

Augmenting Vehicle Localization with Visual Context

by

Robert Andrew Rae

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2009

© Robert Andrew Rae 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Vehicle self-localization, the ability of a vehicle to determine its own location, is vital for many aspects of Intelligent Transportation Systems (ITS) and telematics where it is often a building block in a more complex system. Navigation systems are perhaps the most obvious example, requiring knowledge of the vehicle's location on a map to calculate a route to a desired destination. Other pervasive examples are the monitoring of vehicle fleets for tracking shipments or dispatching emergency vehicles, and in public transit systems to inform riders of time-of-arrival thereby assisting trip planning. These systems often depend on Global Positioning System (GPS) technology to provide vehicle localization information; however, GPS is challenged in urban environments where satellite visibility and multipath conditions are common. Vehicle localization is made more robust to these issues through augmentation of GPS-based localization with complementary sensors, thereby improving the performance and reliability of systems that depend on localization information.

This thesis investigates the augmentation of vehicle localization systems with visual context. Positioning the vehicle with respect to objects in its surrounding environment in addition to using GPS constraints the possible vehicle locations, to provide improved localization accuracy compared to a system relying solely on GPS. A modular system architecture based on Bayesian filtering is proposed in this thesis that enables existing localization systems to be augmented by visual context while maintaining their existing capabilities.

It is shown in this thesis that localization errors caused by GPS signal multipath can be reduced by positioning the vehicle with respect to visually-detected intersection road markings. This error reduction is achieved when the identities of the detected road marking and the road being driven are known *a priori*. It is further shown how to generalize the approach to the situation when the identities of these parameters are unknown. In this situation, it is found that the addition of visual context to the vehicle localization system reduces the ambiguity of iden-

tifying the road being driven by the vehicle. The fact that knowledge of the road being driven is required by many applications of vehicle localization makes this a significant finding.

A related problem is also explored in this thesis: that of using vehicle position information to augment machine vision. An approach is proposed whereby a machine vision system and a vehicle localization system can share their information with one another for mutual benefit. It is shown that, using this approach, the most uncertain of these systems benefits the most by this sharing of information.

Augmenting vehicle localization with visual context is neither farfetched nor impractical given the technology available in today's vehicles. It is not uncommon for a vehicle today to come equipped with a GPS-based navigation system, and cameras for lane departure detection and parking assistance. The research in this thesis brings the capability for these existing systems to work together.

Acknowledgements

First and foremost, I would like to thank my supervisor Professor Otman Basir for his help and guidance throughout my research, for giving me the opportunity to pursue a PhD with him, and for ensuring that I gained practical knowledge along the way. The experience has been truly rewarding.

I would like to thank the members of my thesis defense committee for taking the time to read my thesis and participate in my defense. Special thanks to my external examiner, Professor Philippe Bonnifait from the Université de Technologie de Compiègne, France, for his helpful comments and insight during his visit to Waterloo.

I would like thank my fellow students in the Pattern Analysis and Machine Intelligence group that I have had the opportunity to work with and know over the past few years. In particular, I would like to thank Mustapha, Srinath, Adams, Nabil, Patrick and Mohammed. I wish you all the best.

My family has been a constant source of support throughout my studies. Although we rarely have the opportunity to see each other, I treasure what time we are able to spend together.

Finally, I would like to thank Natalie for her support and understanding. I could not have achieved this without her.

Contents

List of Tables	xi
List of Figures	xv
Nomenclature	xvi
1 Introduction	1
1.1 Problem Description	2
1.2 Incorporating Visual Context into Localization	4
1.2.1 Challenges of Incorporating Visual Context into Localization	5
1.3 Thesis Objectives	8
2 Background and Literature Review	10
2.1 Vehicle Localization Background	11
2.2 Bayesian Filtering Background	16
2.2.1 Dynamic Bayesian Networks	16
2.2.2 Bayesian Filtering Methods	18
2.3 Vehicle Localization Literature Review	23
2.3.1 Passive Localization	23
2.3.2 Map Matching	24
2.3.3 Vision-based Localization	26
2.3.4 Data Association	29
2.4 Summary	31

3	A System Architecture for Vehicle Localization	33
3.1	Theoretical Foundations	34
3.1.1	Notation	34
3.1.2	Vehicle Localization	35
3.1.3	Data Association	37
3.1.4	Map Matching	38
3.2	System Architecture	40
3.2.1	Passive Localization Module	40
3.2.2	Map Matching Module	41
3.2.3	Vision Module	42
3.2.4	Map Database	42
3.2.5	Data Association Module	42
3.2.6	Visual Context Module	43
3.3	Summary	43
4	Augmenting Vehicle Localization with Visual Context	44
4.1	Modeling Observed Visual Features	44
4.1.1	A Road-Based Coordinate System	45
4.1.2	Measurement Models	49
4.2	Localization by Kalman Filtering	52
4.2.1	Passive Localization by GPS	54
4.2.2	Augmenting Localization using Map Matching	59
4.2.3	Augmenting Localization with Visual Context	64
4.2.4	Discussion	68
4.3	Localization by Particle Filtering	71
4.3.1	Passive Localization by GPS	72
4.3.2	Augmenting Localization using Map Matching	74
4.3.3	Augmenting Localization with Visual Context	76
4.4	Summary	79

5	Data Association: Map Matching and Landmark Identification	81
5.1	Multiple-Hypothesis Tracking using Kalman Filtering	82
5.1.1	Introduction	82
5.1.2	MHT Algorithm Description	85
5.1.3	MHT Performance in Unbiased GPS Conditions	98
5.1.4	MHT Performance in Biased GPS Conditions	108
5.2	Monte Carlo Data Association using Particle Filtering	111
5.2.1	Introduction	111
5.2.2	Algorithm Description	113
5.2.3	Results	117
5.2.4	Discussion	124
5.3	Summary	127
6	Experimental Investigations	128
6.1	Objectives	128
6.2	Experimental Setup	129
6.2.1	Hardware and Installation	129
6.2.2	Data Acquisition	131
6.2.3	Data Collection and Preprocessing	131
6.3	System Components	133
6.3.1	Road Network Database	134
6.3.2	Vision Module	135
6.3.3	Measurement Models	138
6.4	Experiment 1: Best-case localization performance	140
6.5	Experiment 2: Data Association in Unbiased GPS Conditions	143
6.6	Experiment 3: Data Association in Biased GPS Conditions	145
6.7	Summary	148
7	Mutual Influence Between Localization and Visual Feature Estimation	149
7.1	A Mutual Influence Formulation using Kalman Filtering	150
7.1.1	A Dynamic Bayesian Network Model	150
7.1.2	Integration into Modular System Architecture	152

7.1.3	Extended Kalman Filter Inference	154
7.2	Experimental Investigation	157
7.2.1	Experimental Results	157
7.3	Discussion	161
7.3.1	Correcting Detection Errors	161
7.3.2	Uncertainty Tradeoff	162
7.4	Summary	165
8	Concluding Remarks	167
8.1	Summary of Contributions	167
8.1.1	Chapter 3: System Architecture	167
8.1.2	Chapter 4: Visual Context in Localization	168
8.1.3	Chapter 5: Data Association	171
8.1.4	Chapter 6: Experimental Investigations	174
8.1.5	Chapter 7: Mutual Influence	174
8.2	Extensions to Thesis Research	175
8.2.1	Maintaining Accurate Localization Without GPS	176
8.2.2	Generalizing Data Association to Improved Visual Capability	176
8.2.3	Incorporating Uncertainty in Map Information	177
8.2.4	Improved Visual Capability and Environmental Awareness .	177
8.2.5	Using Shared Sensory Capability from Other Vehicles	177
8.2.6	Building and Updating a Map of the Environment	178
8.2.7	Impact of Vision on Other Localization Criteria	178
	Bibliography	179

List of Tables

4.1	A comparison of position error and uncertainty between the raw GPS data and Passive Localization using Kalman filtering.	59
4.2	A comparison of lateral position error and uncertainty between Passive Localization, and Passive Localization and Map Matching using Kalman filtering.	64
4.3	A comparison of longitudinal position error and uncertainty between Passive Localization and Map Matching, and Passive Localization, Map Matching and Visual Context, using Kalman filtering.	67
4.4	Summary of feature rate-of-change dependency on position and variance gains	69
4.5	A comparison of position error and uncertainty between Passive Localization, and Passive Localization and Map Matching, using particle filtering.	74
4.6	A comparison of longitudinal position error and uncertainty between Passive Localization and Map Matching, and Passive Localization, Map Matching and Visual Context, using particle filtering.	76
5.1	Lateral and longitudinal position error and uncertainty for the raw GPS measurements, and for MHT with and without visual context.	104
5.2	Longitudinal position error and uncertainty for the raw GPS measurements, and MHT with and without visual context (VC) while a road marking is visible.	104
5.3	Predicted and observed convergence times for MHT localization with and without using Visual Context.	106
5.4	Lateral and longitudinal position error and uncertainty for particle filtering with Monte Carlo data association.	123
5.5	Comparison of MHT and PF data association techniques.	127
6.1	Confusion matrix for road marking detector.	137
6.2	Pseudomeasurement parameters as a function of longitudinal vehicle position.	139

7.1	Visual feature and localization errors showing the effect of mutual influence filtering.	158
-----	--	-----

List of Figures

1.1	Block diagram illustrating augmentation of an existing localization system.	4
1.2	Illustration of a variety of vehicular applications that can benefit from location and environmental knowledge.	6
1.3	Layout of the remainder of the thesis, showing the material covered in each chapter.	9
2.1	A Dynamic Bayesian Network for the filtering problem.	17
2.2	Example of a hybrid DBN with discrete nodes affecting continuous ones.	22
2.3	Search trees representing different approaches to data association.	30
3.1	The DBN for the general Bayesian filter.	35
3.2	State Prediction and Measurement Update modules for Bayesian filtering.	36
3.3	A DBN for Bayesian filtering with independent measurement vectors.	36
3.4	A modular update scheme.	37
3.5	A hybrid DBN for Bayesian filtering with data association.	38
3.6	Addition of map matching to the system.	39
3.7	System architecture.	40
4.1	A portion of a map showing piecewise-linear road segments.	46
4.2	Converting between the world and road coordinate systems.	47
4.3	A visually-detected intersection road marking and the associated distance estimate.	50
4.4	A model of intersection road marking locations.	52
4.5	A block diagram for Kalman filter localization.	52

4.6	The path driven by the vehicle during collection of the example data set.	54
4.7	Using the Kalman filter to mitigate missing GPS position measurements.	57
4.8	A comparison of vehicle position error in the raw GPS measurements and the output of the Passive Localization module using Kalman filtering.	58
4.9	An illustration of the vehicle pose relative to the bounds of the road segment.	60
4.10	An illustration of vehicle pose estimates matched to the known road segment.	62
4.11	A comparison of vehicle position error when using Passive Localization, and when using Passive Localization and Map Matching with Kalman filtering.	63
4.12	A comparison of vehicle position error when using Passive Localization and Map Matching, and when using Passive Localization, Map Matching and Visual Context with Kalman filtering.	66
4.13	An illustration of longitudinal position error converging to zero using Kalman filtering, while a road marking is continually observed.	68
4.14	Position and variance gain versus feature rate of change.	70
4.15	A comparison of vehicle position error in the raw GPS measurements and the output of the Passive Localization module using particle filtering.	75
4.16	A comparison of vehicle position error when using Passive Localization, and when using Passive Localization and Map Matching with particle filtering.	77
4.17	A comparison of vehicle position error when using Passive Localization and Map Matching, and when using Passive Localization, Map Matching and Visual Context with particle filtering.	78
4.18	An illustration of longitudinal position error converging to zero using particle filtering, while a road marking is continually observed.	79
5.1	A block diagram for Kalman filter localization with Multiple Hypothesis Tracking.	83
5.2	Decision trees for the optimal MHT implementation, and the proposed MHT implementation.	86
5.3	A validation region in used to find possible road segments.	87
5.4	An illustration of road network topology constraints on vehicle orientation.	89

5.5	A validation region is used to identify a detected road marking.	91
5.6	An illustration of data association ambiguity resulting in redundant hypotheses.	94
5.7	The vehicle path for the unbiased data set.	99
5.8	An illustration of MHT localization without visual context.	100
5.9	An illustration of MHT localization with visual context.	101
5.10	Average position error at each time instant using MHT localization.	102
5.11	A plot of the entropy among the hypothesis weights, and the weight assigned to the correct hypothesis, for MHT localization.	103
5.12	An illustration of MHT localization with visual context in biased GPS conditions.	109
5.13	An illustration of MHT localization with visual context in biased GPS conditions using a tuned filter.	110
5.14	Average position error at each time instant using MHT localization in biased GPS conditions.	112
5.15	Identifying road segment candidates using the space occupied by the particle set.	114
5.16	Identifying detected road markings for each road segment candidate.	115
5.17	An illustration of PF localization using Monte Carlo data association without visual context.	119
5.18	An illustration of PF localization using Monte Carlo data association with visual context.	120
5.19	Average position error at each time instant for PF localization using Monte Carlo data association.	121
5.20	A plot of the entropy among the hypotheses in the particle set, and the total weight assigned to the correct hypothesis, for PF localization using Monte Carlo data association.	122
5.21	An illustration of PF localization with visual context in biased GPS conditions.	124
5.22	Average position error at each time instant using PF localization with Monte Carlo data association, in biased GPS conditions.	125
6.1	Hardware used in data acquisition.	130
6.2	Examples of successful road marking detections.	136
6.3	Examples of incorrect road marking detections.	137
6.4	Path driven during collection of Data Set 1.	141

6.5	Localization accuracy for known data association case, using Data Set 1.	142
6.6	Localization accuracy for known data association case while road markings are visible, using Data Set 1.	142
6.7	Path driven during collection of Data Set 2.	144
6.8	Localization accuracy and data association success in unbiased GPS conditions, using Data Set 2.	145
6.9	Path driven during the portion of Data Set 1 exhibiting GPS measurement bias.	146
6.10	Localization accuracy and data association success in biased GPS conditions, using the portion of Data Set 1 exhibiting GPS measurement bias.	147
7.1	Example of arc consistency.	150
7.2	Dynamic Bayesian Network models for the mutual constraint problem, and the localization problem.	151
7.3	New visual context component that achieves mutual influence.	153
7.4	Position error in KF localization with known data associations.	159
7.5	Error in the visual estimate of the distance to an intersection road marking.	160
7.6	Tradeoff in gain applied to position and feature uncertainty.	165

Nomenclature

η	Normalization constant.
γ_t	The orientation of the vehicle on the current road segment.
\hat{f}_t	Visual feature measurements.
\hat{p}_t	Vehicle position measurements.
λ_t	Measurement likelihood $p(z_t z_{1:t-1}, c_{1:t}, u_{1:t}, m)$.
$\overline{bel}(x_t)$	Shorthand for the predicted Bayesian filtering posterior $p(x_t z_{1:t-1}, u_{1:t})$.
ζ_t	Pseudomeasurement vector derived from the vehicle pose and map data.
$bel(x_t)$	Shorthand for the Bayesian filtering posterior $p(x_t z_{1:t}, u_{1:t})$.
f_t	Visual feature state vector at time instant t .
o_t	The set of labels corresponding to visually-detected objects.
$p(z_t x_t)$	Measurement probability.
P_t	Vehicle pose covariance matrix in the world coordinate system.
p_t	Vehicle position in the world coordinate system.
P_t^r	Vehicle pose covariance matrix in the road coordinate system.
p_t^r	Vehicle position in the road coordinate system.
r_t	The identity of the current road segment driven by the vehicle.
t	Time index.
u_t	Control inputs at time index t .
v_t	Vehicle velocity in the world coordinate system.
v_t^r	Vehicle velocity in the road coordinate system.

w_t	Importance weight.
x_t	Vehicle pose in the world coordinate system at time index t .
x_t^r	Vehicle pose in the road coordinate system.
z_t	Measurement vector at time index t .

Chapter 1

Introduction

Our society is becoming increasingly mobile, and with that mobility comes increased traffic, increased congestion and increased uncertainty when traveling in unfamiliar places. Intelligent Transportation Systems (ITS) aim to use intelligent technology to make our roads safer and more efficient. Areas where ITS can make a difference are traffic management systems, traveler information systems, and vehicle navigation systems. In each of these areas, it is required that knowledge of vehicle location be available [21]; localization is the term used to describe the process of estimating the vehicle location based on available sensory information. Localization systems are often building blocks in more complex systems in ITS and in telematics. However, vehicle localization has the potential to become a very active research area in its own right, due to its association with ITS and the increased importance that ITS is expected to have in future transportation endeavors.

Vehicle localization is commonly achieved using satellite positioning technology, of which the Global Positioning System (GPS) is the most established. However, GPS technology is known to produce inaccurate position estimates in certain conditions. The most significant contributors to these inaccuracies are: satellite visibility, where the location cannot be found because satellite signals cannot reach the receiver due to large surrounding structures or dense foliage; and multipath conditions, where signal reflections off of large buildings create large errors in the location estimate [1]. The result is that GPS technology has difficulty providing accurate position estimates in urban environments, where the ITS systems employing this technology would arguably provide the most benefit.

Improving the robustness of vehicle localization to GPS inaccuracies would be a major benefit to ITS, telematics, and other application areas that rely on accurate

vehicle location information. Accurate and robust vehicle localization is achieved by augmenting GPS technology with complementary sources of information, including dead-reckoning sensors, digital maps and machine vision. This thesis investigates the improvement in accuracy and robustness that machine vision can provide to GPS-based vehicle localization.

1.1 Problem Description

This thesis deals with the problem of localization. In a general sense, localization is the problem of tracking the location of an object of interest within a defined coordinate space [67]. Therefore, the concept of localization can be applied to anything we may want to track, such as a vehicle driving in a city, a robot exploring a room, or a person in a public space.

Furthermore, we can speak of localization in global and local contexts. The former involves locating the object within the coordinate space, while the latter is relative to another entity, such as a landmark or the object itself at a previous time instant. When we speak of vehicle localization we are normally considering a global localization restricted to the surface of the Earth, thus a spherical latitude-longitude coordinate system is often used.

Vehicle localization is achieved with the help of satellite positioning systems, of which the most established and widespread is the Global Positioning System (GPS) operated by the United States Department of Defense. A GPS receiver determines its position by knowing the distance from itself to at least four satellites of known position. For this purpose each satellite broadcasts its orbit and a navigation signal. The time it takes this signal to reach the receiver is used to calculate this distance, known as a “pseudorange.” A pseudorange measurement is affected by noise from a number of sources, for example receiver clock bias, atmospheric transmission errors, error in the broadcasted satellite orbit (ephemeris), and multipath (signal reflections off of nearby objects) [54]. These noise sources will induce errors in the calculated receiver positions.

The most persistent errors are multipath and reduced satellite visibility. Others, such as atmospheric and ephemeris errors that can contribute tens of meters of error to the estimated position, can be compensated for by differential means including Differential GPS (DGPS) or the Wide Area Augmentation System (WAAS). Multipath effects are a function of the immediate environment of the receiver, while

DGPS and others are designed to work over a wider range [21]. Using knowledge of the receiver’s environment may thus be a worthwhile way of compensating for multipath, and also maintaining accurate location estimates when there are an insufficient number of satellites visible.

Methods known as “dead reckoning” and “map matching” are commonly used to compensate for satellite visibility and multipath. Dead reckoning uses measurements of the vehicle’s motion from on-board sensors, such as accelerometers and gyroscopes, to extrapolate from the last-known vehicle position, heading and/or speed [1, 52, 64]. The major challenge when using dead reckoning is the problem of accumulated error; small sensor measurement errors accumulate over time into large position errors [21]. This is especially problematic when GPS position estimates are unavailable for a long period of time, as there is no capability to correct the accumulated error.

Map matching involves using a map of the road network to restrict the vehicle position (provided the correct road can be identified). Assuming this is the case, the vehicle location can be corrected to lie on the road, allowing the error in GPS estimates to be determined [65]. This correction will be predominantly in the lateral direction (perpendicular to the road), unless the vehicle trajectory matches with a distinct map feature such as a bend in the road or an intersection. Machine vision may provide the means to correct the vehicle position in the longitudinal (along-road) direction.

Vision provides the ability to sense the surrounding environment of the vehicle to alleviate many of the vehicle localization issues identified above. By positioning the vehicle with respect to objects in the environment, it will be possible to reduce errors resulting from GPS multipath and dead reckoning accumulated errors. Also, by observing objects in the surrounding environment, it should be easier to identify which road the vehicle is on as well as correct the vehicle position in the longitudinal direction. The use of vision represents a link between the vehicle’s location and its surrounding environment that is not present in conventional vehicle localization approaches.

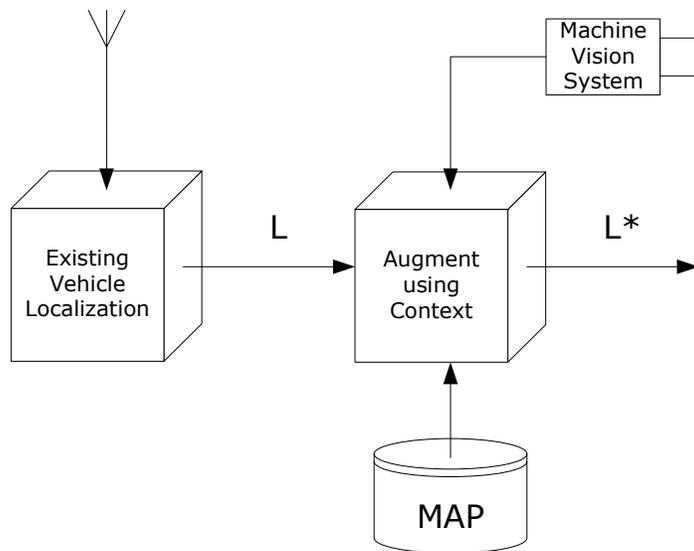


Figure 1.1: Block diagram illustrating augmentation of an existing localization system.

1.2 Incorporating Visual Context into Localization

I am proposing to augment GPS-based localization systems by sensing the surrounding environment of the vehicle for the purpose of improving the accuracy and robustness of vehicle localization. This approach uses visual context to provide constraints on vehicle position that would lead to accurate localization estimates. The approach also provides a connection between the environment surrounding the vehicle and its location estimate, a capability that is not possible using a standalone GPS system. A block diagram in Figure 1.1 illustrates this concept.

More specifically, I am proposing to augment existing localization systems using environmental context, in the form of visually-detected features and a map of the environment [25, 75]. The major advantage of this approach is that feature detection is not affected by any potential GPS issues such as multipath. Therefore the information provided by vision offers a competing estimate of vehicle location that can be used to reduce the effect of GPS inaccuracies. It is also to our advantage to augment existing localization systems as it allows us to complement and enhance their existing capabilities. For instance, it is very common to augment a GPS receiver with an inertial measurement unit (IMU) [13, 31, 37, 52, 64] to maintain continuity of the localization estimates when GPS cannot provide a position, such

as when the vehicle is in a tunnel or surrounded by tall buildings.

The proposed approach is analogous to mobile robot localization systems, which primarily use landmarks and features of the environment to determine the position of the robot in the world [12, 19, 41, 60]. Vehicle localization has been slow to adopt these concepts due to the direct and convenient solution offered by GPS, and the complexity involved with detecting and extracting features in a dynamic environment with moving objects and extreme illumination changes, weather conditions, and appearances.

It is important to stress that I am not trying to solve the vision problem, but rather to provide the framework for augmenting an existing localization system with visual information. As machine vision improves in capability – and it will, based on the amount of attention given to it in recent years – so will the effectiveness of visually-augmented localization.

A question one may ask is: Why not avoid the use of GPS for localization, and instead rely on vision? The answer is that machine vision has its own challenges, which in our case can be alleviated using information obtained from GPS. For example, machine vision cannot easily be used to provide an absolute position; most objects in the world are generic and instances occur in different places. Thus, despite its inaccuracies, GPS can still give an initial estimate of the vehicle position which can help identify more generic landmarks and thereby alleviate ambiguity. It can thus be said that machine vision and GPS have a mutually-beneficial relationship. Both sides of this relationship are explored in this thesis to varying degrees.

In fact, the combination of location technology and environmental sensing technology, including vision, sonar, *et cetera*, would be beneficial in creating more intelligent and autonomous vehicles [9]. Examples include vehicles capable of automatic cruise control, lane departure detection, platooning, autonomous steering and navigation, and obstacle avoidance. Figure 1.2 shows a number of such applications.

1.2.1 Challenges of Incorporating Visual Context into Localization

From the above discussion it is clear that there are significant challenges to incorporating visual context into vehicle localization. I see these challenges falling into

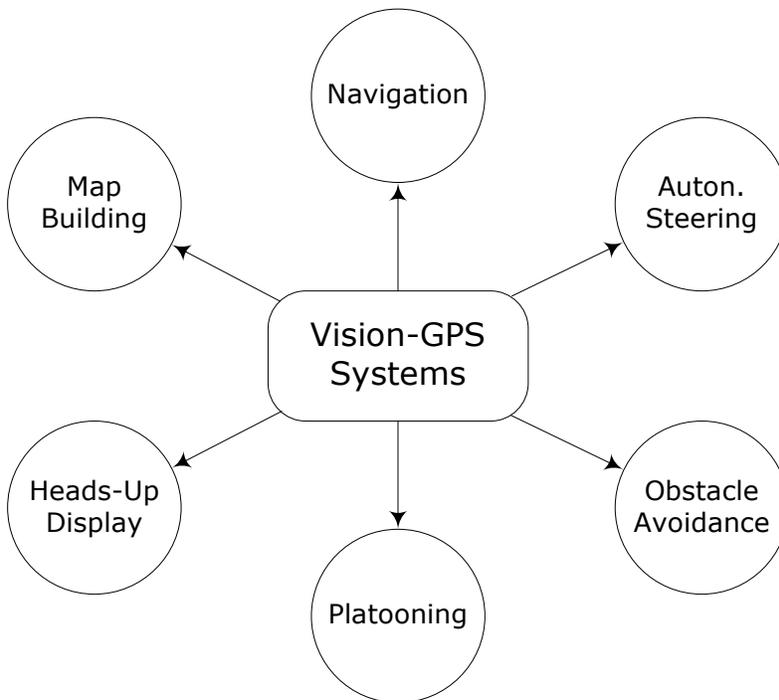


Figure 1.2: Illustration of a variety of vehicular applications that can benefit from location and environmental knowledge.

three broad categories: vision, data association, and data fusion. I discuss these categories in more detail in the following.

Vision

Vision has the potential to be a powerful sensory modality for improving localization, however sensing the environment visually is not a trivial task. Vision must be able to detect features and objects in the surrounding environment, so as to use these objects as landmarks to position the vehicle. To be reliable, this detection must account for variations in object appearance, the presence of moving objects such as other vehicles and pedestrians, occlusions created by a cluttered environment, and varying illumination and weather conditions.

In addition to being able to detect objects, there must be a mechanism for measuring features of that object that are relevant to the vehicle’s position. For example, an estimate of the distance from the vehicle to the detected object, or its angular displacement relative to the vehicle’s heading vector.

Data Association

Data association is the problem of identifying the objects detected by vision using a map of the environment. This can be ambiguous when similar objects occur in close proximity – houses are one example – with the result that more than one object is a plausible identity for the detected object. This has a significant impact on localization, as an incorrect association means that the vehicle is localized with respect to the wrong object, potentially leading to large localization error.

What objects will likely be seen by the vehicle is also dependant on what road the vehicle is driving on. Knowing this information beforehand may reduce the ambiguity of data association in identifying visually-detected objects. The opposite may also be true, namely, knowing which objects have been detected may help in determining which road the vehicle is driving on. This would be a significant help to the map matching problem, where the identity of the road the vehicle is driving on can often be ambiguous due to inaccurate GPS position measurements.

Data Fusion

Data fusion is needed to combine the visual information and the GPS measurement data to achieve a more accurate and robust vehicle localization. This task is not trivial, as these are two heterogeneous data sources. In this regard a map of the environment is necessary to “translate” information from the position space to the visual space, and *vice versa*, so that this information can be fused. The map allows us to determine where the vehicle must be, given that the vehicle can see certain objects. Conversely, it also allows us to determine what the vehicle is likely to be seeing, given the location of the vehicle.

Fusing this information together to improve vehicle localization can be achieved by a number of existing means. Bayesian filters (such as the Kalman filter and particle filters) are popular due to, in part, their recursive nature and ability to integrate measurements from many sources into the vehicle state estimate. Other approaches may include Maximum Likelihood (ML) estimation, and constraint propagation [26].

1.3 Thesis Objectives

Being mindful of the challenges involved with vehicle localization and incorporation of visual context into it, I have set the following objectives for this thesis.

1. Design a system architecture that will incorporate multiple information sources to achieve vehicle localization. Furthermore, to have as one of these information sources a machine vision system to provide visual context to the localization estimate.
2. Propose a method of fusing localization estimates with visual feature data, and verify that the accuracy of vehicle localization is improved as a result. Accuracy will be defined as error relative to the ground truth vehicle position.
3. Examine the effect that visual context has on the map matching problem. Verify that the addition of visual information improves the ability of the system to identify the road being driven by the vehicle.
4. Demonstrate with a real system implementation the improvement to vehicle localization accuracy and map matching that results from the addition of visual context.
5. Determine if a mutually-beneficial relationship between vision and location information can be established. The rest of the objectives are related to using vision to assist localization, but can we simultaneously use location to assist vision?

Where each of these objectives is addressed in the remainder of the thesis is illustrated in Figure 1.3, where the contents of each chapter are displayed in brief.

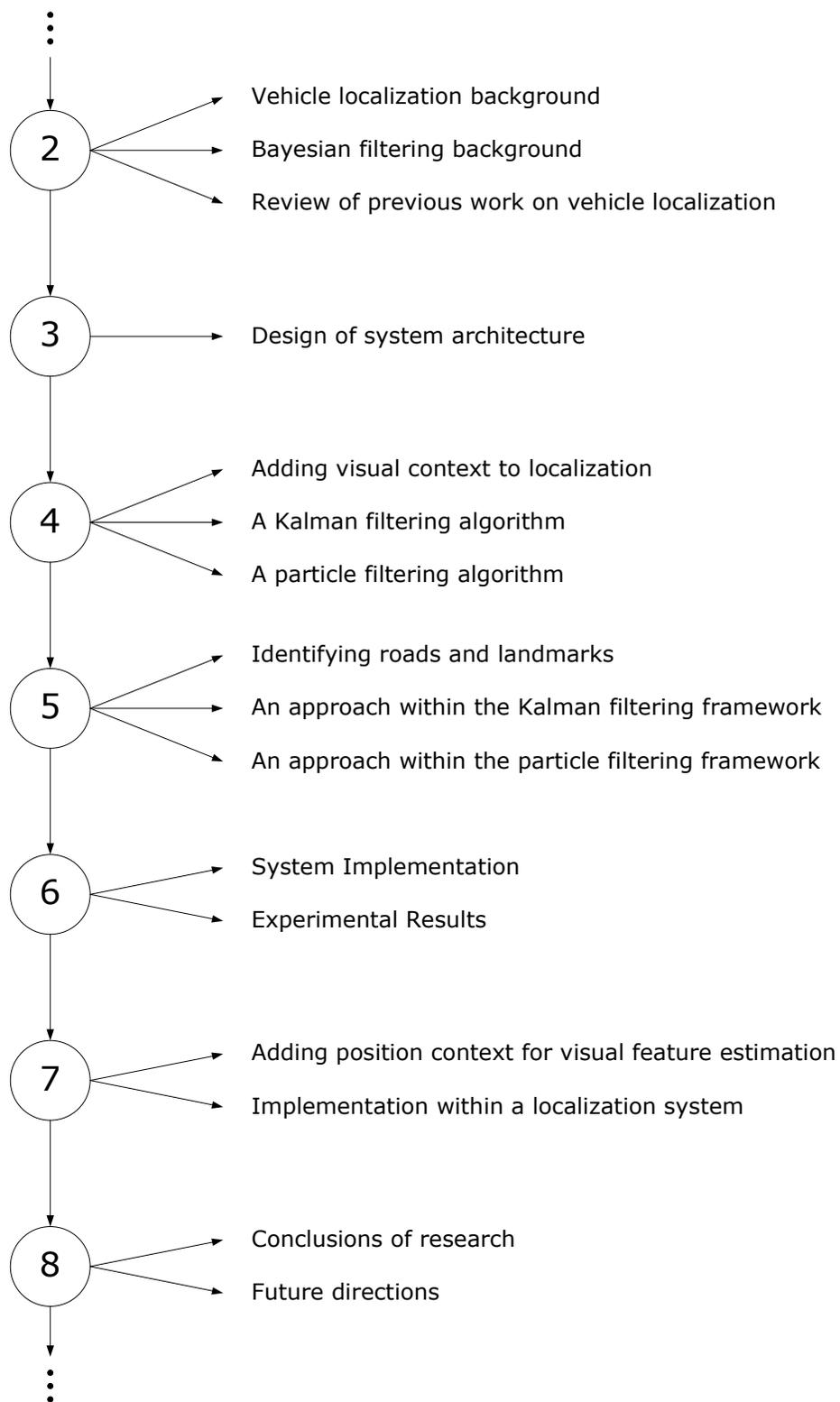


Figure 1.3: Layout of the remainder of the thesis, showing the material covered in each chapter.

Chapter 2

Background and Literature Review

This chapter provides to the reader background information on localization and Bayesian filtering to establish a context for the remainder of the thesis. The localization background will discuss the problem within the vehicular domain. Much of the work on vehicle localization involves compensating for the faults inherent in GPS by using additional sensory information. This generally requires a combination of the following:

- dead reckoning to compensate for GPS being temporarily unavailable or producing unreliable estimates,
- map matching to adjust the estimated vehicle position based on the location of roads, and
- external sensors such as vision to position the vehicle with respect to its environment.

Bayesian filtering is the most popular approach taken for localization. It is also the approach taken in this thesis. As a result, I provide background on Bayesian filtering to enable the reader to better understand the concepts introduced later in the thesis. I also discuss Bayesian filtering as a method of inference in Dynamic Bayesian Networks (DBNs), graphical tools which combine probability theory, graph theory and the notion of time dependency, and are well suited to modeling the dependency between variables.

Finally, I then give a review of past accomplishments in vehicle localization, to establish to the reader where my work fits with respect to the rest of the field.

2.1 Vehicle Localization Background

Vehicle localization systems have been centered primarily around GPS receivers to provide direct measurements of the vehicle position on the Earth. The various shortcomings of this technology, which will be explained, have resulted in many attempts to augment these measurements in a number of ways, including additional infrastructure as well as within the vehicle itself.

The Global Positioning System

The Global Positioning System (GPS) is a satellite-based positioning system operated by the United States Department of Defense (DoD), made available for use by the general public. The full system is composed of the Space Segment, User Segment and Control Segment [21]. The Space Segment consists of a minimum of 24 satellites orbiting the Earth and the signals they transmit. The Control Segment consists of supervision and maintenance required for day-to-day operation of the Space Segment. The User Segment consists of receivers and the many applications of this technology.

