ransit and **H**ighways

NAHSC

The NAHSC (**N**ational **A**utomated **H**ighway **S**ystems **C**onsortium) conducted extensive research on intelligent vehicles from 1994-1997 which resulted in a demonstration of various systems in San Diego in August 1997. This demonstration included the work within the PATH project. Both systems exploiting external infrastructure and systems relying on local sensing were presented. The Robotics Institute of the Carnegie Mellon University presented several vision-based intelligent vehicle applications with a series of demonstrators [Thorpe et al. 97].

After an intermission of two years, continuing work is performed under the project IVI[6]. So far, only conceptual work has been done.

2.2.3 Japan

The most money for intelligent vehicle research is provided in Japan. There, the very densely populated country suffers from chronic traffic congestion. The focus is on infrastructural technologies where most funds are allocated.

Advanced Safety Vehicle (ASV)

Japanese automobile manufacturers have developed Advanced Safety Vehicles (ASV) in cooperation with MITI[7] emphasizing research in safety. An overview of the activities including research vehicles and sensor investigations can be found in the Proceedings of the Intelligent Vehicles 96 Symposium [Hashimoto 96]. Besides CCD cameras, radar sensors and infrared cameras receive a considerable amount of attention. Algorithmic details are not given.

AHSRA

AHSRA[8] was founded in 1996 with the purpose of coordinating research activities in Japan. Numerous universities, 21 industry partners and the employees at AHSRA work closely together on advanced highways. The emphasis lies on research to make the road infrastructure intelligent, not the vehicle [oC98]. Current research activities are summarized in [Iwasaki 99].

[6] **I**ntelligent **V**ehicle **I**nitiative
[7] **M**inistry of **I**nternational **T**rade and **I**ndustry
[8] **A**dvanced cruise-assisted **H**ighway **S**ystems **R**esearch **A**ssociation

Chapter 3
System Environment

This chapter addresses the experimental setup used for the thesis. First, the employed hardware is described, followed by a description of the software architecture, the image processing algorithms, the planning & decision algorithms, and the controller algorithms. The remaining sections are devoted to the development and simulation environment.

3.1 Hardware

Our vehicle-following system is implemented on a Mercedes Benz E-class 420. The research vehicle is equipped with throttle, brake, and steering wheel that can be set electronically. The sensor system, i.e. the stereo camera system, is mounted on the rearview mirror. The image processing algorithms run on an Intel-based computer in the trunk. A Motorola-based computer hosts all interfaces to actuators (brake, throttle, steering wheel) and runs the controller software.

The Intel Pentium computer connected to the cameras is linked via PCI bus to a Stemmer RGB frame grabber where the images are acquired.

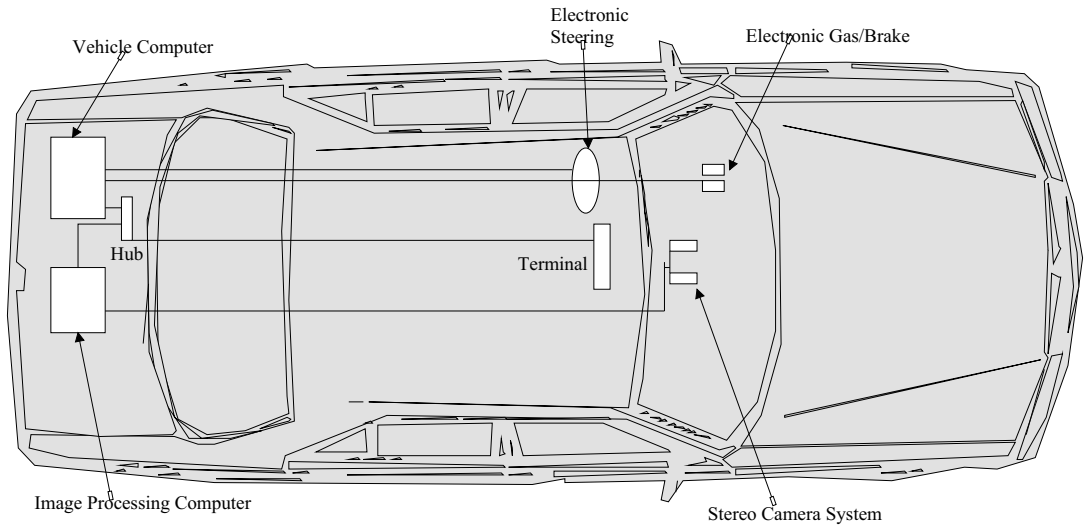

Figure 3.1: The E-class Mercedes with sensors, actuators and computers.

As shown in Figure 3.1, the trunk of the Mercedes E-class car hosts the vehicle computer (Motorola 604 computer with operating system Lynx OS), and the image processing computer (Intel Pentium II computer with operating system Linux). They are controlled from an X-terminal attached to the dashboard. The video output of the frame grabber is displayed on a monitor. The stereo cameras are located under the rearview mirror towards the wind shield. This spot provides a wipeable area for the cameras guaranteeing adequate viewing conditions. The two computers are connected via Ethernet over a communication hub.

Important vehicle data (e.g. vehicle velocity, steering wheel angle) are received from the vehicle's controller area network (CAN) bus.

3.2 Software

For our research vehicle we have two separate operating systems on separate computers for the controller and for the image processing algorithms. The communication between the two hosts is performed via PVM (Parallel Virtual Machine [Geist 94]). The controller algorithms running under LynxOS communicate with the CAN-Bus where the vehicle data is available. The image processing algorithms run under Linux. They are written in ANSI-C and are compiled using a GNU C-Compiler. The controller algorithms need the longitudinal and lateral location of the leader vehicle and the relative velocity. Extensions to this simple vehicle-following scheme are presented in subsequent chapters.

Figure 3.2 shows the processes involved in the complete system. The ellipses represent processes, whereas the lines represent the communication channels.

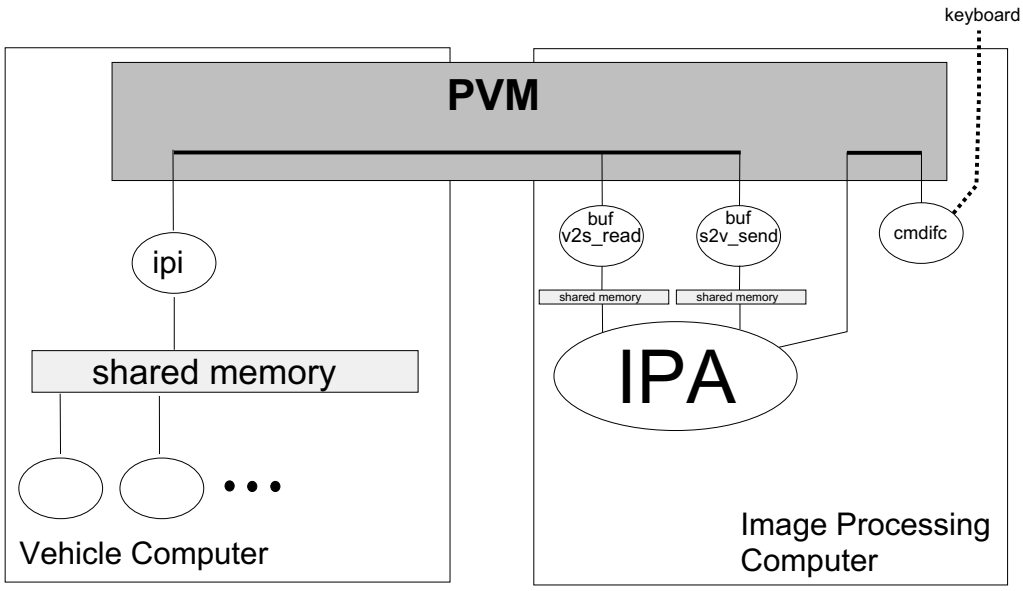

Figure 3.2: Processes in the complete system.

cmdifc Process on the image processing computer serving as the interface to the user.

ipa The process running the stereo vision algorithms.

s2v_send Buffer process for decoupling the communication from the "sensor" to the "vehicle".

v2s_read Buffer process for decoupling the communication from the "vehicle" to the "sensor".

ipi Process on the vehicle computer serving as the interface to the stereo vision algorithms.

Supplementary to the real-time environment, a standalone software system of the image processing algorithms is available that receives its image input from files (see Section 3.6).

Figure 3.3 shows a data-flow diagram of the whole system. Image acquisition, object detection, control, and the actuators are described in this chapter. The planning and decision module is the main contribution of this thesis and receives considerable extensions in subsequent chapters.

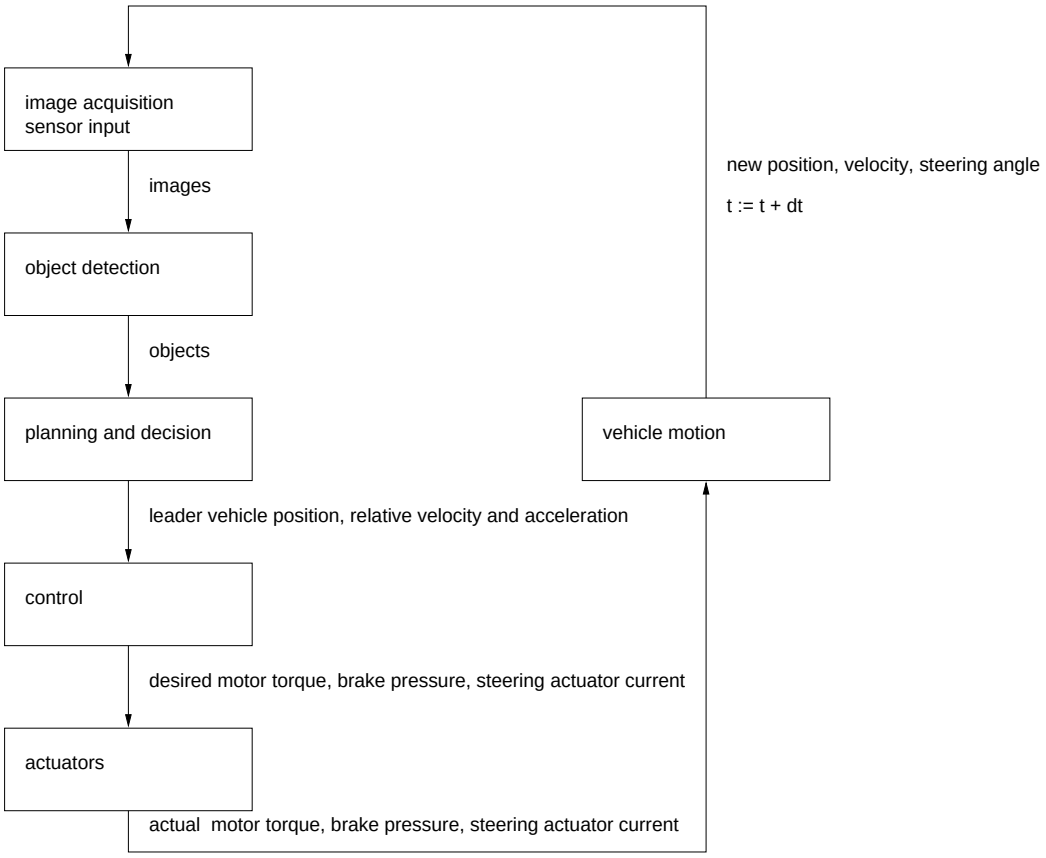

Figure 3.3: Data-flow diagram of the vehicle-following system.

3.3 Image Processing Algorithms

In this section the basic object detection algorithms are described. The main parts are image acquisition, feature extraction, feature matching, cluster analysis of the 3D points, and temporal analysis to confirm clusters over time. Additional algorithms are described that estimate the road geometry and detect lane markings. Camera calibration is covered in the final subsection. The result of the image processing algorithms is a consistent situation assessment which yields a list of detected objects with metric information.

3.3.1 Image Acquisition

The images are taken using a grayscale CCD camera made by COBRA with 768 by 568 pixels. The images are grabbed in interlaced mode at half the video frame rate ($12Hz$). Only half-images are analyzed. Images of the two cameras are grabbed pixel synchronously, using two out of the three channels of the RGB frame grabber.

The shutter time of the camera is automatically adjusted to the brightness of the recorded scene. Hence controlling the shutter is not necessary in the image processing step.

The focus is adjusted to infinity and the aperture is set accordingly to map all objects further away than $2m$ sharply. Closer objects appear blurry which is indeed desired for objects on the windscreen such as fly debris or raindrops. The focal length varied in the experiments between $12mm$ and $7.5mm$ which is equivalent to a 28° to 48° field of view.

3.3.2 Stereo Camera Geometry and Camera Model

Camera Arrangement

The stereo camera system used here has a geometry as described in Figure 3.4. The calibration of the external camera parameters (orientation) and internal camera parameters is described in Section 3.3.10.

The sensor coordinate system is illustrated in Figure 3.4. The origin of the coordinate system is in the midpoint between the two cameras. The Y axis points upwards. X and Z axis are illustrated in Figure 3.4, where the Z axis is parallel to the optical axes.

The Pin-Hole Model

A simple approximation for mapping rays of the real world into a CCD camera is the pin-hole model. (U,V) represents the CCD sensor coordinate system, the sensor center represents the origin of the image plane, U points to the right, V downwards, and (X, Y, Z) represents the camera coordinate system with its origin at the focal point (see Figure 3.5).

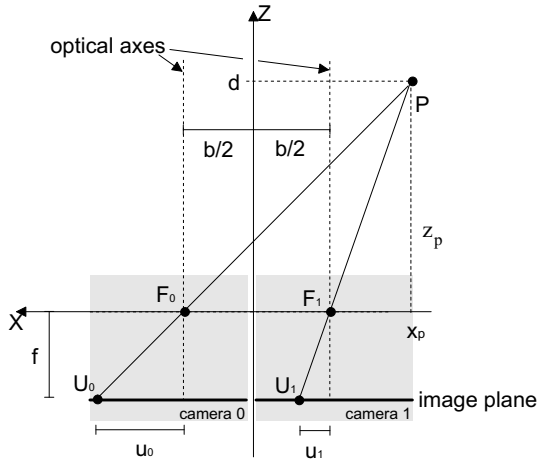

Figure 3.4: The sensor system viewed from above.

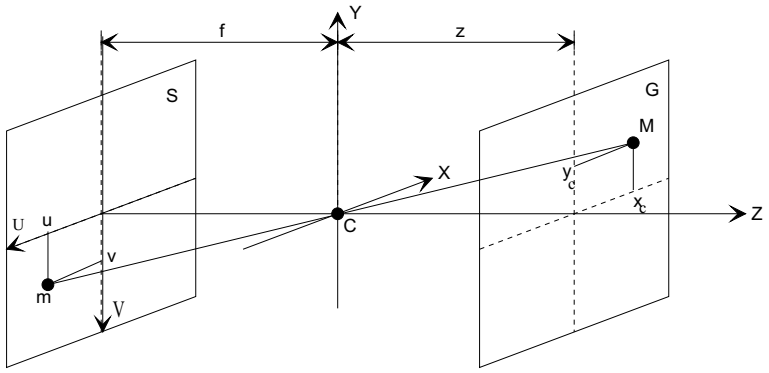

Figure 3.5: Central projection when imaging with a pin-hole camera. Note that the X-axis is offset by half the baseline from the sensor coordinate center which is not reflected here.

A point M (x_c, y_c, z_c) is mapped onto m on the sensor plane with the coordinates (u_s, v_s). The intercept theorem yields:

$$\begin{aligned} u_s &= \frac{x_c \cdot f}{z_c}, \\ v_s &= \frac{y_c \cdot f}{z_c}. \end{aligned} \tag{3.1}$$

The pixel densities in U- and V direction (k_u, k_v [Pixel/mm]) are obtained from the data sheet of the sensor chip. With the coordinate system described above, the image coordinates (u, v) of m are computed via

$$\begin{aligned} u &= u_s \cdot k_u = \frac{x_c \cdot f \cdot k_u}{z_c}, \\ v &= v_s \cdot k_v = \frac{y_c \cdot f \cdot k_v}{z_c}. \end{aligned} \tag{3.2}$$

One point in space is mapped onto two different locations in the two cameras. This

difference is called disparity. A more accurate camera model is introduced in Section 3.3.10.

Selection of the Camera Parameters

The selection of the focal length is determined by the required field of view, i.e. opening angle. Having the cameras at the rearview mirror, it is important to cover the full area in front of the bumper. Another parameter to tune is the base line between the two cameras. Too long a baseline leads to a large blackout zone, where no stereo correspondence can be established. On the other hand, a disparity of more than 1 or 2 pixels should be available for all interesting distance domains to maintain an adequate resolution.

A baseline of the size of the rearview mirror with approximately an opening angle of 40° results in a reasonable tradeoff of large field of view on one hand and accuracy in distance determination and detection reach on the other hand. With our camera chip, this is equivalent to a focal length of about $8mm$.

3.3.3 Feature Acquisition

Before performing the search for corresponding points in a stereo image pair, it has to be tested whether there is enough structure in a given area. Areas without structure (e.g. plain white walls) cannot be matched. An interest operator has to be applied in order to determine areas of sufficient structure. [Brandes 95] used a variance operator to find structure. Here we use the horizontal Prewitt operator, a simple edge detector. Only areas with enough structure undergo feature matching.

The exact location of the edge is irrelevant because the edge only triggers an area-based correspondence search where the exact locations of two areas in the respective images are found.

3.3.4 Feature Matching

Stereo correspondence analysis is greatly simplified by exploiting the so-called epipolar constraint [Faugeras 93]. This constraint limits the search for corresponding points to a one-dimensional search along the epipolar line, which is defined as the intersection of the image plane with the epipolar plane, defined by the optical centers and the image point. Taking one point in the left image, only the same line of the right image has to be analyzed for correspondence with our camera arrangement.

Two types of correspondences can be used: Feature-based and area-based. With a feature-based correspondence one performs a feature extraction on both images and tries to find feature correspondences in the next step. Area-based correspondences correlate the grayscale values directly.

Several correspondence measures are applicable. Typically correlation functions are evaluated. The sum of absolute differences of grayscale values or the sum of squared differences would be examples.

According to [Aschwanden 93], the most suitable correlation measure for image processing is the normalized mean-free cross-correlation function (CCFMF) defined as

$$CCFMF = \frac{\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\left(\overline{P^l(u_p^l+i,v_p^r+j)}\cdot\overline{P^r(u_p^r+i,v_p^r+j)}\right)}{\sqrt{\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\overline{P^l(u_p^l+i,v_p^l+j)}^2\cdot\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\overline{P^r(u_p^r+i,v_p^r+j)}^2}}. \quad (3.3)$$

Here $P^l(u_p^l, v_p^l)$ denotes an intensity value of a window in the left camera image and $P^r(u_p^r, v_p^r)$ a value of a window in the right camera image of size $n \cdot m$ along the epipolar line. The values $\overline{P^l(u_p^l+i, v_p^l+j)}$ and $\overline{P^r(u_p^r+i, v_p^r+j)}$ represent intensities of mean-free windows

$$\begin{aligned}\overline{P^l(u_p^l+i,v_p^l+j)} &= P^l(u_p^l+i,v_p^l+j)-\mu^l\\ \overline{P^r(u_p^r+i,v_p^r+j)} &= P^r(u_p^r+i,v_p^r+j)-\mu^r\end{aligned} \quad (3.4)$$

with μ representing the mean value of the window,

$$\begin{aligned}\mu^l &= \frac{1}{n\cdot m}\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}P^l(u_p^l+i,v_p^l+j),\\ \mu^r &= \frac{1}{n\cdot m}\sum_{j=0}^{n-1}\sum_{i=0}^{m-1}P^r(u_p^r+i,v_p^r+j).\end{aligned} \quad (3.5)$$

[Aschwanden 93] shows that the CCFMF is optimal with respect to Gaussian image noise. The result of the CCFMF is a value between -1 and 1. Higher values represent higher similarity. The window pair with the highest correlation value is the determined correspondence point. To suppress erroneous correspondences, a threshold is applied to the correlation values. Erroneous correspondences might occur due to occlusions.

To increase the accuracy of the correspondence, a widely used sub-pixel interpolation is applied (e.g. [Brandes 95]). The correlation function is approximated around the maximum by a parabola. The exact location of the maximum is the maximum of the fitted parabola

$$\overline{u_p^r} = u_p^r + \frac{\frac{1}{2}(corr_1 - corr_{-1})}{2\cdot corr_0 - corr_1 - corr_{-1}}, \quad (3.6)$$

where $corr_0$ is the maximum correlation value and $corr_1$, $corr_{-1}$ are the values around the maximum. This method increases correlation accuracy from 1 pixel to 0.25 pixel [Aschwanden 93].

The stereo analysis is performed on all pixels within the region of interest which is the whole image in the horizontal direction. Some information in the vertical direction is skipped where only the hood or the sky is imaged. To limit computation time, only a certain disparity range is searched for, typically from $2.5m$ on.

In addition, once a stereo correspondence is found, the disparity area to be searched in the next frame is limited to an area around the previously found disparity.

Feature Matching Problems

Stereo vision has some inherent disadvantages that also apply to human vision to a large extent.

Occlusion: Occlusion is due to the occurrence of a depth discontinuity which causes an obstructed view of the part of the scene behind the occluding edge, and thus is "observed" by only one of the cameras. Accidental correspondence might be established due to the texture of the part occluded by one camera. Occluded situations might be resolved using motion-based grouping [Beymer et al. 96]. In simple situations, a counter-check starting with the right image and looking for correspondences in the left images can help reduce false matches.

Repetitive Texture: When the texture is repetitive, such as the bricks in a brick wall, multiple possible correspondences exist. This can lead to wrong matches.

Lack of Texture: In real world scenes, most objects and surfaces are textured. This forms a basis for the strength of an area-based match by providing a statistically matchable pattern. However, part of the scene may be without texture and, in that area of the scene, no match will be possible.

Mirror: There may be a highly reflective mirror plane in front of a car, e.g. a large glass window. In that case, due to the reflection of the light, the distance to the reflected object will be measured, not the distance to the mirror itself.

3.3.5 3D Reconstruction

In order to extract depth from the found correspondences, a simple 3D reconstruction algorithm is applied, based on the simple geometry of the camera arrangement (see Figure 3.4). The distance of point P (x_p, y_p, z_p) w.r.t. the sensor coordinate system is computed via

$$\begin{aligned} z_p &= \frac{f \cdot b}{u_p^r - u_p^l}, \\ y_p &= \frac{(v_p^l + v_p^r) \cdot z_p}{2 \cdot f}, \\ x_p &= \frac{u_p^l \cdot z_p}{f} + \frac{b}{2} = \frac{u_p^r \cdot z_p}{f} - \frac{b}{2}, \end{aligned} \tag{3.7}$$

where (u_p^r, v_p^r) and (u_p^l, v_p^l) constitute the pixel location in the left and right camera of the corresponding point. v_p^l and v_p^r must be the same value due to our camera arrangement but when taking lens distortions into account the values might differ (see Section 3.3.10). Consult Figure 3.4 to view the underlying triangulation performed here.

In order to transform this point into the vehicle coordinate system, a simple rotation around the pitch axis and a translation from the sensor center to the road surface is performed. The essential underlying model to extract height y and lateral position x

is a flat road assumption. The orientation between the optical axis of the camera and the road, the pitch angle θ, is hereby assumed fixed, neglecting pitch movements of the car. This is a weak underlying model, where all objects are defined as structures protruding from the road. The z axis is parallel to the vehicle's longitudinal axis, the y axis points upwards, and the x axis runs left. The point $P\ (x, y, z)$ w.r.t. the vehicle coordinate system is computed via

$$\begin{aligned} z &= y_p \cdot sin\theta + z_p \cdot cos\theta, \\ y &= y_p \cdot cos\theta - z_p \cdot sin\theta + y_{sens}, \\ x &= x_p, \end{aligned} \tag{3.8}$$

where y_{sens} is the height of the camera sensor above the road. The vehicle coordinate system is the coordinate system used throughout the thesis.

The above stereo reconstruction is performed for all found stereo correspondences and the output is a list of 3D points. The CCFMF proves to be a very robust and accurate measure for stereo correspondence which yields very few mismatches and small errors.

3.3.6 Feature Combination

Once a list of reconstructed 3D points is obtained, preliminary objects can be formed.

To extract objects from 3D measurements, we apply a spatial clustering method to all 3D points except for the ones on or below the ground. Here the weak model of a flat road is used. All 3D points below a certain height are omitted. Note that beam reflections on a wet road surface appear below the ground and are thus discarded. The list of 3D points is traversed and two 3D points are connected when their Euclidian distance is below a certain threshold. This cluster connectivity is protocoled using coloring schemes known from Graph theory [Cormen et al. 90]. The straightforward implementation of this procedure has a complexity of $O(n^2)$ with n being the number of 3D points which is prohibitive for real-time applications when n is large. Using some heuristics (e.g. a 3D point on the far left part of the image is most likely not close to a 3D point at the far right of the image), the order of the algorithm stays the same but n becomes an acceptable number.

Cluster attribute data include dimensions (length, width, and height) represented as a bounding box, and a center location. The center location is computed by averaging all contributing 3D points. When computing the distance to an object, the contributing 3D points far away are weighted much less than the close ones since we want to measure the rear of the object.

3.3.7 Object Detection

Object Formation

One difficulty in combining data from different frames is the matching between consecutive frames, also called the correspondence problem. How can we match data from one frame to the next? Closely related to that issue is the tracking problem (see e.g.

[Sobiesk et al. 98]). We match objects from different frames by requiring them to be close together in space. This procedure works well for small scene changes between consecutive frames.

The clusters of 3D points constitute raw objects. When they are confirmed at geometrically nearby areas in consecutive frames they become an object. This avoids sudden responses to accidental formations of clusters. Once an object is established, the object is tracked with Kalman filters (see Section 5.5) and extrapolation is allowed. This is especially important in rain, when a windshield wiper occludes objects in a single frame.

To measure the center of the found object precisely, the symmetry of the object is determined. The symmetry operator used here is described in [Zielke et al. 93]. The combination of the bounding box center and the location of maximum symmetry is Kalman filtered, which yields the computed lateral position for every detected object.

The distance is computed using the 3D locations of all contributing 3D points with more weight given to the closer ones since we want to measure the rear of the object. The dimensions of the objects receive no further filtering except for extrapolation to keep a reactive behavior.

The whole object detection mechanism is summarized in Figure 3.6. After applying an interest operator, 3D points are generated for areas with sufficient structure (bottom). To form clusters, these 3D points are connected based on Euclidian distance (middle). If these clusters persist over time, objects are formed (top). In this graph, the complexity of the extracted features grows towards the top.

Merging and Splitting of Objects

Using the very basic cluster algorithm to form objects, traffic participants might be merged to one object when driving nearby and split later on again. Merging and splitting is detected by comparing the object data to the previous frame and looking for overlaps. Small interframe changes are assumed here. In addition, a model imposing a maximum width of a road vehicle is introduced and the bounding boxes of the vehicles are split if an extra-large bounding box occurs.

Obstacle Detection Quality

A quantitative evaluation criterion for sensor systems is introduced in [Matthies et al. 94] . For design and comparison, obstacle detectability is evaluated in two stages. First, the quality of the range data from stereo vision itself is investigated. Second, the obstacle detectability for a straightforward obstacle detection algorithm is evaluated. The range data quality is based on a stereo matching algorithm with the sum of squared differences as the correlation criterion. These systematic errors are taken into account in Section 5.6.4. According to [Aschwanden 93], other correlation measures deliver superior measurement quality. For obstacle detection, formulae for detecting false positives or missing false negatives are given. However, these formulae cannot be used for our obstacle detector since the algorithms are different.

Our chosen sensor has proven to be reliable under all reasonable conditions. Experiments included rain, fog, and snow during daytime or nighttime. The obstacle detection

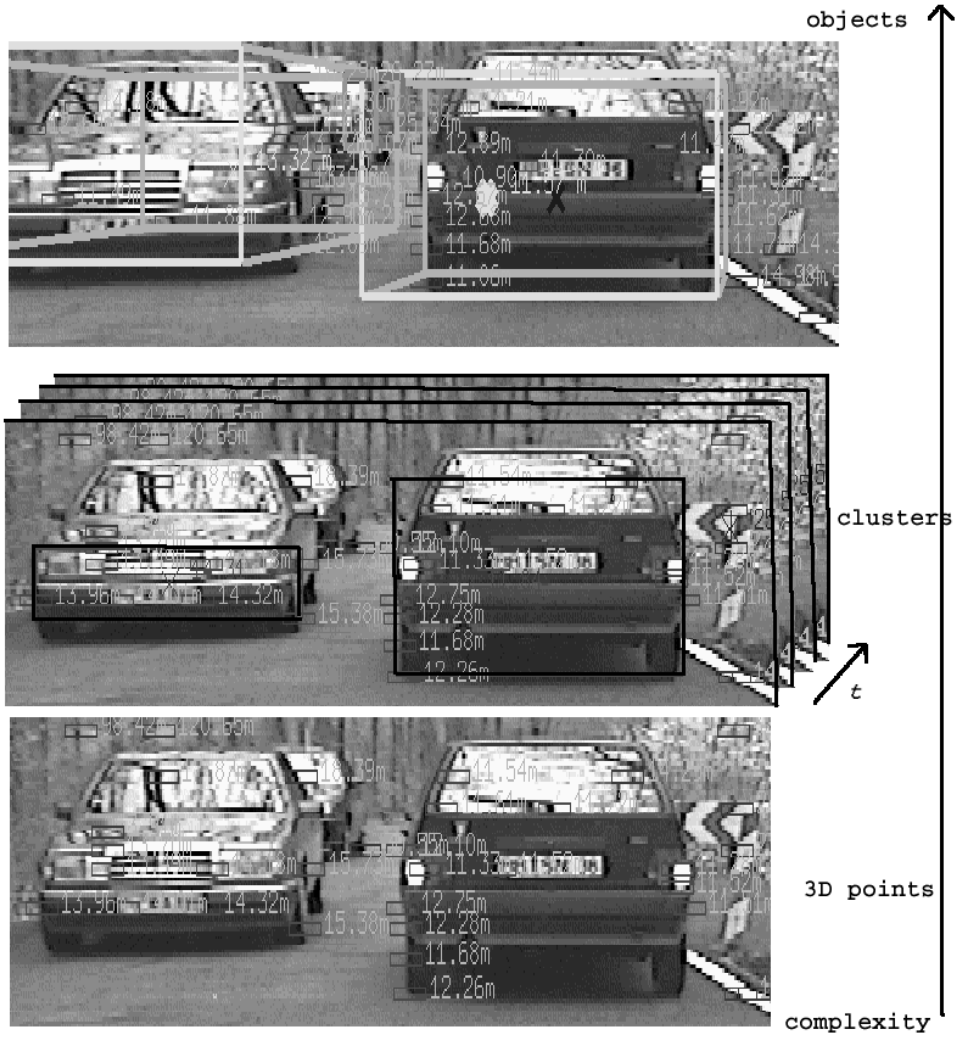

Figure 3.6: Feature hierarchy for the object detection process.

quality has been evaluated empirically. No situations were encountered where nearby obstacles were overlooked. Similarly, no phantom obstacles[1] were observed.

3.3.8 Lane Marking Detection

A lot of literature can be found on lane detection (see e.g. [Dickmanns 98] or [Kreucher et al. 99] for an overview). Since lane markings are of limited importance here, the detection algorithm is only described very briefly.

Lane markings are detected in the same manner as objects using the detected 3D points that lie close to the road surface. Additional road models limit the risk of accepting shadows and the like for the lane detection process. These lane points are connected and Kalman filtered. The final result of lane detection can be described as a polygon. This polygon is input to the collision avoidance algorithm introduced in Chapter 7.

3.3.9 Road Geometry Estimation

The simple flat road assumption is not valid under all circumstances. For one, braking and reaccelerating cause pitch movements. An even more serious effect is the non-flatness of real roads. Roads go uphill and downhill and have transitions. Just before a road goes uphill, structures on the road are mistaken as structures above the road and are treated as obstacles.

To compensate for this effect, a simple pitch angle estimator is implemented. We still assume a flat road which is a valid approximation for short stretches of road. But we estimate our pitch angle while we drive. This produces a line fit through the 3D profile of the road for the range of sight of the system.

The pitch angle estimate is based on the found 3D features on the road, and assumes a good starting value and little interframe changes. With the old pitch angle being θ, the pitch angle change is estimated to be

$$\Delta\theta_{est,f} = \arctan(\frac{y_{sens}}{z_f}) - \arctan(\frac{y_{sens} - y_f}{z_f}), \tag{3.9}$$

where F (x_f, y_f, z_f) is a 3D point considered to be on the road. Taking all n 3D points available on the road yields

$$\Delta\theta_{est} = \frac{\Sigma_{i=1}^{n} \Delta\theta_{est,i}}{n}. \tag{3.10}$$

The new pitch angle is $\theta_{new} = \theta_{old} + \Delta\theta$. This quantity is low-pass filtered via exponential smoothing [Gehrig 91].

3.3.10 Camera Calibration

An abundance of calibration algorithms can be found in the literature (see e.g. [Tsai 87] and [Lenz et al. 89]). For our research, we performed the following calibration steps.

[1] detecting an obstacle where there is none

External Camera Parameters

The external camera orientation of two parallel optical axes is achieved in the following way [Schenk 98]: Looking at a point at a far distance, the rays enter both cameras at the same angle due to

$$\lim_{z_c \to \infty} \left(\tan(\frac{x_c - b/2}{z_c}) - \tan(\frac{x_c + b/2}{z_c}) \right) = 0. \tag{3.11}$$

When superimposing the pictures of the two cameras, rotatory differences are compensated by bringing the horizons of both images into the same location. The yaw angle difference is compensated by looking at a point far away and bringing this point to the same location in both images.

This procedure yields an accuracy of about one pixel. Due to lens distortions and neglect of the principal point, the optical axes might still not be perfectly parallel, i.e. the convergence angle is not zero. A convergence angle of about 1° is realistic taking into account the observations made in [Lenz et al. 89]. In [Overington 92] investigations showed that up to a convergence angle of 6° disparity determination does not suffer significantly when assuming parallel sensor planes.

Internal Camera Parameters

As an extension to the pin-hole model we also took the first order radial distortion and the position of the principal point into account. The transformation equations extending the pin-hole model to a model with radial first order lens distortion can be found in e.g. [Schenk 98] or [Altunbasak et al. 99].

Due to computation time constraints, this transformation is only applied to the already found 3D points. Hence, this transformation can only be applied to a stereo camera system with very similar focal lengths and distortion coefficients. In addition, the epipolar lines are still assumed to be lines, which is only valid up to a certain lens distortion. The radial distortion coefficient is determined empirically by observing reconstructed 3D points of a structured wall. With a distortion coefficient of zero the wall appears pillow-shaped. With the correct distortion coefficient the wall is reconstructed in 3D as a plane.

3.4 Planning and Decision

After perceiving the environment, plans and decisions have to be made to achieve an appropriate response to the situation. This task is performed by the planning and decision module. Another term in the literature for this task is behavior control. This module determines the behavior of the vehicle. For the system described so far, the main task of the planning and decision module is to select the leader vehicle and to send the appropriate data to the controller.

3.4.1 Determining the Leader Vehicle

[Hofmeyer 99] describes an algorithm that finds a leader vehicle for control purposes. It takes the current steering wheel angle as a basis for the corridor in which to look for a leader vehicle.

We take a similar approach. The basic idea underlying our approach is to assume that little steering variations occur until one passes the current position of the leader vehicle. Consequently the path associated with the current steering angle is projected forward via the Ackermann model and the first object to intersect with this corridor is considered the leader vehicle [Gehrig et al. 98a]. All other objects are considered obstacles since they represent locations that have to be avoided. An extension of this leader vehicle determination is described in Chapter 4.

The current leader vehicle position coordinates, the relative velocity, and the relative acceleration are sent to the control module. These values will be modified with additional algorithms in subsequent chapters.

3.4.2 Human Machine Interface

The planning and decision module also hosts the human machine interface. While driving autonomously, no human interaction is required. When driving manually and a leader vehicle is available, i.e. when an object is found within the steering angle corridor, the driver has to engage the system to start the autonomous driving mode.

Control is given back to the driver when an infeasible or dangerous situation is detected, e.g. when the vehicle is not on a collision-free path anymore.

Control is also given back to the driver when the leader vehicle cannot be tracked anymore. Reasons for that might be that the leader vehicle disappeared out of the sensor's field of view or that the leader vehicle was occluded.

3.5 Controller Algorithms

This section describes the controller algorithms used for the vehicle-following system. After solving the recognition task, a desired path to follow has to be chosen. This path has to be converted to control commands for the actuators. The actuators steer the vehicle into the desired direction, closing the loop of recognition, decision, and control. The first part gives an introduction to control theory followed by descriptions of the longitudinal and lateral controllers, the vehicle sensors, and the actuators.

3.5.1 Introduction to Control Theory

[Föllinger 94] gives a general introduction to linear control theory, compares observers and Kalman filters and gives definitions of observability, controllability, and related terms. More advanced topics, especially non-linear control theory, are covered in e.g. [Isidori 95].

The controllers described here have mainly a linear structure relying on the three standard control blocks: Proportional (P), Differential (D), and Integral (I).

In this context, it is sufficient to know that a controller steers a controllable quantity from any currently observed value to a desired value. The relation governing the desired quantity is called the control law. By using a feedback proportional to the difference of current and desired quantity, Δq, a P controller is implemented. In the same manner, I and D controllers and combinations thereof can be created to obtain a feedback of the derivative or the integral of Δq. This is only applicable when the controllable quantity is directly observable, which is the case here.

3.5.2 The Longitudinal Controller

The longitudinal controller receives the distance, the ego-velocity, the relative velocity to the leader vehicle, and the relative acceleration as input. From that a desired acceleration is obtained that keeps a safety distance to the leader vehicle. For comfort, not a strict distance is kept but rather intrusion into the safety zone is allowed. Here, the control quantity is the difference between desired and measured acceleration.

The desired acceleration is fed into a PID controller and this value is converted to desired brake pressure for the brake or desired torque for the motor management. The motor management converts the desired torque to an actual throttle position. A detailed description of the longitudinal controller can be found in [Fritz 98].

The safety distance is defined to be $1.8s$ time distance to the preceding vehicle, i.e. the ego-vehicle always passes through a point on the road $1.8s$ later than the leader vehicle. Vehicles driving closer than $0.9s$ behind the preceding vehicle are subject to fines in Germany (see Bußgeldkatalog, violation number 404021). Thus, although control techniques would allow a much closer inter-vehicle spacing for electronically connected vehicles, a commercially available vehicle-following system would have to obey the safety distance in order to be ready for manual takeovers at any time.

3.5.3 The Lateral Controller

The lateral controller receives distance and lateral offset to the leader vehicle as input. From that a desired steering angle is computed. The difference between desired and measured steering angle constitutes the controllable quantity that is regulated to zero. The basic assumption for following a leader vehicle is that the measured rear of the leader vehicle and the rear wheels of the ego-vehicle follow through the same circular arc. This approach is presented in more detail in [Fritz 99]. Additional introductory information on steering models for cars can be found in [Zomotor 87] or [DeSantis 95]. The resulting steering angle derived from these kinematic relations is fed into a PID controller which yields the desired steering angle as output for the steering actuator.

This control scheme allows accurate vehicle following when going straight and when driving in a curve with constant radius. Deviations from the leader vehicle's path occur in other cases. Modifications and extensions to this lateral control scheme and alternatives are discussed in much more detail in Chapter 4.

3.5.4 Vehicle Sensors

Only the standard velocity sensor displayed at the speedometer is used. It is based on measuring the revolutions of the wheels, which is fairly accurate for velocities above $2m/s$ and good traction.

To sense the wheel angle, the steering wheel position is sensed. With the known transmission ratio, the wheel angle is obtained easily. For sensing the steering wheel position a rough position sensor is used. For more accuracy, an incremental sensor is used that detects the steering wheel movements accurately. However, initialization for the zero position is occasionally inaccurate and leaves a bias on the steering angle measurements.

3.5.5 Actuators

The actuators used to perform vehicle following are an electronic throttle, electronic brake, and an electronic steering wheel. The electronic brake receives a desired brake pressure, and the electronic throttle receives a desired torque for the motor management.

The electronic wheel used for the lateral control is similar to and based on a windshield wiper motor that receives a desired current which it converts to a new steering angle at the wheel.

The desired steering angle from the lateral control uses a vehicle model to determine a desired torque on the steering wheel to hold this wheel position. The current steering angle minus the desired steering angle yields a desired torque on the steering wheel to change to that position. Adding these two values delivers the total desired torque on the steering wheel and is translated to a desired current for the electronic motor. Again through a low-level controller, this desired current moves the steering wheel and hence the wheel. Since this motor is an inexpensive and only moderately accurate solution, latencies and delays must be taken into account when designing the lateral control.

3.6 Off-line Development Environment

The development environment in the laboratory allows the use of the same image processing algorithms off-line with additional debug information. We use Vista [Pope et al. 94] to visualize the images and overlay our detection results. Instead of using the frame grabber as image input we use images recorded while driving.

Whenever we encounter inexplicable detection failures we record the image sequence and analyze the sequence off-line. These failures are reproduced in the off-line environment and enhancements to prevent these failures can be made.

3.7 Simulation Environment

Several simulation models have been developed to verify the algorithms of this thesis. The complexity of the simulation environment gradually increases, putting more

realism into the simulation.

3.7.1 Path-Based Vehicle Following

The basic simulation environment is used to verify the vehicle-following properties. The leader vehicle is simulated as a point mass modeling the vehicle behavior as a one-track bicycle model. The same dynamics apply to the ego-vehicle. See Section 4.5.2 for a detailed description of the vehicle model. An agenda can be pre-selected for the leader vehicle and the ego-vehicle behavior is simulated with the actual controller code and vehicle dynamics.

The sequence in the simulation is identical to the real system depicted in Figure 3.3. The object detection system, i.e. the main sensor system is simulated. The planning & decision, control, actuators and vehicle motion are either simulated or the simulated object detection is applied to the research vehicle and real vehicle motion is generated. When the research vehicle follows the simulated leader vehicle, excellent verification and reproduction opportunities are available to the controller algorithms and for the planning & decision system. In addition, vehicle dynamics models and actuator models can be verified and tuned.

3.7.2 Ego-Position Estimation and Cartography

To verify the ego-position estimation algorithms, an extension to an arbitrary number of obstacles, i.e. landmarks, is performed. In addition, lanes can be added to the simulation.

Moreover, error models for the landmarks, lanes, and for the ego-vehicle sensors are introduced. Both biases and proportional errors are implemented with varying weights. Multimodal error distributions were not implemented. Bias errors are especially considered relevant for the steering angle, and proportional errors for the velocity. Obstacle measurements receive a noise signal uniformly distributed. No Gaussian distribution is used to avoid favorable distributions for the Kalman filter (see Section 5.5). The simulation setup can be used to verify additional trajectory-planning algorithms for parking maneuvers in the research vehicle.

3.7.3 Collision Avoidance

For collision avoidance, the dimensions of the point masses become relevant. So every obstacle receives rectangular dimensions with arbitrary orientation. Thus, the environment around the vehicle is simulated as a box world. The whole image processing part can be skipped and all relevant obstacle data is generated via simulation. The further extension to more complex convex hulls is left to the real image processing system since no principal benefit is expected for the simulation system.

The sensor simulation generates data in the x-z-plane, i.e. along the road surface. No vertical road modeling is performed. Tests in the following chapters justify this decision. Vertical variations of the road shape are accounted for by adapting the pitch angle (see Section 3.3.9). Roll movements (rotations around the vehicle's longitudinal

axis) are negligible when driving a car with lateral accelerations below $4m/s^2$. Laterally slanted roads affect the tracking accuracy of the controller and are considered by allowing a steering angle bias (see Chapter 5).

Chapter 4
Path-Based Vehicle Following

A novel algorithm for the lateral control of vehicle-following systems is introduced in this chapter. At the end of the introduction section, a chapter overview is given.

4.1 Introduction

A crucial task for steering an autonomous vehicle along a safe path in a vehicle-following scenario is the lateral control. The sensory input of such a lateral control are the position coordinates of the leader vehicle (e.g. [Franke et al. 95]). The following problem occurs: Due to the distance between the leader vehicle and the autonomous vehicle, the lateral control has to interpolate a path between the two vehicles. Using as a path either a straight line or a curve of constant curvature causes the autonomous vehicle to deviate from the leader vehicle's path. Before going into a detailed analysis of the problem, the solution is summarized in the following.

Given a system delivering 3D points of the leader vehicle with time tags, one has a handle to reconstruct the leader vehicle's trajectory. In addition, one has to compensate for the motion of the ego-vehicle by using its motion parameters. Once this transformation is performed, the position coordinates of the leader vehicle are available in an inertial reference frame. Knowing the position of the ego-vehicle in that reference frame, one can select any path point of the leader vehicle as input to the lateral controller. This simple approach increases the precision of vehicle following systems significantly and delivers a good starting point for collision avoidance systems that require an initial path.

How is this chapter organized? In Section 4.2, the main problem is stated. Section 4.3 gives an overview of related work. Section 4.4 introduces a new algorithm that solves the problem explained above and allows exact path-following. In Section 4.5 a mathematical treatment of the algorithm is provided. The impact of the new algorithm on the planning and decision module is detailed in Section 4.6. Modifications to the lateral controller are described in Section 4.7. Section 4.8 contains both simulation results and results from our research vehicle. The final section summarizes, discusses the results, and makes comparisons to other systems.

4.2 Problem Statement

Ideally in a vehicle-following scenario, the autonomous ego-vehicle follows exactly the path of the leader vehicle. Hereby, a **path** is defined by a series of points. A **trajectory** is a path that includes time information at every point in space.

Browsing through the described vehicle-following systems in the literature, the lateral control exhibits one serious drawback: With only using the current position information of the leader vehicle, one has to interpolate the path between the two vehicles. Both a straight line (trailer model) and a curve with constant curvature (circular arc) have been tried in previous applications. The arc takes the non-holonomic constraint of car-like vehicles into account. The non-holonomic constraint only allows motion along the vehicle's longitudinal axis according to the steering angle position at the front wheels. These interpolation approaches cause the ego-vehicle to deviate from the leader vehicle's path. The phenomenon scales with the distance to the leader vehicle. In platooning applications this behavior can cause the ego-vehicle to hit an obstacle such as the curb in a curve or a parking vehicle. The deviation Δd of the paths is illustrated in Figure 4.1.

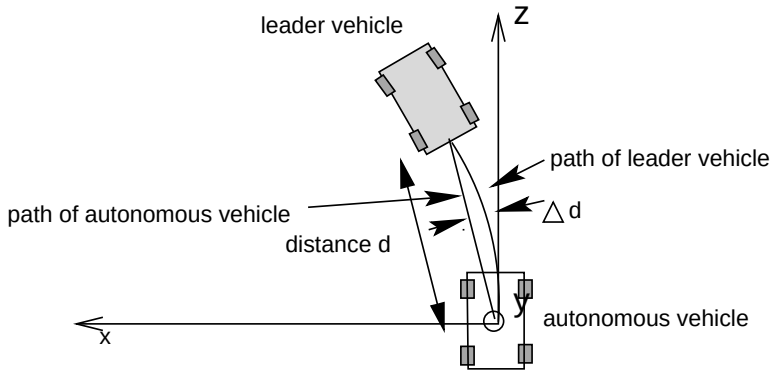

Figure 4.1: Path of the autonomous vehicle contrasted with the leader vehicle path. The autonomous vehicle uses a controller corresponding to the trailer model.

4.3 Related Work

Already an abundance of autonomous vehicle systems have been presented that have vehicle-following capabilities (e.g. [Franke et al. 98, McLauchlan et al. 97]). In these presentations a popular approach to laterally guide the ego-vehicle was either lane following (e.g. [McLauchlan et al. 97]) or vehicle following (e.g. [Franke et al. 95], [Daviet et al. 96]).

The first vehicle-following system based on stereo vision known to the author has been presented in [Kehtarnavaz et al. 91]. Detection of the leader vehicle was facilitated with a unique, actively illuminated pattern. Thus, only a dedicated leader vehicle could be used.

Juberts presents a vision-based road and car-following system in [Juberts et al. 93]. Due to transmission delays, a transformation via inertial sensors of the road model into the current image frame is needed and performed. Results show that up to $90km/h$ autonomous road following is feasible. A velocity-dependent lookahead is needed for the controller to track the lane center (see [Guldner et al. 98] and [Guldner et al. 96] for stability issues of that approach). The planned car-following system uses two different tracking algorithms for short and long (up to $30m$) range and optionally vehicle data from the leader vehicle in order to respond more timely to changes in the leader vehicle path. However, it only tracks the current leader vehicle position.

A different approach in vehicle following is to follow the path of the leader vehicle, not only the current position coordinates of the leader vehicle. This approach has been tried in applications, where vehicle-to-vehicle communication (e.g. [Tang et al. 96]) and typically also some absolute positioning system (e.g. [Kawabe et al. 96]) is available. This is called **path following**.

Platooning systems, that control platoons using absolute positioning methods (e.g. magnetic markers [Guldner et al. 98]) only communicating with the roadway are referred to as **point-following** systems [Shladover et al. 91]. These systems orient themselves only w.r.t. the road and do not allow much flexibility in platoon composition changes.

However, the approach introduced here applies vehicle following using a path to a vision-based autonomous system without using any communication infrastructure or absolute positioning system [Gehrig et al. 98b, Gehrig et al. 98c]. One similar approach can be found in [Sukthantar 93]. There, car-following at night is performed with a monocular camera system. For the lateral control, the trace of the leader vehicle is recorded in a global inertial reference frame. The necessary global position estimation of the ego-vehicle is delivered by an on-board controller in contrast to our research. The tracking strategy is similar to ours using a pure pursuit tracker with a certain lookahead distance [Wallace et al. 85]. The underlying lateral control is based on the so-called trailer model which exhibits the same properties as a trailer following the leader vehicle.

4.4 Introduction to the CUT Algorithm

For the sake of simplicity, we assume perfect sensor data and a very fast controller and actuator to explain the basic idea.

Unlike previous vehicle-following systems, the algorithm makes use of the time history associated with the leader vehicle. The idea of using the history of detected vehicle data has been used in [Schiffmann et al. 97] to determine the lane in front of the vehicle assuming the preceding car drives in the lane.

To determine the leader vehicle's trajectory it is sufficient to store the position coordinates of the leader vehicle and the motion parameters of the ego-vehicle over time. In order to transform the position coordinates into an inertial reference frame, one has to compensate for the motion of the ego-vehicle by using the ego-vehicle velocity and steering angle. Knowing the position of the ego-vehicle in the inertial reference

frame, one can select any path point of the leader vehicle as input to the lateral control.

However, real lateral controllers and actuators have a certain delay to guarantee good damping characteristics and compensate for measurement errors of the sensor data. The next path point to travel through must be a certain distance away from the ego-vehicle in order to allow the desired steering angle to have an impact on the ego-trajectory. To account for that, our algorithm always selects the path point that exceeds a certain lookahead distance measured from the ego-vehicle. A linear interpolation between path points is sufficient since their spacing is narrow. Furthermore, we adjust the lookahead proportional to the ego-velocity which yields a constant time difference between leader vehicle and ego-vehicle for all the path points passed through. This accounts for the delay of the controller and actuator. A constant lookahead part is added to account for the measurement errors of the sensors. Thus, the lookahead distance l has a constant part l_{min} accounting for measurement uncertainties and a part proportional to the ego-velocity t_{inc} to account for the delay incurred by the controller and actuator:

$$l = l_{min} + v_{ego} \cdot t_{inc}. \tag{4.1}$$

The mechanism is illustrated in Figure 4.2. Previous algorithms would have selected the path point at t_n.

This procedure enables us to follow arbitrarily shaped paths within non-holonomic constraints, provided an ideal actuator and ideal measurements are available. Due to filtering and actuator imperfections, path-following performance decreases for paths with highly varying curvatures. Assuming the normal driving behavior of lane following, road geometry guidelines limit the variation in curvature changes on a road (see e.g. [Behringer 94] for the guidelines of German road construction). Instantaneous changes in orientation are not feasible due to the non-holonomic constraint of a car-like vehicle. This non-holonomic constraint applies to the leader vehicle as well. Summarizing one can say, that vehicle kinematics, vehicle dynamics, and actuator delays limit the accuracy of the path-following algorithm.

The algorithm explained above is referred to as the CUT algorithm (CUT - **C**ontrol **U**sing **T**rajectory) in subsequent sections.

4.5 Quantitative Analysis

4.5.1 Transformation into an Inertial Reference Frame

For the subsequent analysis, the reference frame introduced in Section 3.3.2 is used. The origin is located at the sensor center (in the middle between the two CCD cameras) projected onto the x-z-plane on the ground. The global reference frame is equivalent to the local reference frame at the time of initialization and remains stationary. These reference frames are visualized in the previous figures.

The (x, z) coordinates of the leader vehicle, the ego-velocity and steering angle are needed to reconstruct both trajectories. We consider the problem in the x-z-plane. In a first step the coordinates of the ego-vehicle are transformed into an inertial reference

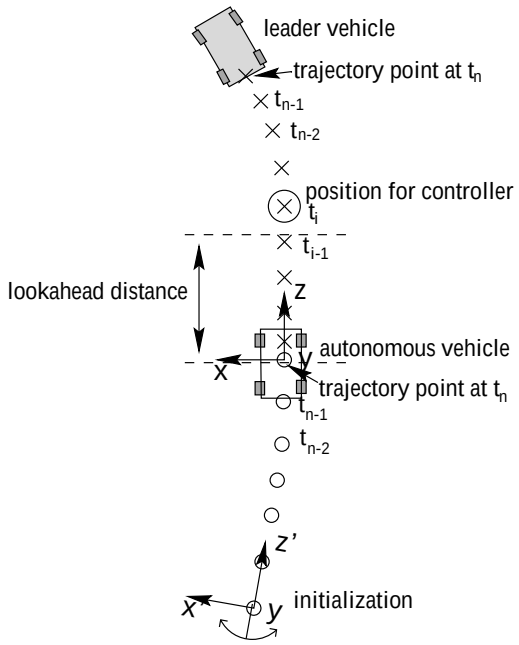

Figure 4.2: Mechanism of the CUT algorithm. Circles represent trajectory points of the ego-vehicle, crosses represent points of the leader vehicle.

frame:

$$\begin{pmatrix} x'_{ego,n} \\ z'_{ego,n} \end{pmatrix} = \begin{pmatrix} x'_{ego,n-1} \\ z'_{ego,n-1} \end{pmatrix} + M(\Phi) \begin{pmatrix} 0 \\ v_{ego,n} \end{pmatrix} \cdot \Delta t,$$
$$M(\Phi) = \begin{pmatrix} cos(\Phi) & -sin(\Phi) \\ sin(\Phi) & cos(\Phi) \end{pmatrix}. \tag{4.2}$$

Here $v_{ego,n}$ denotes the ego-vehicle velocity at time step n and Φ denotes the change of orientation since initialization ($\Phi_0 = 0$):

$$360° \cdot \frac{v_{ego,n} \cdot \Delta t}{r_{ego,n}} = \Phi_n, \Phi = \Sigma_{i=0}^{n} \Phi_i. \tag{4.3}$$

Φ_i reflects the change of orientation due to steering at time step i, n refers to the current time step, $n-1$ to the previous one. $r_{ego,n}$ is the radius of the ego-vehicle connected to the steering angle (see Section 4.5.2). The primed coordinates refer to the inertial reference frame.

In a second step, the coordinates for the leader vehicle are transformed into the same reference frame:

$$(x, z) \rightarrow (x', z')$$
$$\begin{pmatrix} x'_n \\ z'_n \end{pmatrix} = \begin{pmatrix} x'_{ego,n} \\ z'_{ego,n} \end{pmatrix} + M(\Phi) \begin{pmatrix} x_n \\ z_n \end{pmatrix}. \tag{4.4}$$

The above transformations are applied iteratively for every time step allowing the coordinates to be transformed to any point and orientation the ego-vehicle has passed through.

The third and final step consists of transforming the trajectory of the leader vehicle back into the reference frame of the ego-vehicle at the current time step:

$$\begin{pmatrix} x_i \\ z_i \end{pmatrix} = M^{-1}(\Phi)(\begin{pmatrix} x'_i \\ z'_i \end{pmatrix} - \begin{pmatrix} x'_{ego,n} \\ z'_{ego,n} \end{pmatrix}). \tag{4.5}$$

The trajectory points are labeled with i here to emphasize the fact that these trajectory points are associated with time-stamps prior to the current time-stamp ($n > i$).

From these trajectory coordinates (x_i, z_i), one is selected as input to the lateral controller.

4.5.2 The Vehicle Model

The radius of curvature r_{ego} is derived from the steering angle δ using the Ackermann model [Zomotor 87] extended by the self-steering gradient (see Figure 4.3). A detailed description of vehicle models can be found in [Bergman 65] or [Bundorf 67]. The Ackermann model is limited to small velocities and small lateral accelerations. In order to make the Ackermann model also valid for higher velocities, the self-steering gradient of the ego-vehicle is taken into account. This quantity subsumes the effects of moment of inertia around the y-axis of the vehicle and its cornering stiffness. It is easily obtained from data sheets.

$$x_c = \frac{a + b + SSG \cdot v_{ego}^2}{\tan(\delta)}, \tag{4.6}$$

$$r_{ego} = \sqrt{x_c^2 + b^2}. \tag{4.7}$$

SSG denotes the self-steering gradient in $[rad \cdot s^2/m]$. See Figure 4.3 for an explanation of the other used quantities.

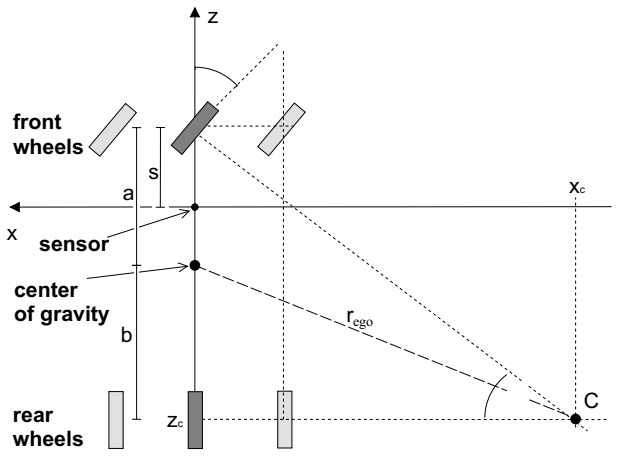

Figure 4.3: Ackermann Model (a: distance center to front axle, b: distance center to rear axle).

The model introduced here covers the steering behavior of a regular car for stationary steering behavior. In the integration step, the side slip angle is neglected. Hence, the vehicle always moves along the vehicle's longitudinal axis. Other models

[Zomotor 87] that include side slip and steering behavior beyond $4m/s^2$ lateral acceleration can be used to compute r_{ego} from δ to extend the validity of the vehicle model.

4.6 Planning and Decision for CUT

The CUT algorithm is an extension to the lateral control of the original algorithm presented in Chapter 3. Thus, the interface to the lateral controller needs to be modified. This is described in the following.

The algorithm presented here enables us to find the leader vehicle from previous frames more easily. Since we know the path of the leader vehicle (up to the previous frame), we use this corridor to look for the new vehicle in the current frame when driving autonomously. This works well for small interframe scene changes. With this strategy, also strongly varying path shapes of the leader vehicle lead to the desired leader vehicle. Previously, the leader vehicle might have been lost since the corridor projected by the steering angle would not have led to the leader vehicle when the path curvature strongly varied.

The driving states and their transition conditions are summarized in Figure 4.4.

The leader vehicle path as the search corridor is only used in the autonomous driving mode. In manual driving mode, the leader vehicle is always selected using the steering angle corridor. The reason is that in manual driving mode, the unknown driver's intentions guide the vehicle. Little changes in steering position are assumed. The same argument applies to the initial detection of a leader vehicle, where also the steering angle corridor is used to find a leader vehicle.

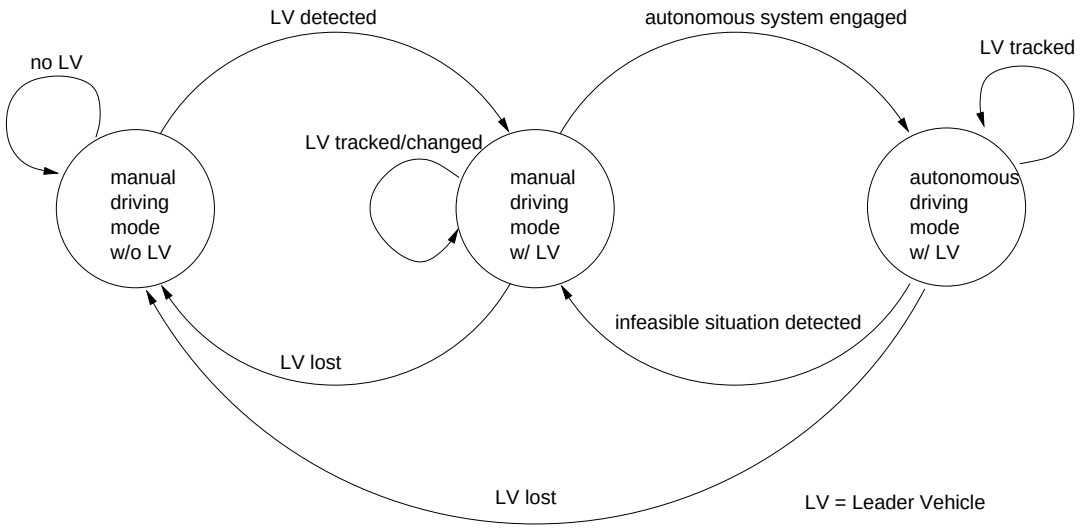

Figure 4.4: State transition diagram of the different driving modes.

4.7 The Lateral Control

4.7.1 Interface of the Lateral Controller

In previously presented lateral control algorithms [Franke et al. 95], the current x and z position of the leader vehicle has been the input to the lateral control. This also holds true for the controller presented in Chapter 3.5.3. These quantities are now replaced by the x and z position of a path point closer to the ego-vehicle determined by the lookahead distance. Using the same control law, the desired steering angle is computed. The output to the actuator is the filtered desired steering angle.

4.7.2 The Lateral Controller

The choice of the lateral control algorithm is uncritical with respect to the interpolation strategy since the distance to the target path point is comparatively small. Hence no significant deviations occur. The path interpolation strategy (e.g. circular arc, straight line) determines the control law.

However, a controller and an actuator with little delay and comparatively high dynamics are needed to follow arbitrary paths. Design issues of such controllers are beyond the scope of this thesis. These issues are addressed for the case of vision-based lane-following in [Kosecka 98] and references therein.

In our simulation and in the research vehicle, the lateral control algorithm interpolates a circular arc in the way described in Chapter 3.5.3.

In robotics, a generalization of the path following method is presented in [DDR99] compared to trajectory tracking. Extending these ideas taking vehicle dynamics into account is very tedious. Consequently, the PID controller described in Chapter 3.5.3 is used with above modifications.

4.8 Results

4.8.1 Simulation Results with an Ideal Controller

For the simulation a typical European street profile including a clothoid followed by a circular arc such as shown in Figure 4.5 is used. The velocity of both vehicles is $10m/s$ approximately constant over time and their distance is $25m$ constant over time.

The chosen street profile reflects a worst-case-scenario for vehicle following in our application with limited opening angle of the sensor and limited velocity range in urban areas.

The simulated controller assumes a circular arc and the actuator regulates the new steering angle within $80ms$. The lookahead distance is $12m$ from the sensor ($11m$ from the front axle). Simulation results are shown in Figure 4.6. Negative deviations are deviations towards the center of the curve. The dashed line denotes the old approach and the solid line marks the deviation using the CUT algorithm. Even less deviation can be achieved by further reducing the lookahead but this also compromises the damping characteristics of a realistic lateral controller.

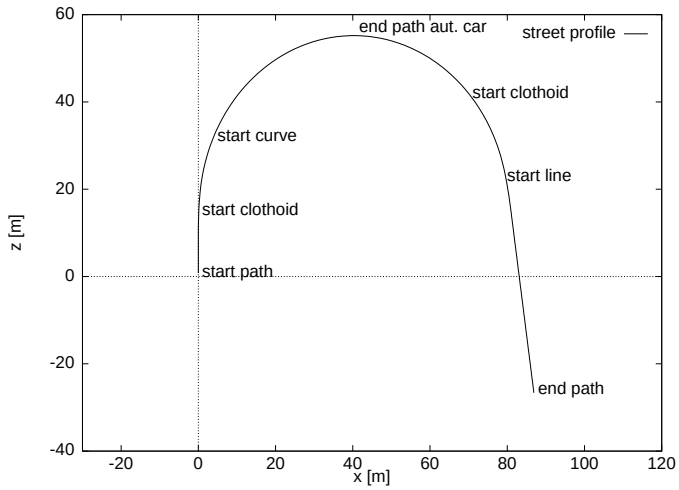

Figure 4.5: Route used for the simulation (coordinates in m).