For a receiver to calculate its position, it is necessary to know the distance from the receiver to at least four satellites of known position. Each satellite broadcasts its own orbit and a pseudorandom navigation signal for this purpose. The delay between the received signal and an internal copy generated by the receiver provides the transmission time for the signal [70]. Multiplying by the speed of light c provides the range from the receiver to the satellite, whose position at the time of transmission can be computed knowing its orbit. This range estimate ρ is known as a “pseudorange” because it is derived from the time delay and is not a direct range estimate. A pseudorange measurement is modeled as the true receiver-satellite range r plus additive noise from a number of sources, for example receiver clock bias, atmospheric transmission errors, error in the broadcasted satellite orbit (ephemeris), and multipath [54].

The receiver clock bias δ_t affects all pseudorange measurements from a given receiver, and is calculated from the following system of non-linear equations (2.1) along with the estimate of receiver position $\{x, y, z\}$ in Earth-Centered Earth-Fixed coordinates. As there are four unknowns to be found, the pseudorange ρ_i and

position $\{X_i, Y_i, Z_i\}$ of at least four satellites must be known [40].

$$\rho_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} - c\delta_t, \quad i \in [1, N], \quad N \geq 4 \quad (2.1)$$

Atmospheric and ephemeris errors will similarly affect all receivers within the same area, since they will have the same satellites in view, receive the same broadcast ephemeris data and be subject to the same atmospheric refraction errors [21]. These errors together result in computed position errors on the order of tens of meters [54].

A number of augmentation systems exist to reduce these errors for all receivers in a common area. Differential GPS (DGPS) is perhaps the most well known. At a base station with a known position, errors in the received pseudoranges can be computed and broadcast so that receivers can correct their own estimates. Position accuracy can be achieved at the submeter level by this method [21].

A related approach is the Wide Area Augmentation System (WAAS). This is a Satellite-Based Augmentation System (SBAS), using satellites rather than base stations to broadcast correction data to receivers within a wider area than a single DGPS base station [27]. Many receivers are capable of using the WAAS service, which improves horizontal position accuracy to approximately 2 meters.

A more persistent error in GPS positioning is due to multipath effects, which can be stated simply as the bias introduced into the pseudorange by satellite signals reflecting off of objects in the immediate environment of the receiver. This makes it appear that the satellite is further away due to the increased time for the signal to reach the receiver. This problem can be alleviated somewhat by better antenna and receiver design [21, 54]; however, many modern receivers still have significant problems computing reliable solutions in “urban canyons” or under dense foliage [1]. These position errors cannot be totally corrected using differential techniques since the multipath effect is heavily dependent on the immediate environment of the receiver, while DGPS and others are designed to work over a range of 100 km from the base stations [21], and WAAS over an even larger area [27]. Using knowledge of the receiver’s environment may thus be a worthwhile way of compensating for multipath effects.

Differential methods are viable means for improving the position estimate that GPS provides, however they cannot mitigate the loss of satellite signals. In urban environments it is common that only a small set of satellite signals are received in

urban canyons. Therefore, a GPS receiver by itself will not be sufficient to provide continuous and accurate localization in a vehicular application. For this reason other complementary sources of location information must be considered, such as dead reckoning, map matching and machine vision.

Dead Reckoning

Dead reckoning is commonly used to supplement GPS, to account for position inaccuracies due to multipath and to mitigate the loss of GPS position fixes when satellite visibility is low [1, 13, 14, 37, 52, 58, 64]. Both of these phenomena are common in urban environments, where satellite signals are often obstructed near tall buildings or in tunnels.

Dead reckoning uses measurements of the vehicle's motion to extrapolate from the last-known position, heading and/or speed. Sensors commonly used for this task are:

- odometers for measuring distance traveled by the vehicle;
- accelerometers for measuring forward and lateral acceleration;
- compasses for measuring vehicle heading, and;
- gyroscopes for measuring the turning rate of the vehicle.

Accelerometers and gyros are commonly found together in what is known as an Inertial Measurement Unit (IMU). An Inertial Navigation System (INS) uses measurements from an IMU to estimate the current vehicle state (position, heading and/or speed) based on the previous state estimate [13]. Wheel speed sensors can also be used to determine the change in vehicle state, with the advantage that they are present on most vehicles in the form of anti-lock braking systems (ABS) [10]. Estimates of vehicle position from dead-reckoning and GPS are often fused using a Kalman filter or extended Kalman filter.

The major challenge when using dead reckoning is dealing with the problem of accumulated error. Small sensor measurement errors will accumulate over time into large position errors the longer that these measurements are used to compute the vehicle position [21]. For this reason, they are primarily used when GPS is not available or in the presence of multipath. GPS position measurements provide a

way of calibrating the dead-reckoning sensors online, by comparing the measured GPS position with the position predicted by dead-reckoning. This approach is one that is used quite often to build high-integrity localization systems [52, 58, 64].

One drawback to the above calibration method would be cost. While it is possible to achieve accurate localization using dead reckoning during long periods of time when GPS signals are unavailable, the cost of dead reckoning sensors with suitably low measurement drift errors would be prohibitively high. At the present time, inexpensive sensors are not capable of achieving this level of accuracy.

Map Matching

Assuming that the vehicle is on the road, a map of the road network can be used to restrict the vehicle position, provided the correct road can be identified. The process of identifying the road driven by the vehicle and adjusting the vehicle position to be consistent with the location of that road is known as map matching [72].

The road network is often contained in a vector database [65]. Roads are represented by piecewise-linear segments modeling the location of the road centerline between two endpoints defined by intersections with other roads or by dead-ends. Each entry in the database represents one such road “segment,” and thus each road segment is uniquely identified by its index in the database.

The identity of the current road segment being driven by the vehicle is often not known *a priori* except in special circumstances or applications. Rather this must be inferred based on the measurement data, and is an example of the data association problem. This problem was introduced in Section 1.2.1 in the context of identifying objects detected using vision. The concept is the same for map matching, instead we are identifying a road instead of a landmark. In both cases we are inferring the identity of something that we cannot measure directly based on the available measurement data.

Identifying the road segment being driven by the vehicle is not a trivial task. It is often the case that multiple road segments are plausible candidates. This can arise because the location estimate is inaccurate, and thus we must consider a wide area containing many roads, or the roads are placed very closely together as is the case in many old European cities. In addition we must contend with GPS measurement inaccuracies such as multipath, which may introduce a significant bias into the estimated vehicle location, thus complicating the map matching problem.

One benefit of map matching is that, provided the road segment can be identified, the vehicle position can be restricted to lie on the road segment. This provides a way of correcting for localization errors caused by multipath or other GPS issues [51, 66]. There is however a limit to the degree of correction that can be achieved. Since roads are modeled by their centerlines, they may not be indicative of the true vehicle path and may introduce a bias on an order as large as tens of meters into the vehicle localization estimate depending on the road width.

Vision in Localization

Vision provides the capability to observe the environment surrounding the vehicle, and the information gained can be used to localize the vehicle with respect to its environment. Vision can therefore be used to create a more robust localization system that improves localization accuracy over GPS-based systems [16] or maintains localization integrity in poor GPS environments [25].

Vision-based localization systems are defined by the visual information they are capable of extracting, and by the information contained in their maps of the environment. For example, one can use vision to measure the position of the vehicle within its current lane [16, 39]. Visually measuring the location of lane boundaries is rather straightforward as they are normally painted a color such as white or yellow that highly contrasts with the dark road surface [15]. Provided the map contains detailed information such as the location of lane boundaries, it would be possible to determine which lane the vehicle is driving in and to precisely locate the vehicle within that lane.

Vision can also be used to localize the vehicle with respect to objects surrounding the vehicle. One simple approach is to use straight-line features [25, 35] which are easily detected in an image using an edge detector and to associate them with objects in the environment, such as buildings, trees or lamp posts, the location of which we know from our map. The estimated position of these objects relative to the vehicle enables us to maintain localization even in poor GPS environments.

Vision may also be used to determine the absolute location of the vehicle when other absolute localization estimates such as those from GPS are not available [33, 75]. This would require using the appearance of the surrounding environment to uniquely identify the vehicle's location. While this may be straightforward in circumstances when unique landmarks are visible (for example, the CN Tower in

Toronto) in most circumstances it would require an exhaustive search of a map database to find compatible scenery.

Regardless of the approaches that can be taken, there are some general guidelines one notices about the use of vision for localization. First, the capability to extract features from images of the surrounding environment is required, and to estimate the position of these features relative to the vehicle. These features should be static, like landmarks or road markings, in order to be used to position the vehicle. Second, the location of these objects must be known, meaning that a map of the environment is required. Third, it must be possible to establish correspondence between the detected objects and those in the map (the problem of data association). Finally, the capability is needed to use the above information to adjust (in the case of relative localization) or determine (in the case of absolute localization) the location of the vehicle in the absolute sense. The majority of methods employ Bayesian filters (specifically Kalman filters and particle filters) for this purpose.

2.2 Bayesian Filtering Background

Bayesian filtering is the term given to a family of probabilistic filtering methods that aim to infer the values of a set of variables of interest, collectively referred to as the state of the system, from a sequence of related observations. In particular, we wish to estimate the posterior probability distribution of the state at time instant t , denoted x_t , conditioned on all measurements made up to t , denoted $z_{1:t}$, that is, $p(x_t|z_{1:t})$. Often the mode of this distribution is of interest, as this indicates the most probable value of the state vector x_t based on the accumulated evidence.

2.2.1 Dynamic Bayesian Networks

A graphical tool known as a Dynamic Bayesian Network (DBN) can be used to model the dependencies between variables in the Bayesian filtering problem. Each node in a DBN represents a random variable, and directed arcs between nodes represent causal or temporal relationships between them. For this reason DBNs are useful for describing tracking problems, where the state of the system changes over time and at each time step we make observations of variables related to the state [48]. A simple DBN is shown in Figure 2.1. The convention used in this diagram is to denote evidence nodes (those whose value has been observed) with

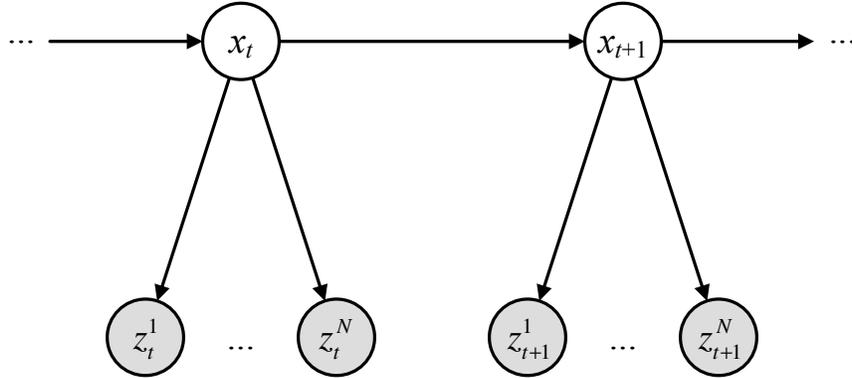


Figure 2.1: A Dynamic Bayesian Network for the filtering problem. The measurement variables $z_t = \{z_t^1 \dots z_t^N\}$ are dependent on the value of the state x_t , which depends on the previous state x_{t-1} . Thus, temporal and causal relationships are captured.

shading and hypothesis nodes (those whose value we wish to infer) without shading and a solid outline. Hidden nodes (those whose value is neither observed or of interest) will be shown with a dashed outline. Furthermore, nodes representing variables with continuous probability distributions are shown as circles, while those with discrete distributions are shown as squares. This convention is maintained throughout the thesis.

Each node in a Bayesian network (BN) is independent of every other node if the values of its parent nodes are known. Thus, by knowing the probability distribution of each node in the network conditioned on its parents, it is possible to describe the joint distribution of all variables in the network [57]. The joint distribution allows us to determine the posterior probability distribution of the subset of variables that we are interested in (the state variables x_t) given observations of a separate subset of variables (the measurement or “evidence” variables $z_{1:t}$). This is the process of inference [63], of which filtering is one type.

In more complex networks, a series of elementary connections (serial, diverging and converging) define the paths by which hypothesis nodes are influenced by evidence nodes. Certain pathways can be “blocked” due to the nature of the connection and the location of evidence nodes. This represents an advantage of BNs as it is known directly from the network structure whether a given node will influence another, regardless of the probability distributions assigned to the various nodes [49].

2.2.2 Bayesian Filtering Methods

Estimating the filtering posterior can be performed recursively with the result that only the current state estimate and set of measurements need to be stored, thereby keeping memory consumption at a minimum. This is a consequence of the Markov assumption – that future states of the system are independent of past states if the present state is known – that is made implicitly in Figure 2.1 by the arc directed from the current state x_t to the next state x_{t+1} .

The recursive update can be developed as follows. The filtering posterior can be factored according to Bayes' rule (2.2). The term $p(x_t|z_{1:t-1})$ is a prediction of the state x_t based on measurements $z_{1:t-1}$. This prediction is updated using the probability of the current measurement $p(z_t|x_t)$ to generate the posterior. The predicted state $p(x_t|z_{1:t-1})$ is found by multiplying the previous estimate $p(x_{t-1}|z_{1:t-1})$ by a state transition probability $p(x_t|x_{t-1})$ and marginalizing x_{t-1} . This is known as the Chapman-Kolmogorov equation [4] and is either a summation, if discrete, or an integration, if continuous (2.3).

$$p(x_t|z_{1:t}) = \frac{p(z_t|x_t)p(x_t|z_{1:t-1})}{p(z_t|z_{1:t-1})} \quad (2.2)$$

$$= \begin{cases} \frac{p(z_t|x_t) \sum_{x_{t-1}} p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})}{p(z_t|z_{1:t-1})} & \text{if discrete} \\ \frac{p(z_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|z_{1:t-1})dx_{t-1}}{p(z_t|z_{1:t-1})} & \text{if continuous} \end{cases} \quad (2.3)$$

Two assumptions of conditional independence are made in arriving at the above expressions. First, z_t is assumed conditionally independent of all previous $z_{1:t-1}$ given the state x_t . This assumption is in fact shown in Figure 2.1, where nodes $z_{t+1}^{1:N}$ are independent of $z_t^{1:N}$ when x_{t+1} is known. Second, the current state x_t is assumed independent of all previous states if its immediate predecessor x_{t-1} is known. This is an example of the Markov property and is also encapsulated by the network structure in Figure 2.1.

Bayesian filtering is used for a multitude of applications, and in fact many well known methods such as Hidden Markov Models (HMM) and Kalman filtering (KF) are special instances of Bayesian filtering. These problems can all be represented

using the common language of DBNs [63]. The following discussion introduces these methods, and groups them according to the type of DBN they correspond to: one with discrete nodes, continuous nodes, or a hybrid of the two.

Discrete Networks

Discrete DBNs are useful for representing classification problems such as speech recognition, and for continuous problems such as localization when the state space is small. Hidden Markov Models (HMM) are an example of a discrete DBN. The purpose of an HMM is to evaluate the probability of a sequence of unobservable (hidden) states in a Markov chain, based on an observed sequence of related events [53]. Discrete environmental representations have been used for localization by segmenting the environment into a grid of cells [11]. This approach is successful in handling arbitrary distributions, however it can be very expensive computationally if the state space is large and the cell size very small [24].

Continuous Networks

Many well-known methods of target tracking are in fact inference schemes for continuous DBNs. Included in this category are Kalman filters, extended Kalman filters, unscented Kalman filters and particle filters.

A Kalman filter (KF) provides the optimal filtering result in the least-squares sense for systems that are linear and Gaussian [36]. This means that the filtering posterior has a Gaussian distribution parameterized by its mean vector $E[x_t|z_{1:t}]$ and covariance matrix $Var[x_t|z_{1:t}]$ (2.4).

$$p(x_t|z_{1:t}) = \mathcal{N}(x_t; E[x_t|z_{1:t}], Var[x_t|z_{1:t}]) \quad (2.4)$$

Furthermore, the next state x_{t+1} and current measurement z_t are linear functions of the current state x_t (2.5)-(2.6), where ω_t and ν_t are zero-mean Gaussian noise terms. The equations for KF prediction and update can be found in many papers and textbooks ([67] provides an extensive derivation), and so are not reproduced here.

$$x_t = A_t x_{t-1} + \omega_t \quad (2.5)$$

$$z_t = H_t x_t + \nu_t \quad (2.6)$$

When these assumptions are violated by a system with non-linear dynamics or a non-Gaussian posterior, the KF framework can still be used however with suboptimal results. A non-Gaussian posterior can be assumed Gaussian for the purpose of filtering (2.7). However, this may not adequately represent the true posterior, particularly if it is multi-modal. A good illustration of approximating a multi-modal distribution with its mean and variance is given in [57], page 558.

$$p(x_t|z_{1:t}) \approx \mathcal{N}(x_t; E[x_t|z_{1:t}], \text{Var}[x_t|z_{1:t}]) \quad (2.7)$$

Extended Kalman Filtering (EKF) can be used when the linearity assumption is violated. Non-linear models are used to predict the values of the state x_t (2.8) and the measurement z_t (2.9). To perform the KF update, however, linear models A_t and H_t are required as in (2.5)-(2.6). Linear approximations to the non-linear models are therefore found as the Jacobian of a_t and h_t with respect to the state x_t . For this reason, if the models are locally-linear near x_t then EKF may approach the optimal filtering result.

$$x_t = a_t(x_{t-1}) + \omega_t \quad (2.8)$$

$$z_t = h_t(x_t) + \nu_t \quad (2.9)$$

To address the shortcomings of EKF, Unscented Kalman Filtering (UKF) was proposed to better handle non-linearities [34]. UKF uses the Unscented Transform, a deterministic sampling method that enables calculation of the statistics of a random variable that undergoes a non-linear transformation [71]. Whereas EKF will capture the posterior mean and variance to 1st order (Taylor series expansion), UKF will capture it to 3rd order using the same level of complexity [71]. This is still dependent on the Gaussian assumption, however, and as a result will have difficulty with multimodal distributions.

Particle filtering (PF) is a different inference method than Kalman filtering, both in terms of how it represents probabilities and how it updates estimates of them. The KF uses a parameterization of the posterior, taking advantage of the fact that a Gaussian distribution is completely specified by its mean vector and covariance matrix. PF represents the posterior non-parametrically as a sum of weighted samples (or particles) (2.10). Each sample or particle can be considered to be a hypothesis of the value of the state x_t . This non-parametric representation enables arbitrary distributions to be handled; dense clustering of particles in one

area of the state space indicates a peak in the posterior distribution.

$$p(x_t|z_{1:t}) \approx \sum_{i=1}^N w_t^i \delta(x_t - x_t^i) \quad (2.10)$$

At each iteration of the filter, the weights and samples are updated to reflect changes to the posterior due to the state transition from x_{t-1} to x_t and any new measurements z_t . Unlike the KF, the state transition and measurement models can be arbitrarily non-linear.

One inherent problem of PF is that after a few iterations of the filter, only a small number of particles have significant weight. This implies that computational resources are wasted updated particles that do not contribute to the posterior. To improve the diversity among particles, resampling is used whenever the set degenerates. A new set of N particles is randomly chosen (with replacement) from the existing set, with the probability of choosing particle i equal to its weight w_t^i . A variety of ways to weight, sample and resample particles means there is no such thing as a “standard” particle filter, although the Sequential Importance Sampling (SIS) and Sampling Importance Resampling (SIR) filters [4] are perhaps the most widely understood. Good overviews of particle filtering are given in [4] and [67].

Hybrid Networks

Hybrid (or mixed) networks contain both continuous and discrete nodes, making the process of inference less straightforward. Examples of hybrid Bayesian networks arise in many practical applications, such as target tracking or fault diagnosis [42], where we are interested in the value of a continuous state that depends on a discrete parameter. A DBN with a mixture of continuous and discrete nodes is shown in Figure 2.2. In target tracking, there can be discrete variables for the identity of the target or the maneuver it is performing, and continuous variables for the position, speed or heading of the target. In fault diagnosis, there can be a Boolean variable representing fault/nofault in a monitoring system, for example detecting pipe burst events from the flow rates through a series of connected water tanks [2, 42]. Problems involving data association fall under this category since the expected value of a sensor measurement will depend on the identity of the detected object [47].

The posterior of the network in Figure 2.2 is $p(x_t, c_t|z_{1:t})$, which is not a conventional probability mass or density, being a mixture of discrete and continuous

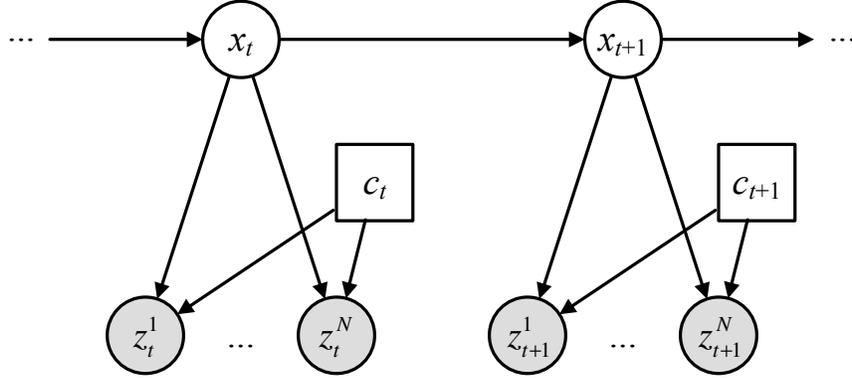


Figure 2.2: Example of a hybrid DBN. Some continuous nodes in the network depend directly on the value of discrete nodes.

variables. In order to create meaningful probabilities, the discrete variables can be marginalized out of this posterior to create the conventional continuous posterior (2.11). This is now a summation of a set of posteriors conditioned on the individual values of c_t . The discrete posterior $p(c_t|z_{1:t})$ is present as well providing the weight to each conditional continuous posterior.

$$p(x_t|z_{1:t}) = \sum_i p(x_t, c_t^i|z_{1:t}) = \sum_i p(x_t|z_{1:t}, c_t^i)p(c_t^i|z_{1:t}) \quad (2.11)$$

This approach is used in multiple-hypothesis tracking (MHT), which is a common method for handling unknown data associations. Each individual posterior is tracked using its own Kalman filter. It is however intractable in practice [67] because the number of discrete hypotheses grows exponentially over time. Assuming there are N values in the domain of c_t , then each of the N hypotheses at time t creates N new hypotheses at $t + 1$ when all possible combinations are considered [47]. Therefore, it is common to prune hypotheses with low probabilities at each iteration to maintain tractability of the filter.

The vehicle localization problem requires a hybrid DBN representation due to the presence of discrete parameters that affect the localization result, such as the identity of the road being driven and the object observed by vision. Data association methods are discussed in more detail in Section 2.3.4.

2.3 Vehicle Localization Literature Review

In this section I will highlight some of the recent work in vehicle localization and related areas. I have organized this work into the following categories: passive localization, referring to the use of GPS, dead reckoning, and other sources that do not use environmental context; map matching, where localization is augmented knowing the road being driven; vision-based localization, involving sensing of the environment surrounding the vehicle; and data association, which refers to the identification of detected objects, and is relevant for the map matching and vision-based localization problems.

2.3.1 Passive Localization

“Passive Localization” is a term I am using to refer to those methods of vehicle localization that do not involve the use of environmental context, in the form of maps or sensing of the environment. This includes the more traditional sources of information such as GPS (and other satellite positioning systems such as Russia’s GLONASS and the European Union’s Galileo), INS and other dead reckoning sensors (wheel-speed sensors, compasses), and emerging technology such as the Global System for Mobile Communications (GSM) cellular telephony standard.

There are numerous examples of approaches that combine satellite positioning with dead reckoning to create a “high-integrity” localization system. These techniques are complementary: dead reckoning can provide relative changes in vehicle location when GPS exhibits an occasional fault or lost signal, while GPS provides reference positions to calibrate the dead reckoning sensor measurements against their inherent bias.

The vast majority of such approaches use KF or EKF to perform localization, although what these filters are used to track varies among the approaches. For example, in [64], a Kalman filter is used to track the bias in positions calculated using an IMU, using measurements from a DGPS receiver. This provides a method of calibrating the IMU online, so that when GPS is not available or exhibits a fault (such as multipath) the IMU can be used to maintain localization integrity for a short period. This approach has been termed “indirect” localization, because the filter does not have the vehicle position among its state variables [52]. So-called “direct” methods, where the vehicle position is present in the state (among other variables, including IMU bias terms) are explored in [13, 52, 58].

To maintain robustness of the localization solution to GPS signal multipath, it is common to monitor the innovation of the Kalman filter for any sudden unexpected changes in the GPS position measurements [13, 58, 64]. How the so-called “erroneous” measurement is used varies among the approaches. For example, in [64] the GPS measurement is discarded and localization continues with the IMU alone. Conversely, in [13, 58] the GPS measurement is allowed to contribute to the localization, however it is given a lower weight relative to the IMU. One advantage I can see in the method of [13] is that faults in the IMU are detected as well, accounting for the case where both sources are giving faulty information.

One deficiency in the studies mentioned above is that, although they discuss detection of multipath events and use dead reckoning to compensate for them, the ability to do so in an environment such as an urban canyon where these events are common is not demonstrated. Testing is instead performed in an environment with brief multipath occurrences [64] or by artificially degrading GPS position measurements with white noise [13, 58]. A possible reason for this deficiency is that dead reckoning would have to operate continuously without calibration in the presence of persistent multipath, thus leading to unbounded growth in the uncertainty of vehicle localization.

One possibility for an alternative localization source in urban environments is using GSM signals. At present, however, the accuracy of GSM-based localization is on average approximately 100m [17]. Therefore, while it may provide a rough estimate of location, currently it is not sufficiently accurate for, say, turn-by-turn vehicle navigation.

2.3.2 Map Matching

Map matching, which has been defined previously as the process of identifying the road driven by the vehicle and adjusting the vehicle location estimated to be consistent with the location of that road [72], is an integral part of a localization system. Without this capability, it is not possible to track the route taken by a vehicle or to navigate a vehicle to a destination.

This process is trivial when there is only one road near the estimated vehicle position. In this situation, or if the road is known *a priori*, the method of Scott [59] can be used to make the position estimate consistent with the road location. It becomes much more complicated when multiple roads lie within the uncertainty

region of the vehicle position estimate. For example, when the vehicle is at an intersection of two or more roads, or when two roads run parallel to each other and the estimated position lies between them [72]. For this reason map matching approaches tend not to make decisions based on a single position estimate, or to consider only one road possibility.

Many map matching approaches use Bayesian filtering in considering multiple road hypotheses. The common element among these approaches is that they maintain multiple possibilities for the road until such time that one becomes significantly more likely than the rest. Pyo [51] uses a Multiple Hypothesis Tracking (MHT) approach with a number of threshold tests to decide among competing road hypotheses. Measurements are obtained from a combined GPS-dead reckoning system providing position and heading information. Gustafsson [28] and Chausse [16] are examples of map matching using particle filtering methods. The former initially spreads the particles over a large number of possible roads surrounding an initial position provided by a GPS receiver. Over time, the particles converge to the correct road while position estimates are updated by dead-reckoning. Fast convergence to the correct position is reported, however the restricted search space raises concerns over the ability to scale up this method to a more general scenario. The latter particle filtering approach [16] weights each particle based on 1) its proximity to the position estimate from GPS, and 2) its proximity to a road.

Taylor [66] matches the trajectory of a series of raw GPS points to the shape of the nearby roads. Roads which differ significantly in their shape from the GPS trajectory can be filtered out, while roads with low error are candidates for the vehicle location. This is determined by a least-squares criterion. In addition, once the road has been identified, the corrected position is used to determine the error in the GPS estimates, thereby providing a calibration for future measurements.

While map matching can improve the estimation of vehicle position, it predominantly does so in the lateral direction (perpendicular to the road). Without additional information about the vehicle's longitudinal (along-road) position, the only adjustment that can be made is in the lateral direction. Distinct features such as a change in the trajectory of the vehicle or landmarks observed using vision can provide this information.

2.3.3 Vision-based Localization

Uses of vision for localization in a vehicular environment and in a robotics environment is discussed here. Also, the literature provides us with some general requirements for localization systems that rely on sensing their environment, by vision or other means.

Vehicle Localization

There are numerous examples of vision being used within the context of vehicle localization. Along the lines of map matching, vision has been used to measure the location and heading of the vehicle relative to the boundaries of its current lane [16, 39]. In [39] the position and heading of the vehicle relative to the current lane are converted to world coordinates using an accurate map of the road. These estimates are combined with position and heading estimates from a GPS-dead reckoning localization system using a Kalman filter. Chausse [16] extends this approach to multi-lane roads using a particle filter in place of the Kalman filter. Reported results indicate that this approach reduces uncertainty in lateral position (perpendicular to direction of the road) while uncertainty in longitudinal position (parallel to direction of the road) is less affected.

A similar investigation [31] uses visual estimates of road shape with GPS and a gyroscope to determine the pose of the camera with respect to the current lane in a digital map. The camera pose is used in a heads-up display to help the driver navigate without distraction in an urban environment.

Georgiev and Allen [25] and Kais *et al* [35] use vision as a complement to GPS and dead reckoning for estimating the pose of a robot in an urban environment where satellite visibility may be limited. Both use straight line features that are easily detectable and abundant in urban environments, such as building edges, doors windows, trees, lamp posts, traffic signs and lane boundaries. In [35], the features to be used for localization are mapped within a Geographic Information System (GIS) database. The locations of GIS features are transformed to the camera coordinate frame to define search regions for these features in the acquired image, driving the operation of the vision system. In [25], the appearances of building facades are modeled by their straight line features. The transformation required to align the detected features with the model gives an indication of the pose of the robot with respect to the building. Both of these approaches [25, 35] are used to control the

accumulation of localization uncertainty when GPS estimates are unavailable due to poor satellite visibility.

Similar investigations have been made into using only vision for performing localization, without the use of GPS. For example, [75] uses images of buildings to determine location. This is accomplished by identifying the most similar image from a database of building images and determining the pose of the camera with respect to the identified building. Johns [33] uses a slightly different approach by extracting a contour of the skyline, and matching this contour against a database of contours.

Robot Localization

Much can be learned from the robotics field, where localization depends strongly on sensing the environment. Due to environmental constraints, such as operating indoors, direct measurement of the state from GPS is often not available. Rather, sensors are used to gather information about the environment, which is used to update the localization state. This is a more general approach that does not rely on the assumption of having a direct state measurement, and will also be relevant regardless of the size of the environment.

Detecting objects in the environment for robot localization is often achieved using laser scanners or sonar. One possible reason for this is that these sources can give accurate localization with simpler maps and less-demanding data association [50]. In an outdoor environment, the versatility of vision would be extremely useful for distinguishing static objects from dynamic ones, and also dealing with clutter. However, the use of vision for localization has been limited to simple features rather than high-level objects.

For example, the localization systems described in [5, 19, 60] all use point features that are invariant to the changing position and orientation of the robot. This implies that these features can be reliably detected as the robot traverses its environment. These types of features are found by low-level feature detectors such as [62, 43].

Perez [50] and Kosaka [38] used vertical line features belonging to doors and walls in hallway images to localize the robot using Kalman filtering. The latter used a model of the feature locations and the uncertainty in the robot position to search for these features. This was accomplished by projecting the robot position

uncertainty into the camera coordinate system and establishing a search region for each feature. It is interesting how this same approach was used for vehicle localization by Kais [35].

There does not appear to be an emphasis on identifying high-level features and using them for localization, for example, tables, chairs, doors, *et cetera*. One explanation may be the difficulty that would be associated with recognizing such objects in a dynamic environment, where occlusions would be common. It may be promising to build upon work using contextual priors for object recognition [68]. The idea is to use position of the robot to constrain the probability of certain object classes. This would be especially promising with a Geographic Information Systems (GIS) database describing the type of objects near a location, such as buildings or street signs, that can be used to tailor the processing needs of the vision system.

General Requirements for Vision-based Localization

Regardless of the type of sources used, some general lessons can be learned from the literature when sensing the environment for the purpose of vehicle localization. First, the environment must contain features that can be detected and whose location is known from a map of the environment. Without these, the robot cannot update its location estimate and its uncertainty will grow without bound [41].

Second, features must be distinct and unambiguous [44, 67] to make them easy to detect, and difficult to confuse with other features. The latter is extremely important, because misidentified features can lead to large localization errors from which it may not be possible to recover.

Third, an environment that is densely populated with such features would provide the best localization result [67]. In a sparsely-populated environment the state is updated only periodically, again leading to uncertainty growth.

The prospects for using vision in vehicle localization are promising when considered in light of these points. Urban environments specifically contain many static features useful for localization, such as street intersections, buildings, signs, telephone poles, among others. A significant challenge will be identifying these features in the presence of moving objects such as people and other vehicles, and also under dynamic lighting and weather conditions. In terms of maps containing the location of features, Geographic Information Systems (GIS) databases are promising as they provide information on the location of roads and buildings, how land is used, *et*

cetera [30]. The main issues may be the public availability and cost of using this data, as well as possible inconsistencies between databases produced by various municipalities [61].

2.3.4 Data Association

Data association involves the association of feature measurements with the objects being measured, that is, identifying the object of a particular measurement. Within the context of vehicle localization, this could be the identification of a detected building or street intersection. The data association problem is considered an extremely difficult problem, especially in dynamic environments [73].

Data association is a necessary evil as it is required to determine which object in the map a measurement refers to in order to use that measurement to update the state. The consequences for incorrect associations can be quite severe. Failing to associate a measurement with an object means that that measurement cannot be used to update the state estimate. Furthermore, associating a measurement with an incorrect object could bias the vehicle location estimate; essentially the vehicle thinks it is seeing something it is not, and there thinks it is located somewhere it is not.

The problem of localization with data association can be modeled using a DBN with continuous and discrete nodes, as depicted in Figure 2.2. Continuous nodes represent the vehicle state and measurements such as from GPS and vision. Discrete nodes represent the identity of detected objects, which directly affect the measurements. The continuous posterior is shown in (2.11), found by summing over all possible values of the discrete node. Various ways of evaluating this sum result in different data association techniques. For example, considering all possibilities for the object identity c_t , and maintaining separate posteriors for each is the premise behind multiple hypothesis tracking (MHT). Examples of MHT in localization are [3, 6, 32].

The full MHT posterior is intractable because the number of hypotheses grows exponentially over time [73]. Low probability hypotheses are often removed from consideration (pruned) to maintain tractability of the filter. The growth of hypotheses over time can be considered as a search tree, with each path through the tree representing one possible history (“track”) of detected object identities. Different data association methods differ in terms of the branching factor of this search

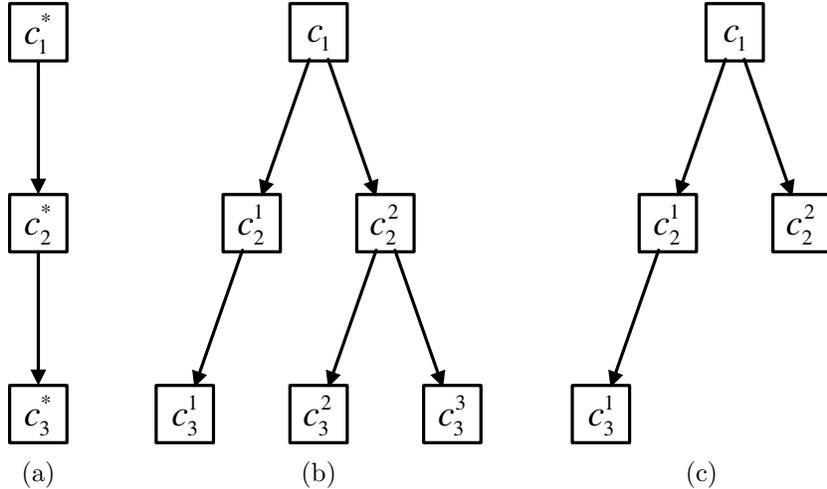


Figure 2.3: Search trees representing different approaches to data association. (a) Maximum Likelihood (b) Multiple Hypothesis Tracking (c) Lazy data association.

tree and the methods of traversing it. Figure 2.3 provides a visual comparison.