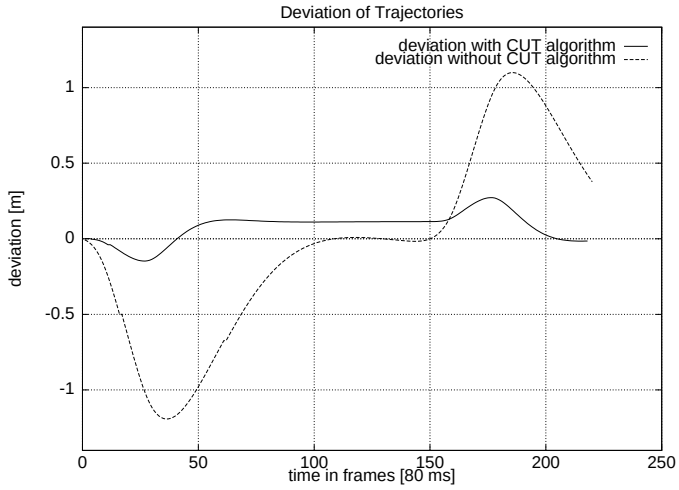

Figure 4.6: Simulated deviation of the trajectories between the ego-vehicle and the leader vehicle. The solid line denotes the deviation using the CUT algorithm, the dashed line indicates the standard approach.

The above simulation did not take any uncertainties into account. For a simple noise model, we take a noise signal of 3% of the distance uniformly distributed in z and of $0.5m$ uniformly distributed in x. That reflects the fact that position measurements with 3D sensors typically have an error proportional to the distance. The noise in x-direction is assumed to be independent of the distance since other effects incurring noise such as occlusion dominate the error. No filtering of the sensor data was performed. Figure 4.7 shows the results of the simulation including noise. No significant changes occur compared to the perfect sensor data.

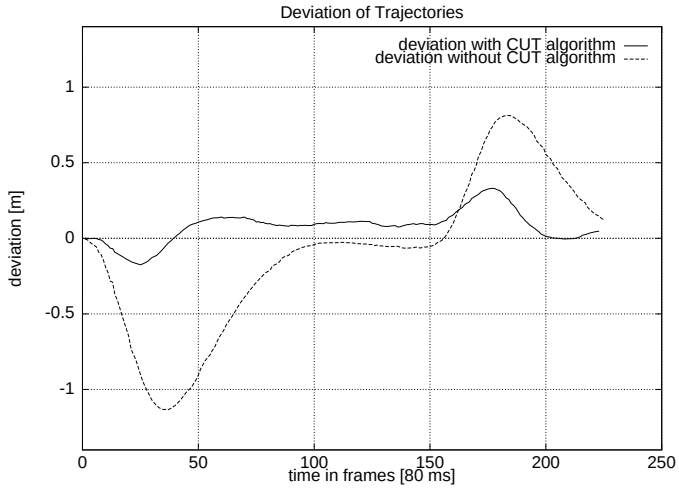

Figure 4.7: Simulated deviation of the trajectories of the ego-vehicle to the leader vehicle with noise. The simulated noise is 3% of the distance uniformly distributed in z and $0.5m$ uniformly distributed in x.

4.8.2 Limitations of the Simulation

Above simulations illustrate the principal mechanism of the CUT algorithm. However, it does not reflect reality in several respects:

- Real 3D sensor data needs filtering. The Kalman filter [Kalman 60] proves to be a good means of filtering for data that can be associated with a dynamic model. It is used for the trajectory measurements of the leader vehicle in our autonomous vehicle. The dynamics are modeled using position, velocity, acceleration and jolt. More details on Kalman filters are presented in Chapter 5.

- The lateral controller and especially the actuator have slower control characteristics than the simulated controller and actuator. Hence more delay occurs than in the simulation.

- The motion parameters used for the transformation have measurement errors. In addition, systematic errors occur when the Ackermann model is violated, e.g. when side slip occurs. Other effects include side winds and roads slanted along the roll (z) axis. These errors are bounded and do not accumulate over time since

only the relative position between ego-vehicle and leader vehicle is used. The ego-velocity and steering angle are quite accurate since effects such as skewing are negligible in the urban velocity range. Although measurements of the steering angle are inaccurate for small velocities and hence yield inaccurate path curvature, their contribution to the uncertainties is small because only the product $v_{ego} \cdot f(\delta)$[1] appears in the transformation equations. Also, the imprecise velocity measurements for velocities close to zero do not contribute much to errors since only the integral of the velocity contributes to the transformation. For higher velocities, the Ackermann model becomes invalid for comparatively small steering angles. With the introduction of the self-steering gradient, this effect is taken into account. To even extend the model beyond the stationary steering behavior, more refined models have to be used. The most dominant error of ego-motion estimation appears in a scenario where the car is driven at a small velocity over an extended period of time. This problem is addressed in Chapter 5.

- 3D imaging devices have limited opening angle. For extremely curved trajectories the leader vehicle might fall out of the field of view of the sensor device. This is not taken into account in the simulation.

- The noise model used in the simulation is not realistic. Effects such as occlusion are not taken into account.

4.8.3 Simulation Results with a Realistic Controller

We used our actual controller code and an actuator model to simulate a more realistic behavior of the lateral control. The delay between desired steering angle and actual steering angle is about $350ms$, and all steering commands are low-pass filtered. Moreover, the real longitudinal controller is used, which leads to shorter distances between ego-vehicle and leader vehicle. This reduces the effect of curve-cutting due to smaller lateral offsets. We used the same profile from Figure 4.5 and applied the new controller model. On average, the distance between ego-vehicle and leader vehicle was about $15m$ being less towards the end of the simulation. Figure 4.8 shows a result of this simulation. The lookahead distance was set to $9m$. The standard approach still cuts the first corner by $50cm$ whereas the CUT algorithm stays on track. While driving through the curve, both approaches exhibit a lateral deviation of about $25cm$ due to the actuator delays. The overshoot at the end of the curve does not occur because the distance between the two vehicles is about the lookahead distance at that point.

4.8.4 Real-World Results

Our test vehicle, the Mercedes-Benz E420 described in Chapter 3, is equipped with actuators for throttle, brake, and steering wheel. The sensor system consists of two CCD-cameras using stereopsis. The results presented here are obtained with cameras having an opening angle of about 30°.

[1] Plugging Equation 4.6 into Equation 4.7 and that result into Equation 4.3 yields Function f.

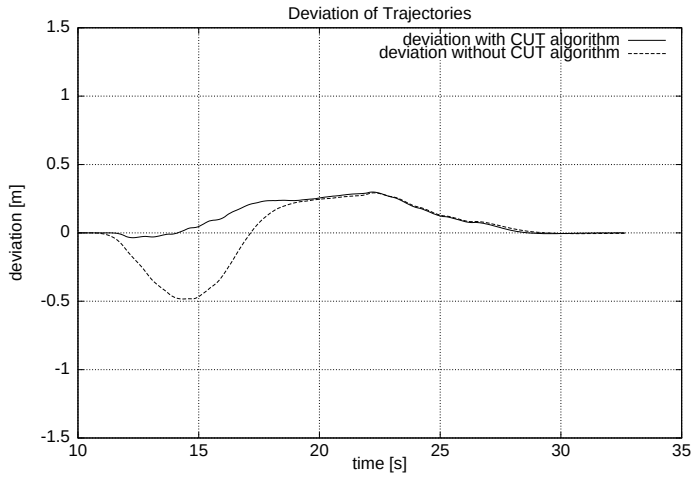

Figure 4.8: Simulated deviation of the trajectories between the ego-vehicle and the leader vehicle using the full controller code. The solid line denotes the deviation using the CUT algorithm, the dashed line indicates the standard approach.

The center of the leader vehicle is measured as described in Chapter 3.3.7. Other approaches in truck platoons use a fixed pattern to determine the center of the leader vehicle [Franke et al. 95]. The lateral x and longitudinal z coordinates of the leader vehicle center are sent to the CUT algorithm.

In order to have the same initial conditions and the same behavior of the leader vehicle we chose to use another simulation in the vehicle that creates a virtual leader vehicle. We pick the same profile as in the off-line simulation for the leader vehicle (only the first part of it due to limited space). The distance to the ego-vehicle is around $25m$. The lookahead distance is set to $11m$.

The controller presented in Section 3.5.3 is used. For passenger comfort and to compensate for measurement errors the controller performs extensive low-pass filtering. The intrinsic frequency of the vehicle of about 6 Hz has to be filtered out to avoid resonances. In general, there is a tradeoff between ride comfort and path tracking precision. In order to guarantee a comfortable ride, extensive smoothing has to be performed which lacks tracking precision for leader vehicle driven in a sporty fashion. The controller and actuator combined have a delay of around $350ms$ between desired steering angle as computed from the x and z coordinates and measured steering angle due to filtering and delay of the actuator.

Figure 4.9 shows a plot of the deviations of the paths between the leader vehicle and the ego-vehicle. Negative deviations refer to deviations towards the inner side of the curve (cutting the corner). As expected the standard approach of vehicle following results in cutting the corner when driving through a clothoid. The path of the ego-vehicle exhibits a slight overshoot due to the actuator delay. A decreased lookahead distance leads to less damping and necessitates a faster actuator response.

In addition, the new CUT algorithm has been tested under real traffic conditions. No instabilities or oscillating phenomena were observed. A comfortable ride was provided at all times. Storing several traffic scenes with lateral and longitudinal controller

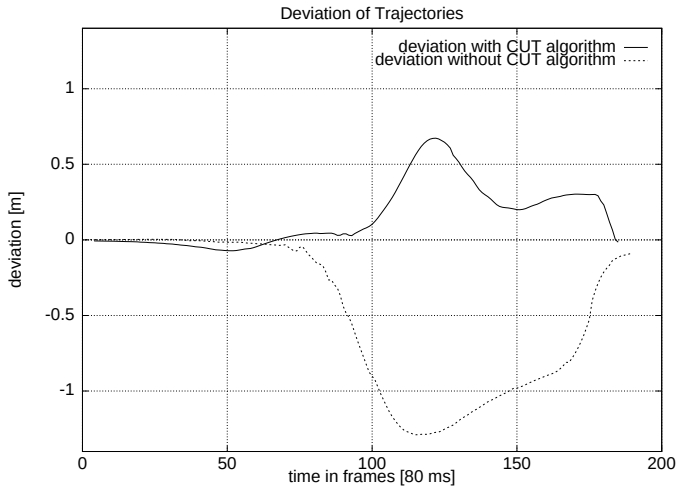

Figure 4.9: Real deviation of the trajectories between the ego-vehicle and the leader vehicle (solid: CUT algorithm, dashed: Controller interpolating an arc), measured in the autonomous test car.

active and reconstructing the trajectories never yielded a deviation above $0.4m$ for the CUT algorithm. Without the CUT algorithm, the observed deviations were much larger in very similar traffic situations. These numbers are imprecise since the measurement errors of the leader vehicle are not known. On the other hand the on-board leader vehicle simulation used above provides perfect sensor data for proper comparison.

The motion parameters of the ego-vehicle were evaluated in a separate analysis. Velocity measurements showed errors of less than 2% above $2m/s$ and the absolute errors for smaller velocities were also within $1m/s$, which does not contribute much to errors of the path reconstruction. The steering angle turned out to be a critical parameter. A steering angle bias of the sensor can be detected by comparing the reconstructed trajectories going through a circle path both clockwise and counter-clockwise. Errors on the steering angle sensor were found to be less than 2% for path reconstruction under low lateral accelerations and dry weather conditions. In one out of ten runs, a significant steering angle bias persisted (see Section 3.5.4). Motion parameter errors can be compensated by measuring stationary objects in the traffic scene (see Chapter 5).

4.9 Discussion

4.9.1 Comparison to Other Vehicle-Following Systems

Only very few quantitative data of vision-based vehicle-following systems are published. Typically, the path deviation from the leader vehicle's path is a good measurement quality. However, tracking performance depends greatly on the maneuvers of the leader vehicle. Tracking performance degrades with increasing steering variations. In addition, the performance also depends heavily on the actuator hardware. So comparisons are hard to make.

[Daviet et al. 96] reports a lateral deviation of up to $60cm$ when following a leader vehicle at $10m$ distance using a trailer model for the controller. [Kehtarnavaz et al. 91] reports deviations of up to $25cm$ when following the car at about $11m/s$ on a $35m$ radius with a distance of about $16m$. Here the Ackermann model controller is used.

Vehicle-following systems using magnetic or electric markers exhibit typically a better performance for path-tracking, e.g. [Guldner et al. 98] performs path-tracking with path deviations of less than $20cm$ at all times.

[Fritz 99] quotes a lateral deviation of less than $30cm$ at all times with a controller very similar to ours. In the tested scenarios, the distance to the leader vehicle is kept comparatively small such that path deviations due to steering variations do not have much impact. Again, a comparison is hard to make since in this application a truck is used and the actuator hardware is different.

4.9.2 Discussion of the Proposed System

The CUT algorithm achieves a significant improvement in precision for the lateral control in vehicle following systems at very little computational expense. To take varying measurement quality of the position into account, the lookahead distance can be adjusted accordingly. The error covariance matrices of the Kalman filtered trajectories provide a good measurement of the quality. High diagonal values in the error covariance matrix lead to an increase of the lookahead distance.

In addition, it is possible to estimate the current deviation of the ego-vehicle from the leader vehicle's path. Since both paths are computed in the same global reference frame, a distance can be calculated. From that quantity one can derive the total uncertainty of the control loop and adjust the lookahead distance accordingly.

Similar to the lateral control, a longitudinal control algorithm can be built: The acceleration of the leader vehicle can be stored at every trajectory point and can be used to obtain the same acceleration at the same position for the ego-vehicle. We decided to keep the current longitudinal safety-distance-keeping control algorithm, because for such an approach no remaining minimum distance is applicable. Future work on path-based vehicle-following is addressed in Section 8.4.1.

The reconstructed leader vehicle path enables us to perform additional path planning taking other obstacles into account (see Chapter 7). The transformation into an inertial reference frame and consequently the measurement of the motion parameters can be improved significantly by using visual cues from the cameras. This is a common approach in the field of robotics (see e.g. [Asada et al. 90]) and is discussed in the following chapter.

Chapter 5
Ego-Position Estimation Using Stereo Vision

Necessitated by the previous chapter, an ego-position estimation algorithm is presented in this chapter to improve self-localization in an inertial reference frame. The chapter overview can be found at the end of the introduction section.

5.1 Introduction

Our main objective in this chapter is to perform an ego-position estimate. We need ego-position estimation to properly associate 3D data among the frames. Ego-position estimation using only standard vehicle odometry (velocity and steering angle) can cause non-negligible errors, especially in situations where side slip or skidding occurs. We use stationary points in the scene to support the determination of our ego-position. Two types of stationary objects are used: First, stationary vertical landmarks such as traffic signs are used to compensate for errors in our localization prediction. Second, lane markings measured in consecutive frames are used to compensate for orientation errors. A precise ego-position estimation for intelligent vehicles is beneficial for the following tasks:

- The CUT algorithm introduced in Chapter 4 benefits greatly from a precise ego-position estimation. The path of the leader vehicle becomes more accurate with a more accurate ego-position (see Equation 4.4).

- For proper mapping of the environment into an inertial reference frame using sensor data from different times and viewpoints, ego-position estimation is of uttermost importance.

- A good ego-position estimation enables us to improve our velocity estimate by simple means (differentiation). This is especially relevant for small velocities where the built-in vehicle sensor for velocity delivers erroneous readings. This data improves the longitudinal control of the vehicle, since the decision whether the car moves or not can be made more precisely.

- For parking maneuvers, a good ego-position estimate is necessary to properly associate data from different frames (see Chapter 6).

- With the approach chosen here, vehicle sensor data is combined with a vehicle model for stationary steering behavior and compared to vision measurements. If a large inconsistency of the two position estimates is found, the stationary steering model no longer applies. The underlying road model implies full traction. Hence the vehicle is most likely found in an unwanted state (skidding) and measures to counteract that can be taken by steer-by-wire action.

How is this chapter organized? Section 5.2 gives an overview of related work both in the field of intelligent vehicles and robotics. In Section 5.3 the necessary transformations and the vehicle model for motion integration are introduced. Ego-position estimation techniques for an unknown environment are described in Section 5.4. The Kalman filter used for that purpose is introduced in Section 5.5. The specific ego-position estimation algorithms using vertical landmarks and lines are presented in Section 5.6 and 5.7. A review on problems of observability connected with the proposed algorithm is covered in Section 5.8. Alternative Kalman filter designs are discussed in Section 5.9. Simulation results and real world results of our ego-position estimation algorithms are shown in Section 5.10. A discussion comprises the final section.

5.2 Related Work

5.2.1 Overview

In the field of robotics, ego-position estimation, also called localization, is necessary for successful navigation. In [Borenstein et al. 96], Borenstein tries to answer the related question "where am I". This book gives a thorough review of the localization methods used to date. The main distinction is made between absolute positioning methods (e.g. GPS) and relative positioning methods (e.g. **dead reckoning**). Dead reckoning is a term borrowed from sailing ("deduced reckoning") and refers to the procedure of finding the current position by knowing a previous position and using velocity and heading direction to deduce the current position.

Ego-position estimation is needed in almost all mobile robot applications. Ego-position estimation in an unknown environment leads to the problem of relative ego-position estimation. This position can be computed from the ego-motion. Integrating the motion of the ego-vehicle leads to proper localization in an inertial reference frame. Due to imperfect motion data localization errors accumulate over time.

The related work described here is limited to ego-position estimation algorithms for unknown environments. A landmark navigation system where a priori knowledge of the landmarks is needed can be found e.g. in [Hock 91]. First, vision-based localization algorithms are described, followed by robot applications and intelligent vehicle applications.

The algorithm described here has no absolute measurements available and relies purely on relative position measurements but it is more than a dead reckoning approach since more than the inertial sensors are used.

5.2.2 Vision-Based Localization

An example of determining motion parameters from image sequences with several simplifications is described in [Yagi et al. 88]. Most of the simplifications do not apply to outdoor environments. [Asada et al. 90] determines motion parameters by matching two consecutive views from a scene recorded with cameras mounted on top of a mobile robot. Matching is performed by matching extracted line segments and the distance is calculated via motion stereo. He also uses that information to create a relational global map. In [Svoboda et al. 98] a vision-based ego-motion estimation algorithm is described. Svoboda uses a panoramic camera view and estimates ego-motion from motion flow of salient points.

A biologically inspired ego-position estimation algorithm for homing (i.e. finding back to your starting point) is described in [Steinhage et al. 98]. Minimalistic visual navigation based on biological evidence is presented in [Franz 98], where a simple and fast optical flow estimator is introduced.

[Adam et al. 99] presents a fusion of odometry and fixation using one camera. Only the viewing angle to one observed landmark supports the odometry measurement. The fusion is performed via a Kalman filter and yields only an improved vehicle dynamics description, not an ego-position estimate directly.

In the robotics realm, [Chenavier et al. 92] introduces a similar approach to ours that uses one camera. Chenavier extracts natural landmarks and uses their direction to localize the mobile robot. In addition, odometry information is used and fused with the landmarks using an extended Kalman filter. Since only the viewing angles of the detected landmarks are used, three landmarks are needed to get an unambiguous ego-position estimate.

A preliminary version of the algorithm described here has been developed in [Penteker 99]. The ego-position estimation approaches of this thesis can also be found in [Gehrig et al. 99a] and [Gehrig et al. 99c].

5.2.3 Localization for Robots

[Yamauchi et al. 96] uses Evidence Grids (see Chapter 6) to support dead reckoning by odometry. Evidence grids from previous frames are matched with the current grid. [Edlinger et al. 91] presents a method based on correlation of laser range data and odometry to obtain an ego-position estimate for a mobile robot. Continuous Localization based on map-matching is presented in [Yamauchi et al. 98]. There, extracted features from different times are matched. Localization via map matching and integration from three different sensor types, infrared, ultrasonic, and vision sensor, is presented in [Courtney et al. 94]. Prior knowledge of the scene is necessary in that approach.

5.2.4 Localization for Intelligent Vehicles

For cars, ego-motion estimation using vision clues has been investigated e.g. in [Uchimura et al. 98] and [Wagner et al. 97]. In contrast to our research, they utilized only

one camera. Both tracked several significant points in the scene and estimated the motion flow. These works make extended use of the optical flow of image sequences.

[Ma et al. 98] describes a motion reconstruction algorithm of images for differential views. Non-holonomic constraints of the motion simplify the algorithm. Comparisons to other ego-motion estimation algorithms are made there and in references therein.

[Hsu 97] describes a method to determine motion parameters from images using Kalman filters. The main purpose here lies in lane detection. Again, significant features are tracked in time to determine the ego-motion.

Although impressive work has been performed in the field of localization, all algorithms presented here put a considerable computational load on the system due to feature extraction. We want to perform an ego-position estimate very quickly. Hence a novel algorithm that uses 3D points already detected during object detection stage is introduced here.

5.3 Ego-Position Estimation via Motion Integration

We do neither use an absolute positioning system such as GPS nor a gyroscope to measure the yaw rate which leaves us with only the steering angle and the velocity to integrate the ego-motion.

For the subsequent analysis, the reference frame introduced in Section 4.5 is used. The global reference frame is equivalent to the local reference frame at the time of initialization and remains stationary.

In order to transform the 3D measurements into the global reference frame, two steps are performed. In the first step the coordinates of the ego-vehicle are transformed into the global reference frame as described in Section 4.5.

In the second step, the 3D measurements are transformed into the global reference frame, similar to Equation 4.4:

$$\begin{pmatrix} x_m \\ z_m \end{pmatrix} = \begin{pmatrix} x \\ z \end{pmatrix} + M(\Phi) \begin{pmatrix} x_{m,l} \\ z_{m,l} \end{pmatrix}, \tag{5.1}$$

where the subscript m refers to measured 3D coordinates and l refers to the measurement in the local moving reference frame. The vehicle model underlying the vehicle motion used here is described in Section 4.5.2. The integration of the vehicle motion in this way yields an ego-position estimate that accumulates error over time. A method to extend the ego-position estimation beyond dead reckoning is introduced in the next section.

5.4 Ego-Position Estimation Using 3D Points

We want to support our ego-position estimation with an additional sensor. Our range sensor, the calibrated stereo camera system, delivers 3D measurement of structured points resulting in a sparse 3D point cloud (less than 1000 3D measurements). However,

the algorithms described in the remainder of the chapter are applicable to any range sensor delivering (sparse) range data.

The ego-motion estimation algorithms of Uchimura [Uchimura et al. 98] and Wagner [Wagner et al. 97] rely heavily on good statistics since they need at least 50 significant points in a scene to produce a good estimation result. Hence, calculation times of these algorithms with a fast processor are high. However, we need to perform an ego-motion estimate very fast. Using the sensor data for velocity and steering angle (see Section 3), our largest uncertainty lies in the estimation of the yaw angle and of small velocities.

We decided to use stationary points in the scene to determine our ego-motion. Two types of stationary objects are used (see Figure 5.1).

Vertical Landmarks: The first type of object are vertical landmarks such as traffic signs, which are used to compensate for errors in our localization prediction.

Lane markings: The second type are lane markings, which are used to compensate especially for orientation errors.

Our basic assumption is that all 3D points close to the ground are stationary (e.g. moving shadows on the ground would violate this assumption). In addition, we assume that a group of 3D points vertically aligned alongside the road are stationary as well. This holds true for traffic lights, traffic signs, reflection posts, and trees, but a skinny pedestrian walking alongside the road might be mistaken as stationary. These primitives are input to our ego-position estimation algorithm.

Recursive matching of the current frame information with previous ones is performed using the extended Kalman filter. This method is explained in the following section.

5.5 Introduction to the Kalman Filter

Since most readers may not be familiar with the Kalman filter, a brief introduction is deemed appropriate here.

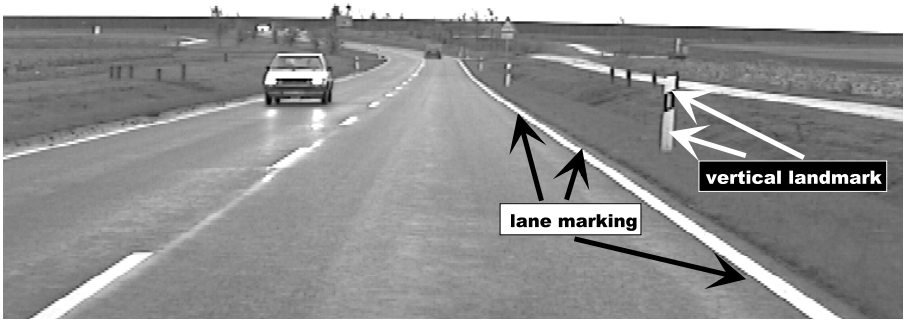

Figure 5.1: Typical traffic scene.

For the purpose of fusing noisy 3D point measurements with noisy velocity and steering angle measurements, an estimator is needed. An optimal linear estimator for Gaussian distributed noise incorporating known system dynamics and noisy measurements is the Kalman filter [Kalman 60].

The Kalman filter [Kalman 60] is a set of mathematical equations that provides an efficient solution to the discrete-data linear filtering problem. In this context, we limit ourselves to a description of the discrete case. The continuous case does not provide further insights. If the measurement or process relations are not linear, the extended Kalman filter is used to attack the problem.

The Kalman filter algorithm consists of two steps: A time update step ("prediction") that uses only the system dynamics for prediction and a measurement update ("correction") step that incorporates the new measurements.

The Condensation-Algorithm as an alternative to the Kalman filter uses a stochastic description and needs a-priori information of the error distributions but performs better for highly non-Gaussian noise distributions (see e.g. [Isard et al. 98a, Isard et al. 98b]).

This Kalman filter introduction sticks to a common notation also found in [Welch et al. 95]. For computer vision applications of the Kalman filter refer to [Faugeras 93] and [Matthies et al. 89].

5.6 Ego-Position Estimation Using Vertical Landmarks

5.6.1 Finding Vertical Landmarks

Vertical Landmarks were chosen as reference objects for ego-position estimation because they appear frequently along the road. Also, they have a unique and simple signature: Their 3D points are vertically aligned. Searching through a pre-sorted list of 3D points and checking for points with similar x and z coordinates is highly discriminative in 3D and very fast. Furthermore, this method is particularly fast if one uses a table of potential windows where to look for vertical alignments of 3D points. Other heuristics such as limiting the search area to the area alongside the road can also be used. Typical vertical landmarks that are suitable for ego-position estimation purposes are traffic signs, reflections posts, and traffic lights. Trees along the road are also used for that purpose. Summarizing, the detection and tracking of vertical landmarks works as follows:

1. Detect structured 3D points in the current frame.

2. Remove 3D points located close to or under the road from the 3D point list.

3. Find vertical structures by comparing 3D locations of the found 3D points. 3D points with similar x and z positions are connected. The position of the landmark is the mean of all 3D points contributing to a vertical landmark. Alternatively, a Hough transform of all 3D points can be performed that yields all line-like structures above a certain threshold and yields their 3D locations by using a

line fit for all contributing 3D points. This approach is computationally more expensive but more accurate. An implementation of this approach is described in [Penteker 99].

4. Once in tracking mode, only a narrow part around the area, where the last landmark has been found, is searched for the landmark in the current frame. In remaining areas new landmarks are searched. The search area for a landmark is predicted via Kalman filter.

5.6.2 System Description

The Kalman filter equations and the variables involved are introduced along with our ego-position estimation algorithm. The basic problem that the Kalman filter solves is the estimation of the state $\vec{x}_k$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$\vec{x}_{k+1} = A_k\vec{x}_k + B\vec{u}_k + \vec{w}_k = \vec{f}(\vec{x}_k, \vec{u}_k, \vec{w}_k). \tag{5.2}$$

Measurements of the measurement vector $\vec{z}$ are governed by

$$\vec{z}_k = H_k\vec{x}_k + \vec{v}_k = \vec{h}(\vec{x}_k, \vec{v}_k). \tag{5.3}$$

The random variables $\vec{w}_k$ and $\vec{v}_k$ represent process and measurement noise, respectively. A_k is the state transition matrix, B is the driving function matrix that relates the inputs u_k to the states, $\vec{z}_k$ is the measurement vector. The index k denotes the sample time step k. For a non-linear system, the matrices are replaced by the arbitrary functions $\vec{f}$ and $\vec{h}$, respectively.

We estimate the following states:

$$\vec{x} = \begin{bmatrix} x \\ z \\ \Phi \end{bmatrix}, \tag{5.4}$$

where x, z, and Φ are the location and orientation of the ego-vehicle.

Measurements are taken for $x_{i,l}$ and $z_{i,l}$, the 3D-points of the local reference frame, i.e. the raw measurement of the vertical landmarks from the moving vehicle:

$$\vec{z} = \begin{bmatrix} x_{1,l} \\ z_{1,l} \\ x_{2,l} \\ z_{2,l} \\ \vdots \end{bmatrix}. \tag{5.5}$$

The measured point refers to the measured vertical landmark projected onto the x-z-plane. x_i and z_i are the global x and z-positions of the measured point.

The steering angle δ constitutes the vector for the driving function

$$u = [\delta], \tag{5.6}$$

which is actually incorporated in $\vec{f}$ in the extended Kalman filter formulation.

5.6.3 Process Description

Using vehicle kinematics and the model derived in Section 4.5, we can formulate the continuous system:

$$\begin{aligned} \dot{x} &= v \cdot \sin\Phi \\ \dot{z} &= v \cdot \cos\Phi \\ \dot{\Phi} &= \frac{v}{r_{ego}(\delta)}. \end{aligned} \tag{5.7}$$

Here, r_{ego} denotes the path radius computed in Section 4.5.2. Φ is the orientation w.r.t. the inertial reference frame. It is equivalent to the yaw angle in vehicle dynamics applications when side slip is neglected. Implicitly stated are the following dynamics equations for the vertical landmark:

$$\begin{aligned} \dot{x_1} &= 0 \\ \dot{z_1} &= 0 \\ \dot{x_2} &= 0 \\ \dot{z_2} &= 0 \\ &\vdots \end{aligned} \tag{5.8}$$

From that we derive the Jacobian $A_{continuous}$ of the system functions

$$A_{continuous} = \begin{bmatrix} 0 & 0 & v \cdot \cos\Phi \\ 0 & 0 & -v \cdot \sin\Phi \\ 0 & 0 & 0 \end{bmatrix}. \tag{5.9}$$

By means of discretization one obtains the transition matrix A with

$$A = e^{A_{continuous}\Delta t}. \tag{5.10}$$

Performing a Taylor expansion yields

$$A = \begin{bmatrix} 1 & 0 & v \cdot \cos\Phi \cdot \Delta t \\ 0 & 1 & -v \cdot \sin\Phi \cdot \Delta t \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.11}$$

The control function B is not needed in the non-linear case. To compute the predicted state $\vec{x}_k^-$ the exact Equation 5.7 is used in discretized form:

$$\vec{x}_{k+1}^- = \vec{f}(\vec{x}_k, \vec{u}_k). \tag{5.12}$$

To compute the error covariances, the transition matrix A is used:

$$P_{k+1}^- = A_k P_k A_k^T + Q_k, \tag{5.13}$$

where P^- is the error covariance projected ahead and Q is the system noise matrix describing the system uncertainty. We assume a linear system noise function w for simplicity and consider the system noise of each state to be independent of the other state, i.e. the states are uncorrelated. Thus all off-diagonal values of Q are set to zero.

The system noise matrix Q could be adjusted in two cases:

- The values of Q are increased when the number of landmarks is increasing. The more landmarks are found, the less important are the system model equations of the system. The data of velocity and steering angle become less relevant.
- When the viewing angles between the landmarks is small, the ego-position estimation using vertical landmarks becomes error-prone (triangulation with a small angle, see Figure 5.3). Thus the values of Q are decreased to increase the confidence in the motion integration.

The Kalman filter mechanism provides an adaptive scheme that covers these effects to a certain extent. A small viewing angle yields a large measurement uncertainty by error propagation and the measurements are automatically given less weight. The number of landmarks is not explicitly taken into account in the Kalman filter, but little variations in Q do not affect the performance of the filter. Hence we keep the Q values constant (see Section 5.10.1).

5.6.4 Measurement Description

The measurement update step incorporates the new measurements. We measure $x_{i,l}$ and $z_{i,l}$, the coordinates of the vertical landmarks. These quantities are expressed in terms of state variables (see Figure 5.2). Every additional landmark yields two new measurement equations. We assume that our measurements are independent from each other, i.e. uncorrelated. The measurement equations are

$$h(\vec{x}) = \begin{bmatrix} x_{1,l} \\ z_{1,l} \\ x_{2,l} \\ z_{2,l} \\ \vdots \end{bmatrix} = \begin{bmatrix} \Delta x_1 \cos\Phi - \Delta z_1 \sin\Phi \\ \Delta x_1 \sin\Phi + \Delta z_1 \cos\Phi \\ \Delta x_2 \cos\Phi - \Delta z_2 \sin\Phi \\ \Delta x_2 \sin\Phi + \Delta z_2 \cos\Phi \\ \vdots \end{bmatrix}, \tag{5.14}$$

where Δx_i and Δz_i are shorthands for $(x_i - x)$ and $(z_i - z)$, respectively. x_i and z_i are the positions of the landmarks in the global reference frame. These values are computed from the initial measurement transformed into the global reference frame. Since a good initial localization leads to a better filter performance, one can re-adjust the global location after a few measurements have been taken.

The Jacobian H of the measurement equations is straightforward to compute:

$$H = \frac{\partial \vec{h}}{\partial \vec{x}} = \begin{bmatrix} -\cos\Phi & \sin\Phi & -\Delta x_1 \sin\Phi - \Delta z_1 \cos\Phi \\ -\sin\Phi & -\cos\Phi & \Delta x_1 \cos\Phi - \Delta z_1 \sin\Phi \\ -\cos\Phi & \sin\Phi & -\Delta x_2 \sin\Phi - \Delta z_2 \cos\Phi \\ -\sin\Phi & -\cos\Phi & \Delta x_2 \cos\Phi - \Delta z_2 \sin\Phi \\ \vdots & & \end{bmatrix}. \tag{5.15}$$

These equations are the input to the Kalman filter. The Kalman gain K is computed using

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}, \tag{5.16}$$

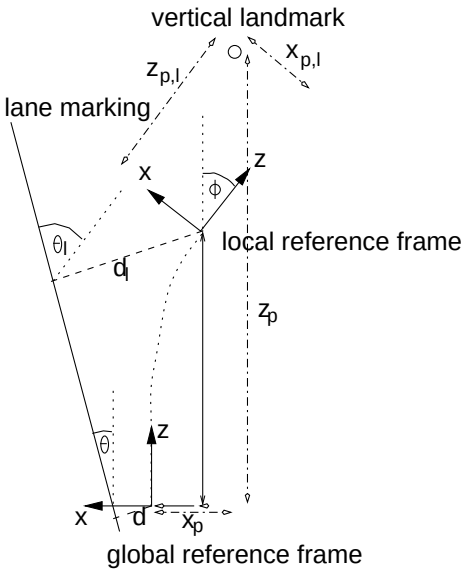

Figure 5.2: Vertical landmark and lane marking measured from the global and the local frame.

where R is the measurement covariance matrix describing the measurement uncertainty. The corrected state estimate is derived from

$$\vec{x}_k = \vec{x}_k^- + K_k(\vec{z}_k - h(\vec{x}_k^-)). \tag{5.17}$$

Equation 5.17 shows that an adjustment of the state prediction is only performed when measurement $\vec{z}_k$ and the predicted measurement of the predicted state $h(\vec{x}_k^-)$ differ. The corrected error covariance matrix is

$$P_k = (I - K_k H_k) P_k^-. \tag{5.18}$$

Time update is performed at each time step. Measurement update steps are performed whenever new measurements are available. When measurements for a landmark are not renewed for a certain number of frames, the landmark is deleted. Measurement variances are estimated using the relations between pixel noise and distance & offset:

$$R_x = R_{2i-1,2i-1} = (\frac{p_{err} \cdot z_{i,l}}{f \cdot k_u})^2, \tag{5.19}$$

$$R_z = R_{2i,2i} = (\frac{p_{err} \cdot z_{i,l}^2}{f \cdot k_u \cdot b})^2, \tag{5.20}$$

where p_{err} denotes the pixel error. Refer to Section 3.3 for an explanation of the other quantities. This non-linear error model for the depth stereo measurement is also presented in [Matthies et al. 87]. Since the measurements are independent of each other, all off-diagonal values are zero.

5.7 Ego-Position Estimation Using Lanes

5.7.1 Finding Lane Markings

Finding lane markings is a standard procedure in intelligent vehicle applications (see e.g. [Dickmanns et al. 94]). For our purpose we use 3D points extracted with our stereo camera system and perform a Hough transform of all 3D points that lie close to the ground plane. Only straight lane markings are considered in the current model.

Summarizing, the detection and tracking of straight lines on the ground works as follows:

1. Detect structured 3D points in the current frame.

2. Remove 3D points located above the road from the 3D point list.

3. Find line structures by performing a Hough transform of all selected 3D points. This yields all line-like structures above a certain threshold and yields their 3D locations by using a line fit for all contributing 3D points. This approach is quite accurate and efficient for the 3D points on the ground. Hough transform and line fit parameters determine the quality of the found lines.

4. Once in tracking mode, only a narrow part around the area where the last line segment has been found is searched for the landmark. The other areas are searched for new lines. The search area is predicted via Kalman filter.

We update line data into our map from the lane markings every frame using only the steering angle and the velocity in the first step (time update). In the second step, we match the old lane information in the map with our current lane information and correct our ego-position accordingly (measurement update).

The system and process descriptions of that approach are identical to the previous one. Only the measurement description changes.

5.7.2 Measurement Description

We measure the distance of each line to the sensor center, $d_{i,l}$, and its orientation, $\theta_{i,l}$, the parameters of each lane marking in the local reference frame described by the line parameters. Their counterparts d_i and θ_i in the global reference frame are related via

$$d_{i,l} = d_i - x \cdot \cos(\theta_i) - z \cdot \sin(\theta_i), \tag{5.21}$$

$$\theta_{i,l} = \theta_i - \Phi. \tag{5.22}$$

Consult Figure 5.2 for the geometrical relations. Again, every additional lane marking yields two additional measurement equations just like in Equation 5.14 for the case of vertical landmarks. The computation of the Jacobian H of the measurement equations

is straightforward:

$$H = \frac{\partial \vec{h}}{\partial \vec{x}} = \begin{bmatrix} -\cos\theta_1 & -\sin\theta_1 & 0 \\ 0 & 0 & -1 \\ -\cos\theta_2 & -\sin\theta_2 & 0 \\ 0 & 0 & -1 \\ \vdots & & \end{bmatrix}. \tag{5.23}$$

Measurement variances of $d_{i,l}$ and $\theta_{i,l}$ are considered roughly constant since the variances are bound by the Hough transform parameters. The global line parameters are initialized by the first measurement and can be adjusted later. The mechanism of the measurement update proceeds in the same way explained in Section 5.6.4.

5.8 Observability and Controllability Issues

A proper Kalman filter design needs to address **observability** and **controllability** issues in order to guarantee convergence of the filter at all times [Kalman 60, Joseph 87]. Controllability is only important for systems where the output is used to control a system. Here we use the Kalman filter as an observer estimating positions. So only observability is relevant here.

5.8.1 Observability

The system model for a Kalman filter has to be observable at all times. Otherwise filter divergence might occur. Intuitively, a system is observable if the state vector and the measurement vector have a one-to-one correspondence, that is if there exists a bijective mapping between state and measurement vector.

More formally, a dynamical system is locally observable, if the observability matrix Q obtained from the observability mapping $\vec{q}$ with

$$\vec{q} = \begin{bmatrix} \vec{h} \\ \dot{\vec{h}} \\ \frac{d\dot{\vec{h}}}{dt} \\ \vdots \end{bmatrix} \tag{5.24}$$

and $Q = \frac{\partial \vec{q}}{\partial \vec{x}}$ has rank n, where n is the number of states. The derivation of this test can be found e.g. in [Hsu et al. 68] or [Isidori 95]. The time derivates of $\vec{h}$ can be computed by substituting the state variable derivatives with the corresponding system equations.

Two measurements from one landmark are not enough:

$$\frac{d\vec{h}}{dt} = \frac{\partial \vec{h}}{\partial \vec{x}}\frac{d\vec{x}}{dt} = \frac{\partial \vec{h}}{\partial \vec{x}} \cdot \vec{f}(\vec{x}, \vec{u}) = \begin{bmatrix} -\frac{v}{r_{ego}}\Delta x_1 \sin\Phi - \frac{v}{r_{ego}}\Delta z_1 \cos\Phi \\ -v + \frac{v}{r_{ego}}\Delta x_1 \cos\Phi - \frac{v}{r_{ego}}\Delta z_1 \sin\Phi \end{bmatrix}. \tag{5.25}$$

Computing $\frac{\partial \dot{\vec{h}}}{\partial \vec{x}}$ yields

$$\frac{\partial \dot{\vec{h}}}{\partial \vec{x}} = \begin{bmatrix} \frac{v}{r_{ego}}\sin\Phi & \frac{v}{r_{ego}}\cos\Phi & -\frac{v}{r_{ego}}\Delta x_1 \cos\Phi + \frac{v}{r_{ego}}\Delta z_1 \sin\Phi \\ -\frac{v}{r_{ego}}\cos\Phi & \frac{v}{r_{ego}}\sin\Phi & -\frac{v}{r_{ego}}\Delta x_1 \sin\Phi - \frac{v}{r_{ego}}\Delta z_1 \cos\Phi \end{bmatrix}. \tag{5.26}$$

The lines in Equation 5.26 are just multiples of the lines in Equation 5.15 which both contribute to the matrix Q. Higher time derivatives of $\vec{h}$ do not increase the rank of Q. Consequently, $rank(Q) = 2 < 3$, which means that the system is not observable when only one landmark is available.

For two landmarks, the rank of Q increases to three just by considering $\frac{\partial \vec{h}}{\partial \vec{x}}$ for all realistic cases (e.g. $\vec{x}_1 \neq \vec{x}_2$), which means that the system is observable. This is also intuitive if one considers the observation of two landmarks. Figure 5.3 shows two landmarks with a radius of their distance to the ego-vehicle. When only one landmark is observed, the vehicle can be on any point of the respective circle. When both are observed, only the two intersection points are feasible solutions. The ambiguity is resolved by the values of the x coordinates of the landmarks.

As shown above for the case of vertical landmarks, both types of stationary objects chosen here yield unobservable systems when only one object is detected. The ego-position cannot be unambiguously computed from that information. This may cause the Kalman filter to become divergent, but this behavior has not been observed. Moreover, the same stable behavior has been observed in a similar approach using one camera [Chenavier et al. 92]. Since we use 3D data, already two landmarks yield a stable and observable localization.

5.8.2 Controllability

Controllability has to be fulfilled for closed-loop systems, i.e. systems that control a system. In our case, the Kalman filter is only used for observation purposes. The problem of controllability is closely related to the Q-Matrix. A very intuitive treatment of the problem of controllability can be found in [Joseph 87].

5.9 Kalman Filter Design Alternatives

5.9.1 Full-State versus Reduced-State Kalman Filter

In below results, we have used the Kalman filter described above. Alternatively, we also estimated the position of the landmark(s) which introduced another two states per landmark (see [Gehrig et al. 99c] for a detailed description). This corresponds to a full-state Kalman filter as opposed to a reduced-state Kalman filter as introduced in Section 5.4. Comparing both approaches we found out that the resulting differences are small. Both approaches result in an unobservable process when using only one landmark, but the error covariance increases very slowly with realistic data. When more than one landmark is used, the reduced-state Kalman filter becomes observable, whereas the full-state Kalman filter needs to estimate the position of the other landmarks as well, which increases the number of states. The results are similar. The key difference is that for stationary situations, the reduced-state Kalman filter reduces its error more than the full-state Kalman filter due to the observability property.

The explanation for the success of the full-state Kalman filter is as follows: The Kalman filter diverges only, if large errors occur. For small bounded errors the least square error minimization within the filter provides a good fit. The difference only

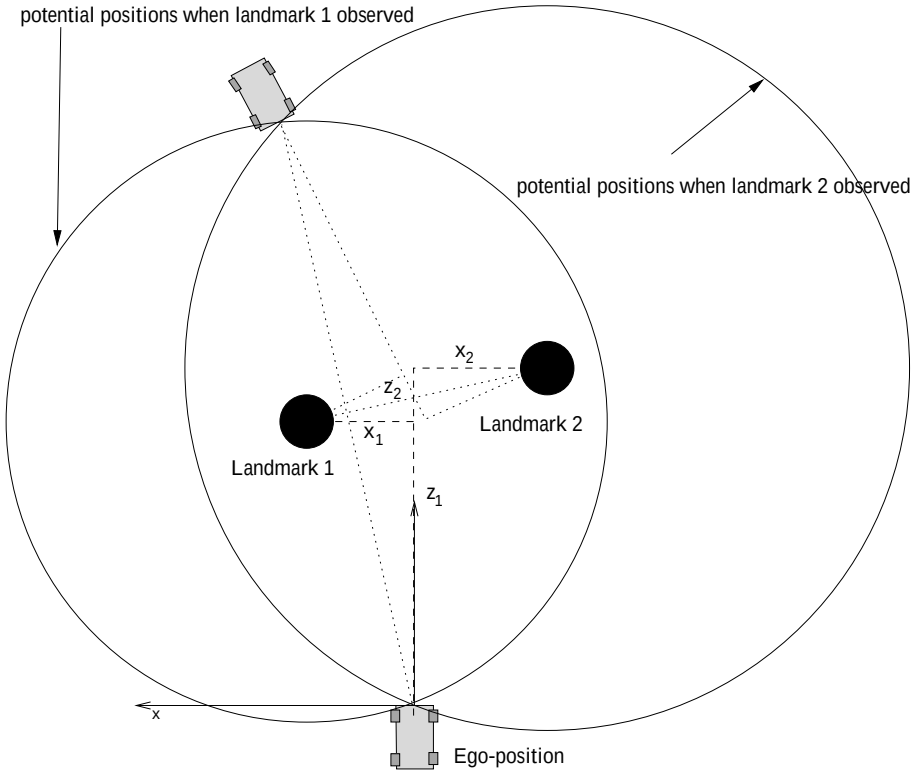

Figure 5.3: Two landmarks and the respective possible ego-vehicle positions. One landmark allows every point on a circle to be the ego-vehicle position whereas two points make the localization unambiguous.

comes into play for large errors, that are compensated by the reduced-state Kalman filter. After a certain time after introducing a large error the estimated value converges to ground truth again. The full-state Kalman filter has no one-to-one mapping to the measurements and keeps the error once introduced by a measurement with a large error.

A drawback of the reduced-state Kalman filter are the limited error model description possibilities. Using the reduced state, only one error model (value in the R-Matrix) for the relative distance between ego-position and landmark is used. The full-state Kalman filter allows separate error models (more states). An alternative filter to include more complex error models without Gaussian error distribution assumptions is described in [Hanebeck et al. 99].

5.9.2 Ego-Position Estimation with More States

Another possibility of making the system description more accurate is the incorporation of vehicle dynamics. Equation 5.7 treats the ego-velocity as a parameter. More realistically, one can incorporate more dynamics by introducing velocity and acceleration as additional states and limiting the dynamics by setting the jolt to zero. The ego-velocity v_{est} is now estimated, whereas before the velocity v was treated as a parameter. The new state vector consists of

$$\vec{x} = \begin{bmatrix} x \\ z \\ \Phi \\ v_{est} \\ a_{est} \end{bmatrix} \tag{5.27}$$

with v_{est} and a_{est} being the estimated ego-velocity and ego-acceleration. The system model extends to

$$\begin{aligned} \dot{x} &= v_{est} \cdot \sin\Phi \\ \dot{z} &= v_{est} \cdot \cos\Phi \\ \dot{\Phi} &= \frac{v_{est}}{r_{ego}(\delta)} \\ \dot{v_{est}} &= a_{est} \\ \dot{a_{est}} &= 0. \end{aligned} \tag{5.28}$$

The new measurement vector includes the measured velocity v and is computed via

$$h(\vec{x}) = \begin{bmatrix} v \\ x_{1,l} \\ z_{1,l} \\ x_{2,l} \\ z_{2,l} \\ \vdots \end{bmatrix} = \begin{bmatrix} v_{est} \\ \Delta x_1 \cos\Phi - \Delta z_1 \sin\Phi \\ \Delta x_1 \sin\Phi + \Delta z_1 \cos\Phi \\ \Delta x_2 \cos\Phi - \Delta z_2 \sin\Phi \\ \Delta x_2 \sin\Phi + \Delta z_2 \cos\Phi \\ \vdots \end{bmatrix}. \tag{5.29}$$

Consequently, the discrete analogies change but all computations are straightforward. Results of that Kalman filter are presented in the next section.

5.10 Results

5.10.1 Ego-Position Estimation Simulation Results

Simulation Scenario

Ground truth of the ego-position is not available for our research vehicle. For that reason we use simulated data than can easily be compared to simulated ground truth to evaluate the Kalman filters.

Our simulation uses the vehicle model as described in Section 4.5.2 and allows for erroneous sensor readings of velocity, steering angle, and 3D measurements.

In our simulated scenario the ego-vehicle takes the path depicted in Figure 5.4. The ego-vehicle starts accelerating from 0 to $1m/s$ within one second and the velocity remains constant for the remainder of the simulation, whereas the steering angle takes values ranging from -8° to 8°, creating a zig-zag path. The algorithm proved to deliver

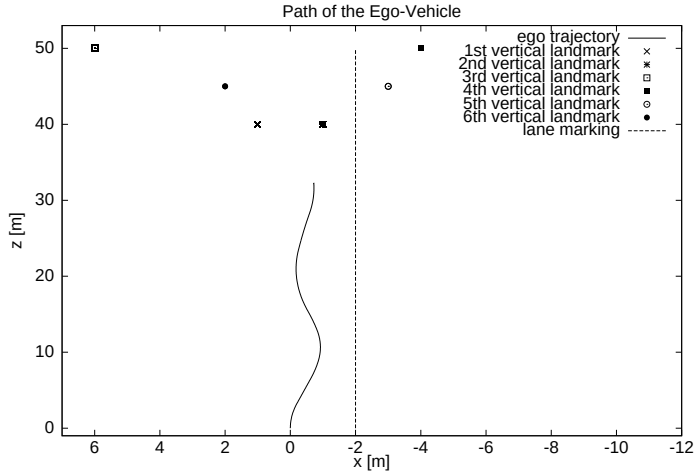

Figure 5.4: Profile of the simulated path (solid line). The isolated points indicate the measured landmarks, the dashed line shows the measured lane.

good results for a vast variety of parameter sets. For all simulation results presented below, the following parameters for the system noise matrix are used:

$$Q = \begin{pmatrix} 0.01 & 0 & 0 \\ 0.0 & 0.01 & 0 \\ 0.0 & 0 & 0.005 \end{pmatrix} \tag{5.30}$$

This is roughly equivalent to an uncertainty of $0.1m$ in x and z and reflects an uncertainty of about 2° for the steering angle. These estimates are conservative and a more rigorous treatment presented in [Chenavier et al. 92] yield varying Q values depending on the experimental setup. Since above parameters already deliver satisfactory performance, no Q value adjustments are performed. The pixel error p_{err} from Equation 5.20 is set to 2 pixels.

Vehicle Sensor Errors

The quality of the vehicle sensors in the research vehicle was investigated in Section 4.8.4. This investigation was initiated due to the fact that vehicle sensors and our used vehicle model deliver unsatisfactory position estimates under certain circumstances. On one hand, the velocity sensor reading at small velocities is unreliable. On the other hand, the steering angle sensor delivers a bias occasionally (see Section 3.5.4). Thus a simulation to investigate the robustness against erroneous vehicle sensor data is conducted. For simplicity, the measurements are assumed to be perfect and two landmarks are used for position estimation.

Table 5.1 shows position deviations for different levels of erroneous velocity data. Even when the velocity sensor delivers a velocity off by a factor of 2, the position estimate error stays within $0.6m$ for a stretch of $50m$ on average. The parameters of Q and R were not modified for this investigation. Experiments with more landmarks yield even better localization (see right columns in below table).

Velocity	deviation for 2 landmarks		deviation for 6 landmarks	
Error [%]	dev_{avg} $[cm]$	dev_{max} $[cm]$	dev_{avg}[cm]	dev_{max} $[cm]$
0	0	0	0	0
10	6	12	3	7
20	11	25	6	14
50	28	62	16	34
100	57	123	32	68
average deviation less than $60cm$ in all scenarios				

Table 5.1: Ego-position estimation errors versus velocity error. dev_{avg} denotes average deviation in cm and dev_{max} denotes maximum deviation.

Table 5.2 shows position deviations in the face of steering angle errors. Even up to a 20° bias of the steering angle, only a deviation of less than $10cm$ occurred. This way, a bias on the steering angle sensor can easily be compensated (see Section 3.5.4).

Steering angle bias [°]	dev_{avg} $[cm]$	dev_{max} $[cm]$
1	< 1	< 1
2	< 1	< 1
5	1	2
10	3	5
20	5	10
deviations less than $10cm$ in all scenarios		

Table 5.2: Ego-position estimation errors versus steering angle error. Note that the deviation stays below $10cm$.

Ego-Position Estimation Using Vertical Landmarks

Figure 5.5 gives an example of the ego-position estimation performance. In the simulation, the velocity sensor always delivers a velocity 20% larger than the actual velocity. In addition, a 5% measurement error on the x and z-position of the vertical landmark was added (uniformly distributed). Moreover, the steering angle was superimposed with a 2° bias. The simulation scenario depicted in Figure 5.4 was employed. The dashed line shows the Kalman filtered result of the ego-position deviation with deviations of less than $20cm$ at all times. Without the measurement feedback, pure motion integration leads to a much larger and increasing error (dotted line). Using two vertical landmarks again improves localization accuracy (solid line). Note that observing one landmark helps to reduce odometry errors already significantly. However, since the system is not observable, the once accumulated error never returns back towards zero. For two landmarks, the improvement in localization is shown in Figure 5.5 and can be performed without the use of odometry information.

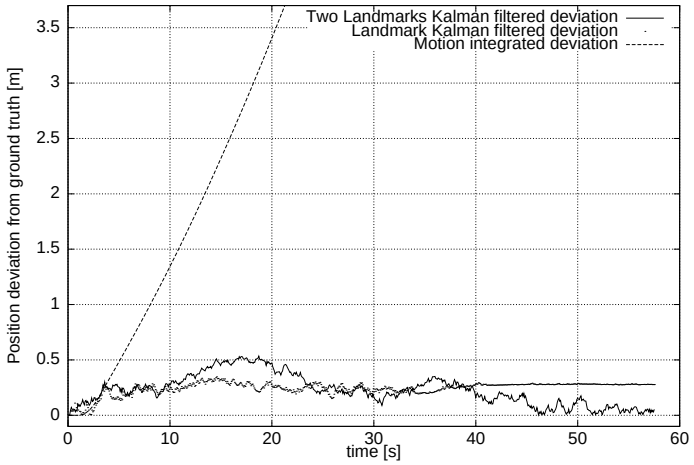

Figure 5.5: Position deviation of the ego-vehicle compared to ground truth (simulated data). See text for details.

In addition, for above scenario, the number of observed landmarks is increased from 1 to 6 for the same path. As expected the localization improves with more landmarks observed. Table 5.3 gives a summary of the results.

Ego-Position Estimation Using Lane Markings

Figure 5.6 shows a result for the lane-based Kalman filter. Here, the steering angle sensor exhibited a bias of 2°. The measured lane marking is located $2m$ to the right with an orientation of 0°. The error on the distance parameter was set to $50cm$ and the orientation error to 2° (uniformly distributed). All other simulation parameters and the ego-vehicle path were kept the same. The steering angle bias can easily be compensated and leads to a stable localization for the ego-vehicle (solid line). Pure motion integration leads to a large and increasing error (dashed line).

Number of observed landmarks	dev_{avg} [cm]	dev_{max} [cm]
1	24.4	35
2	22.7	53
3	11.3	36
4	10.1	33
5	9.3	28
6	8.9	31

Table 5.3: Ego-position estimation errors versus number of landmarks. dev_{avg} denotes average deviation in cm and dev_{max} denotes maximum deviation in cm.

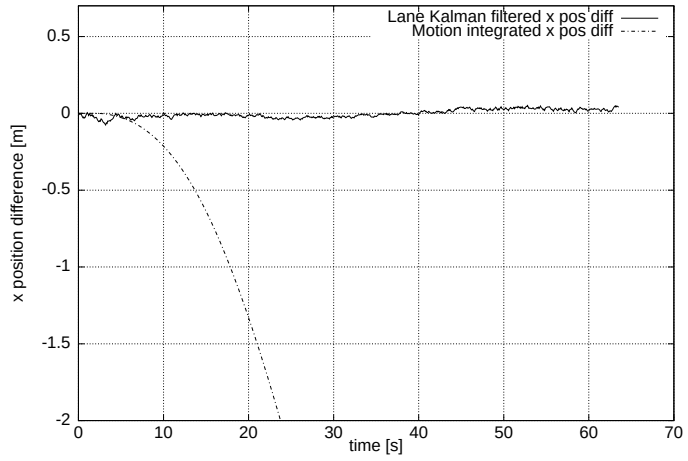

Figure 5.6: x position deviation of the ego-vehicle compared to ground truth (simulated data). See text for details.

Simulation runs incorporating both measurements of lanes and landmarks deliver the more accurate ego-position estimations the more observations are made.

Ego-Position Estimation With Velocity Estimation

Figure 5.7 shows a result for the Kalman filter estimating ego-velocity and ego-acceleration as new states. The same simulation conditions to previous simulation runs apply but the velocity sensor was set to deliver a velocity twice as fast as the real one. The Kalman filtered estimate comes closer to the ground-truth velocity but still exhibits significant deviations to the real velocity. The ego-position estimate is very similar to Figure 5.5.

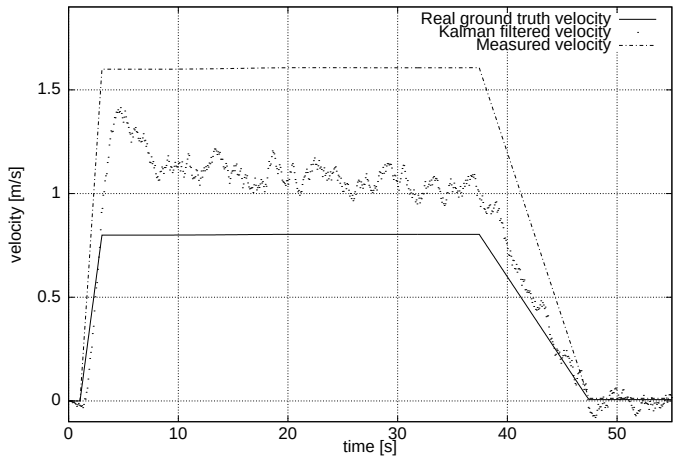

Figure 5.7: Comparison of real velocity, measured velocity and Kalman filtered velocity. The velocity estimate improves significantly with observing only one landmark.

5.10.2 Ego-Position Estimation Real World Results

The ego-position estimation algorithm using vertical landmarks has been run successfully in our research vehicle. The back court depicted in Figure 5.8 served as a test scenario. A zig-zag course similar to the one in the simulation was driven.

Superimposing all 3D points by pure motion integration causes the stationary objects to smear due to accumulation of motion-integration errors (see Figure 5.9, left side). The x-z-view in the figure shows all 3D measurements accumulated for a back court sequence with 100 frames. Ideally, the lamppost would be represented as one small dot in the graph. The "smearing" effect of pure motion integration is reduced significantly with the ego-position estimation algorithm using only one vertical landmark (see Figure 5.9, right side). For this scenario the ground truth is not known. Even for a perfect superposition of the local measurements, a smearing of the stationary objects would still occur due to measurement errors. Only the landmark circled in Figure 5.9 is used for the position estimate. For this run, the full-state Kalman filter described in [Gehrig et al. 99c] has been used.

Figure 5.8: Back court scene to test the ego-position estimation algorithms. The landmark used for ego-position estimation is the lamppost with the black rectangle drawn around it.

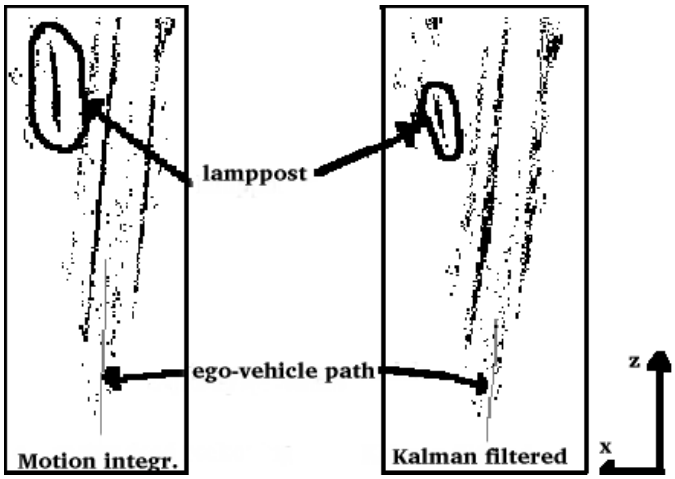

Figure 5.9: Bird view of the stationary scene from Figure 5.8. Ego-position estimation with pure motion integration (left) and with Kalman filtering (right) is depicted. The Kalman filter approach yields a better localization by observing the lamppost.

The computation time to find vertical landmarks and to update the Kalman filter is about $0.5ms$ on an Intel Pentium II 400 MHz PC.

5.11 Discussion

The ego-position estimation algorithms proposed in this chapter improve position accuracy compared to pure motion integration. Real world results show that motion integration errors are significantly reduced for small ego-velocities. The computational burden of these algorithms is minimal taking into account that the necessary correlations for finding the stationary landmarks have already been made.

The algorithm introduced above has similarities to the optical flow approach e.g. in [Uchimura et al. 98]. Instead of tracking many unknown features in 2D and using the power of statistics, our approach looks for more complex features and tracks these features in 3D. In addition, we also use motion information from the vehicle. Alternatively, one could come up with a mixture of both approaches, where many features are tracked in 3D in the picture, yielding a 3D optical flow field. More future work on ego-position estimation is outlined in the final chapter.

Concurrent mapping and localization to minimize errors on both tasks is becoming increasingly popular (see e.g. [Rencken 94]). However, such approaches are computationally expensive since map matching has to be performed. We decided to keep these tasks separate.

Provided an accurate localization in an inertial reference frame, a combination of the measurements from several frames is possible. This procedure to combine measurements from different viewpoints, i.e. mapping or cartography, is covered in the next chapter.

Chapter 6
Cartography Using Stereo Vision

Making use of the results from the previous chapter, a cartography algorithm building a global map in an inertial reference frame is presented. The chapter overview is given at the end of the introduction section.

6.1 Introduction

In this chapter, cartography of a road scene into an inertial reference frame is performed. We need an ego-position estimation to properly associate 3D data among the frames. This is achieved by using the algorithms proposed in the previous chapter.

Several intelligent vehicle systems for regular traffic have already been introduced here. However, none of these approaches known to the author tried to map the environment of the autonomous vehicle into an inertial reference frame.

Cartography is a common procedure in the field of robotics, mostly used for partially known indoor environments. The road scenario differs from the indoor scenario in several ways.

- The environment around a vehicle on the road is mostly unknown unless digital maps and GPS are utilized.
- Many fast moving objects appear in the scene.
- For vision based sensors extremely varying illumination conditions must be handled.
- Only coarse (e.g. GPS) or none positioning information is available, no fine information.
- The system response time must be very fast, for vision systems in the order of the video frame rate (40 ms).

Our main motivation for a cartography into an inertial reference frame in this framework is the opportunity to use 3D points from previous frames. This enables us to extrapolate objects beyond the sensor range by simply using global map data. For a parking assistant with limited opening angle of the sensor, a larger view can be obtained, which is a major benefit for such a system. The mapping procedure described in the following can be applied to a parking assist system without significant changes.

For automated stop-and-go driving, knowledge of stationary objects beyond the sensor range can prevent collisions with these objects in tight turns. Also, most of the benefits described in Section 5.1 also apply to cartography.

For the subsequent discussion we use the following convention: A **global map** is a map integrating information from different times and positions into an inertial reference frame. A **local map**, on the other hand, uses only current information to model the environment and has its origin at the sensor center of the moving vehicle.

How is this chapter organized? Section 6.2 gives an overview of related work both in the field of intelligent vehicles and robotics. Section 6.3 describes the cartography based on 3D points. Cartography results are shown in Section 6.4. A discussion comprises the final section.

6.2 Related Work on Cartography

Intelligent vehicle applications without global positioning systems have mostly used local maps only (see e.g. [Reichardt 96]). A global map used for vehicle following is presented in [Sukthantar 93]. There, only the leader vehicle is registered in the global map.

In robotics, obtaining global maps is crucial since most mission goals are specified using global map data. The main distinction is made between topological and metric maps. Topological or relational maps only give qualitative information about objects in the map, e.g. object 0 is west of object 1. Relational local maps are used e.g. in [Asada et al. 90], where cartography of a global map is achieved by superposition of local maps. Another example for topological map representation is described in [Meikle et al. 97]. There, views are stored at every learned location and topological dependencies are inferred using a probabilistic framework. An overview of mobile robot systems with topological representations of the environment can be found in [Wershofen 95]. Since we want to use the global map for object formation and obstacle avoidance which implies control, metric information is indispensable.