- Maximum Likelihood Data Association chooses the association c_t^* at each iteration with the highest probability (2.12) and discarding the rest. This creates a search tree with a branching factor of one. It can also be considered a Depth-First Search [57].

$$c_t^* = \underset{c_t^i}{\operatorname{argmax}} p(c_t^i | z_{1:t}) \quad (2.12)$$

- MHT discards hypotheses with probabilities below a threshold and keeps the rest [67]. In the case of the full MHT where no hypotheses are discarded the search tree will have a branching factor of M , where M represents the number of possible objects detected at each time step. Since all hypotheses are considered at each iteration, this is analogous to a Breadth-First Search [57].
- Lazy data association is a hybrid of these two approaches [29]. At each iteration it chooses the most probable association, however it maintains past associations and may revisit them if they provide a better explanation of the subsequent measurements received. The tree in this case will have a branching factor of M as with MHT, however the tree is explored by a Greedy Best-First Search [57].

Rather than consider data association hypotheses one at a time, the Probabilistic Data Association Filter (PDAF) [7] weights the contribution from various

hypotheses into a single state estimate. While this gives an improvement in robustness over a Maximum Likelihood data association, it can be said that the state estimate being a weighted average of the data association hypotheses is a drawback compared to those methods that maintain the true hypothesis among a set of candidates.

Two other data association approaches are Monte Carlo and Multidimensional Assignment. Monte Carlo data association is used in [46] with particle filter-based localization. Each particle is assigned a random association, and the weight of that particle is updated to reflect the probability of the current measurement for a robot at that position with that association. The resampling step will then remove those particles with low weights, and therefore unlikely associations. Multidimensional Assignment [73] treats data association as an optimization problem, to find the best assignment of associations for a window of N time instants. This method gives similar performance to MHT but with reduced computational complexity.

2.4 Summary

Vision has the capability to be a significant part of a complete vehicle localization system because it complements the established approaches. GPS is the basis for the majority of vehicle localization systems for good reason; it represents a simple method to determine an absolute location, and is sufficiently accurate in most situations. The most persistent faults with GPS occur in urban environments, where multipath and satellite signal loss make the computed position biased or unavailable, respectively. Dead reckoning provides a way to mitigate these faults over short time intervals, but can not be used indefinitely due to the accumulated error issue. Map matching can provide limited accuracy improvement, however it is often not clear what road the vehicle is driving on. Vision has so far found only limited use, but has been successful in improving the accuracy of vehicle localization, and controlling the accumulation of error when GPS position measurements are unavailable. However, this assumes the vehicle is in a landmarks-rich environment, and that landmarks that are detected can be successfully identified.

In the next chapter, the architecture is presented for a complete vehicle localization system incorporating passive localization, map matching and vision components, in addition to a data association component to identify the road being traveled and the landmarks detected using vision. In subsequent chapters, a sys-

tem based on this architecture will be used to investigate the improvement that the addition of visual context brings to the localization and map matching problems.

Chapter 3

A System Architecture for Vehicle Localization

This chapter presents a general system architecture of a localization system that has the capability to integrate visual context into GPS-based localization. The architecture employs a map that provides *a priori* information concerning features of the surrounding environment that are relevant to localization, including their type and location. This provides the system with the knowledge of what features are expected to be visible from a particular location, as well as where the vehicle must be located to observe these features.

The system architecture is designed to be modular, building on existing methods of localization and feature detection. It is also purposefully designed such that no assumptions have been made concerning the use of GPS or any other specific type of sensor to perform localization. This makes the overall system flexible, and applicable to a larger number of existing localization systems whose capabilities will be complemented and enhanced by the addition of visual context. For example, a system combining GPS and dead reckoning to provide multipath mitigation in urban environments would maintain this capability and gain extra information provided by vision.

The system architecture is designed around a probabilistic foundation in Bayesian filtering. This further simplifies the integration of existing localization systems with vision as they are often built upon the Kalman filter, an example of Bayesian filtering applicable to linear-Gaussian systems. Assumptions of conditional independence among the different sources of information enable the system to be designed in a modular manner, and thereby built upon these existing methods. Section 3.1 discusses the theoretical foundations of the architecture while Section 3.2 defines

the roles for each system component.

3.1 Theoretical Foundations

Bayesian filtering is chosen as the foundation of the system architecture for a number of reasons. Not least among these is the fact that the majority of existing vehicle localization systems employ Bayesian filtering, either in the form of Kalman filtering or particle filtering. A modular architecture with this foundation will enable a wide range of existing localization systems to be augmented using visual context.

3.1.1 Notation

The localization problem will be viewed as a state estimation problem. Consistent with the notation of Burgard [12], the expression $bel(x_t)$ will be used to denote the “subjective belief” of the vehicle state x_t . The vehicle state is unknown and must be inferred from uncertain sensory data. Specifically, this expression is shorthand for the posterior conditional PDF in (3.1), where x_t is the state at time t , and $z_{1:t}$ and $u_{1:t}$ are the set of measurement and control input vectors, respectively, from all time instants up to and including t [67]. The state vector x_t expresses the pose of the vehicle in the world coordinates system using the vehicle position and speed, in (3.2).

$$bel(x_t) = p(x_t | z_{1:t}, u_{1:t}) \quad (3.1)$$

$$x_t = \{p_t \ v_t\} \quad (3.2)$$

The posterior represents the belief in x_t after the current measurement z_t has been used to update the state. As Bayesian filters operate on a predict-update cycle, it is useful to express the belief in the predicted state as well, prior to z_t being incorporated. This is denoted by $\overline{bel}(x_t)$ in (3.3).

$$\overline{bel}(x_t) = p(x_t | z_{1:t-1}, u_{1:t}) \quad (3.3)$$

A DBN for the generic Bayesian filtering problem is shown in Figure 3.1, illustrating variable dependencies and conditional independence assumptions inherent to the filtering problem. Control inputs are omitted from the figure. White

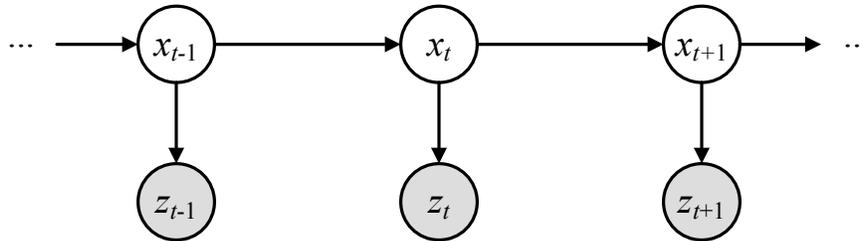


Figure 3.1: The DBN for the general Bayesian filter.

nodes represent state variables, while observed nodes representing measurements are shaded. This simple network is expanded as the complexity of the localization problem grows throughout the chapter.

3.1.2 Vehicle Localization

Letting $z_t = \{z_1, \dots, z_N\}$ be the set of measurements used to update the filter at time t , the update step of the filter can be written as in (3.4). $p(z_t|x_t)$ which follows from the DBN in Figure 3.1 is the measurement probability and reflects the idea that the vector z_t is a function of the state vector x_t . The state vector described the pose of the vehicle by its position and velocity: $x_t = \{p_t v_t\}$. η is a normalization constant. The prediction step of the filter, generating $\overline{bel}(x_t)$ from $bel(x_{t-1})$, is found by the Chapman-Kolmogorov Equation in (2.3).

$$bel(x_t) = \eta \overline{bel}(x_t) p(z_t|x_t) \quad (3.4)$$

From a high-level perspective, any system utilizing Bayesian filtering, including localization, undergoes this predict-update cycle. A simple block diagram illustrating this process is shown in Figure 3.2. The purpose of this diagram is simply to illustrate Bayesian filtering performed by two modules: “State Prediction” and “Measurement Update.” This simple diagram will be gradually expanded into the final system architecture, as shown in Figure 3.7.

It is assumed that the measurement vector z_t is composed of two vectors \hat{p}_t and \hat{f}_t , that is, $z_t = \{\hat{p}_t \hat{f}_t\}$. $\hat{p}_t = \{\hat{p}_t^1, \dots, \hat{p}_t^L\}$ represents a set of vehicle position measurements (such as those from GPS), and $\hat{f}_t = \{\hat{f}_t^1, \dots, \hat{f}_t^M\}$ represents a set of visual feature measurements. It is reasonable to assume that the two sets of measurements are conditionally independent given the vehicle pose as depicted by the network in Figure 3.3, thus creating the factored measurement probability

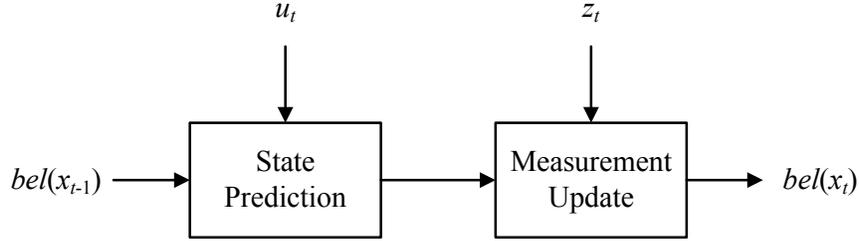


Figure 3.2: State Prediction and Measurement Update modules for Bayesian filtering.

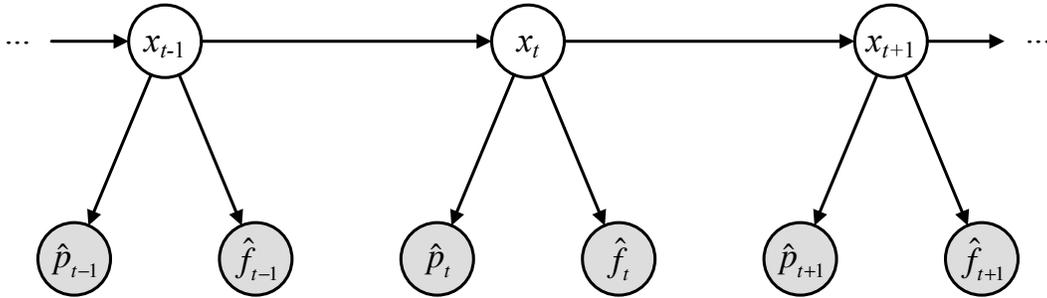


Figure 3.3: A DBN for Bayesian filtering with independent measurement vectors.

in (3.5).

$$p(\hat{p}_t, \hat{f}_t | x_t) = p(\hat{p}_t | x_t) p(\hat{f}_t | x_t) \quad (3.5)$$

This assumption is made based on the rationale that different factors affect the measurement of visual features than affect the measurement of vehicle position. For example, the error in a GPS position measurement would intuitively be uncorrelated with the success rate of detecting a landmark. This is of course an oversimplification, as a number of external factors could result in correlated errors. For example, being in an urban area degrades GPS signal reception, and also overwhelms the vision system as it must deal with the complexity of a more cluttered environment. Also, visual feature estimates are modeled as functions of vehicle position, which is in turn estimated using GPS position estimates; thus the error in this model is related to the error in the GPS position measurement. Modeling such correlations would be rather difficult; therefore, such correlations are disregarded for simplicity in this research.

The “Measurement Update” module is split into “Position Update” and “Visual Context Update” as a result of the factorization of the measurement probability

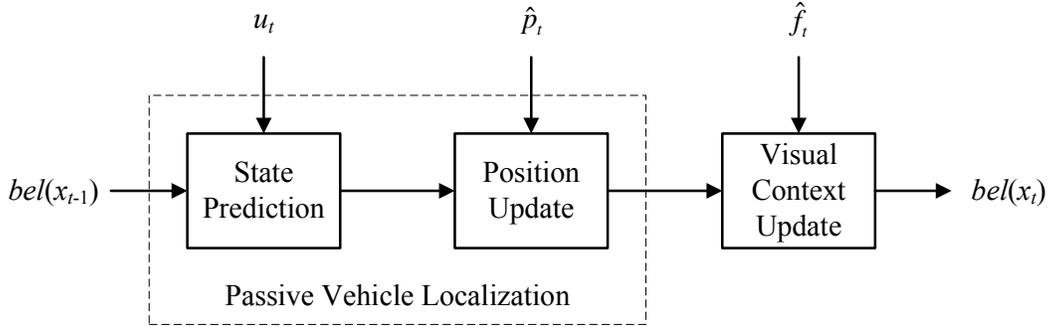


Figure 3.4: A modular update scheme.

depicted in Figure 3.4 and described by (3.6).

$$bel(x_t) = \eta [\overline{bel}(x_t) p(\hat{p}_t|x_t)] p(\hat{f}_t|x_t) \quad (3.6)$$

Recognizing that a localization approach involving only position sensors such as GPS can be achieved using the “State Prediction” and “Position Update” modules, these two modules can be encapsulated into a single “Passive Localization” module. Doing so emphasizes that existing conventional localization approaches (see for example, [13, 64]) can be augmented by a visual context update as a result of the independence assumption in (3.5).

3.1.3 Data Association

The above discussion ignored the problem of data association, which is the association of measurements with the objects in the map they refer to [7]. For example, measuring the distance from the vehicle to a detected landmark provides no indication of vehicle pose unless the identity and position of that landmark are known. However, this is confined to the vision modality, and the lack of data association does not affect the other sensing modalities in the system.

The position of objects in the environment is provided by the map, however the identity of which object is detected is often not known. Identifying objects with a high degree of certainty is difficult in the presence of ambiguity, such as when many similar objects occur in close proximity. The success of vision-based localization can be extremely dependent on the success of data association. Incorrectly identified objects can result in large errors in localization as the system believes it is seeing something it is not, and therefore believes that the vehicle is somewhere it is not.

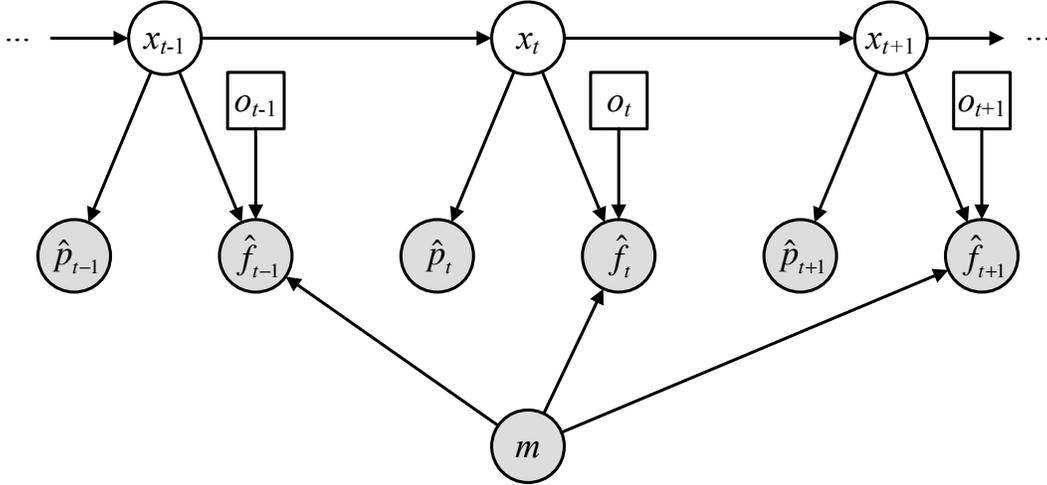


Figure 3.5: A hybrid DBN for Bayesian filtering with data association.

A set of labels $o_t = \{o_t^1, \dots, o_t^M\}$ are used to indicate the identity of objects associated with the feature measurements $\hat{f}_t = \{\hat{f}_t^1, \dots, \hat{f}_t^M\}$. These are discrete values that have been assigned to each object in the map. The measurement probability now depends on the value of the association variable o_t as well as the map: $p(\hat{f}_t | x_t, o_t, m)$ [67]. This is illustrated in a DBN in Figure 3.5, where the square nodes are used to represent the discrete-valued labels; therefore, visual context localization requires a hybrid DBN representation.

3.1.4 Map Matching

What objects are visible by the vehicle at a particular time instant is constrained by: the road the vehicle is traveling on; and the orientation of the vehicle on that road. This knowledge can simplify the identification of detected objects by further restricting the search for plausible measurement associations. This information is also not normally known *a priori*, and must be inferred. The process of identifying the road traveled by the vehicle, and augmenting the vehicle state in light of this information, is known as “map matching” and was introduced in Chapter 2.

It is common for each road segment (a portion of the road defined by intersections with other segments or by dead-ends) to be described separately in the map database and to be assigned a unique label. Thus, map matching is a data association problem as it involves inferring the value of this label. An association variable r_t is used to represent the value of the road segment label at time t .

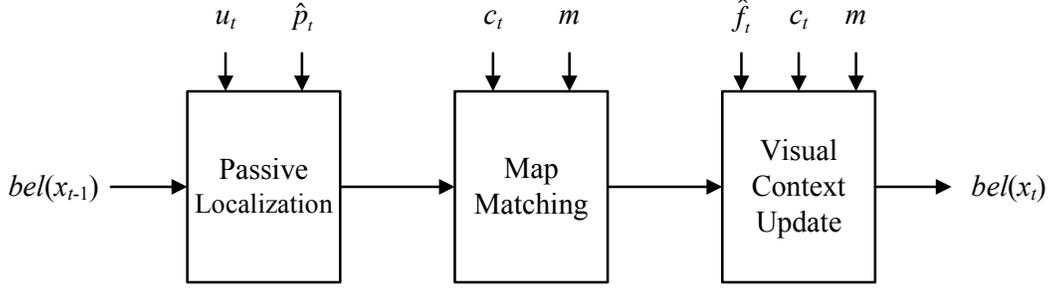


Figure 3.6: Addition of map matching to the system.

Furthermore, a second association variable γ_t is used to specify the orientation of the vehicle on the road segment r_t . The domain of γ_t is $\{-1, 1\}$, which represents a vehicle facing toward and away from, respectively, the beginning endpoint of the segment. Knowing which way the vehicle is facing is important, as it defines the viewpoint of cameras mounted to the vehicle.

The complete data association problem involves identifying the road segment r_t , vehicle orientation γ_t and object associations o_t . For convenience, these can be grouped into a single association vector $c_t = \{r_t, \gamma_t, o_t^1, \dots, o_t^M\}$.

Knowing which road the vehicle is on allows the vehicle pose to be adjusted. There are ways to achieve this adjustment, such as the methods reported in [59, 72], which predominantly involve “snapping” the state estimate to the nearest point (by some criterion) on the road segment. In this thesis, a method is developed to incorporate map data into the vehicle pose using Bayesian filtering. This method is discussed in Chapter 4.

The gist of the map matching approach involves the use of a pseudomeasurement ζ_t , a measurement vector derived from the map data and used to update the vehicle pose. This is assumed conditionally independent of the other measurements. The full measurement vector is therefore now $z_t = \{\hat{p}_t, \zeta_t, \hat{f}_t\}$, and the posterior state update can be written as in (3.7)

$$bel(x_t) = \eta \{ [\overline{bel}(x_t) p(\hat{p}_t|x_t)] p(\zeta_t|x_t, c_t, m) \} p(\hat{f}_t|x_t, c_t, m) \quad (3.7)$$

In light of this multiple measurement update, we can add a “Map Matching” module to the system, as shown in Figure 3.6. This makes for a modular system, with one block for Passive Localization, one for Map Matching to update this initial estimate using the map, and one to update this estimate using Visual Context.

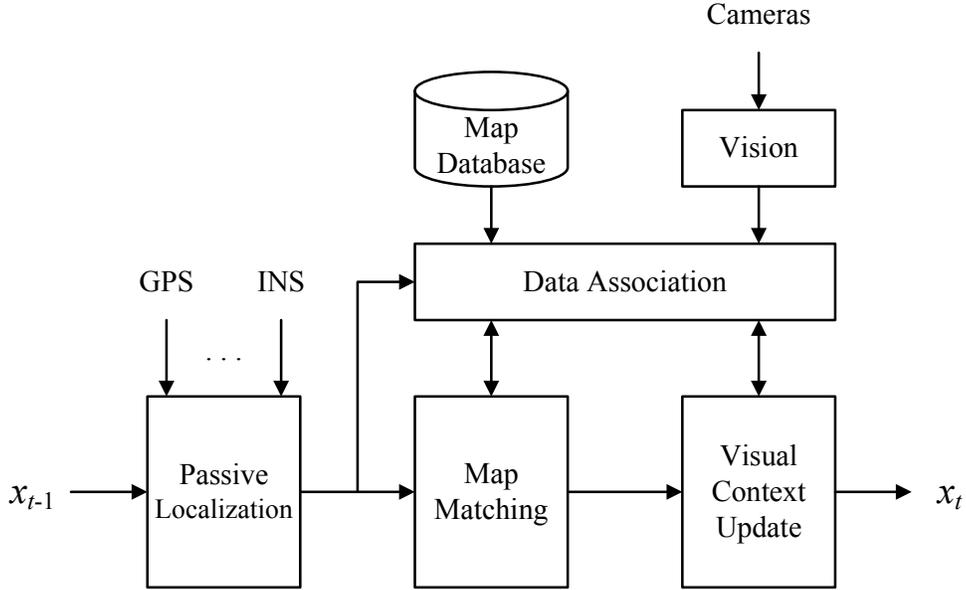


Figure 3.7: System architecture.

3.2 System Architecture

An architecture for a system that implements vehicle localization according to this theoretical foundation is proposed in this section. Six modules are used to define the system, as shown in Figure 3.7. The remainder of this section discusses the role of each module.

3.2.1 Passive Localization Module

The role of this module is to estimate the vehicle pose using traditional road vehicle localization methods. Sensory data can come from satellite positioning systems (GPS, GLONASS, and Galileo), dead reckoning (for example, Inertial Navigation Systems, odometry, wheel speed sensors, steering angle sensors, and magnetic compasses) or other sources (for example, artificial beacons, infrastructure-to-vehicle and vehicle-to-vehicle communications, and GSM).

This module encapsulates the state prediction and position update stages of the localization process introduced in Section 3.1. State prediction takes into account a model of the vehicle dynamics to predict how the vehicle pose changes over time, and also incorporates control inputs u_t to adjust this model. In active localization, the control inputs are used to tell the vehicle to move in a certain direction that facilitates the exploration of the environment [12]. While vehicle localization is

strictly an active process (the driver provides input to the vehicle using the accelerator and brake pedals and the steering wheel) it is assumed to be passive since one rarely has access to these inputs. Dead reckoning sensors provide similar information (distance traveled, turning rates, acceleration) typically with small noise levels, and as a result can be used as control inputs in place of the actual system controls. The reason is that even though they are sensors, dead reckoning sensors measure the effect of control actions [67].

The state update uses vehicle position measurements, from GPS for example, to correct the predicted vehicle pose. These measurements do not depend on the map or data association variables, therefore the measurement probability can be written as $p(\hat{p}_t|x_t, c_t, m) = p(\hat{p}_t|x_t)$. This assumption allows an initial estimate of vehicle localization to be achieved by traditional means. It further enables the system to be modular, building on existing vehicle localization methods by adding the vision element.

3.2.2 Map Matching Module

This module updates the vehicle pose to be consistent with the map and the assumption that a vehicle must be driven on a road at all times. The identity of the road segment and the vehicle orientation will be provided by the Data Association module.

When the road segment identity is known, map matching is simply the process of determining the most probable location of the vehicle on that road. Scott [59] provides a nice explanation of how to achieve this. To this end, map matching to a known road based on Kalman filtering is developed in Chapter 4.

More often than not, the identity of the road is not known. Identifying the road and the pose of the vehicle on it is the subject of many map matching papers [28, 51, 66]. Chapter 5 looks at the map matching and data association problems together, and in this way uses visual information to help identify the road and vehicle orientation. As far as I am aware, this represents a novel approach to map matching.

3.2.3 Vision Module

Cameras must be mounted to the vehicle so as to capture images of the surrounding environment from which features can be extracted. No restriction is placed on the number of cameras used for this purpose or their orientations; however, within this thesis only a single forward-facing camera is used.

The role of the vision module is to process acquired images to detect objects in the environment and estimate their features. Features that are of interest are those that are distinct and unambiguous, ensuring that a detected object can be identified and furthermore provide an indication of vehicle position. It is further advantageous that such features densely populate the environment to facilitate continual updates to the vehicle pose using visual context.

The design of vision algorithms to extract such features is beyond the scope of this thesis. For the purpose of experimental testing, however, an algorithm to detect intersection road markings has been developed. The details of this algorithm are given in Chapter 6.

3.2.4 Map Database

The map provides a model of the environment that enables vision to be used to help localize the vehicle. It provides the location of roads and landmarks that may be used for localization.

The map database is an integral module in the system. It is used by the Data Association module to determine possible roads and the location of detected landmarks. It is used by the visual update module to predict the observed feature values prior to updating the localization estimate. The map is assumed to be error-free, complete, and a current representation of the world. This means that all landmarks that are likely to be detected by the vision system are present and accurately positioned in the map. These assumptions are naïve and unrealistic in practice, yet addressing them is beyond the scope of this thesis.

3.2.5 Data Association Module

This module uses the output of the Passive Localization module, map data and visual data to choose possible road segments that the vehicle may be on and identify

the vehicle orientation on those roads as well as possible objects that have been detected. Under the premise that what features are visible will depend on the road segment and vehicle orientation, these associations are considered together, rather than separately, with the outcome of this block being used by both Map Matching and Visual Context modules.

3.2.6 Visual Context Module

This module uses the features detected by vision, the map database, and the data association result to produce a correction to the localization estimate that results from the map matching module. Two tasks are necessary to achieve this correction. First, a prediction of the value of the observed visual features is generated, which represents how the features are expected to appear based on the current estimate of vehicle pose and the knowledge contained in the map. This is achieved by modeling the feature measurement as a function of the vehicle pose [4]. Second, the localization estimate is updated using the predicted and measured feature values. Achieving these two tasks is the focus of Chapter 4.

3.3 Summary

This chapter has developed a modular system architecture for vehicle localization, based on Bayesian filtering principles. Assumptions of conditional independence allow for a “passive” localization system to be augmented by the addition of map matching and visual context.

In the following chapters, I will propose an implementation of this architecture. Chapter 4 discusses the system for the case of known data associations. Chapter 5 generalizes this to consider the unknown data association case.

Chapter 4

Augmenting Vehicle Localization with Visual Context

This chapter proposes an implementation of the system architecture presented in Chapter 3. Data Association is not addressed. Emphasis is given to the Visual Context and Map Matching modules. Implementations based on Kalman filtering and particle filtering are proposed, although more emphasis is given to the former; the latter is emphasized more in Chapter 5. Data associations are assumed known, to eliminate a significant source of ambiguity and to demonstrate the potential of the proposed system. Chapter 5 will remove this restriction in an effort to enable vision to be employed for localization in a more general and realistic case.

Visual features are modeled as functions of location on the current road. The advantage of this method is that it takes advantage of the vehicle being restricted to travel on the road. A road-based coordinate system is defined to express the vehicle pose in the along-road (longitudinal) and across-road (lateral) directions. This coordinate system is used for map matching, as well as for incorporating visual context. Transformations to and from this coordinate system from world coordinates are also defined.

4.1 Modeling Observed Visual Features

Features observed by vision are not by themselves sufficient to provide an indication of vehicle position. It must be known where the observed features are located in the world to estimate where the vehicle must be for those features to be observed. This motivates the inclusion of a map of the environment in the system, containing

the location and identity of roads and landmarks. From an initial estimate of vehicle location provided by passive sensors, and knowing what road the vehicle is on and what direction it is facing, the map is used to predict the value of feature measurements from the vision system, and from this prediction one can determine the probability of these measurements given the localization estimate, $p(\hat{f}_t|x_t)$.

This section describes the methodology used to model visual features as functions of vehicle position on the current road, thus enabling the value of observed visual features to be predicted. The following introduces a road-based coordinate system and conversion to/from this coordinate system from/to the world coordinate system. A measurement model for intersection road marking landmarks is introduced.

4.1.1 A Road-Based Coordinate System

Roads are represented in the map database as piecewise-linear curves. Each curve is a representation of the centerline of a segment of road between two endpoints, defined by the intersection with another road or a dead-end. Each road segment is a single entry in the database, and is defined by a series of points in world coordinates.

The vehicle pose x_t can be tied to the piecewise-linear road segment r_t by expressing vehicle position and velocity in the lateral or across-road direction (\perp) and the longitudinal or along-road direction (\parallel). The vehicle pose in the road coordinate system for road segment r_t is given by x_t^r in (4.1).

$$x_t^r = \{p_{t,\perp}^r \ p_{t,\parallel}^r \ v_{t,\perp}^r \ v_{t,\parallel}^r\} \quad (4.1)$$

World to Road Coordinates

Each road is defined by a set of points in world coordinates $\{q_1, \dots, q_N\}$, where q_1 is the “beginning” of the segment and q_N is the “ending” of the road segment. Each road is therefore composed of $N - 1$ line segments. Figure 4.1 shows a portion of a map with road points.

Converting from the world coordinate system to the road coordinate system requires finding the closest point on r_t to the vehicle position in the world coor-

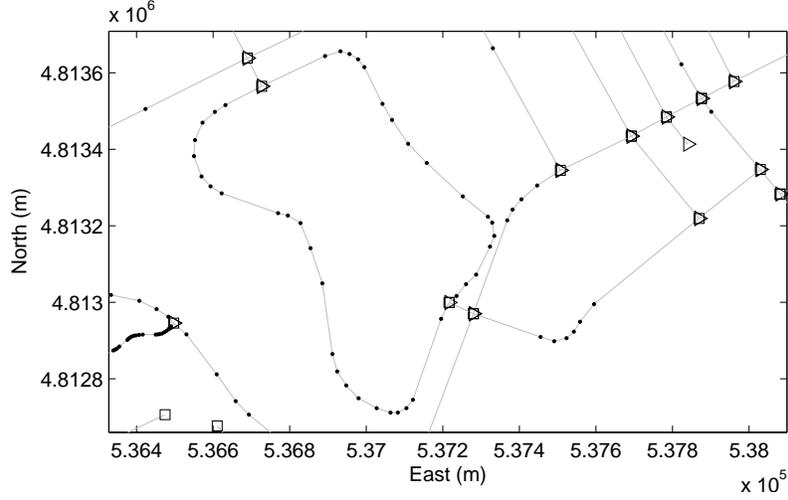


Figure 4.1: A portion of a map showing piecewise-linear road segments. The beginning point of a road is denoted by ‘▷’, the ending point by ‘◻’, and intermediate points by ‘·’. The ending of one segment often coincides with the beginning of another.

dinate system p_t . Let p_t^* denote this point. The conversion can be thought of as “unrolling” the road segment to a straight line, which becomes the abscissa of the road coordinate system along the longitudinal direction; the ordinate is the lateral direction. This is shown in Figure 4.2.

The longitudinal position $p_{t,\parallel}^r$ is defined as the distance from q_1 along r_t to reach p_t^* . This is found by the expression in (4.2), where j is the line segment of r_t that contains p_t^* ; $\|\cdot\|$ represents the Euclidean norm.

$$p_{t,\parallel}^r = \|p_t^* - q_j\| + \sum_{i=2}^j \|q_i - q_{i-1}\| \quad (4.2)$$

The lateral position $p_{t,\perp}^r$ is defined as the distance from p_t to p_t^* , with a sign indicating which side of the road segment p_t falls on. This is found by the dot product of the vector $\overrightarrow{p_t^*p_t}$ and the unit normal $\overrightarrow{n_j}$ to line segment j .

$$p_{t,\perp}^r = \overrightarrow{n_j} \cdot \overrightarrow{p_t^*p_t} \quad (4.3)$$

Lateral and longitudinal velocity are found by projecting v_t onto the unit vectors $\overrightarrow{n_j}$ and $\overrightarrow{u_j}$, respectively. This can be written in matrix notation as in (4.4). Note that this expression is equivalent to applying a rotation matrix $R(\theta_j)$ to the velocity vector, where θ_j is the angle of the line segment j in the world coordinate

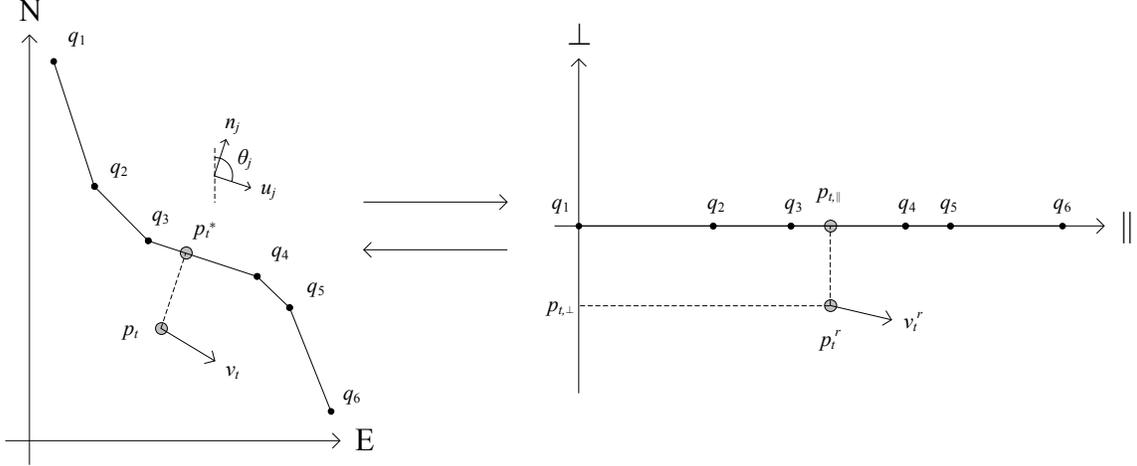


Figure 4.2: Converting between the world and road coordinate systems. The conversion to the road coordinate system can be visualized as an unrolling of the road segment r_t .

system (4.5).

$$v_t^r = \begin{pmatrix} v_{t,\parallel}^r \\ v_{t,\perp}^r \end{pmatrix} = \begin{pmatrix} \vec{n}_j \cdot v_t \\ \vec{u}_j \cdot v_t \end{pmatrix} = \begin{pmatrix} n_e & n_n \\ u_e & u_n \end{pmatrix} \begin{pmatrix} v_{t,e} \\ v_{t,n} \end{pmatrix} \quad (4.4)$$

$$= R(\theta_j)v_t \quad (4.5)$$

The full transformation of the vehicle pose from the world coordinate system to the road coordinate system can be written as follows.

$$x_t^r = \begin{pmatrix} p_{t,\perp}^r \\ p_{t,\parallel}^r \\ v_{t,\perp}^r \\ v_{t,\parallel}^r \end{pmatrix} = \begin{pmatrix} \vec{n}_j \cdot \vec{p}_t^* p_t \\ \|p_t^* - q_j\| + \sum_{i=2}^j \|q_i - q_{i-1}\| \\ \vec{n}_j \cdot v_t \\ \vec{u}_j \cdot v_t \end{pmatrix} \quad (4.6)$$

It would be convenient to convert the vehicle pose covariance matrix from the world coordinate system to the road coordinate system as well, for those systems that use Kalman filtering for localization. A method proposed in [59] can be the basis for this coordinate conversion. Scott [59] provides a method for determining the lateral and longitudinal components of position uncertainty for a long straight road from a position covariance matrix Σ defined in the world coordinate system. The method is equivalent to applying the rotation matrix in (4.7) to Σ , where θ is the angle of the road in the world coordinate system. Equation (4.8) shows the

result of this rotation.