Metric, typically Euclidian, maps reflect the relation between objects in a quantitative way, e.g. object 0 is $1m$ west of object 1. Among Euclidian mapping techniques, grid-based and continuous approaches exist. Metric maps for range data are common for autonomous mobile systems, especially for systems equipped with unreliable range sensors (e.g. sonar sensors). Integrating sensor readings taken at different times/positions and superimposing them makes a compensation of the erroneous measurement readings feasible.

A popular approach for grid-based mapping, evidence grids, was pioneered by Moravec [Moravec et al. 85]. This mapping technique is also referred to as certainty grids or occupancy grids (see e.g. [Borenstein et al. 89] and [Matthies et al. 88]) with slight modifications. Evidence grids model the free space around an autonomous system for navigation and obstacle avoidance. This technique is mostly used for sonar range data. [Moravec 96] presents a 3D mapping algorithm with evidence grids, where a stereo camera system is used as sensory input. Refer also to [Moravec et al. 91] for more details on evidence grids. Borenstein introduced a mapping and collision

avoidance technique based on so-called histogram grids. However, this method is especially tailored to sonar range data to compensate for the low angular resolution [Borenstein et al. 91].

One weakness of above mapping technique is its limited capability to map moving objects. Furthermore, discretization of the environment is necessary which might compromise managing the data if a sufficiently large area is modeled with suitable accuracy.

Other free-space representations besides evidence grids and the like can be found in [Hoppen 92]. There, an overview of existing navigation approaches is given.

In our application grid-based approaches suffer from the large area we have to model (see Section 6.3.1), so a non-grid-based approach is chosen. A very early example of non-grid-based cartography can be found in Moravec's dissertation [Moravec 80]. There, the free spaced is mapped into a graph rather than a grid. Another approach can be found in [Rencken 94]. There, the mapping and localization problem is combined by minimizing errors on both tasks. The intuitively appealing approach suffers from a lot of heuristics that only apply to sonar range data. For an abridged description of the presented cartography algorithm refer to [Gehrig et al. 99b].

6.3 Cartography Based on 3D Points

6.3.1 Selection Criteria for the Mapping Algorithm

In indoor environments a very limited area has to be mapped. When only an especially limited two-dimensional representation of the environment is needed, accurate discrete maps are easily obtainable.

For a vehicle in regular traffic, however, the only limiting factor on the covered area is estimated by the reach of a tank of gas, conservatively estimated about $1000km$. Since the heading direction is unknown we have to map an area of $2000km \cdot 2000km$. The position accuracy of the obstacles around the vehicle should be of the order $10cm$ to allow precise navigation around the obstacles, which yields a prohibitive size of $20000000 \cdot 20000000 = 4 \cdot 10^{14}$ cells.

Obviously, one can reduce this size by several orders of magnitude by some adaptive scaling. Also, one could consider to only discretize the immediate surroundings of the ego-vehicle (e.g. $100m \cdot 100m$), but then discretization has to be redone at every time step since the vehicle is moving. This would involve a prohibitive number of transformations.

The best and simplest choice is to keep the raw 3D point data as a list which has to be scanned through for further tasks such as collision avoidance. Since only the 3D points facing in the forward direction must be kept, this list is relatively short. We decided to keep all 3D point data in an inertial reference frame. This way, only selected 3D points have to be transformed into the local reference frame.

For a parking assistant application, the grid-based mapping approach becomes interesting again, since the area to be mapped can easily be restricted.

6.3.2 Creating the Global Map

Our range sensor delivers 3D measurement of significant points resulting in a sparse 3D point cloud.

The global map is created accumulating 3D points from the current and previous frames. The creation of objects in the local map is described in Section 3.3.7. The same cluster criterion and object formation mechanism is now applied to the global map. This results in a map similar to the local map but also objects beyond the sensor range are kept in the object list.

6.3.3 Aging of 3D Points

To avoid smearing of the moving objects in the global map, old 3D points have to be deleted from the map after a certain time period. 3D points that have other 3D points from a more recent frame in their vicinity are kept longer in the map than "loners". That implicitly removes outliers and accidentals from the map quickly. This approach is also able to handle clipping and occlusions automatically. Every 3D point receives a survival number which is decremented every frame. Once a 3D point is not reconfirmed in the current frame the survival number is decremented a second time. When the survival number is zero, the 3D point is removed from the map.

Clipping of objects no longer occurs because object formation is based on the global map where the data from previous frames are still available. The same applies to occlusions. One could envision a heuristic to keep occluded and out-of-range objects even longer since they cannot be updated with current measurements. But this is dangerous since non-observed traffic participants behave unpredictable. Only predictions of up to two seconds are considered reasonable. For a parking maneuver, longer predictions are deemed appropriate.

6.4 Results

The results are obtained with our research vehicle driving at different speeds in different locations. To show the potential and limitation of the method, three scenarios are selected.

6.4.1 Results with Stationary Objects

Figure 6.1 depicts a scene with only stationary objects. The vehicle moves slowly forward at a speed of about $1m/s$ in this sequence. Figure 6.2 shows that superimposing several frames in the global map facilitates object detection tasks for stationary objects compared to using only the local map (see Figure 6.3). Note that the car has moved in this scene and superposition has been performed correctly by transforming all 3D points into an inertial reference frame. 3D points on or below the ground are not displayed in these figures.

Figure 6.1: Back court sequence with stationary objects.

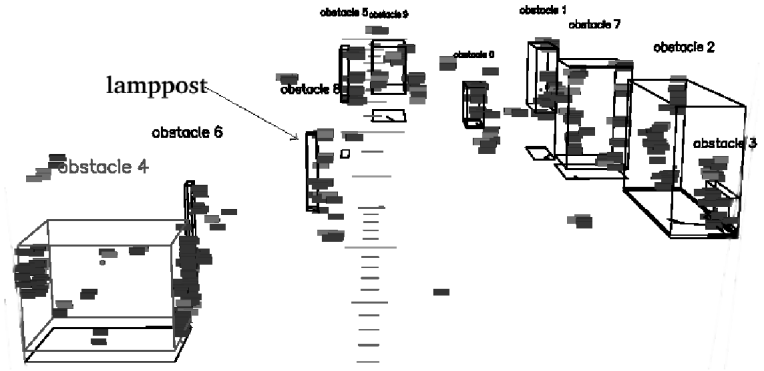

Figure 6.2: Global map representation of the back court sequence. Note that all objects protruding from the ground are detected.

Figure 6.3: Local map representation of the back court sequence. Note that two lampposts on the left are not recognized (too few 3D measurements).

6.4.2 Results with Moving Objects

Looking at a more dynamic scene, Figure 6.4 shows a sequence where clustering on the global map leads to heavy object smearing. A truck traveling at $25m/s$ is depicted there. The ego-vehicle follows at about the same speed. The recognition with our stereo system based on the global map is shown in Figure 6.5 on the left which exhibits a severe smearing effect and which leads to a wrong distance determination of the truck.

Figure 6.4: Dynamic freeway sequence with a truck.

Figure 6.5 on the right shows the sequence with only using the local map. Note, that in the sequences described here, **all** 3D points from 5 frames are accumulated in the global map. For the reason of better visibility, the aging of the 3D points is omitted. One potential remedy of that smearing effect would be to suppress 3D point accumulation for objects moving faster than a certain velocity. In order to use the global map for control, the coordinates of the objects have to be transformed back to the local reference frame of the ego-vehicle [Gehrig et al. 98c].

6.4.3 Results with Extrapolated Objects

The global map also allows to extrapolate objects that disappeared from the field of view. One example is shown in Figure 6.6, where the leader vehicle drives along a line of parking cars at a speed of about $10m/s$. The ego-vehicle follows at about the same speed.

Figure 6.7 depicts the parking cars sequence in the global (left) and local (right) map. Note that the first car on the right in the scene is still visible outside the field of view in the global map. The field of view is depicted as two lines along the road for the left and right camera, respectively. The ego-vehicle is depicted on the bottom left of both figures and the driven distance is depicted as a tail of the ego-vehicle bounding box. Due to a different 3D point distribution in the local map, the second and the third parking car are clustered as two separate objects there. The first parking car is not visible anymore.

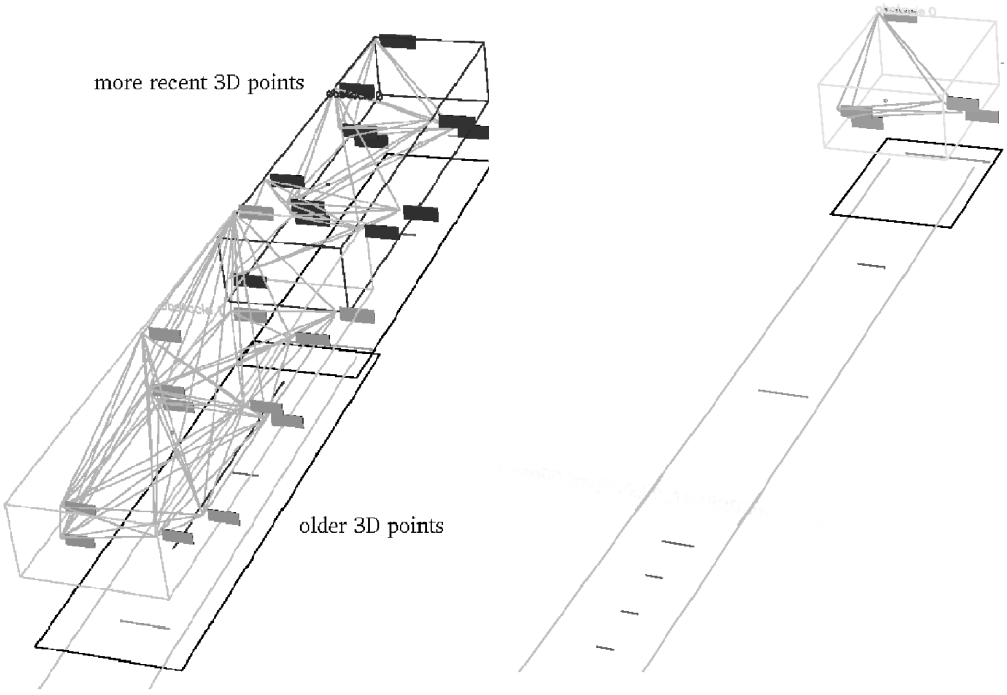

Figure 6.5: Global (left) and local (right) map representation of the freeway sequence. Due to old 3D points from previous frames the position is estimated too close to the ego-vehicle on the left.

Figure 6.6: Sequence driving along parking cars.

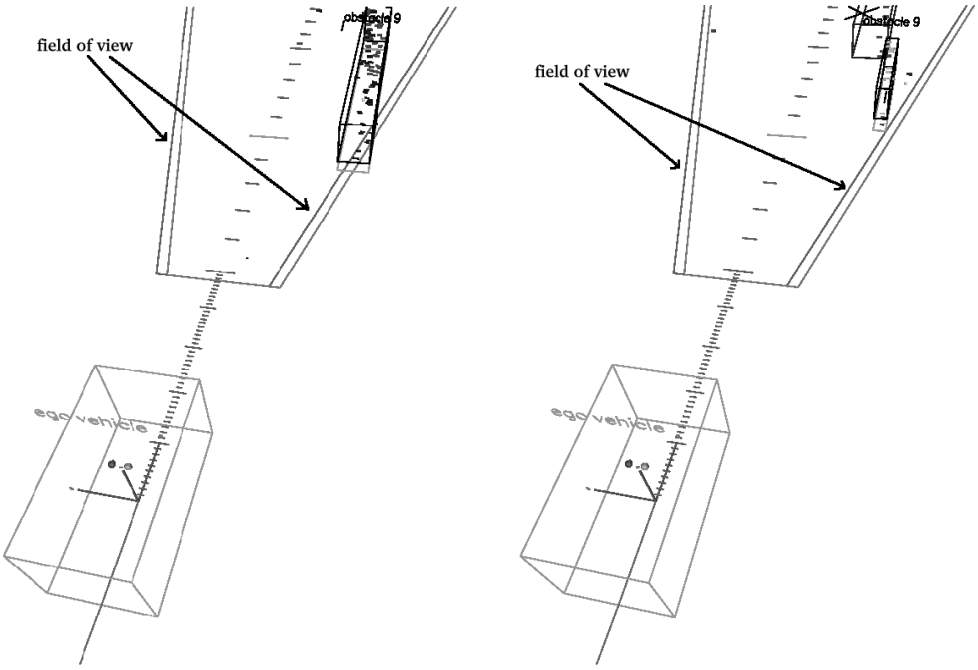

Figure 6.7: Global (left) and local (right) map representation of the parking cars sequence. Due to old 3D points from previous frames the extent of the parking car line stretches beyond the field of view in the global map.

6.5 Discussion

Cartography of the environment of an autonomous car is a beneficial procedure to superimpose vision clues from several frames, especially for stationary objects. More care must be taken for moving objects, since the distance determination of moving object becomes skewed. Due to this object smearing effect, the cartography procedure has not been used for the results presented in other parts of this thesis.

The computational burden of this algorithm is small. Only the clustering on the global map as opposed to the local map takes up to $12ms$ in cluttered scenes since more 3D points must be clustered. Future work on cartography is outlined in the final chapter.

Ego-position estimation and cartography as described here yield an accurate map of the environment. With these ingredients, it is possible to enhance the vehicle-following mechanism from Chapter 4 with a collision avoidance component that takes **all** detected obstacles into account. This approach is detailed in the next chapter.

Chapter 7
Collision Avoidance

This chapter gives an overview of collision avoidance. Specifically, the elastic band framework applied to vehicle following is presented. An overview of the chapter is given in the following section.

7.1 Introduction

Collision avoidance is the task of navigating a vehicle without collision. Collision avoidance has been the subject of extensive research both in the field of robotics and intelligent vehicles. For intelligent vehicle applications, mostly longitudinal collision avoidance, i.e. braking maneuvers, are considered.

Collision avoidance behavior of a completely autonomous car, VITA-II, was presented in [Reichardt 96], where behavioral patterns and potential fields were applied. The potential field method yielded unsatisfactory performance for autonomous navigation. Behavioral patterns were used for real driving maneuvers.

A lot of work on intelligent vehicles benefits from experience with mobile robots. However, robotics ideas applied to intelligent vehicles for regular road traffic must take the differences described in Section 6.1 into account.

In contrast to robotics, collision avoidance algorithms for intelligent vehicles usually do not face maze-like environments on the road. Considering local sensing, there usually is a feasible path and if not it is good strategy to stop. Street layouts of large cities appear to be maze-like sometimes but this kind of path planning is left up to a mission planning module and is not part of a collision avoidance algorithm. In our scenario, the intelligent vehicle is not able to move back autonomously which also limits the search space of feasible paths.

How is this chapter organized? Section 7.2 gives an overview of related work both in the field of intelligent vehicles and robotics. In Section 7.3 the elastic band framework is introduced. Section 7.4 describes the necessary adaptations to the elastic band framework in order to tune the avoidance behavior towards that of human beings. Section 7.5 discusses the incorporation of non-holonomic constraints into the elastic band framework. Necessary modifications to the planning and decision module are described in Section 7.6. Results are detailed in Section 7.7. A discussion comprises the final section.

7.2 Related Work

[Fujimura 91] presents an overview and categorization for planning and navigation algorithms. The main distinction in most autonomous systems is made into planning, navigation and piloting. Planning concerns mostly **mission planning** tasks and is left to the driver in our application. Mission planning is performed off-line before starting the mission. Some related work on mission planning is described in Section 7.2.1.

Piloting on the other end, concerns the control level and tries to convert desired direction and speed commands into actual.

In between is the navigator who is responsible for **collision avoidance** and **motion planning** among other tasks. Collision avoidance is a very important task for autonomous vehicles and includes avoiding obstacles by swerving or braking. Motion planning in dynamic environments should be performed while moving, i.e. online. This limits the computational complexity of such algorithms. Besides collision avoidance behavior it usually also includes path planning towards a goal. Navigation tasks also include mapping and localization (see Chapters 5 and 6). Here, related work on collision avoidance is presented.

The general 2D and 3D motion planning problem in a dynamic environment has been proven to be NP hard when striving for the shortest path or the path of shortest time [Canny et al. 87]. Consequently, all motion planning algorithms make additional assumptions or use heuristics to reduce the computational complexity of the general problem.

Related work on potential fields is listed in Section 7.2.2. Collision avoidance approaches based on physical models are described in Section 7.2.3. Other approaches can be found in Section 7.2.4. The section concludes with an overview of collision avoidance systems for intelligent vehicles (Section 7.2.5).

7.2.1 Mission Planning

In contrast to intelligent vehicles, a lot of robotics applications are designed to give the robot complete autonomy. In order to achieve this, a mission planning level is necessary (e.g. [Brooks 86]).

[Metea et al. 87] introduces a route planning algorithm for intelligent autonomous land vehicles. Dynamic programming is used for the route planning algorithm. For motion execution, a rule-based system is employed.

A path planner based on graph theory is introduced in [Petrov 95]. Here, the emphasis is on finding a feasible path in maze-like environments. For the obtainable field of view of an autonomous vehicle, the visible streets do not exhibit such complicated structures.

Mission planning using topological maps is discussed in [Leven et al. 99]. The area is represented in a graph connecting rooms represented as nodes. Little Euclidian information is used, which reduces the demand for precision for localization.

7.2.2 Potential Field Approaches

A lot of work has been performed on collision avoidance for mobile robots since the early eighties. The most prominent idea are artificial potential fields (e.g. [Krogh 84, Khatib 85]). Hereby the ego-vehicle is modeled as a point charge and obstacles are represented as charges repelling the ego-vehicle. The goal is represented by a point charge of opposite charge. Obstacle charges are modeled as potentials and the gradient of the superimposed potential field yields the direction command for the mobile robot.

[Tilove 90] analyzes different potential field approaches. Convergence and stability issues are discussed. Local and global methods are distinguished by using either a local or global map. Generalized potential fields depend not only on distance to obstacles and can therefore determine irrelevant obstacles pointing away from the ego-path. This method was first applied to velocity-dependent potentials in [Krogh 83]. Path generation is performed either via hill-climbing or via force control. Hill-climbing produces the path directly. Force-control computes an acceleration from the resulting force, where vehicle dynamics can be taken into account.

[Koren et al. 91] shows inherent limitations of potential fields, that are not very relevant for intelligent vehicles. The main problem is a local minimum, where a robot might get trapped (zero resulting force). Other problems include passages between closely spaced obstacles and oscillations in the presence of obstacles. The oscillations can mostly be avoided by sensible parameterization and additional heuristics. The local minima problem is only relevant when potential fields are used for force control. When the output of the potential field method is a path instead of a force, the local minimum problem results in a non-optimal path. [Khatib et al. 95] introduces extensions for potential fields to avoid local minima by superimposing a rotational potential field to the regular field. When a zero resulting force is obtained the rotational part of the field yields a remaining force.

In [Borenstein et al. 89] a combination of occupancy grids and potential field methods is presented, called the Virtual Force Field. An inherent problem, the local minimum trap, is solved here with heuristically switching to wall following behavior.

Krogh combines path planning of critical points via path relaxation and local obstacle avoidance via generalized potential fields in [Krogh et al. 86]. [vP93] describes path planning and collision avoidance based on potential fields in a similar way.

A popular approach to avoid local minima in potential field applications is the use of harmonic potentials. These potential shapes guarantee the non-existence of local minima for arbitrary obstacle shapes. However, the potential shape is limited to the harmonic potential which has an unlimited reach. An implementation of that approach can be found in [Guldner et al. 95] or [Guldner et al. 93]. [Grupen et al. 95] uses harmonic potentials for collision avoidance, where kinematic and dynamic constraints are included. This way, non-holonomic constraints for car-like vehicles are included.

7.2.3 Approaches Based on Physical Models

Other physical models besides potential fields are also popular for collision avoidance within the robotics community.

An alternative approach to the potential field methods is presented in [Decuyper et al. 91], where analogies of the collision avoidance problem to hydro mechanics problems are shown. Fluid dynamics equations are used. The fluid starts at the starting point towards the goal point and obstacles obstruct the flow. From the resulting flow field the planned path can be computed.

In [Steinhage et al. 98] and [Steinhage et al. 97], an approach to behavioral control of robots is described that uses the model of dynamical systems. The path for navigation and obstacle avoidance is generated solving dynamical differential equations. The main advantage to potential fields is that the whole path is represented by steady states. By using relaxation, stochastic noise always brings the system back to a steady state in contrast to potential field approaches. The approach is rather time-consuming due to the necessary relaxations. For this approach, low-level sensory information as opposed to symbolic obstacle information is used.

Quinlan and Khatib present another approach to obstacle avoidance that uses the model of an elastic band for a robot (see e.g. [Quinlan et al. 93a] and [Quinlan et al. 93b]). The elastic band framework is used for our application. An initial path has to be supplied. This path is modeled as an elastic band that is subjected to forces, exerted by obstacles which are represented as potential fields. This approach has also been extended to non-holonomic mobile robots [Khatib et al. 96], where especially the modeling of the free space changes. Brock extends that idea to the many-dimensional configuration space of mobile manipulators [Brock et al. 98]. The resulting configurations are called elastic strips. In [Rieger 00], the elastic band framework is revisited and some preliminary work of this thesis is presented.

7.2.4 Other Approaches

Other approaches often include rule-based systems or systems executing behavioral patterns.

The classic categorization of autonomous system tasks (see e.g. [Bhatt et al. 87]) is into sensing, planning, navigating, motion execution, and control connected in a serial fashion. Here the system can only operate successfully when all modules are functioning. These are functional modules. A different division is presented in [Brooks 86]. Typically, a robotic system can be divided into planning, navigating, and piloting. The subsumption architecture of Brooks divides a robotic system into levels of competence. The main idea is to slice the hierarchy of the robot competences in a different way. Different levels of behavior are designed, the simplest being wandering around. Higher levels include collision avoidance and at some higher level mission planning. So Brooks suggests task-achieving modules that can already operate when only the lowest level is implemented. One collision avoidance example of Brooks using this architecture treats every sonar sensor reading as a repulsive force scaled by the distance reading. The resulting force yields the new heading direction.

[Gat et al. 94] picks up the idea of behavioral patterns of Brooks and applies it to autonomous vehicles in space. There, the main key is to keep programs and robots simple to achieve a low power consumption. One way to navigate with these constraints is programming behavioral patterns. Such a rule-based system exhibits no unifying

concept and is rather built on heuristics.

[Bhatt et al. 87] describes an early real-time system for a mobile robot. The separation of tasks in planner, navigator and pilot is already performed. Higher level tasks and pilot tasks are implemented as rule-based systems. For the navigator, a mix of rules and search for optimal paths is applied to keep computational costs low.

Above mentioned systems use mainly rule bases for collision avoidance. The following systems have similarities to potential field approaches since they take also the obstacle clearance into account quantitatively.

In [vP91] local collision avoidance is performed by building a local sector map. The heading is set to the sector with maximum obstacle clearance. Indoor navigation using dynamic stereo-image sequence analysis can be found in [Schreer 98]. Obstacle forming is avoided. The obtained 3D points and their distribution lead the robot to open terrain. The area of lowest obstacle density is the heading direction.

Another recent approach, the global dynamic window approach performs collision avoidance by searching the space of possible speeds and headings for every computation cycle of the robot [Brock et al. 99]. The objective function in the search space to be minimized takes obstacle clearance and the heading towards the goal into account. The global aspect is introduced by using a global planner that administrates the connectivity of free space. This division makes high-speed navigation feasible.

7.2.5 Approaches for Intelligent Vehicles

Intelligent vehicles are often designed to increase safety and thus have collision avoidance capabilities. Most systems, however, only have warning functions and do not control the vehicle automatically. Here, only systems controlling the vehicle are presented.

[Niehaus et al. 94] describes a rule-based decision system for intelligent vehicles, which makes the safest decision by considering the worst case. Only freeway environments are considered. This work is conceptual and awaits the test in real traffic.

[Freund 88] calculates avoidance trajectories for a single obstacle based on current steering angle yielding a smooth trajectory. The result is always a clothoid, no arbitrary paths are allowed. No implementation of such a system is available yet.

Work on collision avoidance at the University of Armed Forces in Munich is described in [Dickmanns et al. 95] and [Kujawski 95]. By integrating GPS, digital maps and landmarks, whole missions can be performed. The underlying decision system employs a rule base.

[Shiller et al. 96] tries to find emergency collision avoidance maneuvers for intelligent vehicles. The vehicle dynamics is modeled with the one-track bicycle model (see Section 4.5.2). Simulation results are presented for a lane change maneuver when an obstacle suddenly appears in the own lane. The length of the avoidance maneuver in the own lane is minimized. No control algorithms to automate such maneuvers are given.

[Gerdes et al. 99] looks at a potential field application in non-autonomous driving applications. On force-control basis, also vehicle dynamics are taken into account and combinations of different drive-assist systems are feasible. Simulation results are

presented.

A completely autonomous car with obstacle avoidance capability is presented in [Reichardt 96]. The application of potential fields to determine the risk yielded oscillatory behavior in overtaking situations. As a consequence, behavioral patterns were implemented for obstacle avoidance maneuvers.

A very interesting approach both for robotics and intelligent vehicle applications is presented in [Fiorini et al. 98]. In contrast to most other approaches presented here, obstacle avoidance is performed in configuration and in velocity space. Other approaches such as Krogh's obstacle avoidance approach [Krogh 84] take the velocity into account by modifying the potential shape. Dynamic and kinematic constraints are taken into account. The set of feasible velocities and headings are intersected with the set of admissible (non-colliding) velocities and a search yields headings and velocities for either minimum travel distance or minimum travel time. In off-line applications, an exhaustive search can be performed, for online applications several heuristics limit the search space. The admissible velocities are computed by calculating the potential colliding velocities and headings for every obstacle (so-called Velocity Obstacle) in velocity-space and negating that set. For this computation, it is assumed that the obstacle velocity is constant. However, this constraint is not always met. Furthermore, this approach is tailored to circular obstacles and a representation of an obstacle with arbitrary shape by circles yields many circles. Simulation results for a highway scenario are presented. The same obstacle avoidance approach is used in [Prassler et al. 99], where a robot navigates successfully through crowds of moving people. Similar to the potential field approaches above, only distances to the current position are taken into account and distances to a planned path are not available online. These quantities are considered important when modeling human driving behavior (see Section 7.8).

7.3 Elastic Bands

7.3.1 Introduction to Elastic Bands

The original elastic band approach for collision avoidance was proposed by Quinlan and Khatib [Quinlan et al. 93a]. Similar physical properties are used for snakes in computer vision [Kass et al. 88]. An initial, feasible path must be delivered by a path planner. This path is dynamically modified by treating the path as an elastic band that is able to change its shape. The starting and end points are kept fixed. The total energy of the elastic band is minimized yielding smooth paths. Forces acting on the elastic band are computed by taking the gradient of the potential energy at discrete path points. The repelling forces on the elastic band are produced by obstacles in the vicinity of the path. In total, three types of forces are acting on the elastic band. These forces are described in the following sections, where we base our analysis on the thesis of Quinlan [Quinlan 94]. The treatment of the continuous case does not provide any additional insights and can be found in [Quinlan 94].

For the underlying theory of elastic bands the reader is referred to books on Mechanics (e.g. [Brandt et al. 96, Kang et al. 95]).

7.3.2 The Internal Contraction Force

The elastic band is modeled as a series of particles with a series of springs in between in the discrete case. See Figure 7.1 for a representation of an elastic band. Hooke's law would suggest a force proportional to the amplitude of the displacement. However, with defining zero as the unstrained state, the internal force would become larger as the length of the elastic band increases. This would yield an avoidance behavior which depends on the length of the elastic band which is undesirable. Hence we follow Quinlan's approach and let the internal potential be

$$V_{int} = k_c \Sigma_{i=1}^{n-1} ||\vec{q}_{i+1} - \vec{q}_i||, \tag{7.1}$$

where $\vec{q}_i$ represents the i-th configuration particle along the elastic band, k_c the constant of elasticity, and V_{int} the internal potential of the elastic band. The elastic band consists of n particles with $n-1$ springs in between.

Consequently, the internal force $\vec{f}_{int}$ is computed via

$$\vec{f}_{int}(\vec{q}_i) = -\nabla_{\vec{q}} V_{int} = k_c \left(\frac{\vec{q}_{i+1} - \vec{q}_i}{||\vec{q}_{i+1} - \vec{q}_i||} + \frac{\vec{q}_{i-1} - \vec{q}_i}{||\vec{q}_{i-1} - \vec{q}_i||} \right). \tag{7.2}$$

This normalized force ensures the same avoidance behavior independent of the band length.

7.3.3 The External Force

The external force is due to obstacles that are modeled as potentials in the scene. Any potential shape that repels the elastic band from obstacles is conceivable. We decided to use Quinlan's [Quinlan 94] position dependent potential superimposed by Krogh's velocity-dependent potential [Krogh 83]. The gradient of this potential, $\vec{f}_{ext}$, yields the external force. Details on the external force follow in Section 7.4.5.

7.3.4 The Constraint Force

The elastic band potentially reduces its internal energy by moving particles along the elastic band. This is an undesired property since the band might thin out at some parts. To constrain the motion along the elastic band, a constraint force $\vec{f}_{constr}$ is introduced with the direction $\vec{t}$

$$\vec{t} = \frac{\vec{q}_{i+1} - \vec{q}_i}{||\vec{q}_{i+1} - \vec{q}_i||} + \frac{\vec{q}_i - \vec{q}_{i-1}}{||\vec{q}_i - \vec{q}_{i-1}||} \tag{7.3}$$

tangential to the elastic band. The force is computed by projection of the external force along the elastic band,

$$\vec{f}_{constr} = -\frac{< \vec{f}_{ext}, \vec{t} >}{|\vec{t}^2|} \cdot \vec{t}, \tag{7.4}$$

where $<,>$ denotes the scalar product.

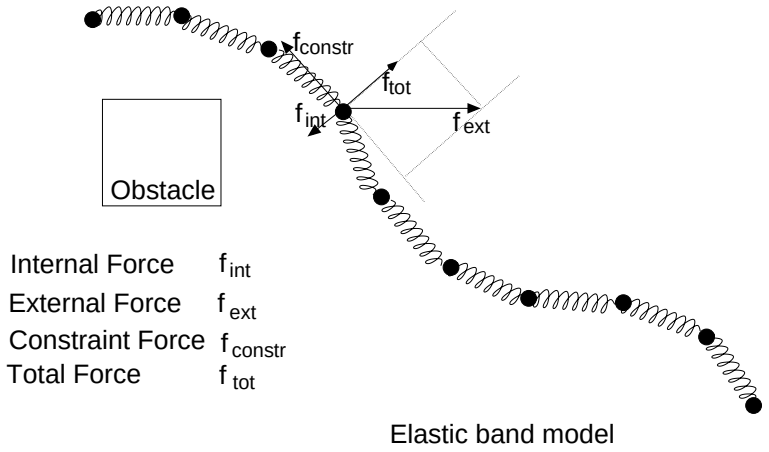

Figure 7.1: Representation of an elastic band as a series of particles with springs in between.

7.3.5 The Elastic Band Algorithm

The elastic band is simulated with reduced dynamics since only the equilibrium state is of interest for obstacle avoidance. For every time step, the algorithm is repeated. The path is represented by particles $\vec{q}_i$ and these particles are subjected to the total force

$$\vec{f}_{tot} = \vec{f}_{int} + \vec{f}_{ext} + \vec{f}_{constr}. \tag{7.5}$$

With our pseudo-static simulation, the particles are moved along the total force to the new position

$$\vec{q}_{i,new} = \vec{q}_{i,old} + \alpha * \vec{f}_{tot}(\vec{q}_i). \tag{7.6}$$

Higher order terms beyond particle acceleration are neglected here. Applying this procedure iteratively yields a linear convergence and proved to be sufficiently fast for our real-time application. The new configurations are recomputed until the total force of all configuration points is sufficiently small, i.e. until the force equilibrium is reached.

Pathological situations such as two obstacles located closely to the left and right of the initial path might yield an oscillatory behavior. We reduce the moving factor α after a certain number of iterations to reduce these oscillations.

[Brent 73] introduces an optimization algorithm that minimizes a function of one variable without computing derivatives and performs very few evaluations of that function. This procedure is used in the original approach to find the force equilibrium but it is less efficient for situations where only very little path modifications occur.

In the original elastic band approach, the moving procedure is supplemented with a procedure of adding and removing particles in order to maintain a collision-free path at all times (see Section 7.4). So-called bubbles model the available free space around a configuration. Adding particles becomes necessary when two adjacent bubbles do not overlap anymore. Bubbles are circles with a radius of the minimum distance to the closest obstacle for our two-dimensional case.

To speed up convergence of the elastic band algorithm, the final configuration from the last time step can be used as the initial configuration for the next time step. Path

transformations have to be performed to account for the interframe ego-motion. The required modifications of the original approach for the vehicle-following scenario are detailed in the next section.