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (4.7)$$

$$\Sigma^r = R(\theta)\Sigma R^T(\theta) = \begin{pmatrix} \sigma_{p_\perp}^2 & \sigma_{p_\perp p_\parallel} \\ \sigma_{p_\perp p_\parallel} & \sigma_{p_\parallel}^2 \end{pmatrix} \quad (4.8)$$

This approach can be extended to also determine the lateral and longitudinal components of velocity uncertainty. The rotation matrix (4.9) is defined, which when applied to the vehicle pose covariance matrix P_t , determines the approximate lateral and longitudinal components of both position and velocity covariance (4.10). The word ‘approximate’ signifies the fact that this conversion is performed using the local heading angle θ_j , which implicitly assumes that the road is straight or at least locally straight. On curving sections of the road, this conversion may not provide a true representation of the vehicle pose covariance.

$$R_P(\theta_j) = \begin{pmatrix} R(\theta_j) & 0_2 \\ 0_2 & R(\theta_j) \end{pmatrix} \quad (4.9)$$

$$\begin{aligned} P_t^r &= R_P(\theta_j)P_t R_P^T(\theta_j) \\ &= \begin{pmatrix} \sigma_{p_\perp}^2 & \sigma_{p_\perp p_\parallel} & \sigma_{p_\perp v_\perp} & \sigma_{p_\perp v_\parallel} \\ \sigma_{p_\perp p_\parallel} & \sigma_{p_\parallel}^2 & \sigma_{p_\parallel v_\perp} & \sigma_{p_\parallel v_\parallel} \\ \sigma_{p_\perp v_\perp} & \sigma_{p_\parallel v_\perp} & \sigma_{v_\perp}^2 & \sigma_{v_\perp v_\parallel} \\ \sigma_{p_\perp v_\parallel} & \sigma_{p_\parallel v_\parallel} & \sigma_{v_\perp v_\parallel} & \sigma_{v_\parallel}^2 \end{pmatrix} \end{aligned} \quad (4.10)$$

Road to World Coordinates

The inverse transformation requires first finding the point p_t^* in world coordinates. As this point lies on line segment j of road r_t , it lies on a straight line between points q_j and q_{j+1} , which are known directly from the map. p_t^* is therefore found by interpolation, given by the weighted average expression in (4.11). This expression is simplified by recognizing that these distance calculations are given by the difference in the longitudinal position of the vehicle $p_{t,\parallel}^r$, and that of the two road points $q_{j,\parallel}$ and $q_{j+1,\parallel}$ (4.12). This can be done since all three of the points are on the same

line segment of road r_t .

$$p_t^* = \frac{\|q_{j+1} - p_t^*\|q_j + \|p_t^* - q_j\|q_{j+1}}{\|q_{j+1} - q_j\|} \quad (4.11)$$

$$= \frac{q_{j+1,\parallel} - p_{t,\parallel}^r}{q_{j+1,\parallel} - q_{j,\parallel}}q_j + \frac{p_{t,\parallel}^r - q_{j,\parallel}}{q_{j+1,\parallel} - q_{j,\parallel}}q_{j+1} \quad (4.12)$$

The vehicle position p_t lies normal to the road at a distance $p_{t,\perp}^r$. Therefore, this can be found using the normal vector \vec{n}_j , as in (4.13).

$$p_t = p_t^* + p_{t,\perp}^r \vec{n}_j \quad (4.13)$$

The velocity vector is returned to the world coordinate system by the inverse of the transformation shown in (4.5). The heading angle θ_j of the line segment j is required to specify the rotation matrix $R(\theta_j)$. The full vehicle pose in the world coordinate system is given by the following.

$$x_t = \begin{pmatrix} p_t \\ v_t \end{pmatrix} = \begin{pmatrix} p_t^* + p_{t,\perp}^r \vec{n}_j \\ R^T(\theta_j)v_t^r \end{pmatrix} \quad (4.14)$$

The covariance matrix is returned to the world coordinate system by the inverse of the transformation shown in (4.10). The rotation matrix $R_P(\theta_j)$ is defined in (4.9). The covariance matrix P_t is thus given by the following.

$$P_t = R_P^T(\theta_j)P_t^r R_P(\theta_j) \quad (4.15)$$

4.1.2 Measurement Models

Predicting the value of observed visual features is achieved by modeling the features as functions of the vehicle pose in the road coordinate system x_t^r . This takes advantage of the fact that the vehicle is restricted to travel on the road, and its orientation is restricted by the lanes of the road. Thus, how the world is seen from the vehicle depends on the vehicle position with respect to the road. Observed feature values f_t are in general a function of x_t^r , noise term ω_t (expressing uncertainty in the model), and the data association parameters c_t consisting of the road segment r_t , the vehicle orientation γ_t and the observed objects o_t . The noise term ω_t is often



Figure 4.3: A detected intersection road marking used to estimate the proximity of the vehicle to an intersection. An estimate of the distance from the camera to the detected marking is overlaid on the image.

assumed to be additive, yielding the following generic feature model.

$$\hat{f}_t = F_t(x_t^r, c_t) + \omega_t \quad (4.16)$$

The nature of the function F_t plays a significant role in how visual features can be incorporated into the localization estimate. When F_t is a linear function of x_t^r , and x_t^r and ω_t have Gaussian distributions, \hat{f}_t will also be Gaussian implying that Kalman filtering can be used to give the optimal update to the vehicle pose. If F_t is non-linear in x_t^r , extended Kalman filtering (EKF), unscented Kalman filtering (UKF) or particle filtering (PF) may be more appropriate, all of which provide sub-optimal solutions in the general case.

Throughout the remainder of the thesis, vision is used to detect landmarks in the form of intersection road markings and to estimate the distance from the camera to the detected markings. The following explains how this visual feature measurement is modeled.

Distance to a Detected Road Marking

Painted road markings are typically present at intersections to delineate where pedestrians may cross or where vehicles should stop. These markings appear at the “entrance” to the intersection rather than within the intersection itself. Furthermore, these markings are painted in a high contrast color (usually white) that

is easily visible on darker asphalt or concrete pavement, and appear as horizontal lines in images acquired using a forward-facing camera. The known orientation and high contrast simplify the visual detection of these markings. An algorithm was developed to perform this detection, and to estimate the distance from the camera to the detected markings under the assumption of a flat road surface. The details of this algorithm are discussed in Chapter 6. Figure 4.3 shows the successful detection of a horizontal road marking with the distance estimate overlaid on the image.

The location of the road markings must be known to use these distance estimates to update the vehicle pose. The road network database does not provide this information directly; instead it provides the center point of the intersection as the point where two or more road segments meet. As road segments are modeled by their centerlines, where these centerlines meet is the center of the intersection. Knowing that intersections are designed and built to specified dimensions, including the location of road markings (given in [45] for the Province of Ontario, Canada), approximate locations for these markings on individual road segments can be determined knowing the center point of the intersection.

The road marking detection algorithm does not distinguish between the three lines typically present at the entrance to an intersection (two crosswalk lines and one stop bar). Instead it returns an estimate of the distance to the marking nearest to the vehicle. Rather than face a complicated data association problem to determine which marking of the three is the one detected, a single point is used to represent the average location of these markings. These points are located on the centerlines of the intersecting road segments, indicating the average distance from the center point of the intersection to the markings. The final result is that the locations of intersection road markings are approximated as shown in Figure 4.4.

The measurement of distance to a detected road marking is modeled as the difference in the longitudinal positions of the vehicle and the landmark on the current road segment. This is shown in (4.17), where $p_{o,\parallel}^r$ is the longitudinal position of the intersection marking on road segment r_t . Recall that γ_t is the orientation of the vehicle; it takes a value from the set $\{-1, 1\}$ to indicate when the vehicle is facing toward or away from the beginning of the road segment, respectively. The use of γ_t ensures that (4.17) produces a positive value whenever the marking is ‘in front’ of the vehicle, regardless of which direction the vehicle is traveling on the road segment. The noise term $\omega_t \sim \mathcal{N}(\omega_t; 0, \Omega_t)$ is modeled as zero-mean Gaussian

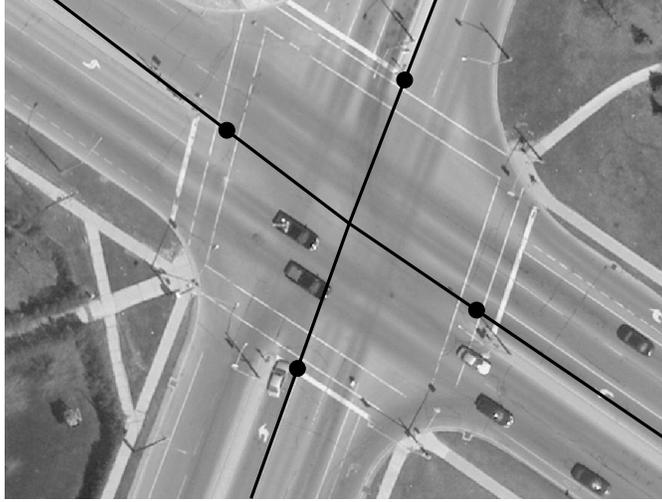


Figure 4.4: A model of intersection road marking locations, generated by knowing the center point of the intersection and the distance from the center point to the road markings.

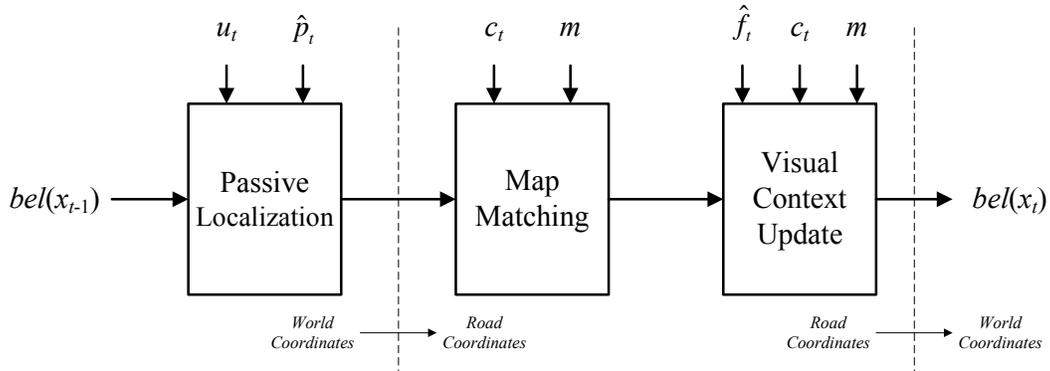


Figure 4.5: A block diagram illustrating the stages of the Kalman filter localization algorithm.

noise.

$$\begin{aligned}\hat{f}_t &= F_t(x_t^r, c_t) + \omega_t \\ &= (p_{o,\parallel}^r - p_{t,\parallel}^r) \gamma_t + \omega_t\end{aligned}\tag{4.17}$$

4.2 Localization by Kalman Filtering

An implementation of a localization system incorporating visual context is proposed in this section. Kalman filtering is used to integrate measurements from GPS and vision, and context from the map as well, into the localization estimate. Figure 4.5

illustrates the flow of information in the localization algorithm. The following system components are discussed.

1. The Passive Localization component which predicts the new vehicle pose and updates this prediction using the GPS position measurements.
2. The Map Matching component which augments the vehicle pose using the known road segment r_t .
3. The Visual Update component which augments the vehicle pose estimate using the context provided by visual feature measurements and the known data associations, including the identities of detected objects o_t .

Experimental results are presented illustrating the improvement in localization accuracy that results from adding context in the form of the map and detected intersection landmarks. These results are found using data collected by a vehicle driving on suburban roads. This provides a number of advantages for the vision system and GPS receiver in terms of clearly-defined road markings, fewer tall buildings and less traffic congestion than in urban area. The GPS receiver has a SiRF Star III receiver chip with a horizontal position accuracy of 10m RMS. A camera with an IEEE 1394 interface is used with a lens having a focal length of 2mm that allows intersection road markings to be detected at a maximum range of approximately 15 meters from the vehicle. The map database is a single-line road network GIS layer for the Regional Municipality of Waterloo made available to the University of Waterloo community by the Region for research purposes. The database is stored in a Shapefile format, developed by the Environmental Systems Research Institute. Intersection models are constructed using the road network data and road marking specifications defined by the Government of Ontario [45]. Finally, a Trimble Pathfinder ProXH GPS receiver with 30cm horizontal position accuracy provides ground truth positions to evaluate the localization accuracy of the proposed approach.

The path of the vehicle for an example data set is highlighted on the map in Figure 4.6, showing the start, finish, and location of any detected intersections. At one of these intersections (Landmark 2) the vehicle was stopped for approximately 20sec, continuously observing intersection markings during this time.



Figure 4.6: The path driven by the vehicle during collection of the example data set. The start (\triangleright) and stop (\square) points of the trip are shown, as well as the location of detected intersection road markings (\star).

4.2.1 Passive Localization by GPS

The Passive Localization component uses Kalman filtering to predict the vehicle pose using a linear model, and update the pose at each time step using position measurements from a GPS receiver. The Kalman filter assumes that the posterior at each time step has a Gaussian distribution, that is, it is fully parameterized by its mean vector and covariance matrix (4.18). The symbols $x_{t|t}$ and $P_{t|t}$ are used to respectively refer to the mean vector and covariance matrix at time t , incorporating measurements from all t time steps.

$$bel(x_t) = \mathcal{N}(x_t; x_{t|t}, P_{t|t}) \quad (4.18)$$

It has been mentioned previously that the pose of the vehicle is defined as its position and speed vector in world coordinates. The Universal Transverse Mercator (UTM) coordinate system is used for this purpose. This defines approximate East-North coordinate systems for a set of zones that are essentially partitions of the Earth's ellipsoid 8° of latitude high by 6° of longitude wide. The Region of Waterloo is in zone 17T, located between latitudes 40°N and 48°N and longitudes 78°W and

84°W. The vehicle pose is defined in this coordinate system as the following 4-tuple (4.19); vertical position and speed are not considered because the vehicle is restricted to the surface of the Earth.

$$x_t = \{p_{t,e} \ p_{t,n} \ v_{t,e} \ v_{t,n}\} \quad (4.19)$$

The constant-velocity model in (4.20) – where the vehicle is assumed to travel along its current velocity vector – is used to describe the vehicle motion. An additive Gaussian noise term $w_t \sim \mathcal{N}(w_t; 0, Q_t)$ captures the uncertainty in this model. While this provides a simple way to predict the next vehicle pose, it may not adapt quickly to vehicle accelerations and decelerations. A more complex model involving vehicle acceleration variables, as considered by Caron [13], may perhaps improve the modeling of the vehicle’s dynamics.

$$\begin{aligned} x_t &= A_t x_{t-1} + w_t \\ &= \begin{pmatrix} I_2 & I_2 \Delta t \\ 0_2 & I_2 \end{pmatrix} x_{t-1} + w_t \end{aligned} \quad (4.20)$$

The mean and covariance of the predicted belief $\overline{bel}(x_t)$, denoted $x_{t|t-1}$ and $P_{t|t-1}$, respectively, are found by the following expressions (4.21)-(4.22).

$$x_{t|t-1} = A_t x_{t-1|t-1} \quad (4.21)$$

$$P_{t|t-1} = A_t P_{t-1} A_t^T + Q_t \quad (4.22)$$

A GPS position measurements $\hat{p}_t = \{\hat{p}_{t,e} \ \hat{p}_{t,n}\}$ is modeled as a linear function of the vehicle pose x_t (4.23), where $v_t \sim \mathcal{N}(v_t; 0, R_t)$ captures the uncertainty in this model. GPS receivers provide position measurements in latitude and longitude coordinates, which are converted to the UTM coordinate system [65] prior to updating the pose.

$$\begin{aligned} \hat{p}_t &= H_t x_t + v_t \\ &= \begin{pmatrix} I_2 & 0_2 \end{pmatrix} x_t + v_t \end{aligned} \quad (4.23)$$

Uncertainty in the GPS position measurement p_t is expressed by R_t , the covariance matrix of the additive noise term v_t . The GPS receiver used throughout this thesis has a horizontal RMS error of 10m. This specification is used to set the

Algorithm 4.1 Pseudocode for GPS localization by Kalman filtering.

```

1. function GPSLocalizationKF(  $x_{t-1|t-1}, P_{t-1|t-1}, \hat{p}_t$  )
2.
3. // Predict the new vehicle pose and covariance matrix
4.  $x_{t|t-1} = Ax_{t-1|t-1}$ 
5.  $P_{t|t-1} = AP_{t-1|t-1}A^T + Q_t$ 
6.
7.  $S_t = HP_{t|t-1}H^T + R$  // Innovation covariance
8.  $K_t = P_{t|t-1}H^TS_t^{-1}$  // Kalman gain
9.
10. if  $\hat{p}_t \neq \emptyset$ 
11. // Update using GPS position measurements, if available
12.  $x_{t|t} = x_{t|t-1} + K_t(\hat{p}_t - Hx_{t|t-1})$ 
13.  $P_{t|t} = (I - K_tH)P_{t|t-1}$ 
14. else
15. // Keep prediction result
16.  $x_{t|t} = x_{t|t-1}$ 
17.  $P_{t|t} = P_{t|t-1}$ 
18.
19. return  $x_{t|t}, P_{t|t}$ 

```

magnitude of R_t , which is kept at a constant value and is referred to as simply R . Other parameters that can be kept constant are A_t , H_t and Q_t . GPS localization is achieved by Algorithm 4.1, which takes only three arguments: the previous vehicle pose, its covariance matrix, and the current GPS measurement.

The principal benefit of using a Kalman filter with GPS position measurements is to maintain localization integrity when the GPS receiver does not provide a measurement as a result of poor satellite visibility. When this occurs the posterior belief $bel(x_t)$ is considered equal to the predicted belief $\overline{bel}(x_t)$, since no measurement is available to perform the update. Thus the filter is more robust than a standalone receiver to missing position measurements.

This result can be seen when the filter is applied to GPS measurement data gathered from a vehicle-mounted receiver. In cases where satellites are not visible for long periods, the localization error and covariance will increase dramatically since no measurement is available to correct the predictions. Localization error in this case may be controlled somewhat through the use of dead reckoning, however this is not investigated here. Figure 4.7 shows a magnified section of the path traveled by the vehicle plotted on a map of the roads. Missing GPS measurements are handled by the prediction process, and consequently the vehicle position estimates

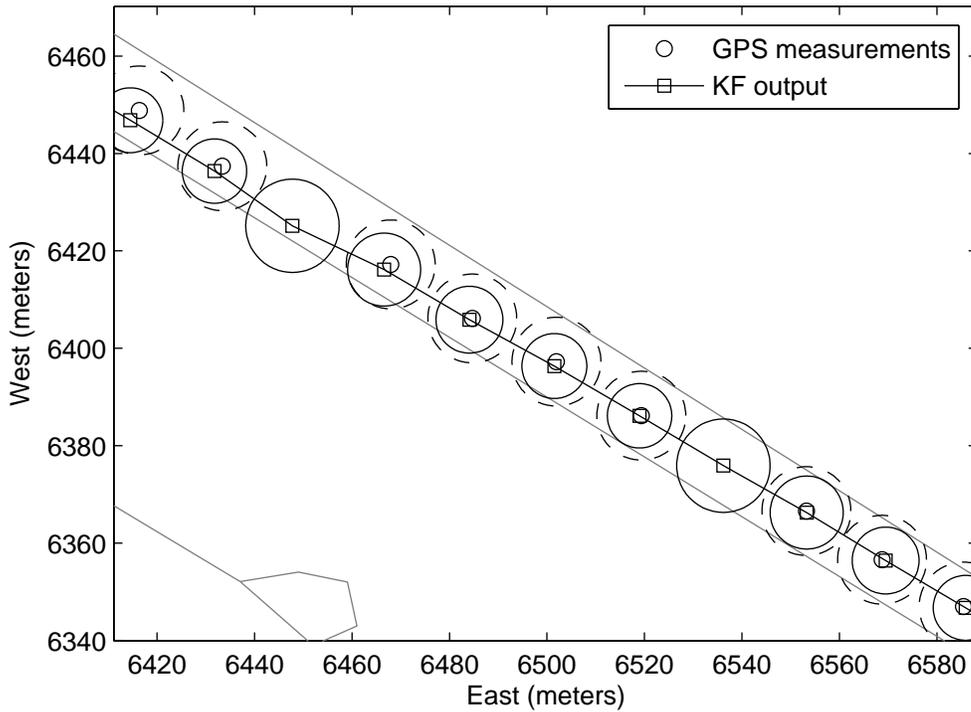
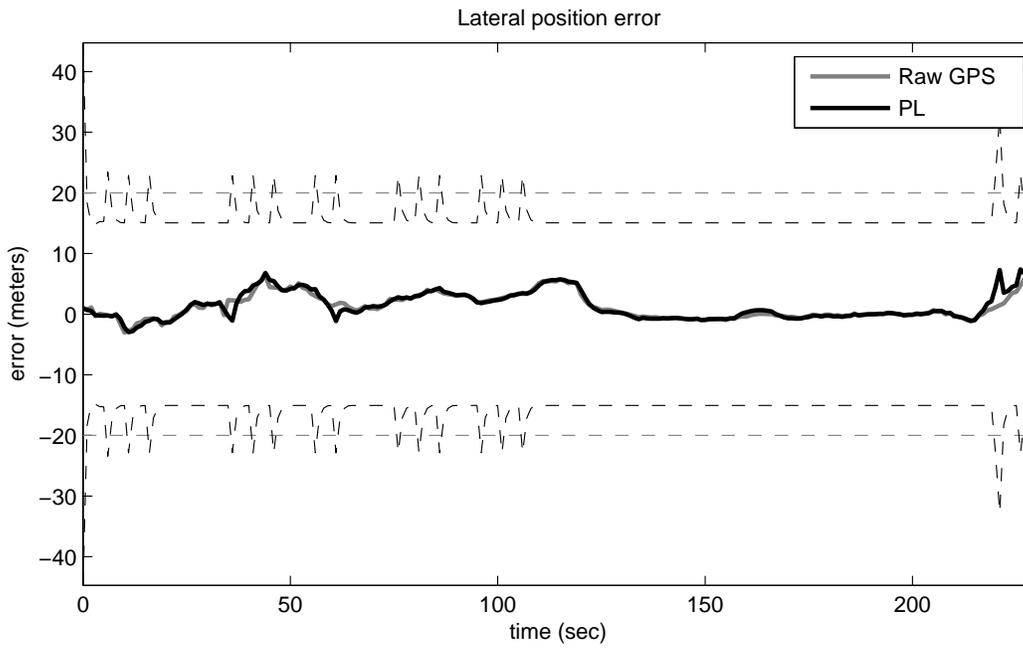
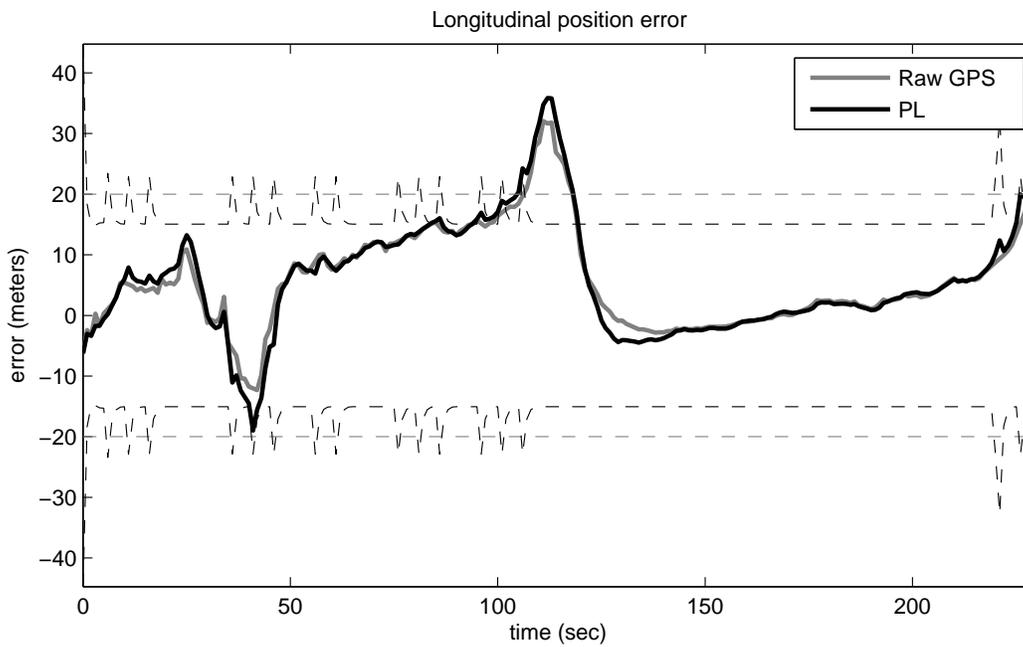


Figure 4.7: An illustration of using the Kalman filter to mitigate occasional missing GPS position measurements.

at these time instants show larger error ellipses. Figure 4.8 shows the error in the GPS measurements compared to the filtered results in the lateral and longitudinal directions relative to the ground truth road segment at each time instant, for the trip shown in Figure 4.6. Notice that neither error nor uncertainty are dramatically different compared to the raw GPS measurements. This is expected since the Kalman filter is simply tracking these measurements over time. These observations are confirmed by Table 4.1, which summarizes the error in the raw GPS measurements and in the output of the Kalman filter. This table contains the mean and standard deviation of the position error ($\bar{\epsilon}$ and s_{ϵ}), and the mean and standard deviation of the position uncertainty ($\bar{\sigma}$ and s_{σ}). The average position error from the Kalman filter is slightly larger than from the raw data, because the filter is following the raw measurements and is slow to react to sudden changes.



(a)



(b)

Figure 4.8: A comparison of vehicle position error in the raw GPS measurements and the output of the Passive Localization module using the Kalman filtering implementation. a) Error in the lateral direction. b) Error in the longitudinal direction.

	$\bar{\varepsilon}$ (m)	s_ε (m)	$\bar{\sigma}$ (m)	s_σ (m)
\hat{p}	7.4	7.1	10	0
p_{PL}	8.5	7.6	8.0	1.5

Table 4.1: A comparison of position error and uncertainty between the raw GPS data (\hat{p}) and Passive Localization (p_{PL}) using Kalman filtering.

4.2.2 Augmenting Localization using Map Matching

Incorporating context into the localization estimate from the map and from vision is performed in the road-based coordinate system introduced in Section 4.1.1. As shown in Figure 4.5 the vehicle pose is converted to road coordinates x_t^r after Passive Localization. The road segment r_t required for this transformation is assumed known *a priori* at this time. This is a strong assumption to make because such knowledge will not be available in general. Chapter 5 relaxes this assumption to make the localization approach more applicable in the general case. The approach presented here is still applicable to situations where the road segment is known *a priori*, for example, the localization of a public transit bus with a fixed route.

The map m used to update the vehicle pose is a GIS road network layer. The vehicle pose x_t^r is updated to be consistent with the location and heading angle of the road currently being driven by the vehicle. Assuming normal driving behavior – that 1) the vehicle is on the road network, and 2) the vehicle follows the heading of the road – using the map effectively restricts the set of plausible locations for the vehicle thereby keeping localization uncertainty low.

A Kalman filtering approach is used to update the vehicle pose with the road network data, according to the two assumptions above. A pseudomeasurement vector ζ_t is created based on these assumptions and used to update the vehicle pose x_t^r . ζ_t is defined in one of two ways, depending on the outcome of the conversion from the world coordinate system to the road coordinate system. Specifically, it depends on whether or not the longitudinal position $p_{t,\parallel}^r$ falls within the bounds of the road segment r_t .

Case 1: Vehicle within the road segment bounds

The vehicle pose x_t^r falls within the bounds of the known road segment r_t if the condition $0 \leq p_{t,\parallel}^r \leq L_r$ is satisfied, where L_r is the length of road segment r_t . This

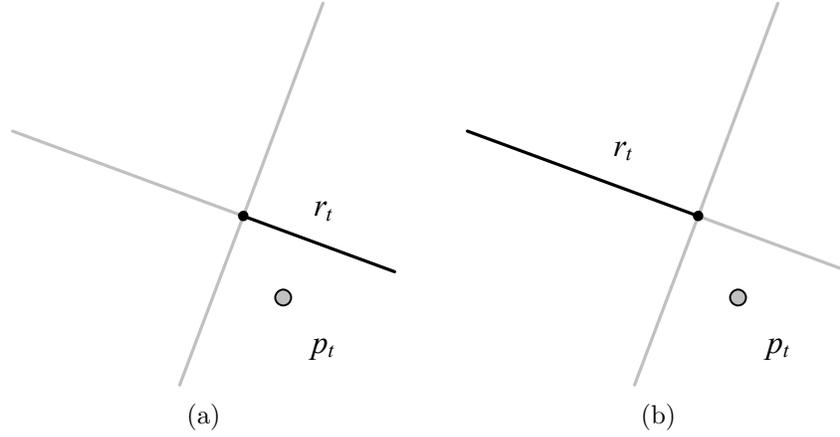


Figure 4.9: An illustration of the vehicle pose relative to the bounds of the road segment. Vehicle pose can be a) inside the bounds of the road segment, or b) outside the bounds of the road segment.

is illustrated in Figure 4.9a. The pseudomeasurement ζ_t in this case is defined as the expected lateral position and speed of the vehicle relative to the road segment r_t (4.24). Both are defined to be zero in accordance with our assumptions, meaning that the vehicle position is restricted to lie on the road centerline with no lateral offset, and that its velocity is aligned with the heading of the road segment. The covariance matrix Z_t may be tuned to reflect the level of confidence in the map data. These values are set arbitrarily to $\sigma_{p_\perp} = 5\text{m}$ and $\sigma_{v_\perp} = 2\text{m/s}$.

$$\zeta_t = \begin{pmatrix} \hat{p}_{t,\perp}^r \\ \hat{v}_{t,\perp}^r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (4.24)$$

$$Z_t = \begin{pmatrix} \sigma_{p_\perp}^2 & 0 \\ 0 & \sigma_{v_\perp}^2 \end{pmatrix} \quad (4.25)$$

Case 2: Vehicle outside the road segment bounds

The vehicle pose x_t^r falls outside the bounds of the known road segment r_t if either condition $p_{t,\parallel}^r < 0$ or $p_{t,\parallel}^r > L_r$ is satisfied. This is illustrated in Figure 4.9b. The pseudomeasurement ζ_t in this case incorporates the longitudinal position $p_{t,\parallel}^r$ as well (4.26). This takes on either a value of 0 or L_r if the vehicle pose x_t^r is closer to the beginning or ending of the road segment, respectively. The purpose of this pseudomeasurement is to bring the vehicle pose closer to the nearest endpoint on

Algorithm 4.2 Pseudocode for map matching using Kalman filtering.

```

1. function MapMatchingKF(  $x_{t,p}^r, P_{t,p}^r, r_t, m$  )
2.
3.  $L_r = \text{GetLengthOfRoad}( r_t, m )$ 
4. // Create pseudomeasurement, covariance and measurement matrices
5. if  $x_{t,p}^r \geq 0$  &&  $x_{t,p}^r \leq L_r$ 
6.    $\zeta_t = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ,  $Z_t = \begin{pmatrix} \sigma_{p,\perp}^2 & 0 \\ 0 & \sigma_{v,\perp}^2 \end{pmatrix}$ ,  $M_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ 
7. else
8.   if  $x_{t,p}^r < 0$ ,  $\zeta_t = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ 
9.   else,  $\zeta_t = \begin{pmatrix} 0 \\ L_r \end{pmatrix}$ 
10.   $Z_t = \begin{pmatrix} \sigma_{p,\perp}^2 & 0 & 0 \\ 0 & \sigma_{p,\parallel}^2 & 0 \\ 0 & 0 & \sigma_{v,\perp}^2 \end{pmatrix}$ ,  $M_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ 
11.
12. // Kalman gain
13.  $S_{t,m} = M_t P_{t,p}^r M_t^T + Z_t$ 
14.  $K_{t,m} = P_{t,p}^r M_t^T S_{t,m}^{-1}$ 
15.
16. // Update vehicle pose and covariance matrix
17.  $x_{t,m}^r = x_{t,p}^r + K_{t,m}(\zeta_t - M_t x_{t,p}^r)$ 
18.  $P_{t,m}^r = (I - K_{t,m} M_t) P_{t,p}^r$ 
19.
20. return  $x_{t,m}^r, P_{t,m}^r$ 

```

the known road segment. For tuning Z_t we set $\sigma_{v,\parallel} = 5\text{m}$.

$$\zeta_t = \begin{pmatrix} \hat{p}_{t,\perp}^r \\ \hat{p}_{t,\parallel}^r \\ \hat{v}_{t,\perp}^r \end{pmatrix} = \begin{cases} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \text{if } p_{t,\parallel}^r < 0 \\ \begin{pmatrix} 0 \\ L_r \\ 0 \end{pmatrix} & \text{if } p_{t,\parallel}^r > L_r \end{cases} \quad (4.26)$$

$$Z_t = \begin{pmatrix} \sigma_{p,\perp}^2 & 0 & 0 \\ 0 & \sigma_{p,\parallel}^2 & 0 \\ 0 & 0 & \sigma_{v,\perp}^2 \end{pmatrix} \quad (4.27)$$

The map matching procedure is shown in Algorithm 4.2. The KF update approach using a pseudomeasurement vector essentially matches the initial localization estimate from the passive localization system to the nearest point on the known

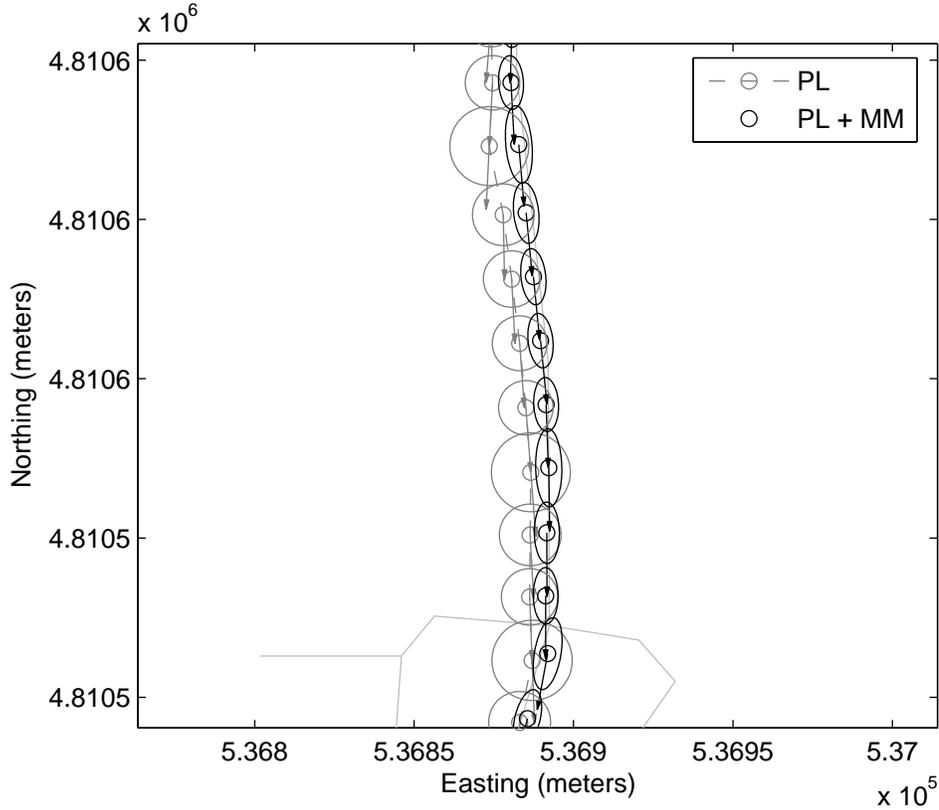
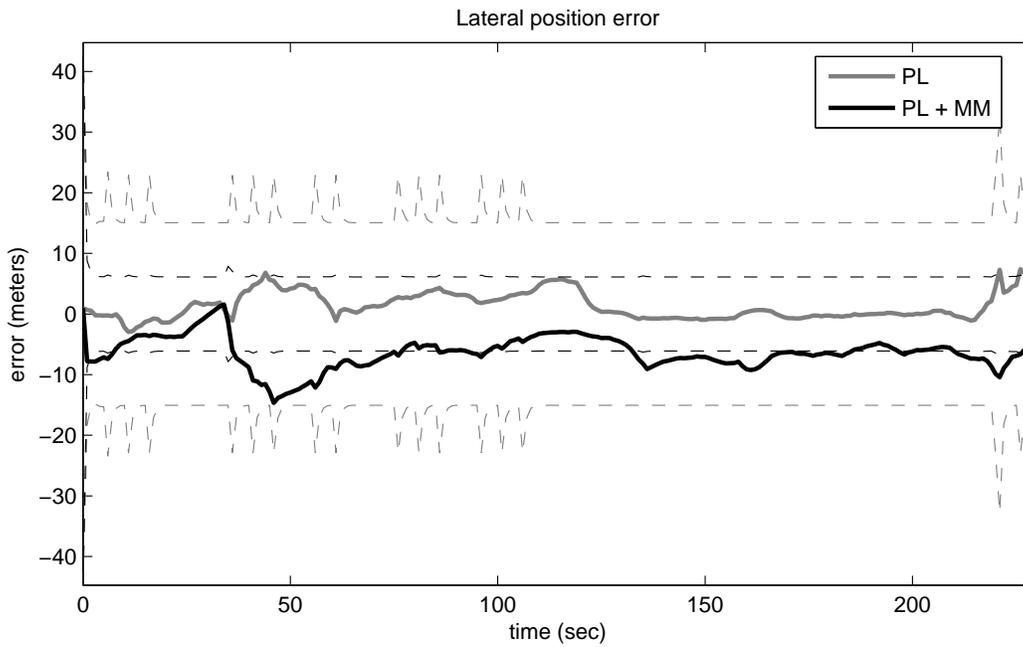


Figure 4.10: An illustration of vehicle pose estimates matched to the known road segment. Note that uncertainty is reduced in the lateral direction.

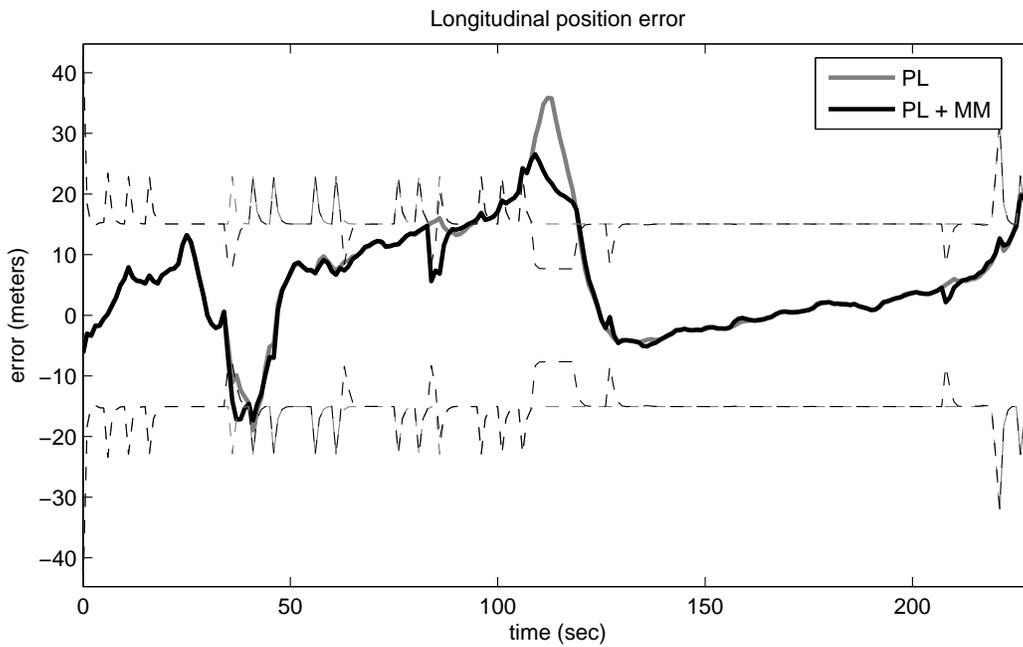
road segment and aligns the heading vector with the heading of the road. In the former respect it is similar to some previous map matching approaches [59, 72]. Figure 4.10 illustrates these updates to the vehicle pose, compared with the Passive Localization results. What is most obvious in this figure is that map matching brings the localization estimate closer to the road centerline and reduces uncertainty in the lateral direction, directing it along the road segment.

Figure 4.11 shows the error in the map matching result compared to Passive Localization results in both the lateral and longitudinal directions. It is not surprising that the longitudinal position error is unaffected for the most part since the pseudomeasurement update occurs in the lateral direction except in rare circumstances. The effect of map matching on lateral position estimates is summarized in Table 4.2. This table shows the sample mean and standard deviation of the lateral position error ε_{\perp} ($\bar{\varepsilon}$ and s_{ε}) and the same for the estimated uncertainty $\sigma_{p_{\perp}}$ ($\bar{\sigma}$ and s_{σ}).

The results show that average uncertainty in the lateral position estimate is



(a)



(b)

Figure 4.11: A comparison of vehicle position error when using Passive Localization (PL), and when using Passive Localization and Map Matching (PL+MM) in the Kalman filtering implementation. a) Error in the lateral direction. b) Error in the longitudinal direction.

	$\bar{\varepsilon}$ (m)	s_{ε} (m)	$\bar{\sigma}$ (m)	s_{σ} (m)
$p_{\perp,PL}$	1.3	2.2	7.9	1.2
$p_{\perp,MM}$	-6.4	2.7	3.1	0.12

Table 4.2: A comparison of lateral position error and uncertainty between Passive Localization ($p_{\perp,PL}$), and Passive Localization and Map Matching ($p_{\perp,MM}$) using Kalman filtering.

reduced compared to Passive Localization, and is close to a constant value having sample standard deviation close to zero. However, the average error in the lateral position estimate is in fact larger than with Passive Localization. This illustrates an important point of matching to road centerlines that should be a cause for concern. This increase in error is due to the fact that the road centerline does not precisely represent the location where the vehicle is actually driving: within a lane offset from this centerline. In fact, the reduced lateral uncertainty is misleading as it does not account for the bias introduced by the map matching process.

It will not always be the case that map matching introduces error into the localization result; in this data set the amount of lateral position error in the GPS measurements is already very small. It is expected that map matching will be able to reduce localization error in environments where extreme multipath creates errors in GPS position measurements larger than the 6 meters or so of bias that matching to the centerline introduces.

In cases where a detailed map is available showing the boundaries of specific lanes on the road, matching to the correct lane can improve accuracy over a standalone GPS receiver. This was performed in [16] using vision to help determine the correct lane.

4.2.3 Augmenting Localization with Visual Context

Vision is used to provide further context for vehicle localization. The vehicle pose x_t is updated by incorporating the measured visual features \hat{f}_t . By positioning the vehicle with respect to its environment in this way, the set of plausible vehicle locations is further restricted. The map is required for this purpose, to provide knowledge of the locations of objects in the surrounding environment.

What objects are in the field of view of the camera at a given time depends on

the position and orientation of the vehicle, assuming that the camera is mounted to the vehicle and has no moving parts. An initial estimate of the vehicle position is provided by the Passive Localization and Map Matching components. Vehicle orientation is given by the discrete variable γ_t . As stated previously, the value of this parameter indicates that the vehicle is facing toward the beginning point of the road segment r_t ($\gamma_t = -1$) or facing toward the ending point of r_t ($\gamma_t = 1$). The vehicle heading is not used to define the orientation since it is possible for the vehicle to be facing in one direction but moving in another (in reverse, for example). Where the vehicle is facing defines the view of the camera. As with the other discrete variables (road segment r_t and detected objects o_t), γ_t is assumed known at this time.

Vision is used to detect intersection road markings and estimate the distance to these markings from the camera. This distance estimate is modeled using (4.17) as a function of the vehicle pose x_t^r , the discrete parameters c_t and the map m . As this model is strictly a non-linear function of the vehicle pose (due the constant offset term $p_{o,\parallel}^r$) an EKF update is used to incorporate the measurement \hat{f}_t into the vehicle pose estimate.

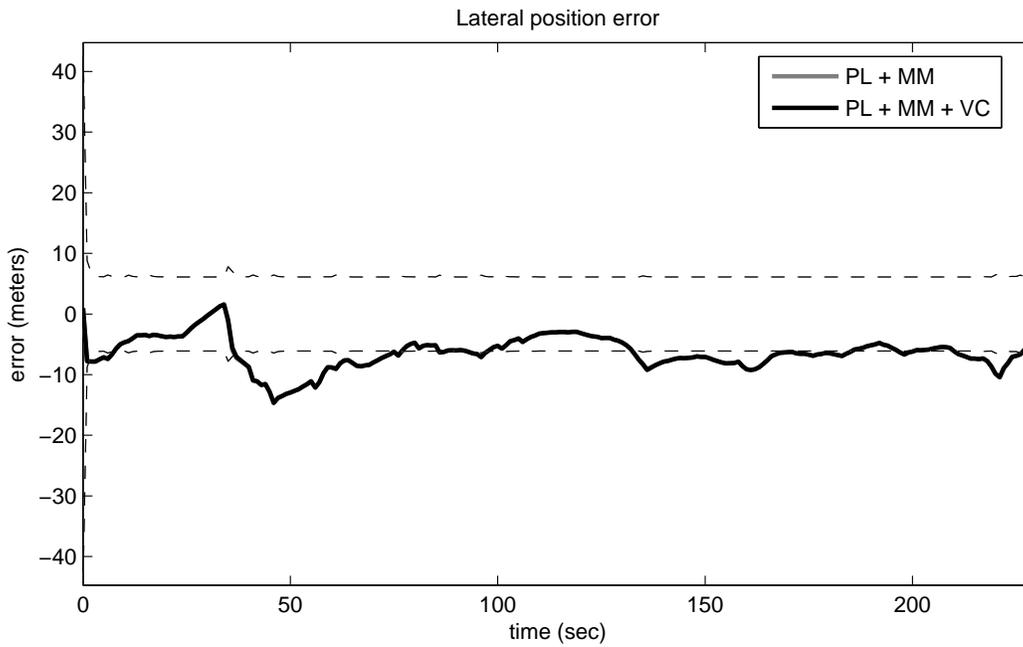
The EKF update requires the expected value of the feature measurement \hat{f}_t as predicted by the feature model (4.28), and the Jacobian matrix G_t , being a linearization of the feature model about x_t^r (4.29).

$$E[f_t] = (p_{o,\parallel}^r - E[p_{t,\parallel}^r]) \gamma_t \quad (4.28)$$

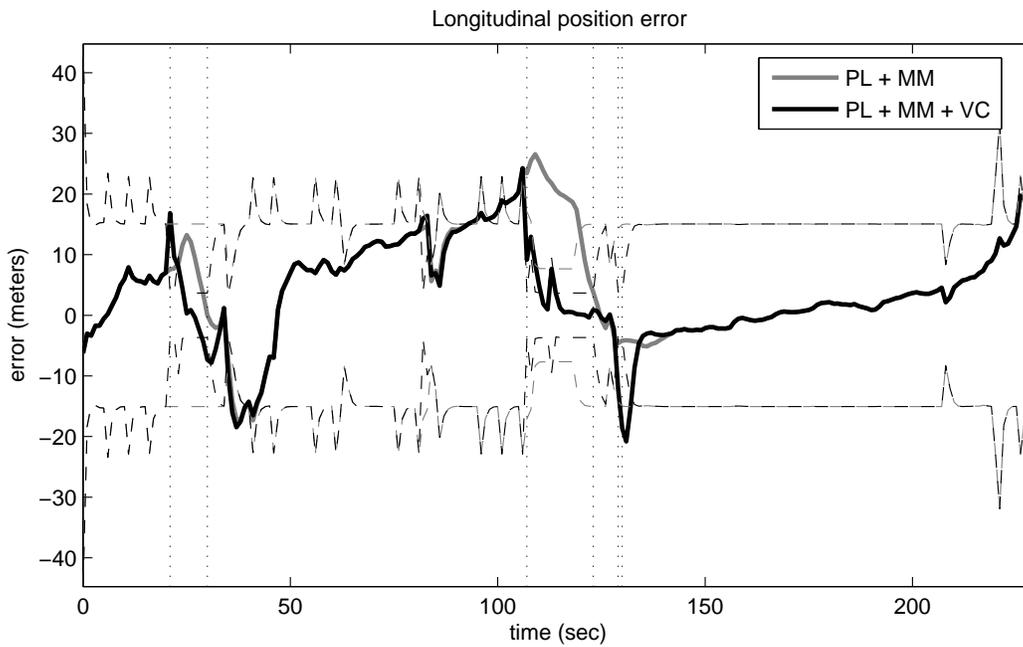
$$G_t = \frac{df_t}{dx_t^r} = \begin{pmatrix} 0 & -\gamma_t & 0 & 0 \end{pmatrix} \quad (4.29)$$

The linearization that results from using the Jacobian should not cause divergence of the Kalman filter gain (which is a concern when measurement models are highly non-linear) because the measurement model in this case is a first-order equation in x_t^r (4.17). The pseudocode for the visual context update is shown in Algorithm 4.3.

Figure 4.12 shows plots of the localization error using visual context in the lateral and longitudinal directions. Results are shown for map matching as well for comparison. Longitudinal position error and uncertainty are reduced **at times when landmarks are detected** (these instances are marked with three pairs of dotted vertical lines in Figure 4.12b) compared to the map matching result at the same time instants. This is summarized in Table 4.3, which shows the sample



(a)



(b)

Figure 4.12: A comparison of vehicle position error when using Passive Localization and Map Matching (PL+MM), and when using Passive Localization, Map Matching and Visual Context (PL+MM+VC) in the Kalman filtering implementation. a) Error in the lateral direction. b) Error in the longitudinal direction.

Algorithm 4.3 Pseudocode for adding visual context to the localization estimate.

```

1. function KFVisionUpdate(  $\mathbf{x}_{t,-}$ ,  $P_{t,-}$ ,  $\hat{\mathbf{f}}_t$ ,  $\mathbf{c}_t$  )
2.
3.  $\mathbf{f}_{t,-}$  = FeatureModel(  $\mathbf{x}_{t,-}$ ,  $\mathbf{c}_t$  )
4.  $\mathbf{G}_t$  = Jacobian( FeatureModel,  $\mathbf{x}_{t,-}$ ,  $\mathbf{c}_t$  )
5.
6.  $\mathbf{S}_t$  =  $\mathbf{G}_t P_{t,-} \mathbf{G}_t^T + \mathbf{R}_t$  // Innovation covariance
7.  $\mathbf{K}_t$  =  $P_{t,-} \mathbf{G}_t^T \mathbf{S}_t^{-1}$  // Kalman gain
8.
9. // Localization update
10.  $\mathbf{x}_t$  =  $\mathbf{x}_{t,-} + \mathbf{K}_t (\hat{\mathbf{f}}_t - \mathbf{f}_{t,-})$ 
11.  $P_t$  =  $(\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) P_{t,-}$ 
12.
13. return  $\mathbf{x}_t, P_t$ 

```

	$\bar{\varepsilon}$ (m)	s_ε (m)	$\bar{\sigma}$ (m)	s_σ (m)
$p_{\parallel, VC}$	3.5	6.9	2.8	2.2
$p_{\parallel, MM}$	14.3	8.7	12.6	4.0

Table 4.3: A comparison of longitudinal position error and uncertainty between Passive Localization and Map Matching ($p_{\parallel, MM}$), and Passive Localization, Map Matching and Visual Context ($p_{\parallel, VC}$), using Kalman filtering.

mean and standard deviation of the longitudinal position error ε_{\parallel} ($\bar{\varepsilon}$ and s_ε) and the same for the longitudinal position uncertainty $\sigma_{p_{\parallel}}$ ($\bar{\sigma}$ and s_σ). These results indicate that the addition of vision improves localization accuracy, and can compensate for significant bias in the GPS position measurements. The reduction in uncertainty is expected since the additional visual update step will further contract the posterior PDF.

Elaborating on the performance of the filter within the time interval $t = 107\text{sec}$ and $t = 123\text{sec}$ (detail shown in Figure 4.13). During this time interval, the vehicle was stopped at an intersection, and the vision system was able to successfully detect the intersection road markings at all but two time instants. It can be seen in Figure 4.13 that the filter converges to zero error in the longitudinal direction by continually observing this landmark, thereby correcting a GPS measurement bias as large as 40m. In effect a feedback mechanism has been created that gradually corrects for the bias in the GPS position estimates that could not be corrected using map matching.

In the proposed implementation, visual context updates localization estimates

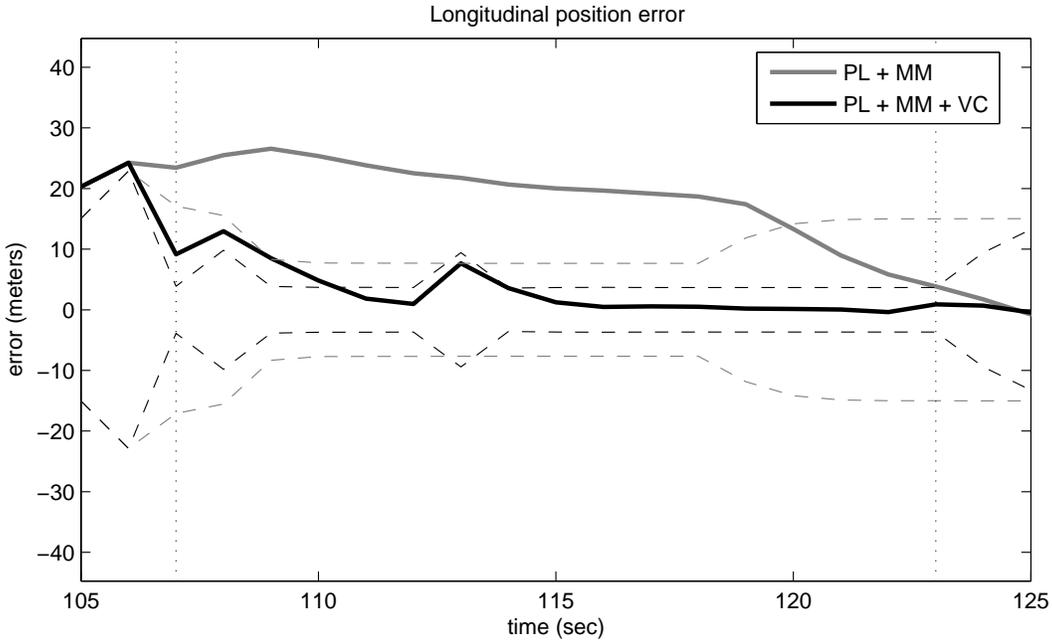


Figure 4.13: An illustration of longitudinal position error converging to zero using Kalman filtering, while a road marking is continually observed during the time interval $t = 107\text{sec}$ and $t = 123\text{sec}$.

in the longitudinal direction, while map matching updates primarily in the lateral direction. They are therefore complementary operations. This should not be considered a general rule, as this is the direct result of modeling map pseudomeasurements as functions of lateral position and landmark distance measurements as functions of longitudinal position.

4.2.4 Discussion

The results show that localization can be improved through the observation of a specific feature: intersection road markings. To better understand the general case, consider what will happen for a generic visual feature modeled as a non-linear function of vehicle position. For simplicity it is assumed this model is a function of longitudinal position only. The model and its Jacobian will be (4.30) and (4.31), respectively, where α_t is used to represent the value of the partial derivative.

$$\hat{f}_t = F_t(x_t^r) + \omega_t = F_t(p_{t,\parallel}^r) + \omega_t \quad (4.30)$$

$$G_t = \begin{pmatrix} 0 & \frac{\partial F_t}{\partial p_{t,\parallel}^r} & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \alpha_t & 0 & 0 \end{pmatrix} \quad (4.31)$$

α_t	0	$\pm\sqrt{\rho}$	$\pm\infty$
K_p	0	$\pm\frac{1}{2\sqrt{\rho}}$	0
K_σ	1	$\frac{1}{2}$	0

Table 4.4: Summary of feature rate-of-change dependency on position and variance gains

The EKF equations result in the following expressions for updating the longitudinal position estimate and variance, where the subscript ‘-’ is used to denote the prior value of a quantity before it is updated.

$$p_{t,\parallel}^r = p_{t,\parallel,-}^r + K_p \left(\hat{f}_t - F_t(p_{t,\parallel,-}^r) \right) \quad (4.32)$$

$$\sigma_{p,\parallel}^2 = K_\sigma \sigma_{p,\parallel,-}^2 \quad (4.33)$$

The terms K_p (4.34) and K_σ (4.35) are gain terms determining the change in the position estimate and variance, respectively. These terms depend on the feature rate of change α_t , the feature measurement uncertainty σ_f^2 and the prior position uncertainty $\sigma_{p,\parallel,-}^2$. The term ρ is the ratio of these uncertainties, used to simplify the following expressions.

$$K_p = \frac{\alpha_t \sigma_{p,\parallel,-}^2}{\alpha_t^2 \sigma_{p,\parallel,-}^2 + \sigma_f^2} = \frac{\alpha_t}{\alpha_t^2 + \rho} \quad (4.34)$$

$$K_\sigma = \frac{\sigma_f^2}{\alpha_t^2 \sigma_{p,\parallel,-}^2 + \sigma_f^2} = \frac{\rho}{\alpha_t^2 + \rho} \quad (4.35)$$

It is apparent from these expressions that the rate of change of the feature model, α_t , influences how the position estimate is updated. A plot of the two gain terms (4.34) and (4.35) versus α_t is shown in Figure 4.14 for $\rho = 1$. Table 4.4 summarizes the gains at pertinent values of α_t .

The parameter α_t describes how the feature model is currently changing with longitudinal vehicle position. It is an indication of how distinct the feature is. What is particularly interesting is that when $\alpha_t = 0$, there is no contribution made to the longitudinal position estimate ($K_p = 0$) and the estimated variance does not change ($K_\sigma = 1$). This can be rationalized by considering a feature with a constant value; measurements of this feature always return the same value no matter where the vehicle is located; therefore observing the feature provides no indication of the

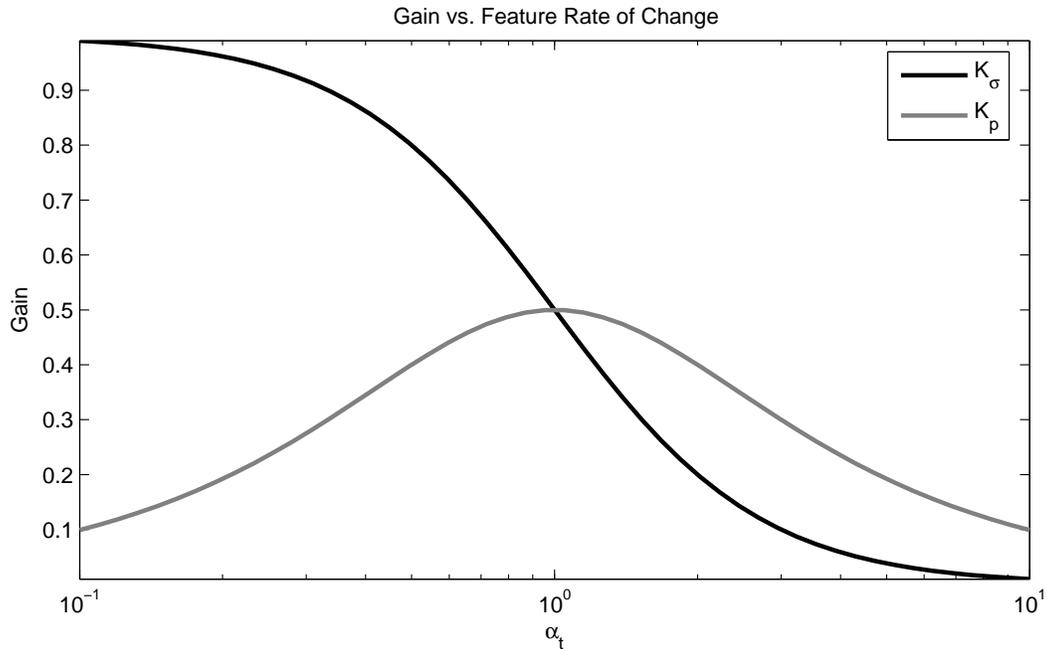


Figure 4.14: Position gain K_p and variance gain K_σ versus feature rate of change α_t .

vehicle position.

There is also a maximum contribution to the position estimate that can be made by a feature, occurring when $\alpha_t = \pm\sqrt{\rho}$. While α_t is a product of the environment and cannot be adjusted, it would be possible to adjust the ratio ρ by designing vision algorithms with a desired σ_f . Thus, one can to design a vision algorithm to best use a particular feature.

Looking Ahead

The assumption of a correct world model, meaning road segments and landmark locations, may contribute to localization error. In modeling the road marking locations, all intersections are assumed to have the same dimensions for simplicity. In practice this is rarely the case, as factors such as lane width, crosswalk width, turning lanes, bicycle lanes, the angle of the road intersection, among others, create highly variable intersection geometries. Such modeling errors manifest themselves as localization errors since the vehicle is positioned with respect to an incorrect reference point. This may become a limitation of a localization system if the error in landmark location is larger than the desired accuracy of the system.

On a larger scale, accurately mapping landmark locations would be a prohibitively large undertaking to perform manually. It would thus be advantageous to be able to update the landmark locations in the map in addition to using them for localization. This would be an application of Simultaneous Localization and Mapping (SLAM). The same approach would be useful to refine the road segment locations as well, which may change due to alterations to existing roads or the addition of new roads. An advantage to correcting this disparity automatically is that it avoids the need to repeatedly purchase updated maps.

The results obtained in this section have assumed that the data association parameters (the road segment identity, vehicle orientation and landmark identity) are known, and these can therefore be considered best-case results. Achieving similar results for the case of unknown associations is necessary for this approach to be applicable in a general sense. Possible approaches are discussed in Chapter 5.

4.3 Localization by Particle Filtering

This section investigates using particle filtering in place of the Kalman filter for localization. Particle filtering has the benefit that features are not required to be linear functions of vehicle position as with the Kalman filter, and that the Gaussian restriction is no longer present. However, the cost of this flexibility is losing the optimality that Kalman filtering offers, as well as an increase in computational complexity.

The particle filter represents the main change in the implementation of the system architecture from the last section. The measurement models are still linear functions of the vehicle pose, and Gaussian distributions are still assumed. As a result, it is not expected that the outcome of using a particle filter will improve upon the Kalman filter at this time. Rather, the benefit of particle filtering will become more apparent when data associations are unknown, and the random sampling of the state space allows for a more comprehensive consideration of plausible hypotheses. This will not be explored until the next chapter; however, the particle filter implementation for known data associations is presented here to simplify its introduction in Chapter 5.

The key design decisions to make for implementing the particle filter are the choice of the importance density and the resampling strategy. Samples are drawn

from the importance density, denoted $q(x_t|x_{t-1}^i, z_t)$, to generate a new set of particles $\{x_t^i\}_{i=1}^N$ from the previous set $\{x_{t-1}^i\}_{i=1}^N$ [4]. Choosing the importance density to be the state transition prior $q(x_t|x_{t-1}^i, z_t) = p(x_t|x_{t-1}^i)$ is a very popular choice of importance density because of its simplicity. This choice also simplifies the update of the particle weights, which can be achieved according to (4.36), by calculating only the measurement probability for each particle.

$$w_t^i \propto w_{t-1}^i p(z_t|x_t^i, c_t^i) \quad (4.36)$$

The factorization of the measurement probability performed in (3.7) can be used with (4.36) to update the particle weights in a modular manner. Thus, the modular measurement update that characterizes the KF implementation can be maintained.

The set of particles representing the vehicle pose will be denoted as follows (4.37).

$$\{x_t^i\}_{i=1}^N = \{p_{t,e}^i \ p_{t,n}^i \ v_{t,e}^i \ v_{t,n}^i\}_{i=1}^N \quad (4.37)$$

The localization procedure is illustrated in Algorithm 4.4. Each measurement update reweighs each particle based on how well it agrees with the measurement data. The resampling step removes low-weight particles by drawing N samples with replacement from the set $\{x_t^i\}_{i=1}^N$, where the probability of particle i being drawn is proportional to the weight of that particle w_t^i . After resampling, all particles receive equal weight, $w_t^i = \frac{1}{N}$. Resampling is performed every T time steps, where $T > 1$. Letting $T = 1$ was found to create sample impoverishment; an insufficient number of unique particles remained after resampling to adequately cover the state space.

4.3.1 Passive Localization by GPS

As with the Kalman filter implementation, the purpose of this module is to perform prediction of the vehicle pose at t based on the posterior at $t - 1$ using a motion model for the vehicle, and to update the predicted pose using passive position measurements \hat{p}_t . The prediction process generates a set of particles $\{x_t^i\}_{i=1}^N$ from the previous set $\{x_{t-1}^i\}_{i=1}^N$. This is achieved using (4.38), where v_t^i is a sample drawn from the process noise, assumed to have a Gaussian distribution $v_t \sim \mathcal{N}(v_t; 0, Q_t)$,

Algorithm 4.4 Pseudocode for particle filtering localization with known data associations.

```

1. function PFLocalization(  $x_{t-1}^{1:N}, w_{t-1}^{1:N}, z_t, c_t$  )
2.
3. // Vehicle pose prediction and GPS update
4.  $x_t^{1:N} = \text{PFpredict}( x_{t-1}^{1:N} )$ 
5.  $\lambda_{t,p}^{1:N} = \text{GPSWeight}( x_t^{1:N}, \hat{p}_t )$ 
6.
7. // Map matching update
8.  $\lambda_{t,m}^{1:N} = \text{MapMatchingWeight}( x_t^{1:N}, c_t )$ 
9.
10. // Visual Context update
11.  $\lambda_{t,v}^{1:N} = \text{VisualContextWeight}( x_t^{1:N}, \hat{f}_t, c_t )$ 
12.
13. // Update particle weights
14.  $w_t^i = w_{t-1}^i \lambda_{t,p}^i \lambda_{t,m}^i \lambda_{t,v}^i \quad \forall i \in [1, N]$ 
15.
16. if  $t \% T == 0$ 
17.    $x_t^{1:N} = \text{ResampleParticles}( x_t^{1:N}, w_t^{1:N} )$ 
18.    $w_t^i = 1/N \quad \forall i \in [1, N]$ 
19. end if
20.
21. return  $x_t^{1:N}, w_t^{1:N}$ 

```

and A is the constant velocity state transition model defined in (4.20).

$$x_t^i = A_t x_{t-1}^i + v_t^i \quad (4.38)$$

The weight w_t^i for each particle x_t^i is updated using the measurement probability $p(\hat{p}_t | x_t^i)$. Particles located near to the GPS position measurement $\hat{p}_t = \{\hat{p}_{t,e} \hat{p}_{t,e}\}$ will receive higher weight, and therefore have a higher probability of being resampled. Weights are determined by the measurement probability (4.39), where H_t is defined as in (4.23).

$$w_{t,p}^i = p(\hat{p}_t | x_t^i) \sim \mathcal{N}(\hat{p}_t - H_t x_t^i; 0, R_t) \quad (4.39)$$

This simple particle filter is applied to the same data set used in Section 4.2 with the Kalman filter. The result of using this particle filter with $N = 200$ particles for Passive Localization, incorporating measurements only from the GPS receiver, is displayed in Figure 4.15. This figure shows the localization error at each time step based on the mean of all particle locations, after resampling. The sub-optimality of

	$\bar{\varepsilon}$ (m)	s_ε (m)	$\bar{\sigma}$ (m)	s_σ (m)
$p_{\perp,PL}$	1.5	4.2	12.6	2.5
$p_{\parallel,PL}$	5.9	13	12.7	2.7
$p_{\perp,MM}$	-6.6	3.0	8.9	2.2
$p_{\parallel,MM}$	5.9	13.6	12.6	2.6

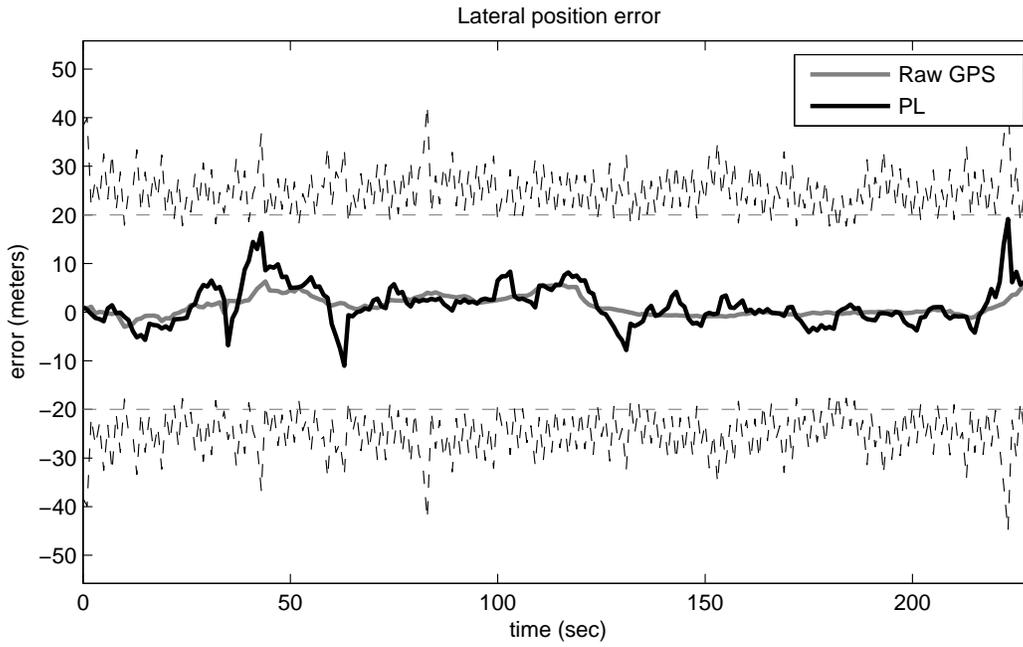
Table 4.5: A comparison of position error and uncertainty between Passive Localization (PL), and Passive Localization and Map Matching (MM), using particle filtering.

the particle filter can be readily observed by comparing this figure with Figure 4.8. The output of the Kalman filter smoothly follows the GPS measurements, while the output of the particle filter resembles a random walk as it does so. Furthermore, the PF appears to react slower than the KF to sudden measurement changes from the GPS. Being a non-deterministic filter, running the filter twice with the same measurements and initial state returns two different results. The results shown are chosen as “typical” results. The localization error for this filter is summarized in Table 4.5.