7.4 Adaptations for Vehicle Following

7.4.1 The Basic Idea

The elastic band approach is adapted to the vehicle-following scenario here. The path of the leader vehicle constitutes the initial path. Obstacles in the environment exert forces on the band and move it into a final configuration. This is the path the ego-vehicle tracks subsequently. There are configurations where the final path is still infeasible in contrast to the original elastic band approach. In order to find these configurations, a final clearance check along the envisioned ego-vehicle path is performed. If the clearance check fails, the autonomous driving mode is switched off and the driver has to take over the control of the vehicle manually.

Driving situations that necessitate this dynamic modification of the leader vehicle path are

- passing a cyclist or pedestrian along the road while following a leader vehicle, or
- driving too close to parking cars due to the smaller size of the leader vehicle, or
- driving around a car that comes to a stop slightly inside the road intersection, or
- driving around a pedestrian that moved one step into the road.

Many more scenarios can be identified. In addition, further extensions of the autonomous driving capability of such a vehicle definitely need this dynamic path planning component.

The following subsections cover extensions to the sensor data acquisition, details on the distance computation to obstacles, algorithmic modifications to the elastic band approach, potential shapes for obstacles and lanes, and the actual integration of the approach in the existing system.

7.4.2 Object Formation

Our range sensor delivers a list of obstacles as described in Section 3.3.7. However, the elastic band algorithms in this chapter are applicable to any sensor delivering range data.

We assume a flat environment and use only the x and z coordinates of the objects for collision avoidance.

One object corresponds to one clustered 3D point cloud projected to the x-z-plane represented by its convex hull for collision avoidance purposes. A **convex hull** of a set of points is defined as the smallest convex polygon containing the points. Basically, the convex hull describes the contour of the obstacle. The obstacle representation as a

convex hull was chosen instead of its bounding box since significant deviations from its true obstacle shape occur in curves for the bounding box. Moreover, objects such as a crowd of people standing close to each other, exhibit arbitrary shapes in the x-z-plane and cannot be adequately described by rectangles. The convex hull is computed with a recursive algorithm described in [Edelsbrunner 87] that works in $O(n \cdot log(n))$, where n is the number of points.

The relative velocities of the objects w.r.t. the ego-vehicle are computed based on the center of the obstacles bounding boxes using an extended Kalman filter. One of the extracted objects is the leader vehicle supplying the initial path. The other detected objects constitute obstacles.

7.4.3 Distance Computation

The distance of the ego-vehicle to the obstacles is the most important quantity to determine the external force. Since the distance is computed thousands of times in one cycle, it is of uttermost importance to compute it fast.

If the bounding boxes of obstacle and ego-vehicle are more than the reach of the potentials apart, no refined distance computation is necessary and the distance of the bounding boxes is sufficient. The algorithm described below is only necessary for obstacles being close to the ego-vehicle. This coarse to fine strategy saves a significant amount of computation time.

Distance computation between two convex polyhedra is described e.g. in [Bobrow 89]. This algorithm is comparatively fast in general, but solves the distance computation in more than two dimensions. [Lin et al. 91] has a good incremental algorithm for the distance computation but the initial distance computation is slow. Faster implementations are available in two dimensions.

In our approach the distance computation between a rectangle (our ego-vehicle) and a convex hull (obstacle) is reduced to a distance computation between two line segments. The ego-vehicle's orientation is parallel to the tangent on the elastic band. When the ego-vehicle and the obstacle do not intersect, the distance is computed from all points of the ego-vehicle to all obstacle points. The points with the minimum distance are taken and the distance of the two adjacent line segments on both sides is computed subsequently. Thus, a total of four line-segment-to-line-segment distance computations is necessary.

The distance between the obtained obstacle line segments and the ego-vehicle line segments is computed using Lumelsky's algorithm [Lumelsky 85]. This algorithm is about five times faster than the straightforward approach.

7.4.4 Modifications to the Original Elastic Band Approach

The vehicle-following application of the elastic band differs from the original idea in the following ways:

- The initial path cannot be guaranteed to be collision free. Since the initial path is created by the path of the leader vehicle, changes in the scene might have

occurred since the leader vehicle has passed. It is possible that another car or a cyclist/pedestrian has approached that path in the meantime. In addition, the leader vehicle might be smaller in width than the autonomous vehicle and might have chosen a path very close to an obstacle.

- The equilibrium position of the configuration particles is the leader vehicle path. We want the autonomous vehicle to follow exactly this path in the absence of obstacles. Hence the internal forces of the initial configuration receive an offset yielding zero in the absence of obstacles. The internal force used for our application is thus computed as

$$\vec{f}_{int}(\vec{q}_i) = k_c \left(\frac{\vec{q}_{i+1}-\vec{q}_i}{||\vec{q}_{i+1}-\vec{q}_i||} + \frac{\vec{q}_{i-1}-\vec{q}_i}{||\vec{q}_{i-1}-\vec{q}_i||} - \frac{\vec{q}_{init,i+1}-\vec{q}_{init,i}}{||\vec{q}_{init,i+1}-\vec{q}_{init,i}||} - \frac{\vec{q}_{init,i-1}-\vec{q}_{init,i}}{||\vec{q}_{init,i-1}-\vec{q}_{init,i}||} \right). \quad (7.7)$$

  This also results in an offset term for the total energy of the band. This behavior was chosen to maintain the intuitive vehicle-following behavior. Hence the best strategy is to follow the leader vehicle path unless obstructions occur. In the original approach, the elastic band was assumed to have zero length and would always collapse to a straight line in the absence of obstacles.

- The concept of bubbles of free space disappears automatically by allowing initial trajectories that are not collision free. By giving up on the bubble concept we need an additional algorithm to reassure a collision-free path after the equilibrium position has been found. This is done by geometrically checking for overlaps along the final configuration of the elastic band. The procedure of adding and removing particles becomes superfluous as well. More general, a collision-free path can never be guaranteed in the presence of moving obstacles. A drastic example would be a faster car than the own that intentionally wants to provoke a collision. Provided sufficient driver skills in the "hunter" car, a collision is unavoidable.

- A regular car is an intrinsically non-holonomic vehicle. The original algorithm applies only to holonomic robots. This limitation has not been explicitly taken into account. Hence the paths created by the elastic band could be infeasible for non-holonomic cars. Since uncertainties in the 3D measurement of objects and uncertainties in the vehicle control require additional slack around the planned path, this is not considered a serious problem. In addition, the non-holonomic constraint also applies to the leader vehicle and is reflected in the leader vehicle path. Work on elastic bands for non-holonomic vehicles has been presented in [Khatib et al. 96]. Section 7.5 is devoted to this problem.

- In regular traffic situations, lane markings are also used for vehicle guidance. The lane markings are detected and are also modeled as virtual obstacles with repelling forces pointing away from the lane boundaries towards the lane center. Lanes are modeled as polygons so the same distance computation algorithm as for obstacles is used. More details on the potential shape are presented in Section 7.4.6.

- When a classification for the obstacle into pedestrian, car, truck, bicyclist, or other traffic participant is available, the repulsion factor for obstacles can be varied according to the classification result. This way, an additional safety envelope can be applied to pedestrians.
- The external force of the elastic band must be shaped to comply with natural driving behavior. Drivers keep more distance from obstacles at high speed than at low speed (e.g. parking situations). Therefore, a velocity-dependent potential shape must be used. This is detailed in the following section.

7.4.5 The Potential Shape for the Obstacles

We have experimented with a lot of potential shapes and came to the conclusion that potentials with limited reach best model human driving behavior. Usually, events and objects further away than a certain distance from our planned path do not have an impact on our actually driven path.

The original approach for elastic bands is designed for mobile robots that rarely exceed 2 m/s. So a position-dependent potential is sufficient. We modified this approach in a way that the effective reach of Quinlan's potential, d_{eff}, becomes a function of velocity,

$$V_{ext1,j}(\vec{q}) = \begin{cases} \frac{1}{2}k_r(d_{eff} - d_j(\vec{q}))^2 & \text{if } d_j(\vec{q}) < d_{eff} \\ 0 & \text{otherwise} \end{cases}, \tag{7.8}$$

$$d_{eff} = d_0 + d_{inc} \cdot v_{ego}, \tag{7.9}$$

where $V_{ext1,j}$ denotes Quinlan's potential of obstacle j at position $\vec{q}$. $d_j(\vec{q})$ denotes the distance between obstacle j and particle $\vec{q}$ of the elastic band, k_r is the repulsion gain. d_{inc} increases the effective reach of the potential for increasing velocities.

In addition, we superimpose a velocity-dependent potential, introduced by Krogh [Krogh 83]. Since we already know an initial path, only obstacles close to that path are considered. A fading function for every obstacle j, $f_{fad,j}$, is multiplied with the Krogh potential $V_{ext2,j}(\vec{q})$,

$$V_{ext2,j}(\vec{q}, v_{rel,j}) = \begin{cases} f_{fad,j}\frac{\overline{a}v_{rel,j}}{2 \cdot d_j \cdot \overline{a} - v_{rel,j}^2} & T_j > \tau_j, v_{rel,j} > 0 \\ 0 & v_{rel,j} \leq 0 \\ \infty & \text{otherwise} \end{cases}, \tag{7.10}$$

$$f_{fad,j} = \left(\frac{d_{krogh} - d_{j,min}}{d_{krogh}}\right)^2. \tag{7.11}$$

v_{rel} is the projection of the relative velocity on the direction vector between ego-vehicle and obstacle, $\overline{a}$ represents the maximal possible deceleration. $d_{j,min}$ denotes the closest distance of the j-th obstacle to the initial elastic band path. T_j and τ_j are minimum (τ_j) and maximum (T_j) avoidance time as defined in [Krogh 84]:

$$T_j = \frac{2d_j(\vec{q})}{v_{rel,j}},$$

$$\tau_j = \frac{v_{rel,j}}{\overline{a}}. \tag{7.12}$$

Note that $d_{j,min}$ is only available for obstacle avoidance algorithms that plan the whole path. Algorithms only yielding current heading and accelerations do not deliver the information to compute that quantity.

The idea behind Krogh's velocity-dependent potential is the following: The minimum avoidance time represents a braking maneuver with maximum deceleration $\overline{a}$. The maximum avoidance time uses the full distance to the obstacle and brakes with a constant deceleration. The reserve avoidance time T_{res} is $T_{res} = T_j - \tau_j$. The inverse of that reserve time to respond to an obstacle represents the potential computed in Equation 7.10. Note that the velocity-dependent potential described here inherits its name from the fact that the potential magnitude and shape depends on the relative velocity between ego-vehicle and obstacle. However, within one time step, this velocity is constant.

Figure 7.2 shows an example of the chosen potential. In that scenario, the obstacle on the right is stationary, whereas the left obstacle moves at half the speed of the ego-vehicle which is $10m/s$ (upper left). Quinlan's position-dependent potential is shown in the upper right. In the lower left side, the Krogh potential is depicted which is very similar to the superimposed potential. The most prominent effect of the Quinlan potential is a potential barrier sideways of the obstacle. The Krogh potential immediately drops to zero sideways of the obstacle when the vehicle moves only perpendicular to the obstacle. The Quinlan potential drops gradually to zero.

The total external force per particle is thus

$$f_{ext,j}(\vec{q}) = k_r(d_{eff} - d_j(\vec{q})) \cdot \frac{\vec{d_j}(\vec{q})}{||d_j(\vec{q})||} + 2 \cdot \frac{V_{ext2,j}(\vec{q}, v_{rel,j})^2}{v_{rel,j}} \cdot \frac{\vec{d_j}(\vec{q})}{||d_j(\vec{q})||}, \tag{7.13}$$

where $\vec{d_j}(\vec{q})$ is the direction vector from obstacle j to particle $\vec{q}$. The total external energy is computed from

$$E_{ext} = \Sigma_j \Sigma_i (V_{ext1,j}(\vec{q_i}) + V_{ext2,j}(\vec{q_i})). \tag{7.14}$$

When an obstacle is unavoidable for or inside a particle of the elastic band, a maximum force is exerted towards a direction, where obstacle clearance is obtained on the shortest way. Details on how to find this direction are presented in [Rieger 00].

Several restrictions regarding building the gradient of the potential field apply, e.g. the potential might not even be differentiable at some positions [Quinlan 94]. This might cause the algorithm to compute a wrong force in one iteration but because of a movement of most particles in the elastic band, the situation is resolved in the next iteration. Pathological cases might occur where a minimum energy configuration is not achieved but additional checks on obstacle clearance always provide backup solutions.

7.4.6 The Potential Shape for the Lanes

Lane markings provide an important guidance for driving. Hence we use lanes to support the vehicle-following procedure by modeling them as virtual obstacles.

To detect lanes, the 3D points already determined for object detection are used. 3D points below a certain height are accounted for lane markings. Using some additional

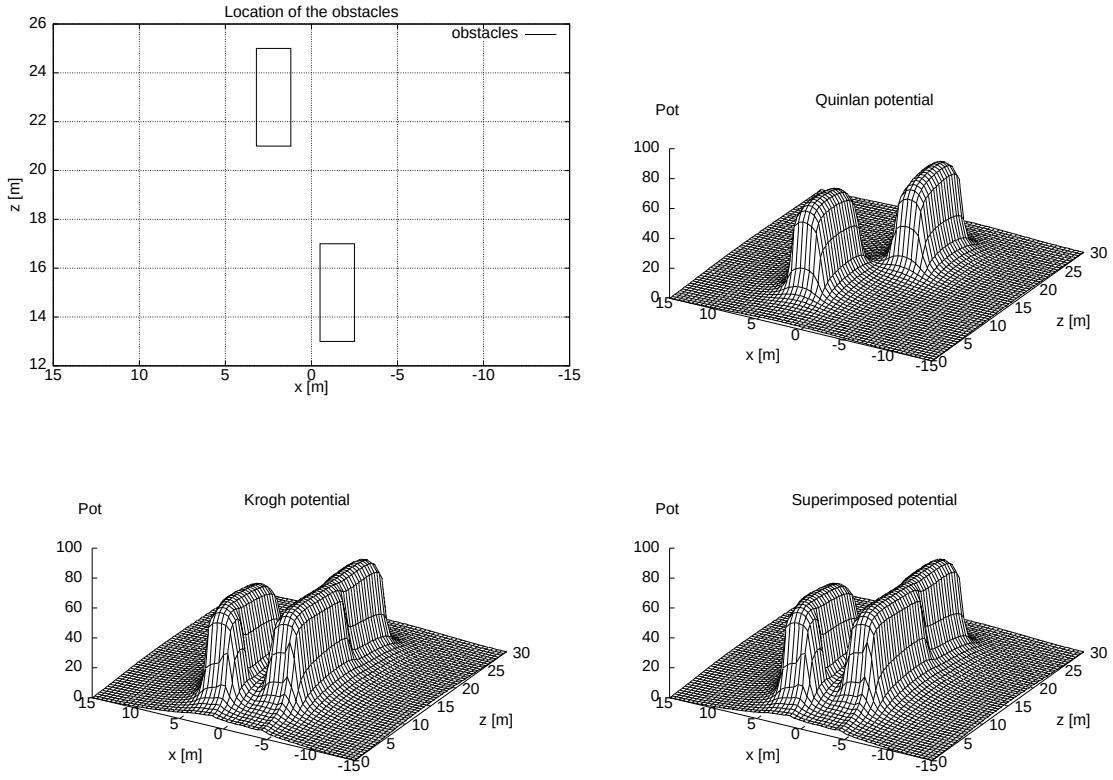

Figure 7.2: Visualization of the chosen obstacle potential shape. For the situation depicted in the upper left, the Quinlan potential can be viewed in the upper right. Below the velocity-dependent Krogh potential and the superimposed potential are shown.

models such as lane width, a lane is fitted through the 3D points. The lane model estimates current lane position, orientation, curvature, and change of curvature with a Kalman filter. This result is sampled at discrete points which yields a polygon. The polygon is input to the elastic band algorithm.

Only one lane to the left and one lane to the right (if existent) of the ego-position are used for the algorithm. The potential for lanes is modeled with the Quinlan potential. Being outside the lane yields a maximum potential, the potential in the middle between the lanes is zero. Figure 7.4.6 shows an example of a lane potential.

When driving close to the lane center, no force results from the lane potential. Only when driving close to the marking or even outside the lane, a force towards the lane center is exerted.

7.4.7 Control with the Elastic Band

The result of the elastic band for the control algorithm is a modified path. The controller is kept the same way as described in Chapter 4. Instead of the leader vehicle path, the elastic band path is used for control. Hence, a change of shape of the leader vehicle path only affects the actually driven path from that moment on, when the lateral position for the current lookahead distance changes. The control behavior of negotiating a path at a certain lookahead distance can also be found when observing humans driving through a curve and tracking their fixation point [Land et al. 94].

Since the elastic band algorithm operates statically on the current scene snapshot and is re-run at every time step, no connection between the elastic band results from one frame to another exist. Sudden changes in the controller input might cause an oscillatory behavior. Hence, a low-pass filter is applied to the lateral position delivered by the elastic band path.

7.5 Extensions to Non-Holonomic Vehicles

The elastic band framework presented so far only delivers feasible paths for holonomic vehicles. A car obviously is a non-holonomic vehicle since it has a minimum turning radius that limits the heading changes.

This restriction is taken into account in [Khatib et al. 96]. The induced metric not only considers the Euclidian distance to an obstacle but also takes the current heading direction into account. This metric satisfies the requirement of a minimum turning radius and also applies to the internal force. It tends to contract, but the zero position is modified: A straight line in an obstacle-free environment cannot be obtained. The result is a circular arc. The external force has a non-convex potential shape around the border of the minimum turning radius, which leads to stability problems.

A special interpolation and smoothing algorithm is needed in this implementation since the number of bubbles of free space must be kept small. The real-time implementation is made feasible through elimination of configuration points along the path. Therefore, interpolation and smoothing is needed.

The result is an algorithm taken far away from the intuitive elastic band idea. The

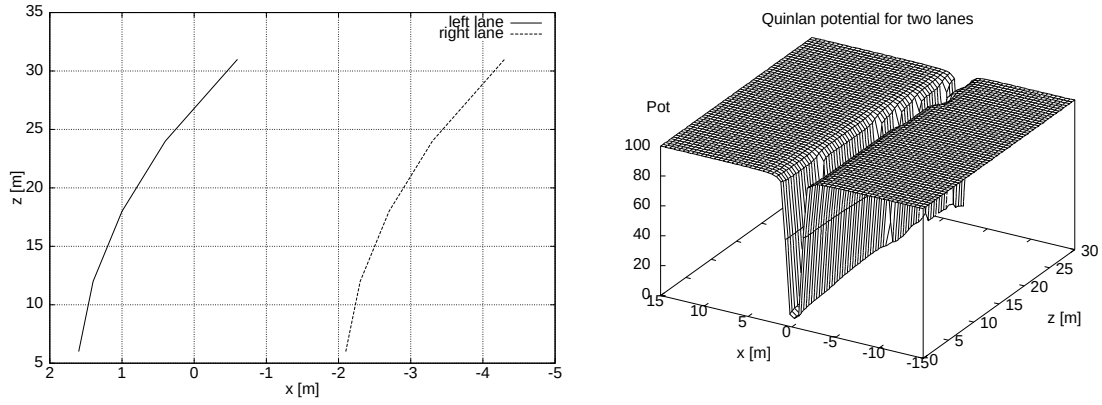

Figure 7.3: Visualization of the chosen lane potential shape. For the situation depicted in the upper left, the Quinlan potential for lanes is depicted on the right.

backbone of this approach is the representation of free space which is not applicable to the vehicle-following scenario.

So we decided to allow crooked paths implying turns below the minimum turning radius and let the vehicle dynamics smooth out the path. With sensible parameterization of the internal elastic band force, a feasible path for all typical traffic situations is obtained. In addition, tight turns are suppressed by the lateral controller anyway to provide sufficient ride comfort. When the vehicle is forced to an infeasible turn due to obstacles, the deviations to the planned path are noticed immediately since the elastic band algorithm runs in real-time. As a consequence, the control of the vehicle is given back to the driver.

7.6 Planning and Decision for Elastic Bands

The elastic band framework is integrated easily using the path-based CUT algorithm presented in Chapter 4.

Only the leader vehicle path of the CUT algorithm has to be exchanged with the path resulting from the elastic band algorithm. The elastic band algorithm is invoked right after the CUT algorithm has generated the leader vehicle path. The longitudinal control remains unchanged, i.e. we still follow the leader vehicle at a safe distance.

The elastic band path is also used to check for overlaps with obstacles since we drive along that path. Hence, when small intersections with an obstacle occur, the elastic band algorithm bends the path such that no intersection with the obstacle occurs and a collision-free path is obtained. As a safety measure the bent of the elastic band path is monitored. In addition, when deviations of the leader vehicle path and elastic band path beyond a certain threshold occur, the control is returned to the driver. Although the elastic band path is still feasible in such a case, the behavior of the ego-vehicle would already be far from that of a typical vehicle-following system.

The initial detection of the leader vehicle remains unchanged.

7.7 Results

7.7.1 Energy Minimization for Elastic Bands

Performing an iteration process for discretized steps that continues until force equilibrium is obtained does not guarantee to find a position of minimum energy due to discretization. To investigate the effect, we computed the total energy $E = E_{int} + E_{ext}$ of the elastic band for every iteration step. As expected, the energy always decreases unless a situation occurs where the algorithm oscillates around the position of minimum energy. This might happen due to a too large moving factor α for the particles. The differences in position are subtle and this situation is lead into equilibrium by gradually reducing α. Note that this procedure does not guarantee a configuration of minimum energy but a configuration close-by.

An example of the convergence quality and energy-minimizing procedure can be viewed in Figure 7.4. The traffic scene depicted in Figure 7.9 and 7.10 is used to produce the energy plot. It only takes 10 iterations to find the equilibrium state. The simple constant step size provides a sufficiently fast convergence.

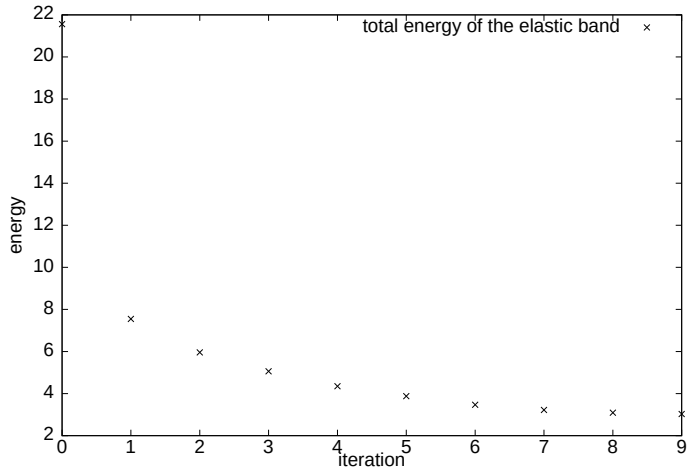

Figure 7.4: Energy of the elastic band vs. iteration number. As desired, the total energy decreases as we perform our gradient-descent algorithm.

A consequence of choosing the leader path as the zero-strained state is that the smoothness of the resulting path depends on the smoothness of the leader vehicle path. A smoothing function guarantees that the leader vehicle path is feasible for the ego-vehicle.

7.7.2 Simulation Results

Simulation Results Without Lateral Control

The initial verification of the implemented algorithms was performed in an off-line environment, where only a snapshot of a scene was analyzed. Parameter tuning was performed in this environment.

In Figure 7.5 a situation is shown, where the leader vehicle is slightly driving out of lane and other cars drive to the left and right. Hence a slight path modification to the right occurs due to the obstacles and also slightly due to the lane markings with the elastic band algorithm. Obstacles are modeled as rectangles for simplicity. The equilibrium state is found within 18 iterations.

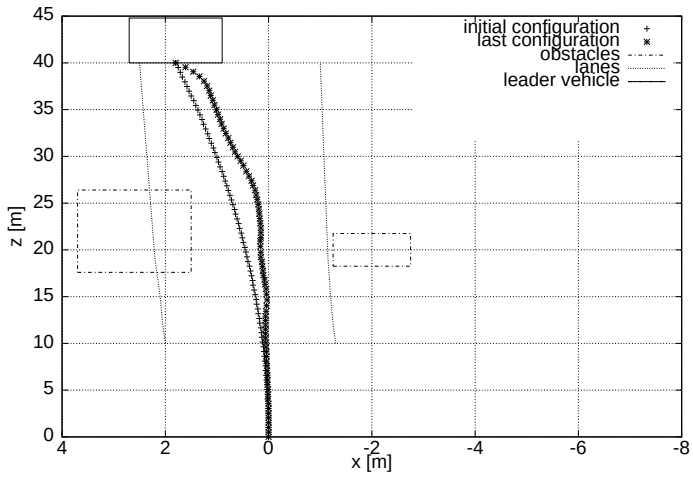

Figure 7.5: Simulation result of the elastic band algorithm (bird view).

Figure 7.6 shows an example, how the algorithm works in the absence of obstacles but with lanes. When the leader vehicle slightly drives out of lane, the ego-vehicle is guided back into the lane. No modification of the leader vehicle path occurs in the absence of obstacles and lanes.

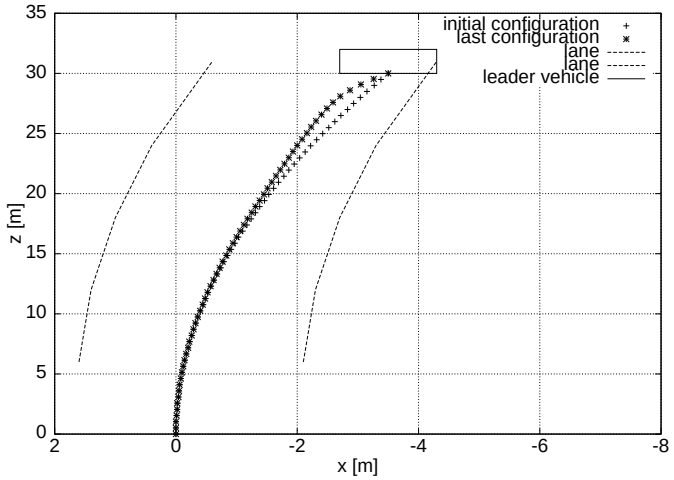

Figure 7.6: Simulation result of the elastic band algorithm using lanes only (bird view).

Simulation Results With Lateral Control

Collision avoidance maneuvers are dangerous to conduct in the real-world. It is not safe to interfere with a car's path when it moves. Maneuvers with especially little clearance to obstacles are critical. So we decided to perform some of these maneuvers in simulation first.

In Figure 7.7 an avoidance maneuver is presented, where a pedestrian slightly overlaps with the driving corridor. Hence a swerving maneuver is performed. This maneuver is performed in a simulation where the full controller code and an accurate vehicle model of the ego-vehicle is employed. Due to the delay of the actuator a very close maneuver resulted. To account for this delay, the required distance to the obstacle can be increased to a certain extent. That yields an additional safety margin. The deviation between desired path and actual path indicates that the algorithm necessitates a faster actuator when using the simple control law introduced in Chapter 4.

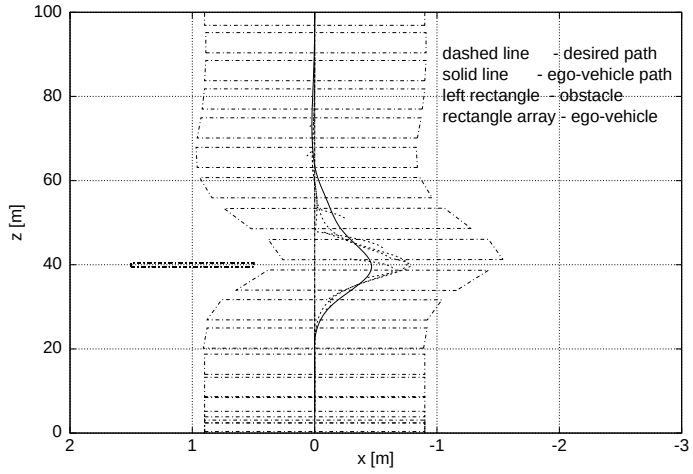

Figure 7.7: Bird view for a situation where a pedestrian slightly enters the driving corridor which necessitates a swerve maneuver. The steering actuator described in Chapter 4 is used. The ego-vehicle and the leader drive at $10m/s$ and they are about $20m$ apart.

Figure 7.8 shows the same situation with a model of a faster steering actuator. Note that the deviation of the actual path to the desired path is significantly smaller.

7.7.3 Real World Results

Real World Results without Lateral Control

We integrated our elastic band framework in the basic vehicle-following system. Computing the modified path runs in real-time without much optimization. With our current parameterization, the elastic band finds its equilibrium state within 20 iterations for most situations. In order to suppress oscillations around the equilibrium state, the moving factor α is reduced after a certain amount of iterations (see Section 7.7.1).

Figure 7.9 shows a typical traffic scene with counter traffic on the left and a parked obstacle (a dredger) on the right.

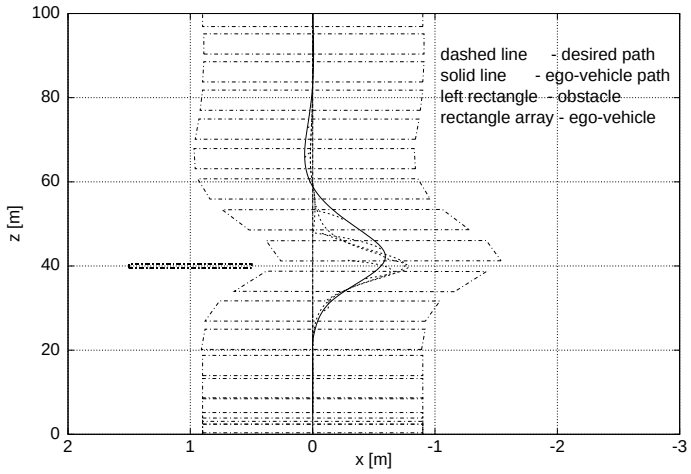

Figure 7.8: Bird view for a situation where a pedestrian slightly enters the driving corridor which necessitates a swerve maneuver using a faster actuator. A steering actuator with about $150ms$ delay is used. The ego-vehicle and the leader drive at $10m/s$ and they are about $20m$ apart. An avoidance maneuver with a comfortable safety margin results.

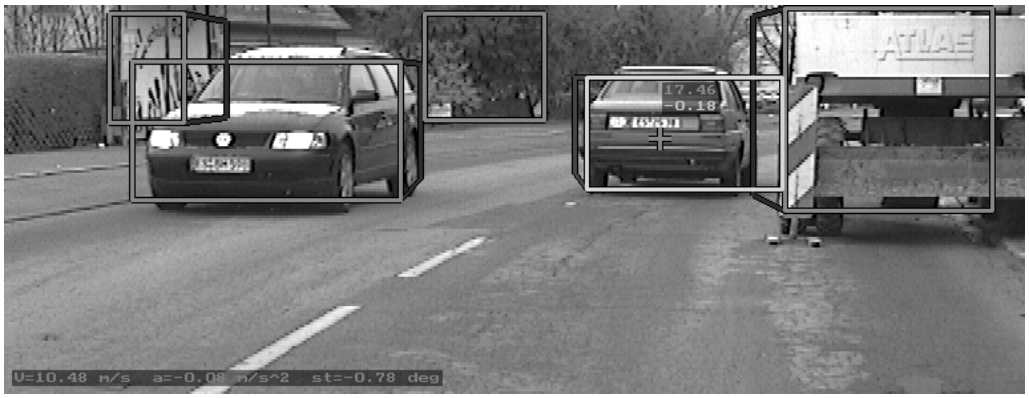

Figure 7.9: Traffic scene with a dredger at the right side of the street.

The leading vehicle drove very close to the dredger (see initial path in Figure 7.10) and has a smaller car than our research vehicle. Hence following the initial path brings us dangerously close to the dredger. The elastic band approach yields the dashed path in Figure 7.10 which avoids the dredger and keeps a safety distance from the car on the left at the same time. Lane markings were not used here.

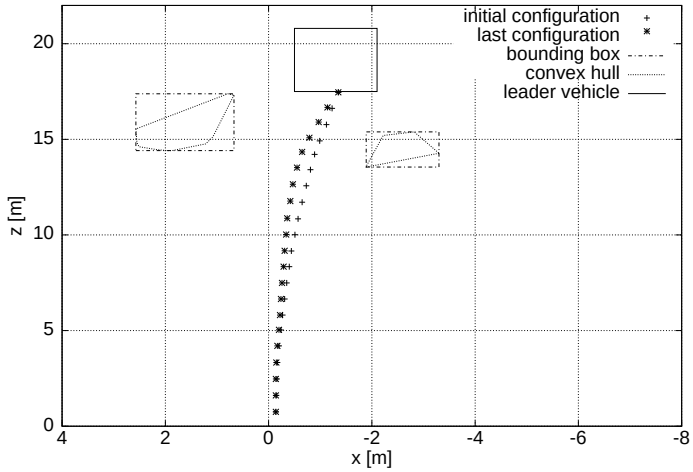

Figure 7.10: Bird view for the dredger scene. The leader vehicle is located at the end ($z = 17.5m$) of the path and smaller than the ego-vehicle.