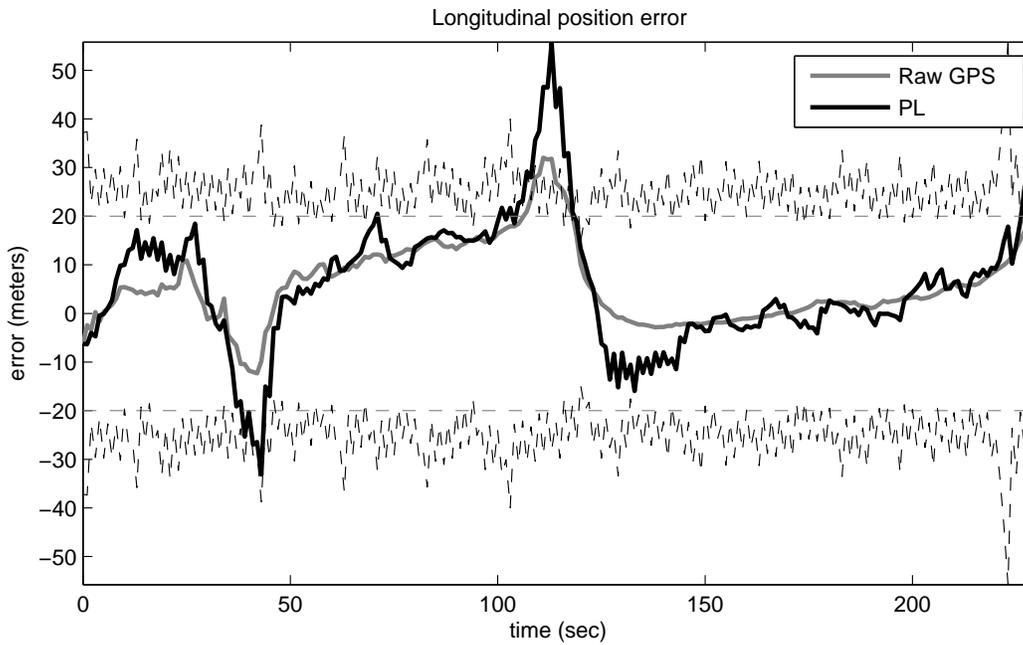
Resampling of particles is performed every $T = 2\text{sec}$. Resampling has the effect of contracting the set of particles by discarding unlikely ones far from the true vehicle position. The effect of resampling is visible in Figure 4.15; the uncertainty envelope widens and contracts every two seconds, creating a sawtooth pattern. This represents a drawback of using this resampling strategy; the most certain estimates will occur when resampling occurs because outliers have been removed, however this does not occur at every time instant.

4.3.2 Augmenting Localization using Map Matching

Map matching is implemented as with the KF implementation, except that it is now on a per-particle basis. Each particle is reweighted to reflect how close it is to the known road segment, and how similar its heading is to the angle of the road. The pseudomeasurement ζ_t^i is again used to represent the expected position and heading of the vehicle relative to the known road segment. ζ_t^i was defined previously in (4.24)-(4.27). The major distinction between the KF and PF approaches is that a pseudomeasurement ζ_t^i is defined for every particle x_t^i . Each particle is first expressed in road coordinates $x_t^{r,i}$, and then reweighted using the pseudomeasurement



(a)



(b)

Figure 4.15: A comparison of vehicle position error in the raw GPS measurements and the output of the Passive Localization module using the particle filtering implementation. a) Error in the lateral direction. b) Error in the longitudinal direction.

	$\bar{\varepsilon}$ (m)	s_ε (m)	$\bar{\sigma}$ (m)	s_σ (m)
$p_{\parallel,VC}$	9.4	12.2	8.8	2.8
$p_{\parallel,MM}$	24.5	17.1	12.0	2.3

Table 4.6: A comparison of longitudinal position error and uncertainty between Passive Localization and Map Matching ($p_{\parallel,MM}$), and Passive Localization, Map Matching and Visual Context ($p_{\parallel,VC}$), using particle filtering.

probability (4.40).

$$w_{t,m}^i = w_{t,p}^i p(\zeta_t^i | x_t^{r,i}) \quad (4.40)$$

$$p(\zeta_t^i | x_t^{r,i}) \sim \mathcal{N}(\zeta_t^i - M_t x_t^{r,i}; 0, Z_t)$$

Results from a typical run of the particle filter incorporating map matching are shown in Figure 4.16, and the summarized in Table 4.5 with the Passive Localization results. The mean lateral position error becomes more consistent as a result, and the standard deviation is also reduced in the lateral direction. The bias shown is again due to the fact that the lane the vehicle is driving in is offset from the road centerline. As expected from the KF results, the error in the longitudinal direction (Figure 4.16b) is similar to the passive localization case (see Figure 4.15b). Again, these results will not be identical due to the non-deterministic nature of the filter.

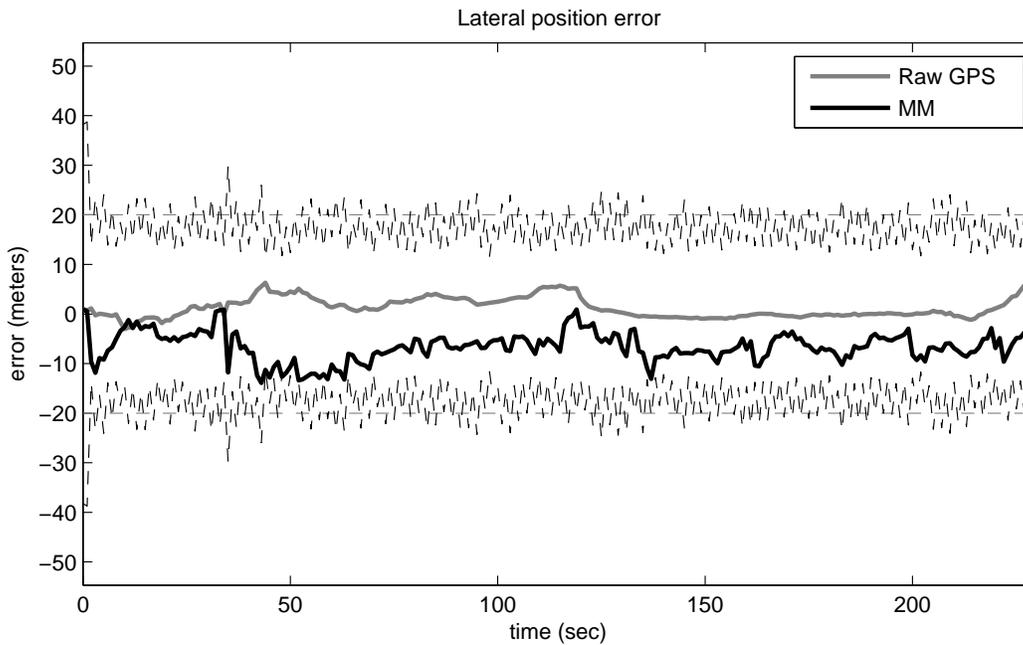
4.3.3 Augmenting Localization with Visual Context

Visual context is added to the localization estimate by reweighting the particles using the measurement probability of the observed visual features. Again, the vision system is used to detect intersection road markings, which are used as landmarks to localize the vehicle. The same feature model as the EKF (4.17) is used to predict the feature measurement for each particle. The measurement probability $p(\hat{f}_t | x_t^{r,i})$ for each particle is used to reweight that particle, as in (4.41).

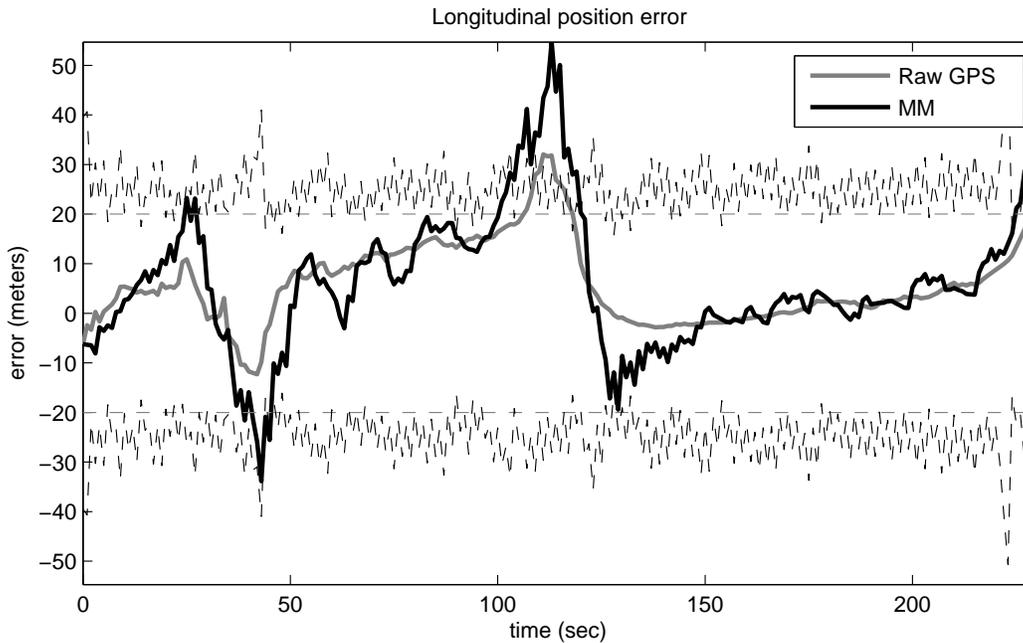
$$w_{t,v}^i = w_{t,m}^i p(\hat{f}_t | x_t^{r,i}, c_t, m) \quad (4.41)$$

$$p(\hat{f}_t | x_t^{r,i}, c_t, m) \sim \mathcal{N}(\hat{f}_t - F_t(x_t^{r,i}, c_t, m); 0, \Omega_t)$$

Results for the particle filter in this scenario are shown in Figure 4.17, and the improvement over map matching demonstrated by Table 4.6. Immediately

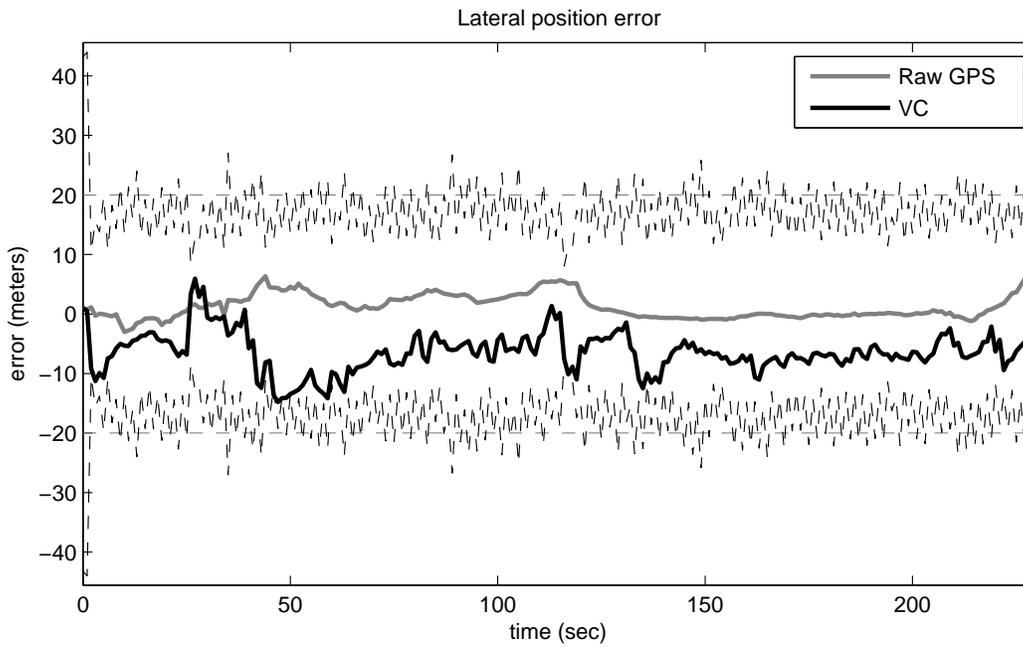


(a)

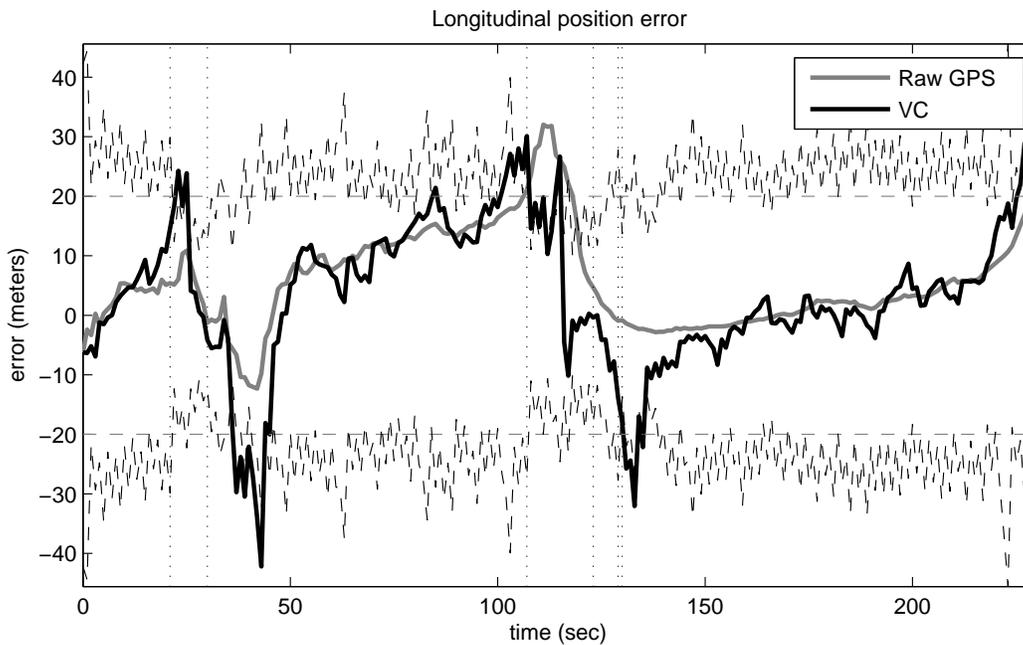


(b)

Figure 4.16: A comparison of vehicle position error when using Passive Localization (PL), and when using Passive Localization and Map Matching (PL+MM) in the particle filtering implementation. a) Error in the lateral direction. b) Error in the longitudinal direction.



(a)



(b)

Figure 4.17: A comparison of vehicle position error when using Passive Localization and Map Matching (PL+MM), and when using Passive Localization, Map Matching and Visual Context (PL+MM+VC) in the particle filtering implementation. a) Error in the lateral direction. b) Error in the longitudinal direction.

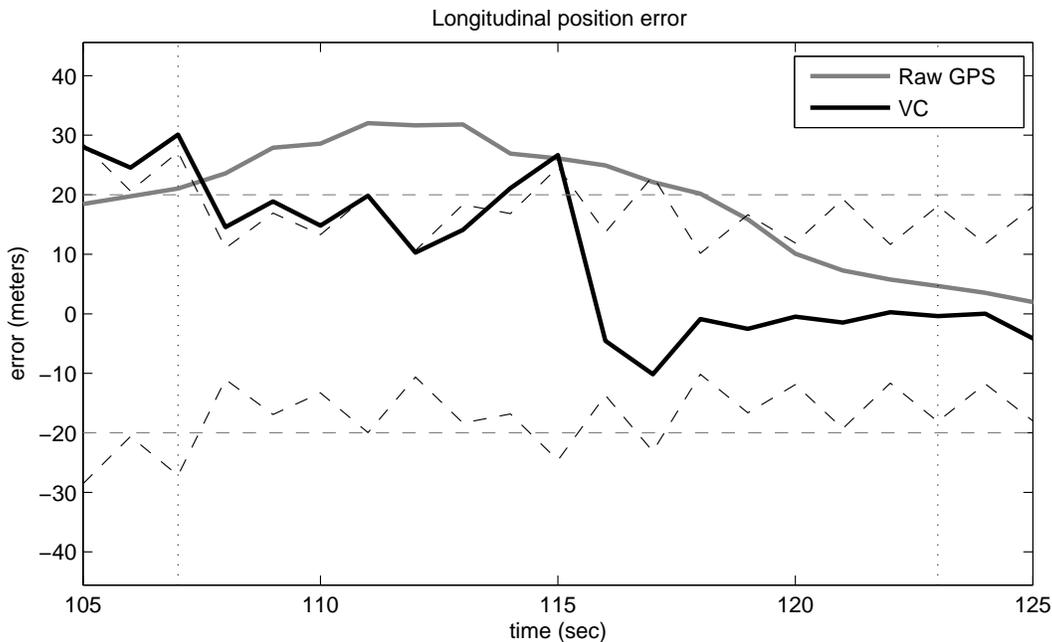


Figure 4.18: An illustration of longitudinal position error converging to zero using particle filtering, while a road marking is continually observed during the time interval $t = 107\text{sec}$ and $t = 123\text{sec}$.

noticeable is that the longitudinal error component (Figure 4.17b) becomes very consistent as a result of positioning the vehicle with respect to the intersection road marking. While the vehicle was stopped and thus was continually detecting the road marking, it was able to correct its position using this visual context. This is shown in Figure 4.18. Thus, the longitudinal position error is quite small and also very consistent. This is also reflected by the small standard deviation of the particles in the longitudinal direction.

The results signify that by continually positioning the vehicle with respect to its surroundings while the vehicle is stopped, it is possible to counteract GPS measurement bias, using either a Kalman filter or particle filter.

4.4 Summary

This chapter has presented two implementations of the system architecture proposed in Chapter 3: the first based on Kalman filtering, the second based on particle filtering. These implementations use a propose road-based coordinate system to perform map matching and to incorporate visual context.

With both implementations, it is shown that positioning the vehicle with respect to detected landmarks improves vehicle localization accuracy. These results are arrived at by assuming data association parameters (the road segment identity, vehicle orientation and landmark identity) are known *a priori*.

The next chapter aims to relax this assumption of known data association using methods of Multiple Hypothesis Tracking (MHT) and Monte-Carlo data association.

Chapter 5

Data Association: Map Matching and Landmark Identification

In this chapter the assumption of known data associations is removed. Such information will not be known *a priori* in the majority of applications, and therefore by removing this assumption the localization approach becomes more practical.

From a high-level point of view, the data association problem within the Bayesian filtering paradigm can be considered to be that of inferring the value of the discrete parameters that affect the estimation of the continuous state variables. For the vehicle localization problem these parameters are the identity of the road segment where the vehicle is located, the orientation of the vehicle on that segment, and the identity of any detected landmarks. This is discussed in Chapter 4. These parameters are not measured directly, but instead their values must be inferred based on other available information, including measurements from GPS and vision, the map database, and state estimates from previous time instants. However, their value affects the state estimate and can cause a divergence if incorrect[46].

In particular, the two former discrete parameters (road segment and orientation) are important for navigation of the vehicle, to tell the driver where to turn or when they have reached their destination. They also allow localization to be improved using the road network map, by restricting the position and heading to be consistent with the road segment and the orientation of that segment. This process is typically known as “map matching” or “road matching.”

The main question being investigated in this chapter is the following: How does the addition of vision affect inference of these discrete parameters? Does the detection of visual features make the choice of road segment and/or orientation

more obvious? Does it improve robustness when faced with problems such as drift in GPS position estimates?

Two data association approaches are investigated in this chapter:

1. Multiple Hypothesis Tracking (MHT) with Kalman filtering; a number of data association hypotheses are carried through to address the ambiguity problem. Each hypothesis has its own state vector and covariance updated by Kalman filtering. Managing the set of hypotheses is the major challenge of this method.
2. Monte Carlo Data Association with particle filtering; each particle represents a possible state of the vehicle with its own set of parameters. The resampling step removes unlikely particles, and as such particles with the most plausible parameters should survive. Adequately covering the sample space, and maintaining reasonable computational complexity are challenges.

5.1 Multiple-Hypothesis Tracking using Kalman Filtering

5.1.1 Introduction

The first approach investigated is Multiple Hypothesis Tracking (MHT) based on the Kalman filter. MHT maintains multiple data association hypotheses and updates each of them independently over time using Kalman filtering. The motivation behind MHT is that there is insufficient information available to choose the correct hypothesis, so the decision is deferred until a later time when enough information becomes available. MHT is often labeled a “deferred logic” method for this reason [73]. A weight assigned to each hypothesis indicates the relative importance of each hypothesis. Those hypotheses with very low weights are considered implausible data associations, and are discarded in favor of the higher-weight hypotheses.

A decision tree is often used to display the evolution of hypotheses over time in MHT (see for example [6, 29]). A single hypotheses at time instant t may result in two probable hypotheses at $t + 1$, each of which may result in two more hypotheses at $t + 2$. As enough sensory information is gathered over time, the hope is that one hypothesis becomes obvious and all others are discarded.

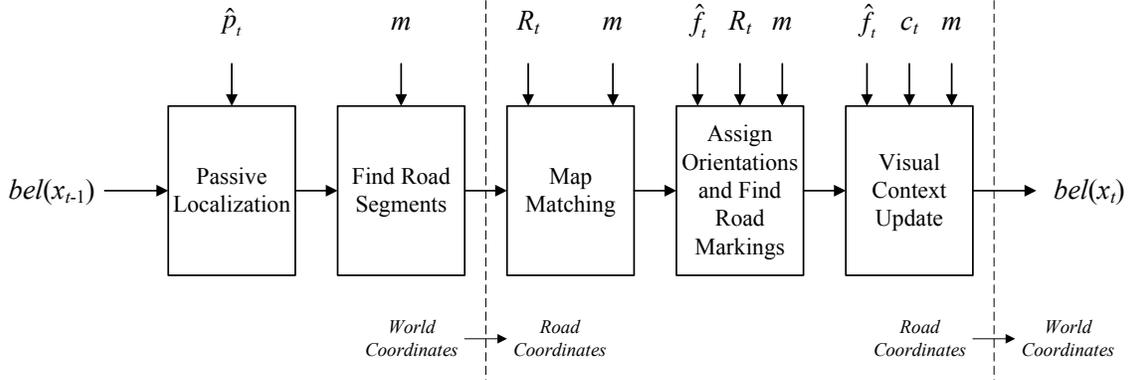


Figure 5.1: A block diagram for Kalman filter localization with Multiple Hypothesis Tracking.

It is interesting to note that the Maximum Likelihood (ML) approach to data association can be considered a special case of MHT. Both MHT and ML choose N_t hypotheses at time t ; however with ML, $N_t = 1$ for all t . This can be illustrated in a trivial decision tree with a branching factor of 1. The analogy between data association and tree search is illustrated in Figure 2.3 of Chapter 2.

The motivation for using MHT is to deal with ambiguous data association situations while maintaining a Kalman filtering framework. The latter is desirable because it is a simple and efficient localization method that was found to be successful in integrating visual context in the known data association case in Chapter 4. Therefore, an MHT approach will build upon this previous work. A block diagram for the proposed MHT method is shown in Figure 5.1.

MHT allows the state of the system to be modeled using Gaussian distributions as in standard Kalman filtering, with the caveat that multiple possibilities are considered for the true state. Each such possibility is a state estimate dependent on a unique set of discrete parameters, representing one data association hypothesis. The full posterior is modeled as a mixture of Gaussians (5.1) rather than a single Gaussian as in the standard KF.

$$bel(x_t) = \sum_i w_t^i bel_i(x_t) \quad (5.1)$$

This mixture of Gaussians representation can be derived from the Bayesian filter posterior by considering the full posterior $bel(x_t)$ to be a weighted summation over all data association hypotheses (5.2). $bel_i(x_t)$ represents the posterior of a single

hypothesis $i \in [1, N_t]$ conditioned on the series of data association variables $c_{1:t}^i$. Considering all $bel_i(x_t)$ to be Gaussian, (5.2) represents a mixture of Gaussians formulation.

$$\begin{aligned}
 bel(x_t) &= p(x_t | z_{1:t}, u_{1:t}, m) \\
 &= \sum_{i=1}^{N_t} p(x_t, c_{1:t}^i | z_{1:t}, u_{1:t}, m) \\
 &= \sum_{i=1}^{N_t} p(c_{1:t}^i | z_{1:t}, u_{1:t}, m) p(x_t | z_{1:t}, u_{1:t}, m, c_{1:t}^i) \\
 &= \sum_{i=1}^{N_t} w_t^i bel_i(x_t)
 \end{aligned} \tag{5.2}$$

The weight w_t^i assigned to each hypothesis i is updated at each time instant using the measurement likelihood probability and the weight of the parent hypothesis $j \in [1, N_{t-1}]$ at the previous time step (5.3). The measurement likelihood can be found by introducing the state variable x_t and then marginalizing over it (5.4) [67].

$$w_t^i = w_{t-1}^j p(z_t | z_{1:t-1}, c_t^i, c_{1:t-1}^j, u_{1:t}, m) \tag{5.3}$$

$$\begin{aligned}
 p(z_t | z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) &= \int p(x_t, z_t | z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) dx_t \\
 &= \int p(z_t | x_t, c_t^i, m) \overline{bel}_i(x_t) dx_t
 \end{aligned} \tag{5.4}$$

When the measurement probability $p(z_t | x_t, c_t^i, m)$ and predicted belief $\overline{bel}_i(x_t)$ are Gaussian, the resulting likelihood is also a Gaussian (5.5) whose mean is the predicted measurement and whose covariance is the innovation covariance matrix [67], used to calculate the Kalman gain.

$$p(z_t | z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) = \mathcal{N}(z_t; h_t(x_{t|t-1}, c_t^i, m), S_t^i) \tag{5.5}$$

In keeping with the modular approach used to update the vehicle state with the various measurements, the hypothesis weights can be updated as well at each measurement update step. This follows from the factorization of the measurement

likelihood in (5.6).

$$\begin{aligned}
 p(z_t | z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) &= p(\hat{p}_t | z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) \\
 &\quad p(\zeta_t^i | \hat{p}_t, z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) \\
 &\quad p(\hat{f}_t | \zeta_t^i, \hat{p}_t, z_{1:t-1}, c_{1:t}^i, u_{1:t}, m)
 \end{aligned} \tag{5.6}$$

The following section discusses the specifics of the MHT algorithm implemented to perform localization in the unknown data association case. Hypothesis initialization, termination and merging are discussed, and comparisons are made with other MHT localization approaches, mostly within the realm of robotics.

5.1.2 MHT Algorithm Description

MHT as described above is suboptimal in the sense that it discards hypotheses with low weight. An optimal MHT filter that maintained all hypotheses would be intractable, however, because the number of hypotheses never decreases but instead grows exponentially over time. This concept can be illustrated in the decision tree in Figure 5.2a. Each node of the tree represents a separate hypothesis, and as such is shown as a Gaussian. The height of the Gaussian is proportional to its weight in the mixture. Over time, the weight of certain hypotheses becomes negligible, however they are still maintained for optimality. The cost of optimality is thus an increase in memory required to store the hypotheses at each step, and an increase in computation to process each hypothesis.

Making MHT tractable involves removing hypotheses that are incorrect; the policy whereby hypotheses are managed thus defines the MHT implementation. Often this involves determining when to add a new hypothesis and when a hypothesis becomes sufficiently unlikely that it can be pruned. A further step is also taken to recognize when there are redundant hypotheses and to merge them together. The decision tree may have converging nodes as illustrated in Figure 5.2b. This hypothesis management policy is now described.

Each iteration of the localization filter requires the generation of new data association hypotheses. For each hypothesis $j \in [1, N_{t-1}]$ from iteration $t - 1$ (that is, for each “parent hypothesis”) plausible data associations for iteration t are determined (“child hypotheses”). The following describes how these child hypotheses are generated.

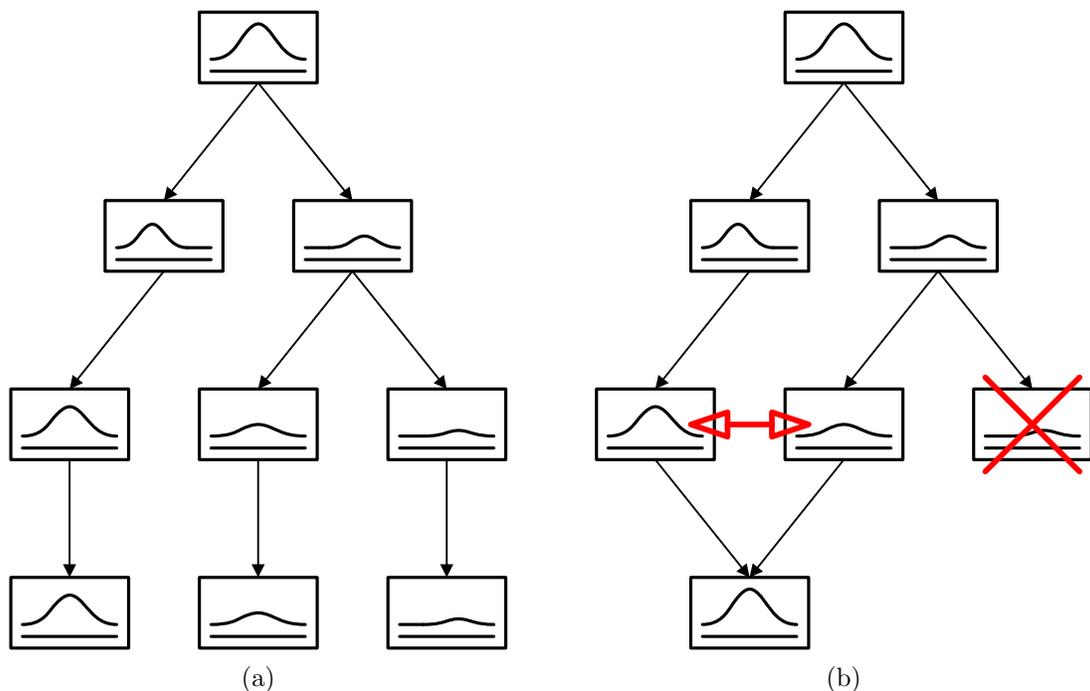


Figure 5.2: Decision trees for a) the optimal MHT implementation, and b) the proposed MHT implementation.

Passive Localization Update

As the GPS position measurement \hat{p}_t does not depend on any data association, it can be incorporated into each parent hypothesis prior to searching for new associations. This has the advantage of contracting the PDF of each hypothesis j , making it more obvious where the vehicle is located and therefore what are the possible data associations. This is simply the straightforward application of Passive Localization described in Section 4.2.1 to each parent hypothesis j . The resulting state estimate and covariance matrix are denoted $x_{t,PL}^j$ and $P_{t,PL}^j$, respectively.

The weight of each hypothesis is updated using the measurement likelihood as described in (5.5). The contribution $\lambda_{t,PL}^j$ made by the GPS measurement to the hypothesis weight in (5.6) is given by (5.7). Hypotheses closer to the position measurement \hat{p}_t are rewarded with a higher weight. The measurement model matrix H_t is defined in (4.23), while $S_{t,PL}^j$ is the innovation covariance matrix. η is the normalization term.

$$\begin{aligned} \lambda_{t,PL}^j &= p(\hat{p}_t | z_{1:t-1}, c_{1:t}^j, u_{1:t}, m) \\ &= \eta \exp \left[-\frac{1}{2} (\hat{p}_t - H_t x_{t|t-1}^j)^T S_{t,PL}^{j-1} (\hat{p}_t - H_t x_{t|t-1}^j) \right] \end{aligned} \quad (5.7)$$

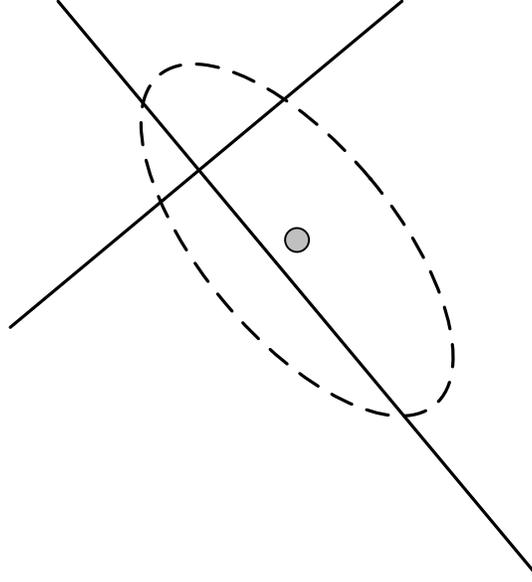


Figure 5.3: A validation region in used to find possible road segments. This region is defined as the 2σ contour around the vehicle pose estimate.

Identifying Candidate Road Segments

The first step in identifying possible associations is to search the area surrounding the newly-updated state estimate for nearby road segments. This involves a search of the road network database to find those near to the state estimate. A validation region around the state estimate and topology constraints of the road network are used to determine what road segments are plausible locations for the vehicle. In order for a road segment r to be accepted, it must lie within the 2σ error ellipse surrounding the state estimate $x_{t,PL}^j$ and defined by the state covariance matrix $P_{t,PL}^j$, as illustrated in Figure 5.3. Furthermore, this road segment must be connected to the previous road segment r_{t-1}^j . Road segments meeting these criteria form the set of child segments R_t^j for hypothesis j . Hypothesis j is then split into $|R_t^j|$ copies, and each copy assigned one road segment from R_t^j . In the event that $R_t^j = \emptyset$, it is assumed that no road segment is near the vehicle; the hypothesis j is maintained, but will not be updated by map matching or visual context. This process is repeated for all N_{t-1} parent hypotheses.