Real World Results with Simulated Obstacles

Further experiments have been conducted using the elastic band path as input to the lateral control. Results using simulated obstacles in the research vehicle are described in the following.

The steering actuator described in Chapter 4 was upgraded for the following experiments. The delay from desired to actual steering angle could be reduced to about $150ms$. With this actuator, we simulated obstacles in the research vehicle to test the algorithm safely. In addition, the experiments can be reproduced exactly. Figure 7.11 shows a scene, where a small leader vehicle drives in a straight way very close to an obstacle on the left. The ego-vehicle avoids this obstacle by performing a swerve maneuver. Figure 7.12 shows the sequence of the swerve maneuver in three snapshots. The desired path as output from the elastic band algorithm is depicted in a dashed line. Note that the ego-vehicle does not follow exactly the elastic band path due to non-holonomic constraints and due to the simple controller design.

In Figure 7.13 the leader vehicle drives straight. Another vehicle in the right lane recognizes an obstacle ahead and starts to change its lane. When recognizing the autonomous vehicle it brakes but overlaps slightly with the other lane. Hence a swerve maneuver of the autonomous vehicle results similar to a maneuver a human would conduct in such a situation.

In Figure 7.14 the leader vehicle passes a bicyclist and cuts back in too early for pure vehicle following. Due to the elastic band path enough room is left for the bicyclist.

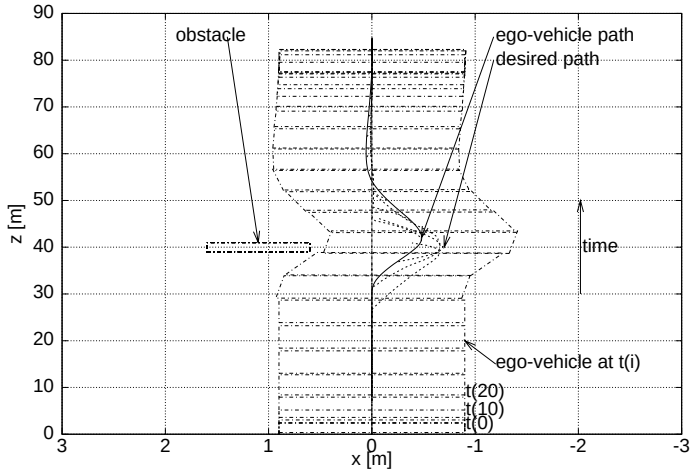

Figure 7.11: Bird view for an actually driven situation where a pedestrian slightly enters the driving corridor which necessitates a swerve maneuver. The ego-vehicle and the leader vehicle drive at $6m/s$ and they are about $15m$ apart.

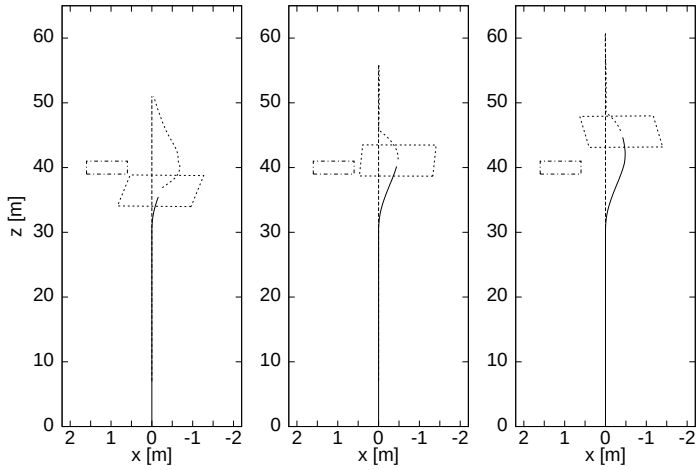

Figure 7.12: Snapshots for the scene depicted in Figure 7.11.

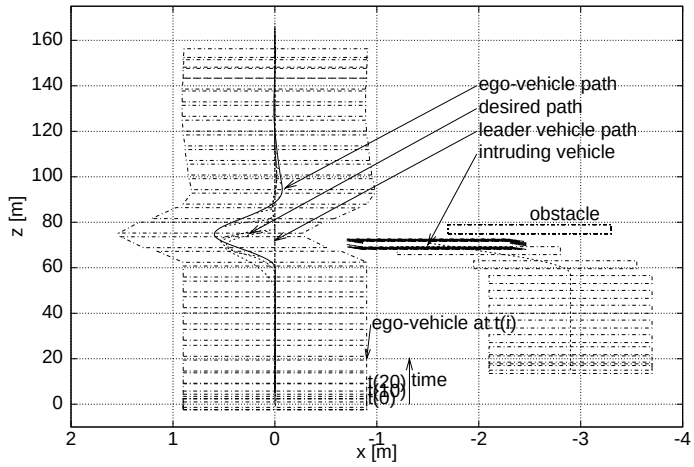

Figure 7.13: Bird view for a situation where a car slightly enters the driving corridor which necessitates a swerve maneuver. The ego-vehicle and the leader vehicle drive at $8m/s$ and they are about $18m$ apart.

Figure 7.14 shows three snapshots of the scene that are $0.8s$ apart each. The ego-vehicle leaves sufficient room for the bicyclist when returning to its lane.

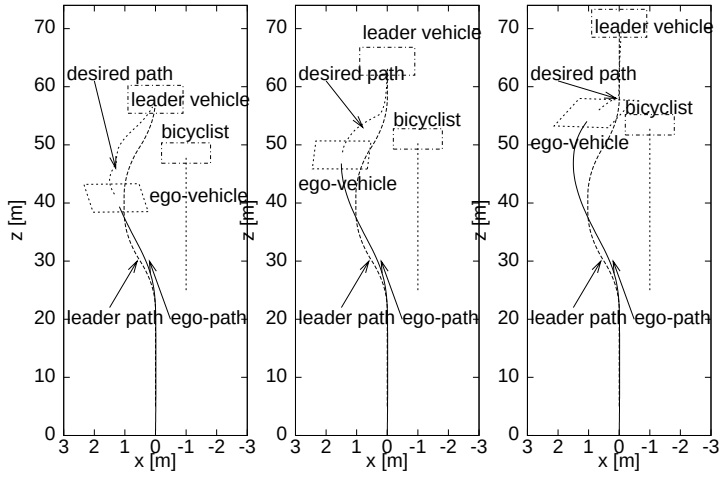

Figure 7.14: Bird view for a situation where the leader vehicle passes a bicycle and the ego-vehicle must avoid the bicycle to not cut it. The ego-vehicle and the leader vehicle drive at $8m/s$ and they are about $18m$ apart. The snapshots show that the bicycle is avoided successfully.

Real World Results Using Stereo Vision

After extensive tests with simulated obstacles, we performed avoidance maneuvers with real image processing data in the research vehicle. A correct mapping of the whole avoidance maneuver in an inertial reference frame is difficult to establish due to localization and measurement uncertainties. In our chosen scenario, a human as a

thin leader vehicle runs on a straight path at about $5m/s$. A cardboard box is located next to that path. The ego-vehicle has to avoid the box while following the human. Figure 7.16 shows the deviation between leader vehicle and ego-vehicle path. The deviation increases at the cardboard box location due to the avoidance maneuver. Figure 7.15 shows a snapshot of the scene. The depicted corridor is the corridor modified by the elastic band algorithm. In a simple vehicle-following scenario, the car would have touched the cardboard box.

Figure 7.15: Avoidance scene with a human as a leader vehicle passing close to a cardboard box on the left. Due to the larger width, the ego-vehicle performs a swerve maneuver to avoid the box. The elastic band path is depicted projected to the ground. The leader vehicle and the ego-vehicle are about $12m$ apart.

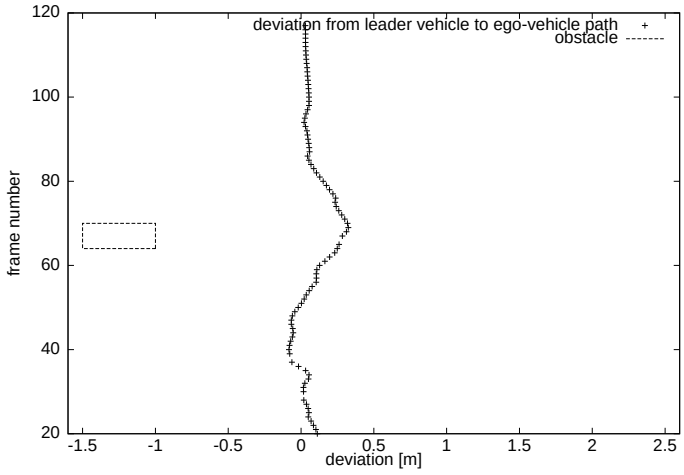

Figure 7.16: Deviation plot for the scene in Figure 7.15. The frame number roughly corresponds to the z position in the global reference frame (constant velocity assumed). Compared to the image data, the cardboard box location is depicted in the graph. Assuming the leader vehicle path runs along the z axis at $x = 0$, the deviation curve also shows the path of the ego-vehicle.

We performed further tests in real traffic using the elastic band framework. The desired avoidance maneuvers were properly performed in all encountered dynamic situations.

Results under Adverse Visibility Conditions

Feasibility studies are usually concerned with default situations. The system presented here is designed to keep a high performance level under all circumstances.

Figure 7.17 shows a scene where the leader car is far away and the vehicle center determination delivers a slightly wrong measurement due to adverse visibility conditions. Since the car is driving between two trucks and the elastic band framework delivers a path with sufficient obstacle clearance, the erroneous center determination is corrected and the ego-vehicle follows the car accurately (see Figure 7.18).

Figure 7.17: Traffic scene with trucks to the left and right of the leader vehicle. Back light, wet street, and a large distance deteriorate detection.

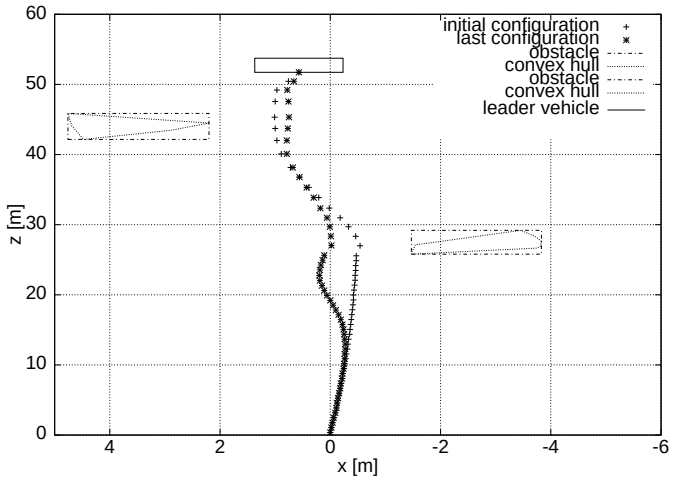

Figure 7.18: Bird view for the scene in Figure 7.17. The leader vehicle path becomes pushed towards a straight path which is desirable for that scenario.

In all realistic scenarios the elastic band algorithm found an equilibrium state within 20 iterations. The average computation time for the algorithm with interfering obstacles is about $10ms$ on a 400 MHz Intel Pentium II PC. Even in cluttered and complex scenes, the maximum computation time always stays below $50ms$.

7.8 Discussion

In this chapter a dynamic collision avoidance component for the standard vehicle-following approach has been introduced. The elastic band framework is used to modify the initial path of the leader vehicle.

The results show, that for standard vehicle-following situations, no modification to the vehicle-following principle is necessary. In dynamic situations, the intuitive obstacle avoidance response is achieved.

Compared to the standard potential field approach of robotics, the global aspect of the whole path constitutes the main difference. So local minima are not a problem here. The feasibility of the whole path is not considered in local potential field approaches and infeasible situations might be detected late. In addition, knowing the planned path is necessary to determine the lateral distance of obstacles to the planned path. This quantity is important, otherwise reflection posts ahead of the ego-vehicle in a curve might cause erroneous responses.

Using potential field approaches only for the lateral control avoids the problem of local minima but above mentioned lateral distance determination problem persists. In general, the elastic band approach comes closer to a human's intuition.

Modeling human driving behavior requires a lot of context knowledge that has to be represented as a rule base in some way. However, we consciously skipped that step and tried to model human driving behavior with a physical model in order to keep things intuitive and to analyze global properties of the planned path (e.g. minimum energy).

All collision avoidance algorithms based on physical models have limited validity in extreme situations. A human driver has a lot more context knowledge and employs many different behavioral patterns. But for the restricted scenario of vehicle following at small to moderate speeds, the above presented algorithm works very well.

Another extension to make the approach more dynamically would be a prediction of the other traffic participant's trajectory. However, other people's intentions and their trajectories are inherently unpredictable. The only restrictions on trajectories can be kinematic or dynamic constraints (e.g. purely lateral motion of cars do not occur). Our approach treats every situation as static but replanning is performed every cycle and hence changing environments become manageable.

An incorporation of traffic sign (especially speed limit signs) and traffic light recognition is possible in the elastic band framework. Systems with these capabilities are listed in Chapter 1. Both would mostly influence the longitudinal control. The lateral control could be supplied by the elastic band framework with no restriction on the end point. Lane following at constant speed without a leader vehicle could be performed without modifications of the current approach. The end point of the elastic band would be the lane center at a certain lookahead distance.

The real-time implementation in the research vehicle has been successfully tested in numerous test drives. Computation time measurements show that real-time performance is achieved without much optimization effort. The result of the elastic band algorithm is successfully put into the feedback loop of the vehicle-following system.

Chapter 8
Conclusions and Outlook

8.1 Summary

Generalizations of and extensions to the vehicle-following problem in regular traffic are presented. This dissertation focuses on the image analysis of traffic scenes based on a stereo camera system as the main sensor. Its main application is autonomous driving in stop-and-go traffic. Within this thesis, a situation assessment is performed. A planning and decision module translates the situation into control commands for the actuators to perform longitudinal and lateral vehicle guidance.

Standard vehicle following is composed of the task of following a leader vehicle at a certain distance guided by only the current position of the leader vehicle. This thesis introduces extensions to that notion which are divided into three parts.

1. An algorithm is developed that allows the autonomous vehicle to exactly follow the path of a preceding vehicle. Unlike other vehicle-following systems, the whole path and not only the current position of the leader vehicle is considered. In the process of reconstructing the path of the preceding vehicle an ego-position estimation in an inertial reference frame is necessary.

2. For the temporal integration of the acquired traffic scene data, a global map in an inertial reference frame of the vehicle's environment is built. This map is the basis for the planning and decision stage that ultimately generates the data for longitudinal and lateral control. For that purpose, a novel ego-position estimation algorithm is developed fusing visual cues with motion parameters. In order to extract symbolic data from the camera data flow, it is necessary to develop some new image processing algorithms. Specifically, a fast algorithm to extract vertical landmarks out of a 3D point cloud is developed.

3. Final control commands to the actuators must guarantee a collision-free path. Pure vehicle-following reaches its limitations when another traffic participant enters the path between the ego-vehicle and the preceding vehicle. This thesis obtains a collision avoidance capability using the notion of an elastic band for the path between the autonomous and the preceding vehicle. The elastic band is deformed by obstacles in its vicinity that exert forces on the band and subsequently deform it. Lane markings constitute virtual obstacles and help the ego-vehicle to stay in its lane. As a result, either a collision-free path is obtained or an infeasible situation is detected. In such a case, a warning to the driver is issued who must take over the control of the vehicle manually.

In this dissertation, design, simulation and implementation of the above mentioned algorithms were presented as well as results with a research vehicle obtained under regular traffic conditions.

8.2 Contributions

Summarizing, the main contributions of this thesis are:

- Design, simulation, implementation, and testing of a path-following algorithm for vehicle-following systems with local sensing. This algorithm can easily be extended to range sensors other than vision. The path-following algorithm described in Chapter 4 is the first algorithm to attempt path following without external infrastructure.
- Design, simulation, implementation, and testing of an ego-position estimation algorithm. This algorithm uses 3D points already extracted from the obstacle detection system. Again, this algorithm can be extended to other range sensor data such as laser range finders. Only the feature extraction part would be different. The ego-position estimation algorithm described in Chapter 5 has the very appealing feature of only using data available from the object detection system. The additional computational burden is minimal and the performance is good when the basic assumption of the landmarks being stationary is not violated.
- Design, simulation, implementation, and testing of a cartography algorithm specifically tailored to increase the density of the 3D point cloud. It is used to extrapolate objects beyond the sensor's field of view. An extension to a parking assistant application is straightforward.
- Design, simulation, implementation, and testing of a collision avoidance algorithm for vehicle-following systems. Again, this algorithm does not explicitly exploit vision sensor data. Any range data could be used. The path of the leader vehicle is treated as an elastic band and is subjected to forces exerted by obstacles in the scene. In this context, lane markings are modeled as virtual obstacles. This way, lane following and vehicle following are combined in our approach. The initial path can also be supplied by a lane detecting system delivering a path in the middle between two lane markings. The extension of the vehicle-following scenario allows even more advanced driving assistance functions. For instance, lane-following with a set speed would be a straightforward application. The intuitively appealing elastic band framework delivers safe modified paths in a vehicle-following scenario and even helps to handle bad measurement data.

8.3 Conclusions

What has been achieved in this thesis? The algorithms developed above are useful for all platooning applications. However, significantly different vehicle dynamics must

be taken into account for trucks. Furthermore, the consideration of all obstacles in the scene is crucial for safety applications and performed here. Even autonomous navigation would be feasible this way provided a panoramic view. In addition to these extended vehicle-following capabilities, warnings of infeasible situations are given earlier, since the whole path is considered and evaluated unlike local approaches.

In this thesis, the two concepts of vehicle following and lane following are combined intuitively using the elastic band framework.

8.4 Future Work

A lot of work remains to be done. Better global goals can be formulated with a global positioning system (e.g. GPS). Digital maps could further help to navigate the vehicle autonomously. Additional recognition modules such as traffic light and traffic sign recognition would contribute towards more autonomy. Elastic bands allow more autonomous driving capabilities even without a leader vehicle. Scenarios such as lane following while keeping a speed limit are feasible. Responses to traffic lights or traffic signs can be integrated. However, the extension of the elastic band framework for longitudinal control is not straightforward since it is a path-planning algorithm. Unlike a trajectory-planning algorithm, no information is given on when to pass a specific point on the path.

The remainder of this section covers future work specifically related to the algorithms developed here.

8.4.1 Future Work for Path Following

The path-following approach introduced in Chapter 4 does not guarantee feasible paths under all circumstances. A kinematic check on the reconstructed path of the leader vehicle is performed but dynamic constraints are not taken into account. This becomes important when the approach is extended beyond the urban velocity range.

Having the whole path of the leader vehicle at hand, it would be beneficial to include the path curvature at the current position in the controller interface. By using such information a reduction of the controller delay might be feasible. Well established lane-following control algorithms also use this information. The lateral control algorithm could be modified such that the path of the preceding vehicle is treated as a lane. Standard lane-following control algorithms could be applied to follow that path.

8.4.2 Future Work for Ego-Position Estimation

The ego-position estimation algorithm using vertical landmarks has been tested extensively in simulation and in regular traffic situations.

Concerning the extraction of vertical landmarks from the list of 3D points, a more accurate method would enhance the quality of the algorithm. A line-fitting algorithm would improve localization of the landmark. Another potential improvement can be achieved by performing temporal matching of the landmark in the grayscale image to improve data association from one frame to the next.

8.4.3 Future Work for Cartography

In order to use this algorithm for highly dynamic scenes, several heuristics must be introduced to suppress old 3D points from fast moving objects. Also, stationary objects beyond the field of view could be kept in the global map for a longer period of time. An application of that algorithm to a parking assistant seems promising.

8.4.4 Future Work for Collision Avoidance

The resulting paths from the elastic band framework do not necessarily comply with the vehicle's dynamic and kinematic constraints. A penalty function for violating these constraints can be envisioned in order to achieve feasible paths.

No intelligent smoothing of the elastic band paths from one frame to another is performed. Obstacles appearing for the first time cause sudden changes to the control input. Smoothing could be achieved by letting the obstacles' impact increase slowly by adjusting the obstacle repulsion gain over a ramp. The same can be done when an obstacle disappears. Clearly, this also increases reaction time in that approach and might result in delayed responses.

In addition, one could investigate the use of the depth map as an alternative to the obstacles as a basis for the potential. This would yield a more reactive response closely related to the sensor data. The number of distance computations would increase significantly but at the same time each computation would degenerate to a distance computation between a point and a line segment.

The elastic band framework for collision avoidance might also benefit from a modification of the lateral control for path following. Treating the elastic band path as a lane and performing lane-following seems promising.

8.5 Outlook

Looking into the future is difficult. Today's sensor and computing technologies allow the transfer of more and more complex functions into the vehicle of tomorrow. Intelligent cruise control functions have already been introduced into the market by several car manufacturers (Mitsubishi, DaimlerChrysler, Subaru). A traffic jam pilot has the very appealing feature of supporting the driver in a task that is tiring and lacks the joy of driving. The algorithms introduced above can be integrated in such a system without any additional hardware costs since they would represent only software in an embedded system.

The algorithms developed in this thesis especially support autonomous driving in regular traffic. The path planning component introduced with the elastic band framework is crucial and superior to local approaches when emulating human driving behavior. Even without a leader vehicle, the framework is still applicable.

Appendix A
Hardware Environment

A.1 The Research Vehicle

The hardware equipment of the research vehicle used for the experiments presented in this thesis is described here. Relevant vehicle data, actuators, the camera system, and the computer equipment is covered.

A.1.1 Physical Dimensions

Our research vehicle is a Mercedes Benz E-class 420 built in 1995. All components in addition to the standard equipment are described below. The relevant physical dimensions of the vehicle are listed in Table A.1.

Distance sensor center to front of car	$1.80m$
Distance sensor center to front axle	$1.00m$
Distance sensor center to cog	$-0.33m$
Distance sensor center to rear axle	$-1.83m$
Car width (tire center to tire center)	$1.60m$
Height above ground of the camera system	$1.17m$
Self-steering gradient	$0.2° s^2/m$

Table A.1: Physical dimensions of the research vehicle used for the experiments. The center of gravity is denoted as cog, a minus indicates that the point lies behind the sensor center.

A.1.2 Actuators

Besides the regular equipment of an E-class model, our research vehicle has additional actuators for throttle, brake and steering wheel.

The throttle is actuated by sending appropriate messages to the ASR system (a drive slip control system). The manual brake system is supplemented with an electronic brake made by ITT (ITT Smart Booster). Brake pressures are limited such that emergency brake maneuvers cannot be performed automatically. However, decelerations up to $8.3m/s^2$ are achieved.

The steering wheel is controlled with an electronic steering mechanism built by Weiss GmbH (Elstar steering control). It provides a mechanical fallback connection to

the steering column. Communication among the various control devices is implemented using two control area networks (CAN). The predecessor of this actuator used in the experiments presented in Chapter 4 is described in Section 3.5.5.

A.1.3 The Stereo Camera System

For the experiments, two cameras made by Cobra were used. The focal lengths of the lenses varied from $12mm$ during the experiments for the CUT algorithm to $7.5mm$ for the elastic band experiments. The technical data of the camera system are listed in Table A.2.

Baseline	$0.23m$
Chip-size	$6.4 \times 4.8mm$
Pixel	$768 \times 568pixel$
Radial distortion coefficient 1st order (12 mm lens)	$0.0023mm^{-2}$
Radial distortion coefficient 1st order (8.5 mm lens)	$0.0035mm^{-2}$
Radial distortion coefficient 1st order (7.5 mm lens)	$0.015mm^{-2}$

Table A.2: Data sheet of the stereo camera system Cobra CS 5132.

A.1.4 Computers

For most of the experiments presented here, the following computer equipment was employed. Table A.3 lists the most important properties of these computers.

Image Processing Computer	Clock rate	$400MHz$
	Operating system	Linux (SuSE 5.3)
	Processor	Intel Pentium II
	RAM	$384MB$
	Frame grabber	Stemmer ITI
Vehicle Control Computer	Clock rate	$200MHz$
	Operating system	Lynx OS 2.09
	Processor	Motorola 68030
	RAM	$64MB$
	Bus	VME Bus
	Modules	CAN module 68333

Table A.3: Data sheet of the vehicle computers.

Bibliography

[Adam et al. 99] A. Adam, E. Rivlin, H. Rotstein. *Fusion of Fixation and Odometry for Vehicle Navigation.* In *Proceedings of the IEEE Conference on Robotics and Automation 99*, Volume 2, pages 1638–1643, 1999.

[Altunbasak et al. 99] Y. Altunbasak, A. J. Patti, O. D. King. *On Global Parametric Motion Estimation with Lens Distortion Correction.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Asada et al. 90] M. Asada et al. *Representing Global World of a Mobile Robot with Relational Local Maps. IEEE Transactions on Systems, Man and Cybernetics*, 20(6):1456–1461, 1990.

[Aschwanden 93] P. F. Aschwanden. *Experimenteller Vergleich von Korrelationskriterien in der Bildanalyse.* Hartung-Gorre Verlag Konstanz, 1993.

[Behringer 94] R. Behringer. *Visuelle Erkennung und Interpretation des Fahrspurverlaufes durch Rechnersehen für ein autonomes Straßenfahrzeug.* PhD Thesis, Universität der Bundeswehr München, 1994.

[Bellon et al. 94] A. Bellon et al. *Real-time Collision Avoidance at Road Crossings on Board the PROMETHEUS-ProLab2 Vehicle.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 56–61, 1994.

[Bergman 65] W. Bergman. *The Basic Nature of Vehicle Understeer-Oversteer. SAE paper 6500085*, 1965.

[Bertozzi et al. 96] M. Bertozzi, A. Broggi. *Real-Time Lane and Obstacle Detection on the GOLD System.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 213–218, September 1996.

[Beymer et al. 96] D. Beymer, J. Malik. *Tracking Vehicles in Congested Traffic.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 130–135, September 1996.

[Bhatt et al. 87] R. Bhatt, D. Gaw, A. Meystel. *A Real-Time System for an Autonomous Vehicle.* In *Proceedings of the IEEE Conference on Robotics and Automation 87*, pages 1785–1791, 1987.

[Blake et al. 92] A. Blake, A. Yuille. *Active Vision.* MIT Press, 1992.

[Bobrow 89] J. E. Bobrow. *A direct minimization approach for obtaining the distance between convex polyhedra. Int. Journal of Robotics Research*, 8(3):65–76, 1989.

[Bohrer et al. 95] T. Bohrer, T. Zielke, V. Freiburg. *An Intelligent Obstacle Detection Framework for Intelligent Cruise Control on Motorways.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 276–281, 1995.

[Borenstein et al. 89] J. Borenstein, Y. Koren. *Real-Time Obstacle Avoidance for Fast Mobile Robots. IEEE Transact. Systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

[Borenstein et al. 91] J. Borenstein, Y. Koren. *Histogrammic In-Motion Mapping for Mobile Robot Obstacle Avoidance. IEEE Transact. on Robotics and Automation*, 7(4):535–539, 1991.

[Borenstein et al. 96] J. Borenstein, H. R. Everett, L. Feng. *Navigating Mobile Robots.* AK Peters Ltd., 1996.

[Brandes 95] M. Brandes. *Entfernungsbestimmung markanter Punkte aus einem Fahrzeug mit Hilfe eines Stereokamerasystems.* Master's Thesis, Fakultät für Informatik, Universität Stuttgart, 1995.

[Brandt et al. 96] S. Brandt, H. D. Dahmen. *Mechanik.* Springer Verlag, 3rd Edition, 1996.

[Brauckmann et al. 94] M. E. Brauckmann et al. *Towards All Around Visual Obstacle Sensing for Cars.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 79–84, 1994.

[Brent 73] R. P. Brent. *Algorithm for Minimization Without Derivatives.* Prentice-Hall Series in Automatic Computation, 1973.

[Brock et al. 98] O. Brock, O. Khatib. *Executing Motion Plans for Robots with Many Degrees of Freedom in Dynamic Environments.* In *Proceedings of the IEEE Conference on Robotics and Automation 98*, pages 1–6, 1998.

[Brock et al. 99] O. Brock, O. Khatib. *High-Speed Navigation Using the Global Dynamic Window Approach.* In *Proceedings of the IEEE Conference on Robotics and Automation 99*, Volume 1, pages 341–346, 1999.

[Broggi 95] A. Broggi. *A Massively Parallel Approach to Real-Time Vision-Based Road Markings Detection.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 84–89, 1995.

[Brooks 86] R.A. Brooks. *A Robust Layered Control System For A Mobile Robot. IEEE Transact. on Robotics and Automation*, 2(1):14–23, 1986.

[Bruyelle et al. 92] J.-L. Bruyelle, J.-G. Postaire. *Disparity Analysis for Real Time Obstacle Detection by Linear Stereo Vision.* In *Proceedings of the Intelligent Vehicles 92 Symposium*, pages 51–56, 1992.

[Bülthoff et al. 89] H. H. Bülthoff, J. Little, T. Poggio. *A Parallel Algorithm for Real-Time Computation of Optical Flow. Nature*, 337(6207):549–553, 1989.

[Bundorf 67] R. T. Bundorf. *The Influence of Vehicle Design Parameters on Characteristic Speed and Understeer. SAE paper 6700078*, 1967.

[Canny et al. 87] J. F. Canny, J. Reif. *New Lower Bound Techniques for Robot Motion Planning Problems.* In *28th Symposium on Foundations of Computer Science*, pages 49–60, 1987.

[Chenavier et al. 92] F. Chenavier, J. L. Crowley. *Position Estimation for a Mobile Robot Using Vision and Odometry.* In *Proceedings of the IEEE Conference on Robotics and Automation 92*, pages 2588–2593, 1992.

[Cormen et al. 90] T. H. Cormen, C. H. Leiserson, R. L. Rivest. *Introduction to Algorithms.* MIT Press, 1st Edition, 1990.

[Courtney et al. 94] J. D. Courtney, A. K. Jain. *Mobile Robot Localization via Classification of Multisensor Maps.* In *Proceedings of the IEEE Conference on Robotics and Automation 94*, pages 1672–1678, 1994.

[Dao et al. 99] K. Dao, S. F. Dow, S. K. Gehrig, R. C. Jared, A. Karcher, J. F. Kral, C. M. LeClerc, M. E Levi, H. von der Lippe, T. H. Liu, K. M. Marks, A. B. Meyer, R. Minor, A. H. Montgomery. *A Binary Link Tracker for the BaBar Level 1 Trigger System. IEEE Transactions on Nuclear Science*, 46(4):928–932, 1999.

[Daviet et al. 96] P. Daviet, M. Parent. *Longitudinal and Lateral Servoing of Cars in a Platoon.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 41–46, 1996.

[DDR99] F. Diaz Del Rio et al. *A Generalization of Path Following for Mobile Robots.* In *Proceedings of the IEEE Conference on Robotics and Automation 99*, pages 7–12, 1999.

[Decuyper et al. 91] J. Decuyper, D. Keymeulen. *A Reactive Robot Navigation System Based on a Fluid Dynamics Metaphor.* In *Proceedings of the Conference on Parallel Problem Solving from Nature.* Springer Verlag, 1991.

[DeSantis 95] R. M. DeSantis. *Path-Tracking for Car-Like Robots with Single and Double Steering. IEEE Transact. on Vehicular Technology*, 44(2):366–377, 1995.

[Dickmanns 98] E. D. Dickmanns. *Vehicles capable of dynamic vision: a new breed of technical beings? Artificial Intelligence*, 103(1-2):49–76, August 1998.

[Dickmanns et al. 94] E. D. Dickmanns et al. *The Seeing Passenger Car VaMoRs-P.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 68–73, 1994.

[Dickmanns et al. 95] E. D. Dickmanns, N. Müller. *Scene Recognition and Landmark Navigation for Road Vehicles.* In *Proceedings of IFAC Conference on Autonomous Vehicles 95*, pages 214–219, 1995.

[Duda et al. 73] R. O. Duda, P. E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[Edelsbrunner 87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer Verlag, 1987.

[Edlinger et al. 91] T. Edlinger et al. *Accurate Position Estimation for an Autonomous Mobile Robot Fusing Shaft Encoder Values and Laser Range Data.* In *IARP, 2nd Workshop on Sensorfusion, Oxford*, 1991.

[Enkelmann et al. 94] W. Enkelmann et al. *Obstacle Detection by Real-Time Optical Flow Evaluation.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 97–102, 1994.

[Estable 96] S. Estable. *Reconnaisance d'Objets En Environnement Dynamique - Application a la Reconnaisance des Panneaux Routiers.* PhD Thesis, Universite Blaise Pascal-Clermont II, Paris, 1996.