Algorithm 5.1 Pseudocode for finding nearby road segments.

```

1. function FindPossibleRoadSegments( x, P )
2.
3.   R = ∅
4.   for all road segments r in the database, do
5.     d = PointToRoadDistance( x, r )
6.     D = dP-1dT // Normalize distance by P
7.
8.     if D < 2
9.       r → R
10.
11.   end for
12. return R

```

Map Matching Update

Recall from Chapter 4 that a road-based coordinate system is used for map matching and adding visual context. As the possible road segments that the vehicle may be driving on have been determined, each child hypothesis i of the N_t generated by the above procedure is now converted to the road coordinate system for its assumed road segment r_t^i . The state vector and covariance matrix in road coordinates are denoted $x_{t,PL}^{r,i}$ and $P_{t,PL}^{r,i}$.

Map matching is now used to update each of the N_t vehicle state hypotheses, as described in Section 4.2.2. The pseudomeasurement ζ_t^i used to perform map matching must be defined for every hypothesis individually, to account for the vehicle state $x_{t,PL}^{r,i}$ being inside or outside the bounds of the hypothesized road segment r_t^i . Each hypothesis therefore has its own measurement model M_t^i and measurement covariance matrix Z_t^i , defined by Algorithm 4.2.

The resulting state estimate and covariance matrix for hypothesis $i \in [1, N_t]$ are denoted $x_{t,MM}^{r,i}$ and $P_{t,MM}^{r,i}$, respectively. The weight of each hypothesis is updated using the map matching measurement likelihood $\lambda_{t,MM}^i$ given by (5.8), where $S_{t,MM}^i$ is the innovation covariance matrix of the map matching update. A high weight will be given to a hypothesis whose position estimate is close to the assumed road segment r_t^i – that is, it has a lateral position close to zero – and also whose velocity vector is aligned with the road – that is, it has a lateral velocity close to zero.

$$\begin{aligned}
\lambda_{t,MM}^i &= p(\zeta_t^i | \hat{p}_t, z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) \\
&= \eta \exp \left[-\frac{1}{2} (\zeta_t^i - M_t^i x_{t,PL}^{r,i})^T S_{t,MM}^{i-1} (\zeta_t^i - M_t^i x_{t,PL}^{r,i}) \right] \quad (5.8)
\end{aligned}$$

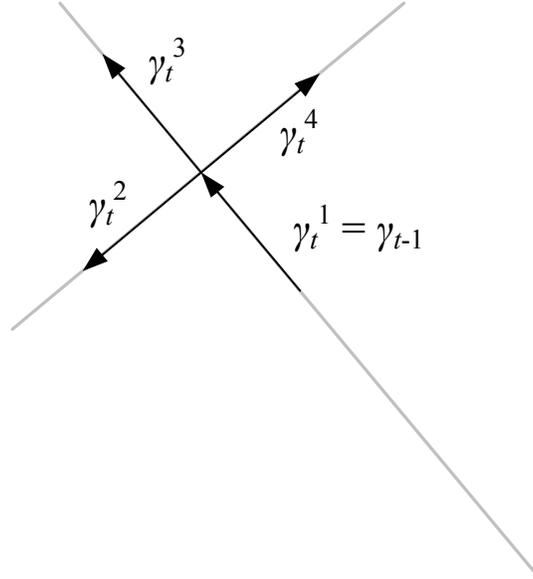


Figure 5.4: The topology of the road network is used to restrict the possible orientations on the road segment candidates.

Assigning Vehicle Orientations

The orientation of the vehicle on each road segment is determined by using constraints imposed by the road network topology. The data associations assigned to the parent $pa(i)$ of each hypothesis $i \in [1, N_t]$ are used in addition to a set of heuristic rules to assign a vehicle orientation to each hypothesis. These rules are as follows.

1. If a hypothesis has the same road segment association as its parent ($r_t^i = r_{t-1}^{pa(i)}$), then it also has the same orientation ($\gamma_t^i = \gamma_{t-1}^{pa(i)}$).
2. Otherwise, the orientation γ_t^i must point away from the intersection of r_t^i and $r_{t-1}^{pa(i)}$, as illustrated in Figure 5.4. In this way the topology of the road network can be used to restrict the possible vehicle orientations.

These rules implicitly make the assumption the vehicle does not make any U-turns, since that would result in a change of orientation on the same road segment. While this assumption is not realistic in the general case, it should be valid for the majority of driving situations. It also keeps the number of hypotheses low by not considering both possible orientations for each road segment, thereby helping to maintain tractability of the MHT approach.

Algorithm 5.2 Pseudocode for determining vehicle orientation.

```

1. function FindPossibleOrientations(  $R_t, R_{t-1}, \Gamma_{t-1}$  )
2.
3.  $R^* = \emptyset, \Gamma_t = \emptyset$ 
4. for  $i = 1: \{R_t\}$ 
5.   if  $R_t^i == R_{t-1}$ 
6.      $R_t^i \rightarrow R^*, \Gamma_{t-1} \rightarrow \Gamma_t$ 
7.   else if CommonEndpoint(  $R_t^i, R_{t-1}$  )
8.     Choose  $\Gamma$  to point away from the common point
9.      $R_t^i \rightarrow R^*, \Gamma \rightarrow \Gamma_t$ 
10.  else, this road segment  $R_t^i$  is rejected
11.
12. end for
13. return  $R^*, \Gamma_t$ 

```

Both possible orientations of the vehicle on a particular road segment r_t^i are only considered if $r_{t-1}^{pa(i)} = \emptyset$. This could occur because the filter is being initialized or no roads were found in the previous iteration.

Identifying Detected Landmarks

The identity of a detected landmark is determined by using a validation region on each candidate road segment. This validation region is defined by rearranging the landmark measurement model (4.17) to solve for the landmark position in road coordinates $p_{o,\parallel}^r$ (5.9). The expected longitudinal position of the landmark is given by (5.10) using the longitudinal vehicle position estimate $p_{t,\parallel}^{r,i}$, the distance measurement to the landmark \hat{f}_t and the assumed orientation γ_t^i . The uncertainty in this expected value is denoted $\sigma_{o,\parallel}$ and is given by (5.11), where it is assumed that \hat{f}_t and $p_{t,\parallel}^{r,i}$ are uncorrelated. The validation region is established as the 2σ interval $E[p_{o,\parallel}^r] \pm 2\sigma_{o,\parallel}$, as illustrated in Figure 5.5.

$$p_{o,\parallel}^r = \hat{f}_t \gamma_t + p_{t,\parallel}^r + \omega_t \quad (5.9)$$

$$E[p_{o,\parallel}^r] = \hat{f}_t \gamma_t + E[p_{t,\parallel}^r] \quad (5.10)$$

$$\begin{aligned} Var[p_{o,\parallel}^r] &= Var[\hat{f}_t] + Var[p_{t,\parallel}^r] \\ &= \sigma_f^2 + \sigma_{p,\parallel}^2 = \sigma_{o,\parallel}^2 \end{aligned} \quad (5.11)$$

The location of the intersection road marking occurring at the endpoint of the assumed road segment r_t^i in the direction γ_t^i is compared with the validation region.

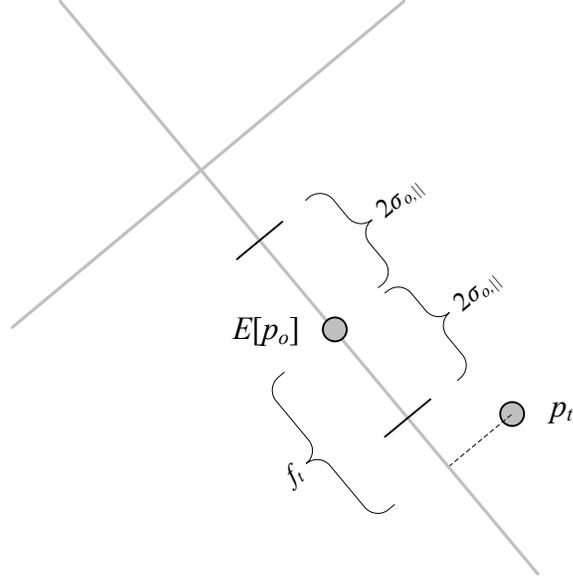


Figure 5.5: A validation region is used to identify a detected road marking.

The locations of these markings are determined using road construction standards defined by the Government of Ontario [45], as discussed in Section 4.1.2. At most one landmark will be found within the defined validation region, unless this region is sufficiently large (or the length of the road segment r_t^i is sufficiently small) to encompass the entire road segment.

If a landmark is found within the validation region, its longitudinal position $p_{o,\parallel}^r$ is used to update the state estimate for that hypothesis as described in Section 4.2.3. The weight of the hypothesis is updated using the visual context measurement likelihood $\lambda_{t,VC}^i$ given by (5.12), where $S_{t,VC}^i$ is the innovation covariance matrix of the visual context update. A high weight will be given to those hypotheses whose relative position to the identified landmark is similar to the estimated distance \hat{f}_t .

$$\begin{aligned} \lambda_{t,VC}^i &= p(\hat{f}_t | \zeta_t^i, \hat{p}_t, z_{1:t-1}, c_{1:t}^i, u_{1:t}, m) \\ &= \eta \exp \left[-\frac{1}{2} \left(\hat{f}_t - F_t(x_{t,MM}^{r,i}, c_t^i) \right)^T S_{t,VC}^i{}^{-1} \left(\hat{f}_t - F_t(x_{t,MM}^{r,i}, c_t^i) \right) \right] \end{aligned} \quad (5.12)$$

If a landmark is not found for a given hypothesis, then that hypothesis cannot be updated using visual context. A disparity in the weight assigned to the various hypotheses can therefore develop, as those without valid landmarks will not receive a contribution from the visual measurement \hat{f}_t . Balancing this weight disparity is addressed next.

Algorithm 5.3 Pseudocode for finding landmarks.

```

1. function FindPossibleLandmarks( R,  $\Gamma$ , z,  $\sigma_z$ , x, P )
2.
3.    $\Lambda = \emptyset$ 
4.    $\forall r \in R, \gamma \in \Gamma$ 
5.     [ $p_r, \sigma_r$ ] = WorldToRoadCoords( x, P, r )
6.     [ $p_{\lambda, \max}, p_{\lambda, \min}$ ] =  $p_r + z\gamma \pm 2(\sigma_r + \gamma)$ 
7.     [ $\lambda_r, p_{\lambda, r}$ ] = GetIntersectionLandmark( r,  $\gamma$  )
8.     if  $p_{\lambda, \min} < p_{\lambda, r} < p_{\lambda, \max}$ 
9.        $\lambda_r \rightarrow \Lambda$ 
10.    else  $0 \rightarrow \Lambda$ 
11.  end  $\forall$ 
12.
13. return  $\Lambda$ 

```

The validation gating and road network topology constraints used in generating data association hypotheses are such that only one hypothesis contains a valid landmark in most situations. Furthermore, this hypothesis will correspond to the road segment upon which the vehicle was driving when it approached the intersection. Thus, detecting a road marking landmark is an indication that the vehicle is still on that road segment, and has not turned onto a different road.

Balancing the Weight Disparity

It is often the case with the procedure just described for generating data association hypotheses that a valid landmark is identified found for only a subset of hypotheses. Because the location of a landmark is required to update the localization estimate using visual context (according to the measurement model (4.17)) those hypotheses without landmarks do not receive this update. It may appear at first that these hypotheses would be at a disadvantage without the extra update step to correct their localization estimates and contract their PDFs. However, the weight adjustment given by the measurement likelihood (5.12) lowers the weight assigned to those hypotheses that receive the update. This means that these hypotheses contribute less to the overall weight, and are therefore prone to pruning.

To balance the weight disparity, an extra term is appended to the hypothesis weights whenever a visual landmark detection is made. If a valid landmark cannot be identified for a given hypothesis i , it is assumed that the detection is a false positive. Thus, the weight is multiplied by the false positive rate α of the landmark detector, as shown in (5.13), where the subscript ‘ $\neg o$ ’ is used to indicate that no

object hypothesis o_t^i was found. Conversely, if a valid landmark is identified, it is assumed that the detection is successful. Thus, the weight is multiplied by the rate of “true positives” $1 - \beta$ of the landmark detector as shown in (5.14), where β is the false negative rate. The subscript ‘ o ’ is used to indicate that an object hypothesis o_t^i was found.

$$\lambda_{t,\neg o}^i = \lambda_{t,gps}^i \lambda_{t,map}^i \alpha \quad (5.13)$$

$$\lambda_{t,o}^i = \lambda_{t,gps}^i \lambda_{t,map}^i \lambda_{t,vis}^i (1 - \beta) \quad (5.14)$$

A well-designed landmark detector will have a low false positive rate and high sensitivity ($\alpha \ll 1 - \beta$), meaning that the vast majority of detections occur when a landmark is present. If this is the case, this may be sufficient to compensate for the weight disparity introduced by the visual likelihood $\lambda_{t,vis}^i$.

The MHT results presented in this thesis use values for α and β determined by processing a set of images taken by the vehicle mounted camera with the road marking detector. In actuality, these parameters do not need to be a function of how well or how poorly the landmark detector performs. They can instead be tuned to create the best filter performance provided that the weight disparity is addressed. Using α and β is simply one way to achieve this, and represents a way of relating the performance of the vision algorithm to the localization result.

Merging Redundant Hypotheses

It often arises that multiple hypotheses exist with identical associations and (nearly) identical state estimates. Often this is the result of multiple hypotheses with distinct associations at one time instant t producing children with identical associations at the next time instant $t+1$. A possible situation is shown in Figure 5.6. If all of these hypotheses have significant weight, none will be pruned and they will likely converge to the same state estimate over time. All but one of such hypotheses are redundant, needlessly consume computational resources, and create an unrealistic distribution of weights among hypotheses.

It is therefore proposed to first determine when two or more hypotheses are redundant, and then to merge them together into a single hypothesis whose weight is the sum of its contributors’ weights. As opposed to discarding one of the hypotheses, merging them together allows each redundant hypothesis to contribute to the outcome in a manner consistent with its weight in the overall mixture.

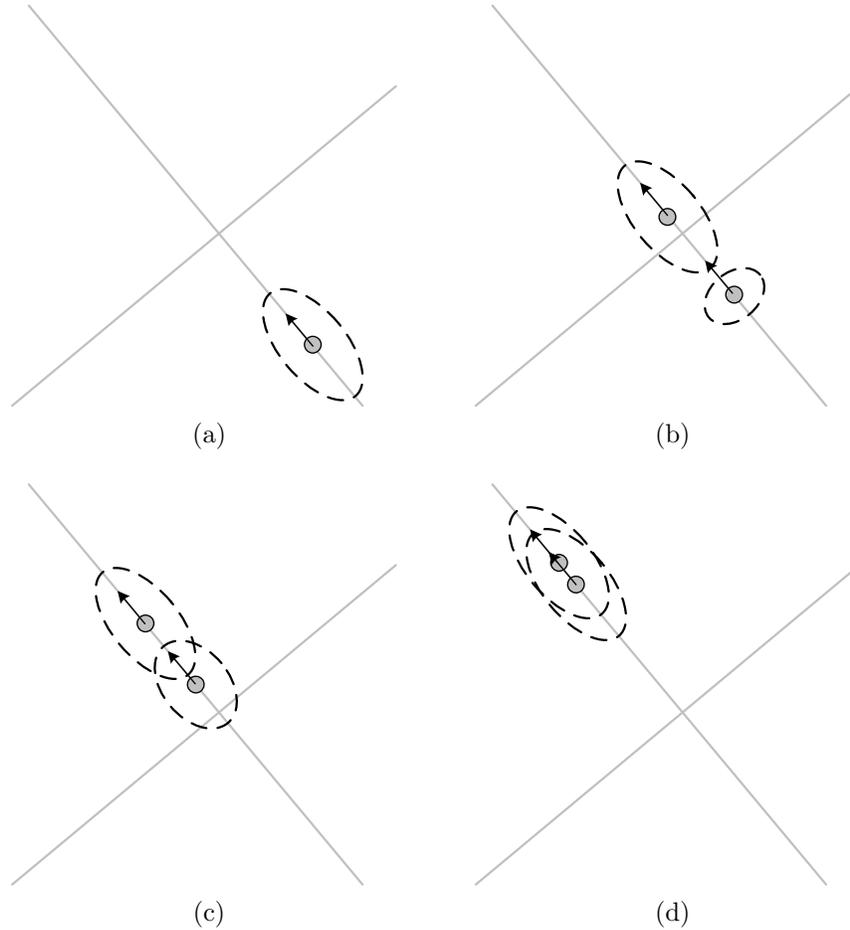


Figure 5.6: An illustration of data association ambiguity resulting in redundant hypotheses. Hypotheses with unique data associations (in b) can have children with identical data associations (in c) that converge to similar vehicle pose estimates (in d)

The redundancy of two hypotheses is determined by two conditions:

1. Both hypotheses have the same set of associations (road segment, orientation, and detected road marking).
2. Their distributions are sufficiently similar to one another, as determined by using the Kullback-Liebler divergence.

KL divergence measures the difference between two distributions in a directed (non-symmetric) fashion. It is defined as in (5.15). If the divergence is below a predefined threshold, $D(x_i||x_j) < \epsilon$, the hypotheses are considered sufficiently

Algorithm 5.4 Pseudocode for merging hypotheses.

```

1. function MergeHypotheses(  $\mathbf{x}_{t-1}^{1:M}, \mathbf{P}_{t-1}^{1:M}, \mathbf{c}_{t-1}^{1:M}, \mathbf{w}_{t-1}^{1:M}$  )
2.
3.  $\mathbf{x}_t^* = \mathbf{P}_t^* = \mathbf{c}_t^* = \mathbf{w}_t^* = \emptyset$ 
4.  $\mathbf{c}_{\text{unique}} = \text{FindUniqueCorrespondences}( \mathbf{c}_t^{1:M} )$ 
5.  $\forall \mathbf{c} \in \mathbf{c}_{\text{unique}}$ 
6.    $[\mathbf{x}_t^c, \mathbf{P}_t^c, \mathbf{w}_t^c] = \{\text{states and covariances with associations } \mathbf{c}\}$ 
7.
8.    $[\mathbf{x}_t^k, \mathbf{P}_t^k, \mathbf{w}_t^k] = \text{KLDivergenceTest}( \mathbf{x}_t^c, \mathbf{P}_t^c, \mathbf{w}_t^c )$ 
9.
10.  // Add failed hypotheses to the return variables.
11.   $(\mathbf{x}_t^c - \mathbf{x}_t^k) \rightarrow \mathbf{x}_t^*, (\mathbf{P}_t^c - \mathbf{P}_t^k) \rightarrow \mathbf{P}_t^*, \mathbf{c} \rightarrow \mathbf{c}_t^*, (\mathbf{w}_t^c - \mathbf{w}_t^k) \rightarrow \mathbf{w}_t^*$ 
12.
13.   $\mathbf{w} = \sum_i \mathbf{w}_t^{k,i}$ 
14.   $\mathbf{x} = \sum_i \mathbf{w}_t^{k,i} \mathbf{x}_t^{k,i}$ 
15.   $\mathbf{P} = \sum_i \mathbf{w}_t^{k,i} \mathbf{P}_t^{k,i} + \sum_i \sum_{j, j \neq i} \mathbf{w}_t^{k,i} \mathbf{w}_t^{k,j} \mathbf{x}_t^{k,i} (\mathbf{x}_t^{k,i} - \mathbf{x}_t^{k,j})^T$ 
16.
17.  // Add merged hypothesis to the return variables.
18.   $\mathbf{x} \rightarrow \mathbf{x}_t^*, \mathbf{P} \rightarrow \mathbf{P}_t^*, \mathbf{c} \rightarrow \mathbf{c}_t^*, \mathbf{w} \rightarrow \mathbf{w}_t^*$ 
19. end  $\forall$ 
20. return  $\mathbf{x}_t^*, \mathbf{P}_t^*, \mathbf{c}_t^*, \mathbf{w}_t^*$ 

```

similar to be redundant.

$$D(x_i || x_j) = \int_{-\infty}^{\infty} f_i(x) \log \frac{f_i(x)}{f_j(x)} dx \quad (5.15)$$

The probability distribution of a random variable x_t is modeled as the weighted sum of M Gaussian distributions, which will be denoted $\{f_i(x_t)\}_{i=1}^M$. It is desired to determine the mean μ and covariance matrix Σ of this weighted mixture, which would be sufficient to define a single Gaussian approximation of this mixture. The calculation of these quantities is shown in (5.17) and (5.18), where $W_t = \sum w_t^i$.

$$p(x_t) = \sum_{i=1}^M \frac{w_t^i}{W_t} f_i(x_t) \quad (5.16)$$

$$E[p(x_t)] = \sum_{i=1}^M \frac{w_t^i}{W_t} E[f_i(x_t)] = \sum_{i=1}^M \frac{w_t^i}{W_t} \mu_i \quad (5.17)$$

$$\text{Var}[p(x_t)] = \sum_{i=1}^M \frac{w_t^i}{W_t} \text{Var}[f_i(x_t)] + \sum_{\substack{j=1 \\ j \neq i}}^M \frac{w_t^i w_t^j}{W_t^2} \mu_i (\mu_i - \mu_j)^T \quad (5.18)$$

Pruning Unlikely Hypotheses

The weights for all newly-generated hypotheses are normalized to determine the relative contributions of each of the hypotheses to the overall Gaussian mixture (5.19).

$$w_t^i = \frac{w_t^i}{\sum_k w_t^k} \quad (5.19)$$

Hypotheses that contribute weakly to the overall mixture should be pruned in order to maintain tractability of the MHT filter. A minimum weight threshold w_{min} is used when deciding which hypotheses to keep. This threshold is chosen to be quite small (on the order of 0.001) so as to be conservative in pruning hypotheses. In principle, the number of hypotheses that can exist at a given time is $\frac{1}{w_{min}}$, which would be 1000 for $w_{min} = 0.001$. In practice, the number of hypotheses is limited by the number of road segments in the vicinity, the use of topology constraints and the merging of redundant hypotheses.

A pseudocode description of the MHT approach is given in Algorithm 5.5.

Comparison with Selected Previous Approaches

Austin [6] uses MHT for mobile robot localization. Each robot pose hypothesis is represented using a Gaussian, creating a mixture of Gaussians to approximate the probability distribution of robot pose over the entire pose space. Updating the pose estimates is accomplished using the covariance intersection method of Uhlmann [69], rather than Kalman filtering. In terms of data association, a null hypothesis with a uniform distribution is used to represent the case of sensor failure creating complete ignorance of the robot pose. A novel weight redistribution strategy is implemented as a result of this null hypothesis in order to maintain a constant probability of sensor failure.

Jensfelt [32] uses track splitting and merging to avoid making false associations. Prior to updating a hypothesis, it is split into two identical copies with equal weight. One of these is updated while the other is not. If after updating the two copies are not significantly far apart (based on a Mahalanobis distance criterion), they are merged back together by summing their weights and retaining only the updated copy.

Pyo [51] applies MHT directly to the problem of matching GPS measurements to a map of the roads (map matching). Their approach has many similarities to the

Algorithm 5.5 Pseudocode for MHT algorithm.

```

1. function MultiHypothesisTracking(  $x_{t-1}^{1:M}$ ,  $P_{t-1}^{1:M}$ ,  $c_{t-1}^{1:M}$ ,  $w_{t-1}^{1:M}$  )
2.
3.   for i = 1:M
4.     // State prediction and GPS update
5.      $[x_{t|t-1}^i, P_{t|t-1}^i] = \text{KFPredict}( x_{t-1}^i, P_{t-1}^i )$ 
6.      $[x_{t,\text{gps}}^i, P_{t,\text{gps}}^i] = \text{KFGPSUpdate}( p_t, x_{t|t-1}^i, P_{t|t-1}^i )$ 
7.      $w_{t,\text{gps}}^i = \text{GPSWeightUpdate}( w_{t-1}^{1:M}, p_t, x_{t|t-1}^i, P_{t|t-1}^i )$ 
8.
9.     // Choosing possible correspondences
10.     $R_t^i = \text{FindPossibleRoadSegments}( x_{t,\text{gps}}^i, P_{t,\text{gps}}^i )$ 
11.     $\Gamma_t^i = \text{FindPossibleOrientations}( R_t^i, R_{t-1}^i, \Gamma_{t-1}^i )$ 
12.     $\Lambda_t^i = \text{FindPossibleLandmarks}( R_t^i, \Gamma_t^i, x_{t,\text{gps}}^i, P_{t,\text{gps}}^i )$ 
13.
14.    if  $\{R_t^i\} > 0$ 
15.
16.
17.      // Map update
18.       $[x_{t,\text{map}}^i, P_{t,\text{map}}^i] = \text{KFMapUpdate}( R_t^i, \Gamma_t^i, x_{t,\text{gps}}^i, P_{t,\text{gps}}^i )$ 
19.       $w_{t,\text{map}}^i = \text{MapWeightUpdate}( w_{t,\text{gps}}^i, R_t^i, \Gamma_t^i, x_{t,\text{gps}}^i, P_{t,\text{gps}}^i )$ 
20.
21.      // Vision update
22.       $[x_{t,\text{vis}}^i, P_{t,\text{vis}}^i] = \text{KFVisionUpdate}( \Lambda_t^i, R_t^i, \Gamma_t^i, x_{t,\text{map}}^i, P_{t,\text{map}}^i )$ 
23.       $w_{t,\text{vis}}^i = \text{VisionWeightUpdate}( w_{t,\text{map}}^i, \Lambda_t^i, R_t^i, \Gamma_t^i, x_{t,\text{map}}^i, P_{t,\text{map}}^i )$ 
24.
25.      // Final weight and state
26.       $w_t^i = w_{t,\text{vis}}^i, x_t^i = x_{t,\text{vis}}^i, P_t^i = P_{t,\text{vis}}^i$ 
27.    else
28.       $w_t^i = w_{t,\text{gps}}^i, x_t^i = x_{t,\text{gps}}^i, P_t^i = P_{t,\text{gps}}^i$ 
29.    end for
30.
31.   $[x_t^{1:M}, P_t^{1:M}, c_t^{1:M}, w_t^{1:M}] = \text{PruneHypotheses}( x_t^{1:M}, P_t^{1:M}, c_t^{1:M}, w_t^{1:M} )$ 
32.
33.   $[x_t^{1:M}, P_t^{1:M}, c_t^{1:M}, w_t^{1:M}] = \text{MergeHypotheses}( x_t^{1:M}, P_t^{1:M}, c_t^{1:M}, w_t^{1:M} )$ 
34.
35.  return  $x_t^{1:M}, P_t^{1:M}, c_t^{1:M}, w_t^{1:M}$ 

```

proposed MHT method in terms of integrating map information into the state estimate. First, potential roads are found within an elliptical validation region around the position measurement. Second, for each road segment a pseudo-measurement is created containing the measured position projected onto the road and the heading angle of the road. In addition, the road network topology is used to determine the likelihood of each hypothesis. Pruning of hypotheses is based on a number of pre-defined thresholds rather than a single threshold, which may make the algorithm brittle and difficult to tune.

5.1.3 MHT Performance in Unbiased GPS Conditions

This section presents results of using MHT for localization when data associations are unknown and the position measurements from the GPS are unbiased. Results are shown for two cases: one in which visual context is used and the other in which it is not. This should illustrate the benefit (if any) that visual context provides to data association in vehicle localization.

The results shown are for a trip where the vehicle comes to a stop at an intersection and is waiting at a traffic light. The path taken by the vehicle is shown in Figure 5.7. This trip allows us to show how data association performs in the trivial situation where there is only a single road segment nearby and in the more complicated situation with multiple nearby segments at an intersection.

Results are shown first for the case where vision is not used, and the data association is performed using only GPS and the map. Figure 5.8 plots the vehicle position estimate and its error ellipse on the map at particular time instants. The road and orientation are identified correctly in the trivial case where only one road is nearby (Figure 5.8a). Ambiguity in the data association occurs at the intersection and two competing hypotheses arise (Figure 5.8b). Eventually, the hypothesis closest to the measured GPS position survives (Figure 5.8d) because it receives consistently higher weight than the other at each iteration; this also happens to be the correct hypothesis. It takes 35 iterations of the filter for the filter to converge to a single hypothesis after the second is generated.

Figure 5.9 shows the localization result when vision is added to the system. Since vision is only detecting landmarks at intersections, the performance away from an intersection is the same (Figure 5.9a). A second hypotheses again arises when the vehicle is at the intersection (Figure 5.9b). However, a valid landmark

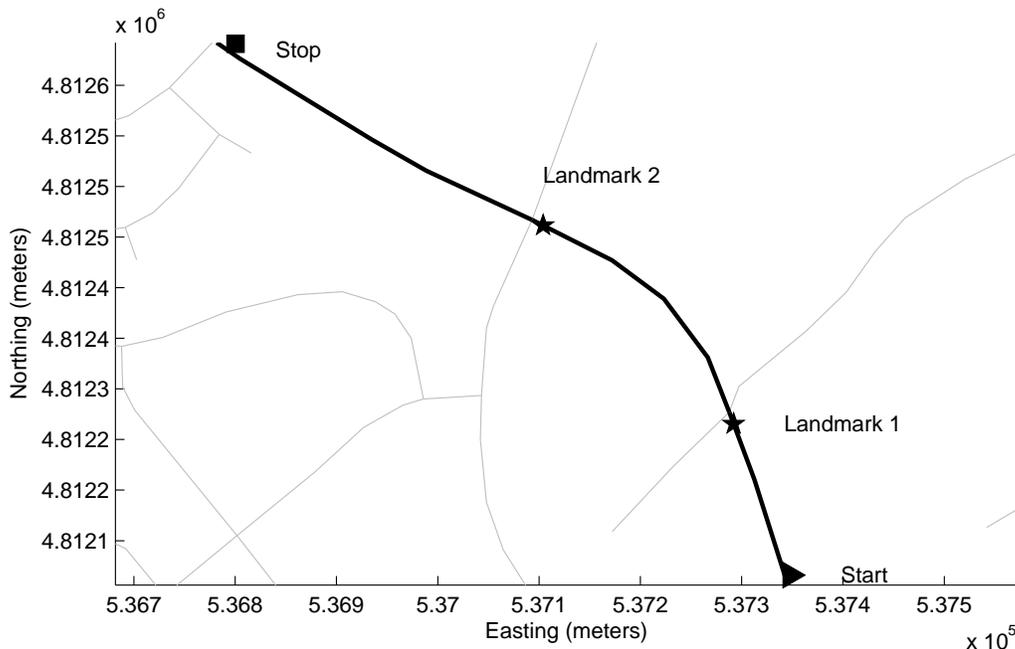


Figure 5.7: The vehicle path for the unbiased data set. Vision observes two intersection road markings; the second of these is visible for 60 seconds while the vehicle is stopped.

can only be found for one hypothesis, and the extra weight given to it causes the other to be pruned after only 6 iterations of the filter (Figure 5.9d).

The error in the MHT localization is shown in Figure 5.10. This is determined by calculating the weighted mean and covariance of all hypotheses using (5.17) and (5.18) respectively, and computing the error between the mean to the ground truth position. Lateral and longitudinal errors are determined relative to the ground truth road segment. Table 5.1 summarizes the localization error for the raw GPS measurements, and the MHT method both with and without visual context.

The longitudinal error, shown in Figure 5.10b, is consistently close to zero within the time interval between $t = 36$ sec and $t = 98$ sec during which the intersection landmark was being observed. Table 5.2 summarizes the longitudinal localization error within this time interval.