[Faugeras 93] O. Faugeras. *3D Computer Vision.* MIT Press, 1993.

[Fenton et al. 91] R. E. Fenton, R. J. Mayhan. *Automated Highway Studies at the Ohio State University — An Overview. IEEE Transact. on Vehicular Technology*, 40(1):100–111, 1991.

[Fiorini et al. 98] P. Fiorini, Z. Shiller. *Motion Planning in Dynamic Environments Using Velocity Obstacles. Int. Journal of Robotics Research*, July 1998.

[Föllinger 94] O. Föllinger. *Regelungstechnik.* Hüthig Verlag, 8th Edition, 1994.

[Franke et al. 94] U. Franke et al. *The Daimler-Benz Steering Assistant - A Spin-Off from Autonomous Driving.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 120–124, 1994.

[Franke et al. 95] U. Franke et al. *Truck Platoon in Mixed Traffic.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 1–6, 1995.

[Franke et al. 96] U. Franke, I. Kutzbach. *Fast Stereo Object Detection for Stop and Go Traffic.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 339–344, 1996.

[Franke et al. 98] U. Franke et al. *Autonomous Driving Goes Downtown. IEEE Intelligent Systems and their Applications*, 13(6):40–48, 1998.

[Franz 98] M. O. Franz. *Minimalistic Visual Navigation.* PhD Thesis, University of Tübingen, 1998.

[Freund 88] E. Freund. *Automatic Anti-Collision Process and Device for Autonomous Vehicles*, 1988. German Patent No 90/02985, European Patent 89/01042.

[Fritz 98] H. Fritz. *Verfahren und Einrichtung zur Regelung der Längsdynamik eines Kraftfahrzeugs*, 1998. Patent DE 196 32 337.

[Fritz 99] H. Fritz. *Longitudinal and Lateral Control of Two Electronically Coupled Heavy-Duty Trucks in the CHAUFFEUR Project.* In *Proceedings of the 6th World Congress on Intelligent Transportation Systems, Toronto*, 1999.

[Fujimura 91] K. Fujimura. *Computer Science Workbench: Planning.* Springer Verlag, 1991.

[Gat et al. 94] E. Gat et al. *Behavior Control for Robotic Exploration of Planetary Surfaces. IEEE Transact. on Robotics and Automation*, 10(4):490–503, 1994.

[Gehrig 91] S. Gehrig. *Erstellung eines heuristischen Verfahrens zur Trendberechnung und Visualisierung am Beispiel "Niedrigwassermenge im Bereich Kernkraftwerk Neckarwestheim".* Master's Thesis, Fakultät für Technische Informatik, Berufsakademie Stuttgart, 1991.

[Gehrig 97] S. Gehrig. *Design and Simulated Performance of the Level 1 Trigger System for the BaBar CP Violation Experiment.* Master's Thesis, Lawrence Berkeley National Laboratory (CA) and University of Tübingen, 1997.

[Gehrig et al. 98a] S. K. Gehrig, F. J. Stein. *An Algorithm for Advanced Lateral Control of an Autonomous Vehicle Applied to Car Following.* In *Proceedings of the Conference on Advances in Vehicle Control and Safety*, pages 188–193, 1998.

[Gehrig et al. 98b] S. K. Gehrig, F. J. Stein. *A Trajectory-Based Approach for the Lateral Control of Car Following Systems.* In *Proceedings of the IEEE Conference on Systems Man and Cybernetics 98*, Volume 4, pages 3596–3601, 1998.

[Gehrig et al. 98c] S. K. Gehrig, F. J. Stein. *A Trajectory-Based Approach for the Lateral Control of Vehicle Following Systems.* In *Proceedings of the Intelligent Vehicles 98 Symposium*, Volume 1, pages 156–161, 1998.

[Gehrig et al. 99a] S. K. Gehrig, F. J. Stein. *Cartography and Dead Reckoning Using Stereo Vision for an Autonomous Vehicle.* In *ISCA International Conference on Intelligent Systems, Denver, USA*, pages 209–212, 1999.

[Gehrig et al. 99b] S. K. Gehrig, F. J. Stein. *Cartography and Dead Reckoning Using Stereo Vision for an Autonomous Car.* In *International Conference on Image Processing, Kobe, Japan*, 1999.

[Gehrig et al. 99c] S. K. Gehrig, F. J. Stein. *Dead Reckoning and Cartography Using Stereo Vision for an Autonomous Car.* In *International Conference on Intelligent Robots and Systems, Kyongju, Korea*, pages 1507–1512, 1999.

[Geist 94] A. Geist. *PVM: Parallel Virtual Machine - A Users' Guide for Network Parallel Computing.* MIT Press Cambridge (MA), London (UK), 1994.

[Gengenbach et al. 98] V. Gengenbach et al. *Model Based Recognition of Intersections and Lane Structures.* In *Vision 98, Stuttgart*, 1998.

[Gerdes et al. 99] J. C. Gerdes, E. J. Rossetter. *A Unified Approach to Driver Assistance Systems Based on Artificial Potential Fields.* In *Proceedings of ASME Int. Mech. Eng. Congress and Exposition, Nashville, TN*, 1999.

[Graefe et al. 96] V. Graefe, W. Efenberger. *A Novel Approach for the Detection of Vehicles on Freeways by Real-Time Vision.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, 1996.

[Grupen et al. 95] R. A. Grupen et al. *Toward a Path Co-Processor for Automated Vehicle Control.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 164–169, 1995.

[Guldner et al. 93] J. Guldner, V. Utkin. *An Integrated System Structure for the Path Control of Mobile Robots. Autonome Mobile Systeme, 9. Fachgespräch*, pages 239–250, 1993.

[Guldner et al. 95] J. Guldner, V. Utkin, R. Bauer. *A Three-Layered Hierarchical Path Control System for Mobile Robots: Algorithms and Experiments. Robots and Autonomous Systems*, 14:133–147, 1995.

[Guldner et al. 96] J. Guldner, H. Tan, S.. Patwardhan. *Analysis of Automatic Steering Control for Highway Vehicles with Look-down Lateral Reference Systems. IEEE Transact. on Vehicle System Dynamics*, 26(4):243–269, 1996.

[Guldner et al. 98] J. Guldner et al. *Robust Steering Control for Automated Highway Systems.* In *Proceedings of the Conference on Advances in Vehicle Control and Safety*, pages 123–129, 1998.

[Hanebeck et al. 99] U. D. Hanebeck, J. Horn. *State Estimations Based on Observations Simultaneously Corrupted by Random Noise with Known Distribution and Uncertainties with Known Bounds.* In *International Conference on Intelligent Robots and Systems, Kyongju, Korea*, pages 665–670, 1999.

[Hashimoto 96] H. Hashimoto, editor. *Proceedings of the Intelligent Vehicles 96 Symposium.* IEEE Catalog Number 96TH8230, 1996.

[Heichel 95] M. Heichel. *Zeitschnittverfahren zur Hinderniserkennung in Straßenfahrzeugen.* Master's Thesis, FHTE Esslingen, 1995.

[Heimes et al. 98] F. Heimes et al. *Model-Based Tracking of Complex Innercity Road Intersections. Math. Comput. Modeling*, 27(9-11):189–203, 1998.

[Heisele 98] B. Heisele. *Objektdetektion in Straßenverkehrsszenen durch Auswertung von Farbbildfolgen.* PhD Thesis, Universität Ulm, 1998.

[Heisele et al. 95] B. Heisele, W. Ritter. *Obstacle Detection Based on Color Blob Flow.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 282–286, 1995.

[Hock 91] C. Hock. *Landmark Navigation with ATHENE 5.* In *International Conference on Advanced Robotics, Pisa*, 1991.

[Hofmeyer 99] A. Hofmeyer. *Verfahren und Anordnung zur Bestimmung eines Regelobjektes*, 1999. Patent DE 197 36 966.

[Hoppen 92] P. Hoppen. *Autonome Mobile Roboter - Echtzeitnavigation in bekannter und unbekannter Umgebung.* BI Wissenschaftsverlag, 1992.

[Hsu 97] J. Hsu. *Estimations of Previewed Road Curvatures and Vehicular Motion by a Vision-Based Data Fusion Scheme. Machine Vision and Application*, 1997.

[Hsu et al. 68] J. C. Hsu, A. U. Meyer. *Modern Control Principles and Applications.* McGraw Hill, 1st Edition, 1968.

[Isard et al. 98a] M. Isard, A. Blake. *CONDENSATION - Conditional Density Propagation for Visual Tracking. International Journal on Computer Vision*, 1998.

[Isard et al. 98b] M. Isard, A. Blake. *A mixed-state CONDENSATION tracker with automatic model-switching.* In *Proceedings of the 6th European Conference on Computer Vision*, 1998.

[Isidori 95] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, 3rd Edition, 1995.

[Iwasaki 99] Y. Iwasaki. *Japan's Policies on Research and Development of the Advanced Cruise-Assist System.* In *Proceedings of the 6th World Congress on Intelligent Transportation Systems, Toronto*, 1999.

[Jähne 91] B. Jähne. *Digitale Bildverarbeitung.* Springer-Verlag, 2nd Edition, 1991.

[Jocoy et al. 99] E. H. Jocoy, H. A. Pirson. *Integration of Radar and GPS/GIS for Intersection Threat Detection and Collision Avoidance.* In *Proceedings of the 6th World Congress on Intelligent Transportation Systems, Toronto*, 1999.

[Joseph 87] P. D. Joseph. *Filtering for Stochastic Processes with Applications to Guidance.* AMS Chelsea Publishing Distributor, 1987.

[Juberts et al. 93] M. Juberts, D. Raviv. *Vision-Based Vehicle Control for AVCS.* In *Proceedings of the Intelligent Vehicles 93 Symposium*, pages 195–200, 1993.

[Kalman 60] R. E. Kalman. *A new approach to linear filtering and prediction problems. Transactions ASME J. of Basic Engineering, Series D*, 82:35–45, March 1960.

[Kanade et al. 95] T. Kanade et al. *Development of a Video Rate Stereo Machine.* In *International Conference on Intelligent Robots and Systems, Pittsburgh, PA*, 1995.

[Kang et al. 95] F. Kang, S. Zhong-Ci. *Mathematical Theory of Elastic Structures.* Springer-Verlag, Science Press Beijing, 1995.

[Kass et al. 88] M. Kass, A. Witkin, D. Terzopoulos. *Snakes: Active Contour Models. Int. Journal of Computer Vision*, pages 321–331, 1988.

[Kawabe et al. 96] T. Kawabe et al. *A Concept of Car in a Virtual Moving Cell Platoon Construction on an Automated Highway System.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 397–402, 1996.

[Kehtarnavaz et al. 91] N. Kehtarnavaz et al. *Visual Control of an Autonomous Vehicle (BART) — The Vehicle-Following Problem. IEEE Transact. on Vehicular Technology*, 40(3):654–662, 1991.

[Khatib 85] O. Khatib. *Real-Time Obstacle Avoidance for Manipulators and Mobile Robots.* In *Proceedings of the IEEE Conference on Robotics and Automation 85*, 1985.

[Khatib et al. 95] M. Khatib, R. Chatila. *An Extended Potential Field Approach for Mobile Robot Sensor-Based Motions.* In *4th Int. Conf. on Intelligent Autonomous Systems*, pages 490–496, 1995.

[Khatib et al. 96] M. Khatib et al. *Dynamic Path Modification for Car-Like Nonholonomic Mobile Robots.* In *Proceedings of the IEEE Conference on Robotics and Automation 97*, Volume 4, pages 2920–2925, 1996.

[Knoeppel et al. 99] C. Knoeppel, U. Regensburger, B. Michaelis. *Erkennung und Bestimmung des Abstandes von Straßenfahrzeugen in großer Entfernung.* In *Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM)*, pages 258–265, 1999.

[Konolige 97] K. Konolige. *Small Vision Systems.* In *Proceedings of the International Symposium on Robotics Research, Hayama, Japan*, 1997.

[Koren et al. 91] Y. Koren, J. Borenstein. *Potential Field Methods and Their Inherent Limitations For Mobile Robot Navigation.* In *Proceedings of the IEEE Conference on Robotics and Automation 91*, pages 1398–1404, 1991.

[Kosecka 98] J. Kosecka. *A Comparative Study of Vision-Based Lateral Control Strategies for Autonomous Highway Driving. IEEE Transactions of Robotics and Automation*, 1998.

[Kreucher et al. 99] C. Kreucher, S. Lakshmanan. *A Frequency-Domain Approach to Lane Detection in Roadway Images.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Krogh 83] B. H. Krogh. *Feedback Obstacle Avoidance Control.* In *21st Allerton Conference on Communication, Control, and Computing*, pages 325–334, 1983.

[Krogh 84] B. H. Krogh. *A Generalized Potential Field Approach to Obstacle Avoidance Control.* In *Conference on Robotics Research: The next five years and beyond, Bethlehem, PA*, August 1984.

[Krogh et al. 86] B. H. Krogh, C. E. Thorpe. *Integrated Path Planning and Dynamic Steering Control for Autonomous Vehicles.* In *Proceedings of the IEEE Conference on Robotics and Automation 86*, pages 1664–1669, 1986.

[Krotkov 89] E. P. Krotkov. *Active Computer Vision by Cooperative Focusing and Stereo.* Springer-Verlag, 1989.

[Kujawski 95] C. Kujawski. *Deciding the Behaviour of an Autonomous Road Vehicle.* In *Proceedings of IFAC Conference on Autonomous Vehicles 95*, pages 404–409, 1995.

[Land et al. 94] M. F. Land, D. N. Lee. *Where we Look When We Steer. Nature*, 369:742-744, 1994.

[Langer et al. 96] D. Langer, T. Jochem. *Fusing Radar and Vision for Detecting, Classifying and Avoiding Roadway Obstacles.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 333–338, 1996.

[Lenz et al. 89] R. Lenz, R. Y. Tsai. *Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, 1989.

[Leven et al. 99] P. Leven, S. Hutchinson, D. Burschka, G. Färber. *Perception-Based Motion Planning for Indoor Exploration.* In *Proceedings of the IEEE Conference on Robotics and Automation 99*, Volume 1, pages 695–700, 1999.

[Lin et al. 91] M. C. Lin, J. F. Canny. *A fast algorithm for incremental distance calculation.* In *Proceedings of the IEEE Conference on Robotics and Automation 91*, pages 1008–1014, 1991.

[Liter et al. 96] J. Liter, H. H. Bülthoff. *An Introduction to Object Recognition.* Technical report, TR-043, Max-Planck-Institut für Biologische Kybernetik, 1996.

[Lorei et al. 99] M. Lorei, C. Stiller. *Visual Sensing in Electronic Truck Coupling.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Ludmann 95] J. Ludmann. *Analysis of ACC Concepts with the Simulation Tool PELOPS.* In *5. Aachener Kolloquium Fahrzeug- und Motorentechnik*, October 1995.

[Lumelsky 85] V. J. Lumelsky. *On fast computation of distance between line segments. Information Processing Letters*, 21:55–61, 1985.

[Lund 91] S. A. Lund. *Intelligent Vehicle/Highway Systems in the US - in Need of a Vision.* SAE Warrendale, 1991.

[Ma et al. 98] Y. Ma, J. Kosecka, S. Sastry. *Motion Recovery from Image Sequences: Discrete Viewpoint vs. Differential Viewpoint.* In *European Conference on Computer Vision (ECCV)*, 1998.

[Malik et al. 97] J. Malik et al. *Development of Binocular Stereopsis for Vehicle Lateral Control, Longitudinal Control and Obstacle Detection.* Technical report, PATH MOU-257 Final Report, 1997.

[Mallot et al. 91] H.A. Mallot, H. H. Bülthoff, J. J. Little, S. Bohrer. *Inverse Perspective Mapping Simplifies Optical Flow Computation and Obstacle Detection. Biological Cybernetics*, 64:177–185, 1991.

[Marks et al. 99] K. M. Marks, S. K. Gehrig, C. Grace, R. C. Jared, A. Karcher, D. Kasen, J. F. Kral, C. M. LeClerc, M. E Levi, H. von der Lippe, T. H. Liu, K. M. Marks, A. B. Meyer, R. Minor, A. H. Montgomery. *Continuously Live Image Processor for Drift Chamber Track Segment Triggering. IEEE Transactions on Nuclear Science*, 46(3):348–353, 1999.

[Marr et al. 79] D. Marr, T. Poggio. *A Computational Theory of Human Stereo Vision. Proc. R. Soc. Lond.*, B 204:301–328, 1979.

[Marr et al. 80] D. Marr, E. Hildreth. *Theory of Edge Detection. Proc. R. Soc. Lond.*, B 207:187–217, 1980.

[Masaki 98] I. Masaki. *Vison-Based Driving-Assistance. IEEE Intelligent Systems and their Applications*, 13(6):24–31, 1998.

[Masaki et al. 94] I. Masaki et al. *Cost-Effective Vision Systems for Intelligent Vehicles.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 39–43, 1994.

[Matthies et al. 87] L. Matthies, S. A. Shafer. *Error Modeling in Stereo Navigation. IEEE Transact. on Robotics and Automation*, RA-3(3):239–248, 1987.

[Matthies et al. 88] L. Matthies, A. E. Elfes. *Integration of Sonar and Stereo Range Data Using a Grid-Based Representation.* In *Proceedings of the IEEE Conference on Robotics and Automation 88*, 1988.

[Matthies et al. 89] L. Matthies, R. Szelinski, T. Kanade. *Kalman Filter Based Algorithms for Estimating Depth from Image Sequences. Int. Journal of Computer Vision*, 3:209–238, 1989.

[Matthies et al. 94] L. Matthies, P. Grandjean. *Stochastic Performance Modeling and Evaluation of Obstacle Detectability with Imaging Range Sensors. IEEE Transact. on Robotics and Automation*, 10(6):783–792, 1994.

[Matthies et al. 95] L. Matthies et al. *Obstacle Detection for Unmanned Ground Vehicles: A Progress Report.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 66–71, 1995.

[Mayr 95] R. Mayr. *Lateral Vehicle Control for Automated Lane Following.* In *Proceedings of IFAC Conference on Autonomous Vehicles 95*, pages 273–278, 1995.

[McLauchlan et al. 97] P. F. McLauchlan, J. Malik. *Vision for Longitudinal Control of Vehicles.* In *Proceedings of the IEEE Conference on Intelligent Transportation Systems, Boston*, 1997.

[Meikle et al. 97] S. Meikle, R. Yates, A. Harris. *Computer Vision Algorithms for Autonomous Mobile Robot Map Building and Path Planning.* In *Intelligent Transportation Systems, Boston, MA*, 1997.

[Metea et al. 87] M. B. Metea, J. J.-P. Tsai. *Route Planning for Intelligent Autonomous Land Vehicles Using Hierarchical Terrain Representation.* In *Proceedings of the IEEE Conference on Robotics and Automation 87*, pages 1947–1952, 1987.

[Moravec 80] H.-P. Moravec. *Obstacle Navigation in the Real World by a Seeing Robot Rover.* PhD Thesis, Stanford University, 1980.

[Moravec 96] H.-P. Moravec. *Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids.* Technical report, CMU-RI-TR-96-34, Carnegie Mellon University, Pittsburgh, 1996.

[Moravec et al. 85] H. Moravec, A. E. Elfes. *High Resolution Maps from Wide Angle Sonars.* In *Proceedings of the IEEE Conference on Robotics and Automation 85*, pages 116–121, 1985.

[Moravec et al. 91] H.-P. Moravec, M. Blackwell. *Learning Sensor Models for Evidence Grids.* Technical report, Annual Research Review, Carnegie Mellon University, Pittsburgh, 1991.

[Moravec et al. 99] K. Moravec, R. Harvey, J. A. Bangham, M. Fisher. *Using an Image Tree to Assist Stereo Matching.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Mori et al. 95] H. Mori et al. *Danger Estimation of Vehicles at Intersection.* In *Proceedings of the IEEE Conference on Robotics and Automation 95*, pages 781–787, 1995.

[Niehaus et al. 94] A. Niehaus, R. F. Stengel. *Probability-Based Decision Making for Automated Highway Driving. IEEE Transact. on Vehicular Technology*, 43(3):626–634, 1994.

[Ninomiya et al. 95] Y. Ninomiya et al. *A Real-Time Vision System for Intelligent Vehicles.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 315–320, 1995.

[oC98] MOC (Ministry of Construction). *1998 ITS Hand Book Japan.* Highway Industry Development Organization, October 1998.

[Overington 92] I. Overington. *Computer Vision: A unified, biologically-inspired Approach.* Elsevier, 1992.

[Paclik et al. 99] P. Paclik et al. *Road Sign Classification using Laplace Kernel Classifier.* In *Scandinavian Conference on Image Analysis, SCIA'99*, 1999.

[Papageorgio et al. 99] C. Papageorgio, T. Poggio. *Trainable Pedestrian Recognition.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Parent et al. 94] M. Parent et al. *Automatic Driving in Stop and Go Traffic.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 183–188, 1994.

[Penteker 99] K.-H. Penteker. *Kartographierung der Fahrzeugumgebung eines autonomen Fahrzeugs mittels Stereo-Bildverarbeitung.* Master's Thesis, Diplomarbeit Nr. 1591, Fakultät für Informatik, Universität Stuttgart, 1999.

[Petrov 95] A. A. Petrov. *Path Planning by Intelligent Autonomous Robotic Vehicles with Growing World Models.* In *Proceedings of IFAC Conference on Autonomous Vehicles 95*, pages 56–61, 1995.

[Pomerleau 92] D. A. Pomerleau. *Neural Network Perception for Mobile Robot Guidance.* PhD Thesis, Carnegie-Mellon University, Pittsburgh, 1992.

[Pope et al. 94] A. R. Pope, D. G. Lowe. *Vista: A Software Environment for Computer Vision Research.* In *Proceedings of Int. Conference on Computer Vision and Pattern Recognition 94*, 1994.

[Prassler et al. 99] E. Prassler, J. Scholz, P. Fiorini. *Navigating a Robotic Wheelchair in a Railway Station During Rush-Hour. Int. Journal of Robotics Research*, May 1999.

[Quinlan 94] S. Quinlan. *Real-Time Modification of Collision-Free Paths.* PhD Thesis, Stanford University, 1994.

[Quinlan et al. 93a] S. Quinlan, O. Khatib. *Elastic Bands: Connecting Path Planning and Control.* In *Proceedings of the IEEE Conference on Robotics and Automation 93*, pages 802–807, 1993.

[Quinlan et al. 93b] S. Quinlan, O. Khatib. *Towards Real-Time Execution on Motion Tasks.* In *Experimental Robotics 2, Springer-Verlag*, pages 241–254, 1993.

[Raboisson et al. 94] S. Raboisson, P. Schmouker. *Obstacle Detection by Colour CCD Camera and Image Processing Prototype Installed in a Vehicle.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 44–49, 1994.

[Reichardt 96] D. Reichardt. *Kontinuierliche Verhaltenssteuerung eines autonomen Fahrzeugs in dynamischer Umgebung.* PhD Thesis, Universität Kaiserslautern, 1996.

[Rencken 94] W. D. Rencken. *Autonomous Sonar Navigation in Indoor, Unknown and Unstructured Environments.* In *International Conference on Intelligent Robots and Systems, Germany*, pages 431–438, 1994.

[Rieger 00] B. Rieger. *Entwurf, Simulation und Implementierung von Hindernisvermeidungsstrategien für ein autonomes Fahrzeug.* Master's Thesis, Fakultät für Physik, Universität Tübingen, 2000.

[Ritter 92] W. Ritter. *Traffic Sign Analysis in Color Image Sequences.* In *Proceedings of the Intelligent Vehicles 92 Symposium*, pages 12–17, 1992.

[Ritter 96] W. Ritter. *Automatische Verkehrszeichenerkennung.* PhD Thesis, Universität Ulm, 1996.

[Saneyoshi 94] K. Saneyoshi. *3-D Image Recognition System by Means of Stereoscopy Combined with Ordinary Image Processing.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 13–18, 1994.

[Saneyoshi 96] K. Saneyoshi. *Drive Assist System Using Stereo Image Recognition.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 230–235, 1996.

[Saneyoshi et al. 93] K. Saneyoshi et al. *3D-Image Recognition System for Drive Assist.* In *Proceedings of the Intelligent Vehicles 93 Symposium*, pages 60–65, 1993.

[Schenk 98] W. Schenk. *Aufbau einer bildgestützten Vermessungsanlage zur Koordination eines Robotergreifarms im Rahmen der Automobilproduktion.* Master's Thesis, Diplomarbeit Nr. 1603, Fakultät für Informatik, Universität Stuttgart, 1998.

[Schiehlen et al. 94] J. Schiehlen, E. D. Dickmanns. *A Camera Platform for Intelligent Vehicles.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 393–398, 1994.

[Schiffmann et al. 97] J. K. Schiffmann, G. R. Widmann. *Model-Based Scene Tracking Using Radar Sensors for Intelligent Automotive Vehicle Systems.* In *Intelligent Transportation Systems, Boston, MA*, 1997.

[Schreer 98] O. Schreer. *Dynamische Analyse von rekonstruierten 3D-Punkten zur Navigation eines Stereo-Vision basierten mobilen Systems.* In *Mustererkennung 98, DAGM Symposium*, pages 523–530, 1998.

[Scrase 99] R. Scrase. *Informed Approach. ITS International - Advanced Technology for Traffic Management and Urban Mobility*, pages 70–72, July/August 1999.

[Shiller et al. 96] Z. Shiller, S. Sundar. *Emergency Maneuvers of Autonomous Vehicles.* In *13th World Congress of IFAC*, Volume Q, pages 393–398, 1996.

[Shirai 87] Y. Shirai. *Three-Dimensional Computer Vision.* Springer-Verlag, 1987.

[Shladover et al. 91] S. E. Shladover et al. *Automatic Vehicle Control Developments in the PATH Program. IEEE Transact. on Vehicular Technology*, 40(1):114–129, 1991.

[Sobiesk et al. 98] E. Sobiesk et al. *Tracking Multiple Targets in Terrain.* In *Proceedings of the IEEE Conference on Systems Man and Cybernetics 98*, pages 2842–2847, 1998.

[Steinhage et al. 97] A. Steinhage, G. Schöner. *Self-Calibration Based in Invariant View Recognition: Dynamic Approach to Navigation. Robotics and Autonomous Systems*, 20:133–156, 1997.

[Steinhage et al. 98] A. Steinhage, G. Schöner. *Dynamical Systems for the Behavioral Organization of Autonomous Robot Navigation.* In *Sensor Fusion and Decentralized Control in Robotic Systems: Proceedings of SPIE Vol. 3523*, 1998.

[Straube et al. 97] M. Straube, M. Pöppel-Decker. *Unfallentwicklung Straßenverkehr.* Faltblatt, Bundesanstalt für Straßenwesen, 12 1997.

[Sub99] Subaru Inc. *Manual Subaru Legacy, ADA - Active Drive Assist*, October 1999.

[Sukthantar 93] R. Sukthantar. *RACCOON: A Real-time Autonomous Car Chaser Operating Optimally at Night.* In *Proceedings of the Intelligent Vehicles 93 Symposium*, pages 37–42, 1993.

[Svoboda et al. 98] T. Svoboda, T. Pajdla, V. Hlavac. *Motion Estimation Using Central Panoramic Cameras.* In *Proceedings of the Intelligent Vehicles 98 Symposium*, pages 335–340, 1998.

[Tang et al. 96] Q. Tang et al. *Automatic Vehicle Following System*, 1996. VI & T Group, US Patent No 5572449.

[Taylor et al. 96] C. J. Taylor, J. Malik, J. Weber. *A Real-Time Approach to Stereopsis and Lane-Finding.* In *Proceedings of the Intelligent Vehicles 96 Symposium*, pages 207–212, 1996.

[Thorpe et al. 97] C. Thorpe, T. Jochem, D. Pomerleau. *Automated Highways and the Free Agent Demonstration.* In *International Symposium on Robotics Research, October*, October 1997.

[Tilove 90] R. B. Tilove. *Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials.* In *Proceedings of the IEEE Conference on Robotics and Automation 90*, pages 566–571, 1990.

[Tsai 87] R. Y. Tsai. *A Versatile Camera Calibration Technique for High-Accuracy Machine Vision Metrology Using Off-the-Shelf TV Cameras. IEEE Transact. on Robotics and Automation*, 3(4), 1987.

[Uchimura et al. 98] K. Uchimura, Z. Hu. *Lane Detection and Tracking Using Estimated Camera Parameters for Intelligent Vehicles.* In *Proceedings of the Intelligent Vehicles 98 Symposium*, pages 11–16, 1998.

[Ulmer 94] B. Ulmer. *VITA II - Active Collision Avoidance in Real Traffic.* In *Proceedings of the Intelligent Vehicles 94 Symposium*, pages 1–6, 1994.

[vP91] E. von Puttkamer et al. *Local Obstacle Avoidance and Acceleration Based Motion Control for an Autonomous Mobile Robot.* In *IARP, 2nd Workshop on Sensorfusion, Oxford*, 1991.

[vP93] E. von Puttkamer et al. *Lokale Kollisionsvermeidung durch Ansteuerung der maximalen Hindernisdistanz für einen autonomen mobilen Robote.* In *9.Fachgespräch "Autonome Mobile Roboter", München*, 1993.

[Wagner et al. 97] R. Wagner, K. Donner, F. Liu. *A "Half-Perspective" Approach to Robust Ego-Motion Estimation for Calibrated Cameras.* Technical report, Universität Passau, 1997.

[Wallace et al. 85] R. Wallace et al. *First Results in Robot Road-Following.* In *Proceedings of the IJCAI*, 1985.

[Weber et al. 95] J. Weber et al. *New Results in Stereo-Based Automatic Vehicle Guidance.* In *Proceedings of the Intelligent Vehicles 95 Symposium*, pages 530–535, 1995.

[Welch et al. 95] G. Welch, G. Bishop. *An Introduction to the Kalman Filter.* Technical report, University of North Carolina at Chapel Hill, 1995. Available at http://www.cs.unc.edu/ welch.

[Wershofen 95] K.-P. Wershofen. *Zur Navigation sehender mobiler Roboter in Wegnetzen von Gebäuden.* PhD Thesis, Universität der Bundeswehr München, 1995.

[Williamson 98a] T Williamson. *Detection of Small Obstacles at Long Range Using Multibaseline Stereo.* In *Proceedings of the Intelligent Vehicles 98 Symposium*, Volume 1, pages 311–316, 1998.

[Williamson 98b] T Williamson. *A High-Performance Stereo Vision System for Obstacle Detection.* PhD Thesis, CMU-RI-TR-98-24, Carnegie Mellon University, Pittsburgh, 1998.

[Wyatt et al. 92] J. L. Wyatt et al. *Small Fast Analog VLSI Systems for Early Vision Processing.* In *Proceedings of the Intelligent Vehicles 92 Symposium*, pages 69–73, 1992.

[Yagi et al. 88] Y. Yagi et al. *Dynamic Scene Analysis for a Mobile Robot in a Man-Machine Environment. Systems and Computers in Japan*, 19(2):1–9, 1988.

[Yamauchi et al. 96] B. Yamauchi, R. Beer. *Spatial Learning for Navigation in Dynamic Environments. IEEE Transactions on Systems, Man, and Cybernetics*, 26(3):496–505, 1996.

[Yamauchi et al. 98] B. Yamauchi, A. Schultz, W. Adams. *Mobile Robot Exploration and Map-Building with Continuous Localization.* In *Proceedings of the IEEE Conference on Robotics and Automation 98*, pages 3715–3720, 1998.

[Yezzi et al. 99] A. Yezzi, A. Tsai, A. Willsky. *Binary and Ternary Flow for Image Segmentation.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Yoda et al. 98] I. Yoda, K. Sakaue. *Utilization of Stereo Disparity and Optical Flow Information for Human Interaction.* In *IEEE International Conference on Computer Vision*, pages 1109–1114, 1998.

[Yoshioka et al. 99] T. Yoshioka, H. Uemura, H. Nakaue. *Development of Detection Algorithm for Vehicles Using Multi-Line CCD Sensor.* In *Int. Conference on Image Processing, Kobe, Japan*, 1999.

[Zamperoni 91] P. Zamperoni. *Methoden der digitalen Bildverarbeitung.* Vieweg-Verlag, 2nd Edition, 1991.

[Zhang et al. 97] Z. Zhang, R. Weiss, A. R. Hanson. *Obstacle Detection Based on Qualitative and Quantitative 3D Reconstruction. IEEE Transact. Pattern Analysis & Machine Intelligence*, 14(12):15–26, 1997.

[Zielke et al. 93] T. Zielke et al. *Intensity and Edge-Based Symmetry Detection with an Application to Car Following. CVGIP: Image Understanding*, 58(1), 1993.

[Zomotor 87] A. Zomotor. *Fahrwerktechnik: Fahrverhalten.* Vogel Buchverlag Würzburg, 1987.