In addition to localization error, it is important to evaluate the data association success of the MHT approach. In this application it is particularly important to identify the road segment and the orientation of the vehicle, as these pieces of information are required for map matching and also enable a comparison between the vision and non-vision approaches to be made as both are common to the two

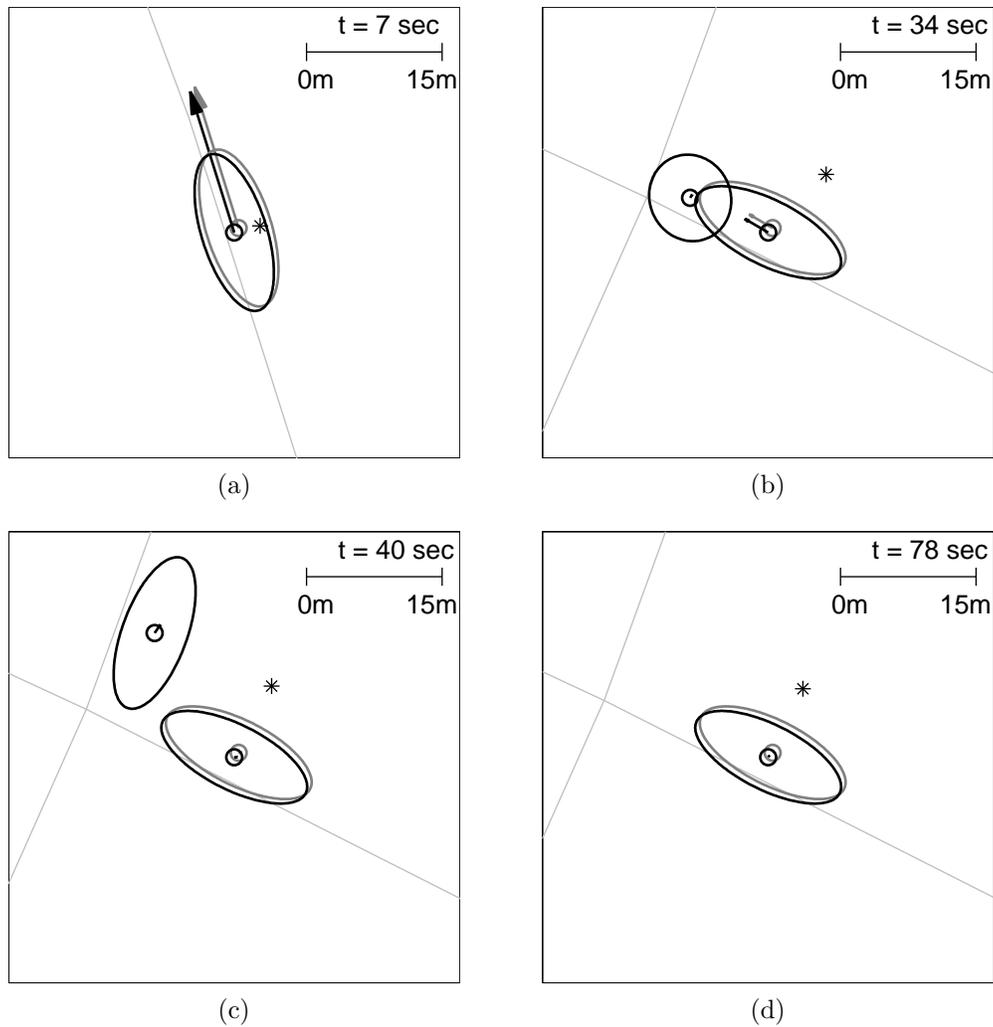


Figure 5.8: An illustration of MHT localization without visual context. The known data association result is shown for comparison in gray, and the GPS measurement is shown as ‘*’.

filter implementations. To evaluate the ability of the MHT to find the correct data association, two measures are used. First, the entropy of the hypothesis weights is calculated to determine how ambiguous the choice of hypothesis is at each time step (Figure 5.11a). A large entropy value will indicate that the weight is distributed rather evenly among many hypotheses, making it unclear which is correct. A low value will indicate that most of the weight is concentrated on one hypothesis. Second, the weight assigned to the hypothesis with the correct data association is examined to determine how successful the filter is at identifying the correct hypothesis. This is shown in Figure 5.11b for both the vision and no-vision situations. What can be observed from these results is that by incorporating vision, the choice

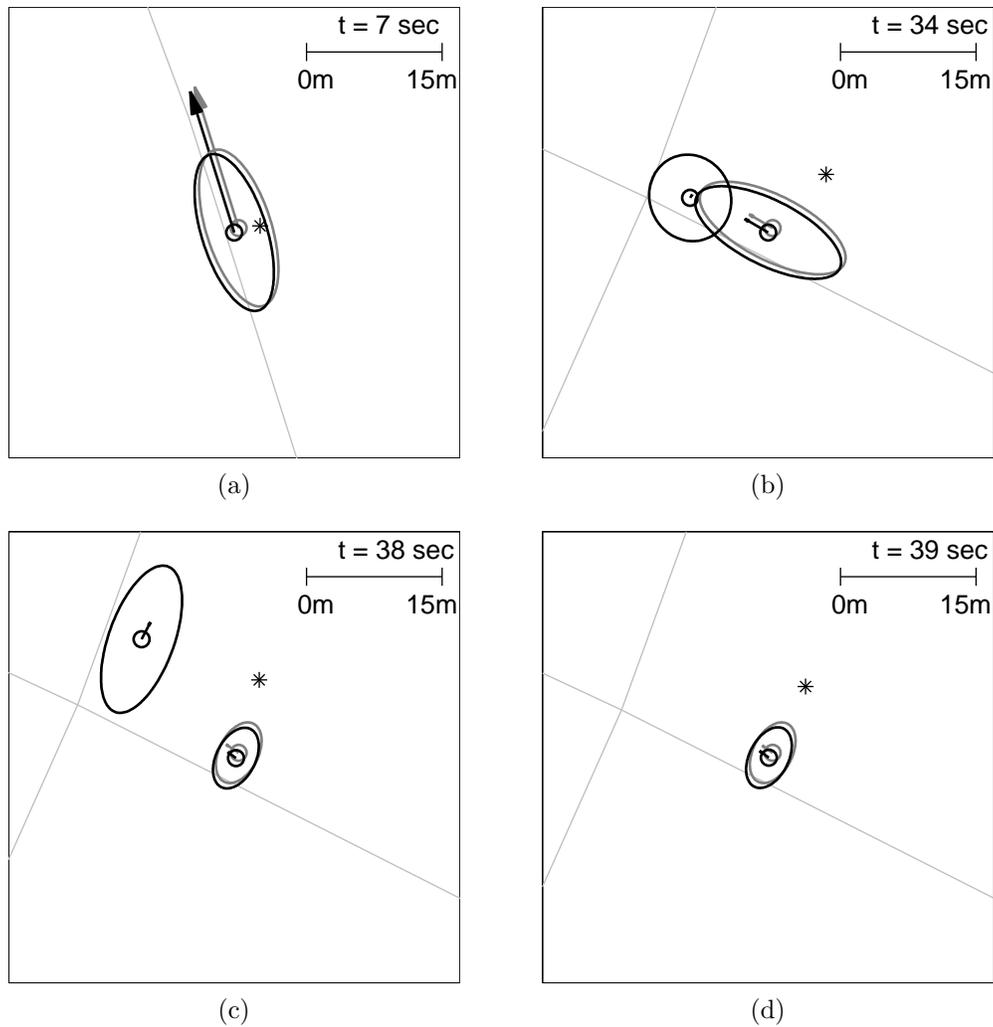
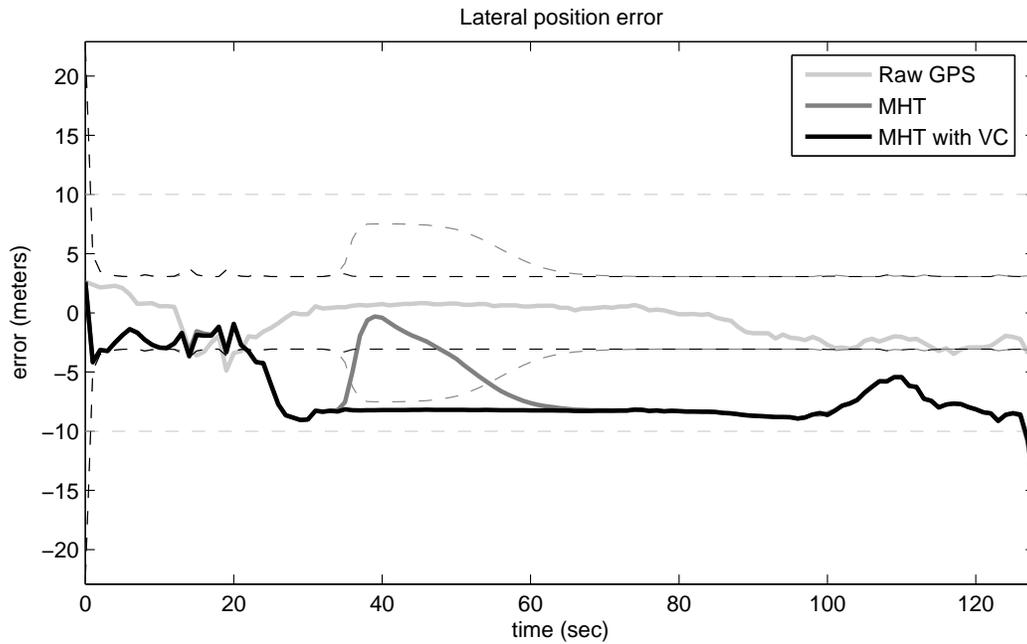


Figure 5.9: An illustration of MHT localization with visual context. The known data association result is shown for comparison in gray, and the GPS measurement is shown as ‘*’.

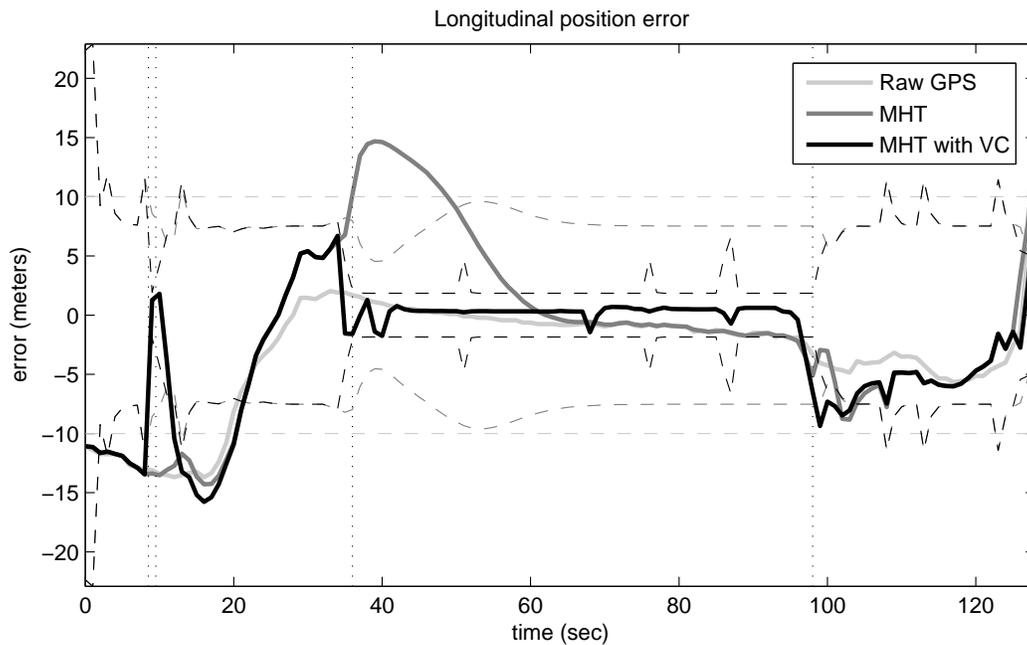
of data association becomes less ambiguous because more weight is given to one hypothesis than the others, and also that the weight in this case is assigned to the correct hypothesis.

Discussion

The results shown above indicate that the addition of vision to the system improves the ability to identify the road segment and orientation in an ambiguous data association condition, such as when stopped at an intersection. In addition, accurate localization is maintained with the use of vision despite this ambiguity. Vision accomplishes this by placing a high weight on successful landmark detections, thus

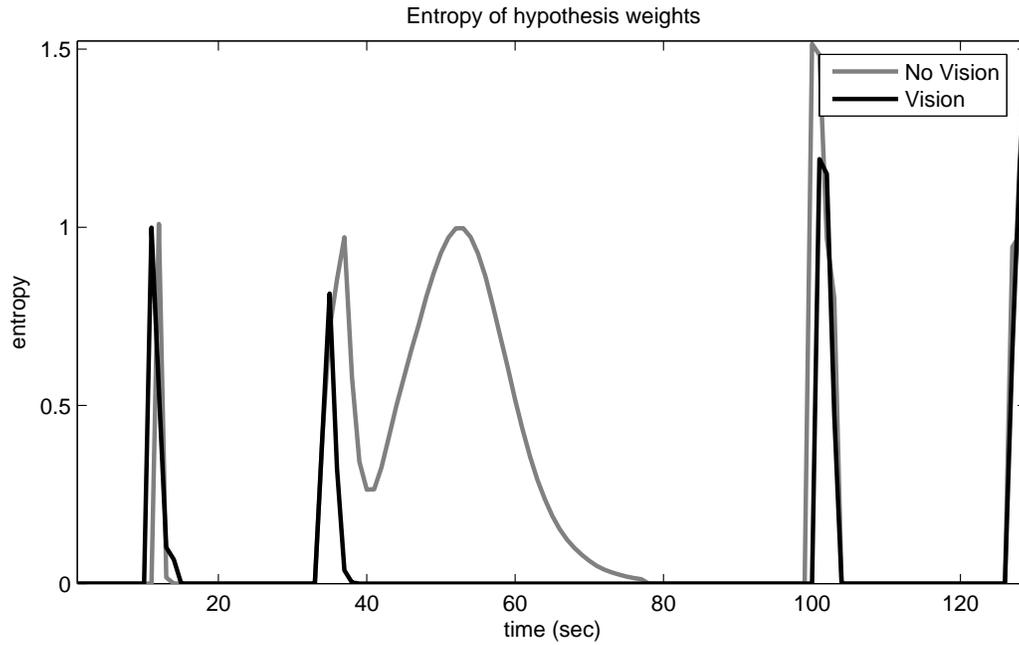


(a)

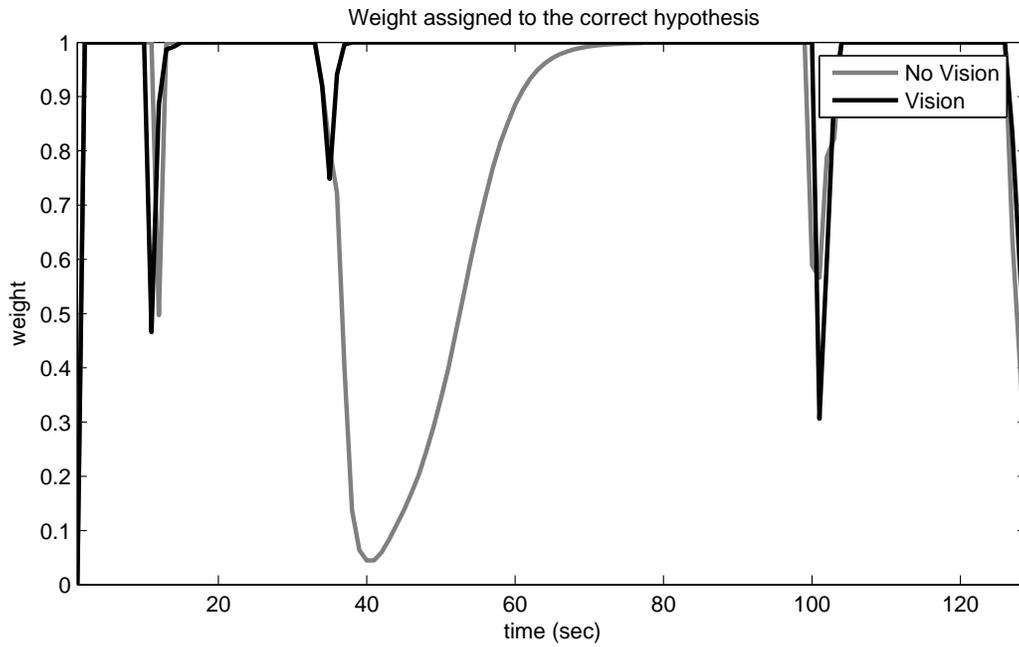


(b)

Figure 5.10: Average position error at each time instant using MHT localization. The error in the raw GPS position measurements are shown, with the average error in MHT localization with and without visual context. a) Error in the lateral direction. b) Error in the longitudinal direction; dotted lines show the beginning and ending of time intervals where road markings are visible.



(a)



(b)

Figure 5.11: A plot of a) the entropy among the hypothesis weights, and b) the weight assigned to the correct hypothesis, for MHT localization.

	$\bar{\varepsilon}$ (m)	s_{ε} (m)	$\bar{\sigma}$ (m)	s_{σ} (m)
$p_{\perp,GPS}$	-0.7	1.6	10	0
$p_{\parallel,GPS}$	-3.0	4.3	10	0
$p_{\perp,MHT}$	-6.2	3.0	2.0	1.1
$p_{\parallel,MHT}$	-1.4	7.5	4.0	1.1
$p_{\perp,MHT+VC}$	-7.1	2.6	1.6	0.85
$p_{\parallel,MHT+VC}$	-2.7	5.2	2.6	1.9

Table 5.1: Lateral and longitudinal position error and uncertainty for the raw GPS measurements, and for MHT with and without visual context (VC).

	$\bar{\varepsilon}$ (m)	s_{ε} (m)	$\bar{\sigma}$ (m)	s_{σ} (m)
$p_{\parallel,GPS}$	-0.90	1.1	10	0
$p_{\parallel,MHT}$	3.1	5.8	7.6	1.1
$p_{\parallel,MHT+VC}$	0.18	0.74	2.1	0.9

Table 5.2: Longitudinal position error and uncertainty for the raw GPS measurements, and MHT with and without visual context (VC) while a road marking is visible.

providing further evidence in favor of a road segment for which a valid landmark can be identified. In particular, continual observation of the same landmark offers repeated evidence for one data association hypothesis over the others. The mechanism by which this occurs is explored in the following.

It can be shown based on some reasonable assumptions that the MHT filter will converge to a single hypothesis, as occurs in the results illustrated in Figures 5.8 and 5.9. As is approximately the case with the investigated data set, I am assuming that the vehicle remains stationary and that there is minimal drift in GPS position measurement error during this time. Under these conditions, the state estimate for each hypothesis will converge to a steady state because the measurements at every iteration are always the same: the GPS error does not drift, the map pseudomeasurement is constant, and the vehicle does not move hence the estimated distance to a landmark is constant. As a result, the measurement likelihood assigned to each hypothesis will also be constant over time in these steady-state conditions.

Each hypothesis $i \in [1, N_t]$ has an initial weight w_0^i , which refers to its weight

when it reaches steady state. The measurement likelihood assigned to hypothesis i at time t is defined as $\lambda_t^i = \lambda_i$ (the time subscript can be dropped because this likelihood is constant). Hypothesis weights are also normalized after every iteration of the filter, making the weight of hypothesis i to be as shown in (5.20). Recursively substituting w_{t-1}^i back to w_0^i gives the expression (5.21).

$$w_t^i = \frac{w_{t-1}^i \lambda_i}{\sum_j w_{t-1}^j \lambda_j} \quad (5.20)$$

$$= \frac{w_0^i \lambda_i^t}{\sum_j w_0^j \lambda_j^t} \quad (5.21)$$

The limit of this expression as $t \rightarrow \infty$ illustrates that one hypothesis will accumulate all of the weight while the rest tend to zero, provided that one hypothesis consistently receives a higher measurement probability.

$$\begin{aligned} \lim_{t \rightarrow \infty} w_t^i &= \lim_{t \rightarrow \infty} \frac{w_0^i \lambda_i^t}{\sum_j w_0^j \lambda_j^t} \\ &= \lim_{t \rightarrow \infty} \frac{1}{1 + \sum_{j \neq i} \left(\frac{w_0^j}{w_0^i}\right) \left(\frac{\lambda_j}{\lambda_i}\right)^t} \\ &= \begin{cases} 0 & \text{if } \exists \lambda_j, \lambda_i < \lambda_j \\ 1 & \text{if } \forall \lambda_j, \lambda_i > \lambda_j \\ \frac{w_0^i}{w_0^i + \sum w_0^k} & \text{if } \exists \{\lambda_k\} \subseteq \{\lambda_j\}, \forall \lambda_k, \lambda_i = \lambda_k \end{cases} \quad (5.22) \end{aligned}$$

The number of iterations of the filter it would require to prune away all but the dominant hypothesis depends on the relative magnitude of the measurement likelihoods as well as the relative magnitude of the initial weights. The condition whereby only one hypothesis i would remain is $w_t^i > 1 - w_{min}$, where w_{min} is the pruning threshold. Using the expression for w_t^i in (5.22), and dropping the limit, this condition can be rearranged to (5.23).

$$\sum_{j \neq i} \left(\frac{w_0^j}{w_0^i}\right) \left(\frac{\lambda_j}{\lambda_i}\right)^t < \frac{w_{min}}{1 - w_{min}} \quad (5.23)$$

The number of iterations T required for condition (5.23) to be met (and therefore for all hypotheses but i to be pruned) is at least as many as the slowest converging sequence, that is the sequence with the ratio $\frac{\lambda_j}{\lambda_i}$ nearest to unity. This provides a lower bound on the number of filter iterations needed before pruning occurs, which

Filtering Method	Predicted time	Observed time
MHT with VC	39 sec	39 sec
MHT without VC	76 sec	78 sec

Table 5.3: Predicted and observed convergence times for MHT localization with and without using Visual Context.

can be found using (5.24).

$$T > \frac{\ln\left(\frac{w_{min}}{1-w_{min}}\right) - \ln\left(\frac{w_0^{j*}}{w_0^i}\right)}{\ln\left(\frac{\lambda_{j*}}{\lambda_i}\right)} \quad (5.24)$$

This lower bound was tested on the experimental results, using measurement likelihoods for GPS, map matching and vision. Predicted and observed convergence times for the MHT filters with and without Visual Context are summarized in Table 5.3. To perform this calculation the measurement likelihoods for both filters were observed until such time that they reached steady-state. The hypothesis weights at that iteration were then used along with the constant likelihoods to predict the time until convergence. For the filter using Visual Context, convergence was predicted at iteration $t = 39$ sec, which agrees with the observation (see Figure 5.9d). Without Visual Context, convergence was predicted at $t = 76$ sec, when it actually occurred at $t = 78$ sec (see Figure 5.8d). Thus, the lower bound on convergence time appears to hold for this data set, indicating that the assumptions of constant measurement likelihood are valid.

The above discussion explains why the MHT filter converges to a single hypothesis. This validates the use of MHT to perform localization in unknown data association conditions, because it is shown that, with enough evidence, the filter will converge to one hypothesis. However, it is important that it does not converge to an incorrect hypothesis, as this may create divergence of the localization estimate. In order for the correct hypothesis to be the one chosen, therefore, it must consistently receive higher weight than the alternatives.

As mentioned previously, the method used in generating data association hypotheses normally gives only one hypothesis with a valid landmark. In addition, this hypothesis corresponds to the road segment upon which the vehicle was driving when it approached the intersection. Thus, detecting the painted road marking

should provide evidence in favor of this hypothesis over the others.

In order for the filter to converge to the “landmark hypothesis,” that hypothesis must have a higher measurement likelihood than the others, according to (5.22). The measurement likelihood for this hypothesis is given by (5.14) because a landmark is successfully identified, while the remaining hypotheses have measurement likelihoods according to (5.13) because the detection is considered a false positive.

In keeping with assumptions made previously, consider that the vehicle is again stationary, and that there is no drift in the GPS position measurements. Further, consider the GPS measurement \hat{p}_t to be equidistant between two road segments. The result is that the two tracks being considered, one for each road segment, receive equal weight from the GPS measurement and from map matching. We will denote one of these by the subscript ‘ o ’ to denote the hypothesis with a landmark, and the other by ‘ $-o$ ’. Therefore, the condition whereby the filter will converge to the landmark hypothesis can be stated as in (5.25).

$$\lambda_o > \lambda_{-o} \quad (5.25)$$

As the measurement likelihoods for GPS and map matching are equivalent for the two hypotheses, the only difference between the two will be weight contributed by vision (5.26), and the parameters α and β .

$$\lambda_{VC,o}(1 - \beta) > \alpha \quad (5.26)$$

The need for these balancing parameters can now be demonstrated. Without the α and β terms, (5.26) would reduce to $\lambda_{VC,o} > 1$, which is not possible because $\lambda_{VC,o}$ is a probability density and therefore limited to $[0, 1]$. Therefore, the filter would always converge to the “non-landmark hypothesis” in the absence of these balancing terms.

This theory can help to explain the results observed. The higher measurement probability consistently applied to the landmark hypothesis results in a higher weight relative to the other hypotheses, as shown in Figure 5.11b. As the weight of one hypothesis increases over the others, the entropy of the set of weights will decrease indicating that the choice of hypothesis is becoming less ambiguous, observed in Figure 5.11a.

5.1.4 MHT Performance in Biased GPS Conditions

This section presents results of using MHT for localization when data associations are unknown and the position measurements from the GPS are biased. Two sets of results are shown: one in which the filter is implemented as previously described, and another in which the filter is tuned to operate in biased GPS conditions. This should illustrate the practical feasibility of using vision for localization within an MHT framework.

The results shown are for the data set used in Chapter 4. This data set contains a bias in the measured GPS positions as large as 40m while the vehicle is stopped at an intersection. The results shown in this section are for the specific portion of the data set when this occurs. The results shown use the entire system (using Visual Context).

Figure 5.12 plots the vehicle position estimate and its error ellipse for each hypothesis on the map at particular time instants. It is observed that the hypotheses are following the biased GPS estimate, and that the use of vision does not help to correct the localization. The reason this occurs is because the bias in the GPS position measurement is larger than the validation region used to identify road segment candidates. Thus, the correct road is not among these candidates, and even with the use of vision, no correction can be made.

This result indicates that the proposed MHT approach is not practical to use in biased GPS conditions. However, as shown in Chapter 4, a significant advantage of using vision is that this bias can be corrected. It should be possible to tune the MHT filter to consider a wider range of road segments to combat this bias problem. The concern in doing so is that the number of hypotheses will increase, thus raising the computational demands of the filter.

Increasing the size of the validation region used to find candidate road segments can be achieved in two ways: directly by increasing the size of the error ellipse, or indirectly by increasing uncertainty in the GPS measurement. The former would involve increasing the validation region to an $n\sigma$ error ellipse around the state estimate, where $n = 2$ for the current method. The latter would widen the size of the current 2σ error ellipse, and put less emphasis on the GPS measurements in the Kalman filter. I chose the latter approach of increasing the uncertainty of the GPS measurements, but the amount of increase was dependant on the speed of the vehicle, as measured by the GPS. The reasoning behind this is that the

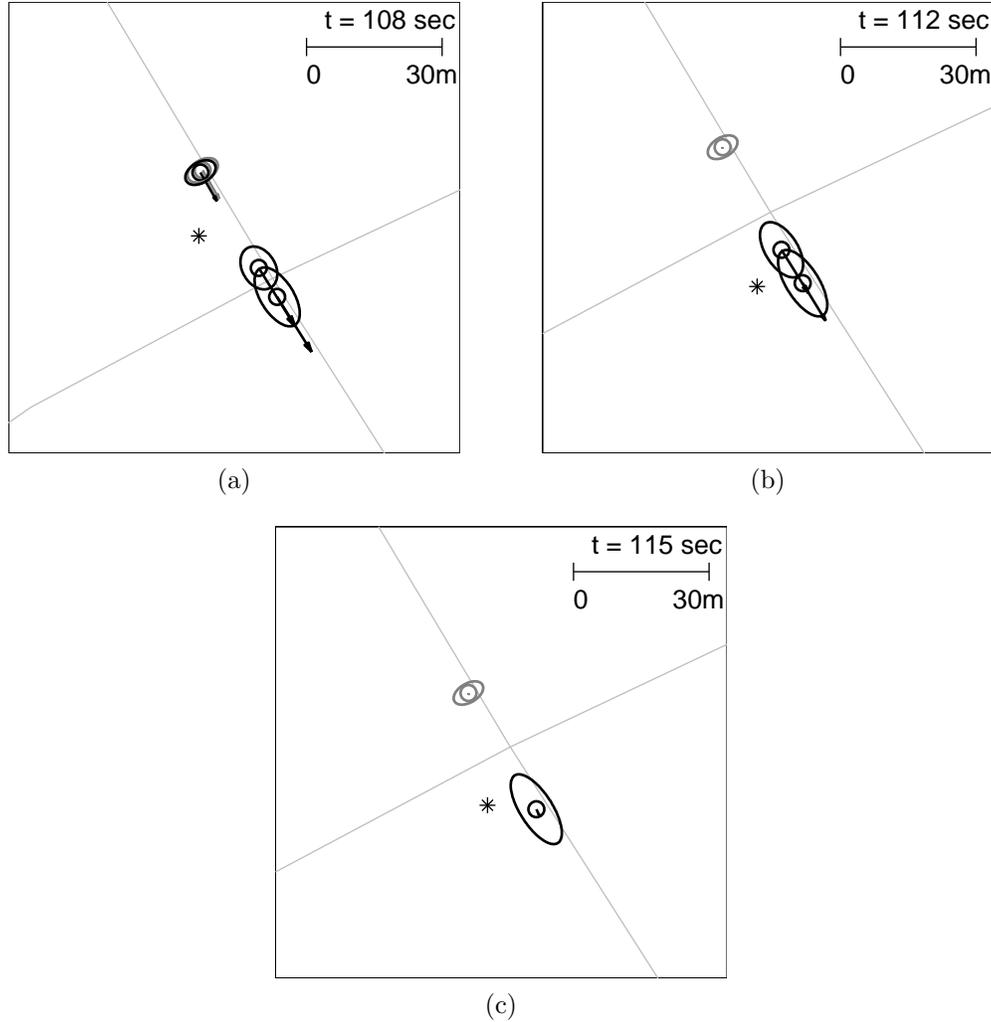


Figure 5.12: An illustration of MHT localization with visual context in biased GPS conditions. The known data association result is shown in gray for comparison, and the GPS measurement is shown as ‘*’.

wider validation region is mostly needed at intersections, where the landmarks can be detected and the vehicle is likely to be stopped. A sigmoid function (5.27) is used to smoothly vary the GPS uncertainty $\sigma_{t,GPS}$ between $\sigma_{min} = 10\text{m}$ and $\sigma_{max} = 80\text{m}$ as a function of the vehicle speed \hat{v}_t measured by the GPS receiver. The GPS measurement covariance R_t is then given by (5.28).

$$\sigma_{t,GPS} = \sigma_{max} - \frac{\sigma_{max} - \sigma_{min}}{1 + \exp\left[-\frac{1}{2}(\hat{v}_t - v_0)\right]} \quad (5.27)$$

$$R_t = \begin{pmatrix} \sigma_{t,GPS}^2 & 0 \\ 0 & \sigma_{t,GPS}^2 \end{pmatrix} \quad (5.28)$$

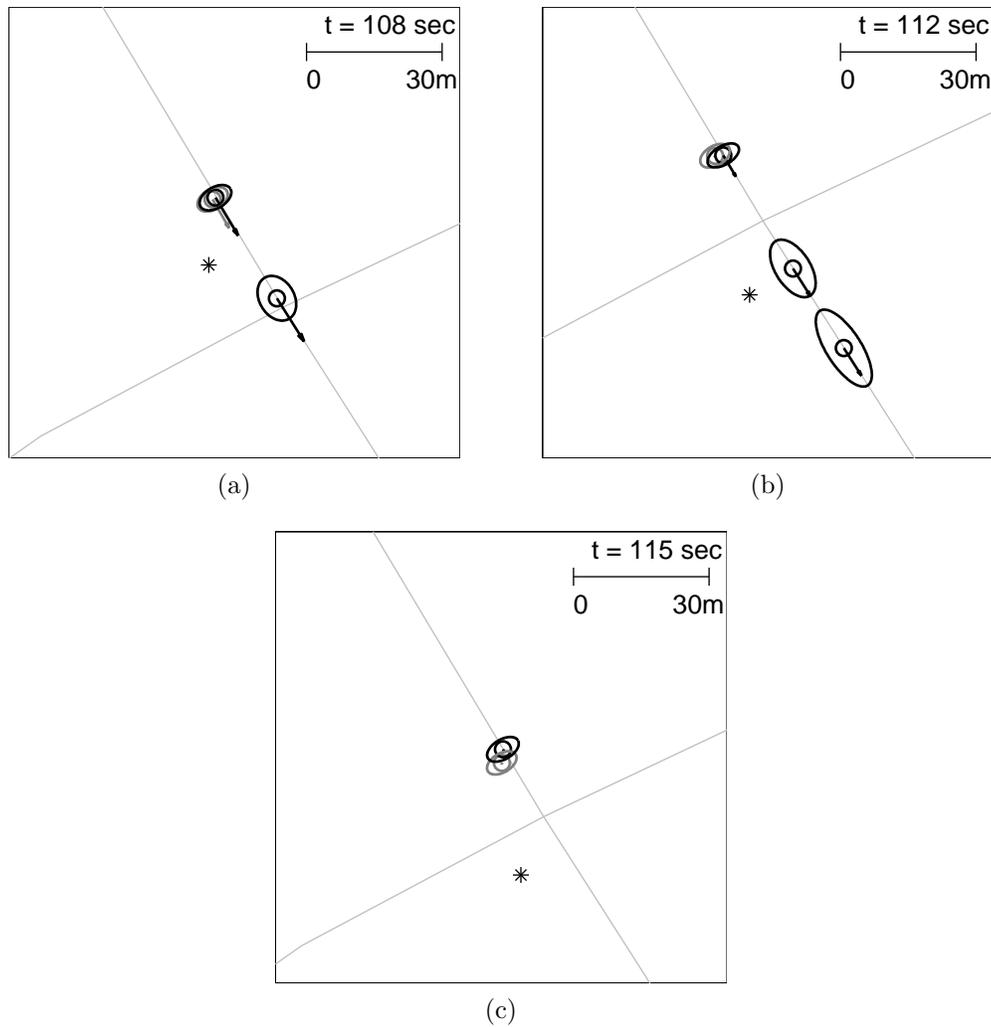


Figure 5.13: An illustration of MHT localization with visual context in biased GPS conditions using a tuned filter. The known data association result is shown in gray for comparison, and the GPS measurement is shown as ‘*’.

The outcome of this change is shown in Figure 5.13, where the vehicle position estimate and its error ellipse for each hypothesis are plotted on the map at the same time instants shown in Figure 5.12. It is observed that the correct road segment hypothesis is maintained despite the drift in the GPS measurements. Furthermore, the filter converges to the correct hypothesis despite it being much further away from the GPS position measurement.

The localization error for the two realizations of the MHT filter is shown in Figure 5.14. For the tuned filter, localization error approaches zero in the longitudinal direction as the intersection landmark is continually observed. This agrees with the result from the known data association case, shown in Figure 4.13. Lateral position

error is largely unaffected in this case.

Tuning the MHT filter to operate in a biased GPS environment was performed by trial-and-error on this specific data set, and as a result the parameters chosen may not work for every data set. There are other methods that can be attempted to improve the robustness of MHT to GPS measurement bias. Track splitting is a method commonly used in multiple-target tracking to ensure that a new data association hypothesis is not spurious. Whenever the data association changes, a copy of the hypothesis is made. The original maintains the old data association while the copy is assigned the new association [32]. If the new association is the result of an incorrect detection (or a temporary GPS drift), it should be pruned rather quickly. The old hypothesis is thus still present should it prove to be correct. Lazy data association is similar to a Maximum Likelihood approach, except that it allows past data association decisions to be revisited and changed, similar to a tree search [29]. This could be applied to MHT, allowing parent states to be revisited should the children prove to be poor candidates. The benefit of these two approaches is that they can be implemented within the current MHT approach, although they may not remove the need to tune the filter as this is a common task when using Kalman filters, and one that can be quite delicate [64]. One reason for this tuning sensitivity is that Kalman filters are often employed to systems that violate the linear-Gaussian assumptions.

Another approach to data association that may prove beneficial is particle filtering. By sampling the state space randomly and assigning random associations to each particle, it is expected that the particles with the correct associations should receive higher weight and thus improve their chances of survival [46]. The drawback of this approach is that it requires a redesign of the localization method. This is the topic of the next section.

5.2 Monte Carlo Data Association using Particle Filtering

5.2.1 Introduction

A method based on particle filtering (PF) is investigated as an alternative to the Kalman filter-based MHT approach in the hope that it will prove more flexible

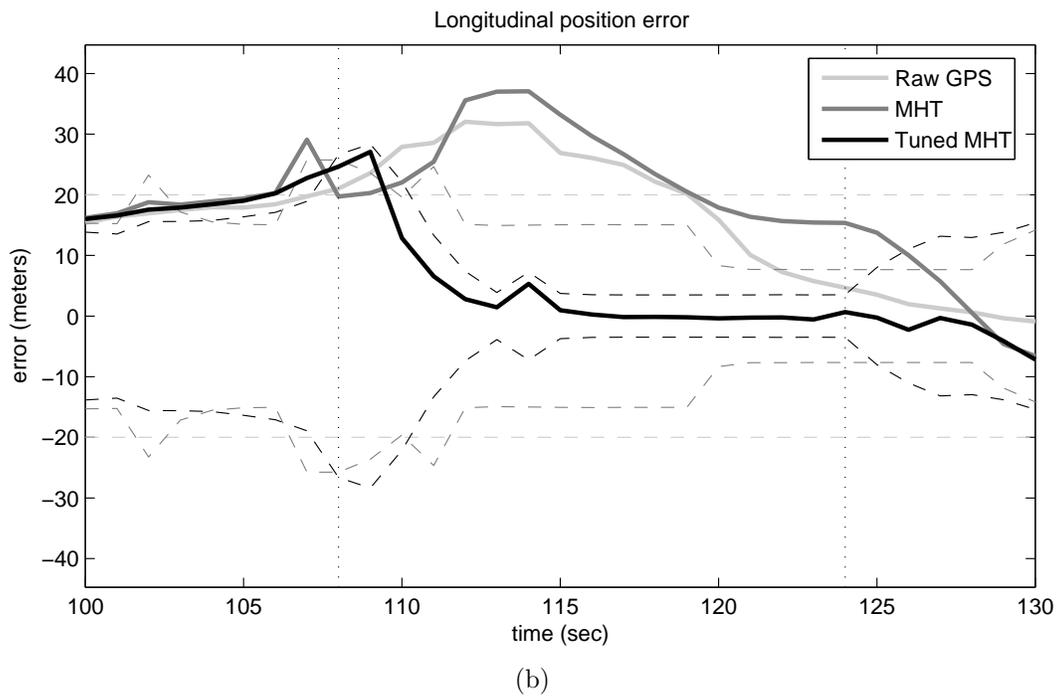
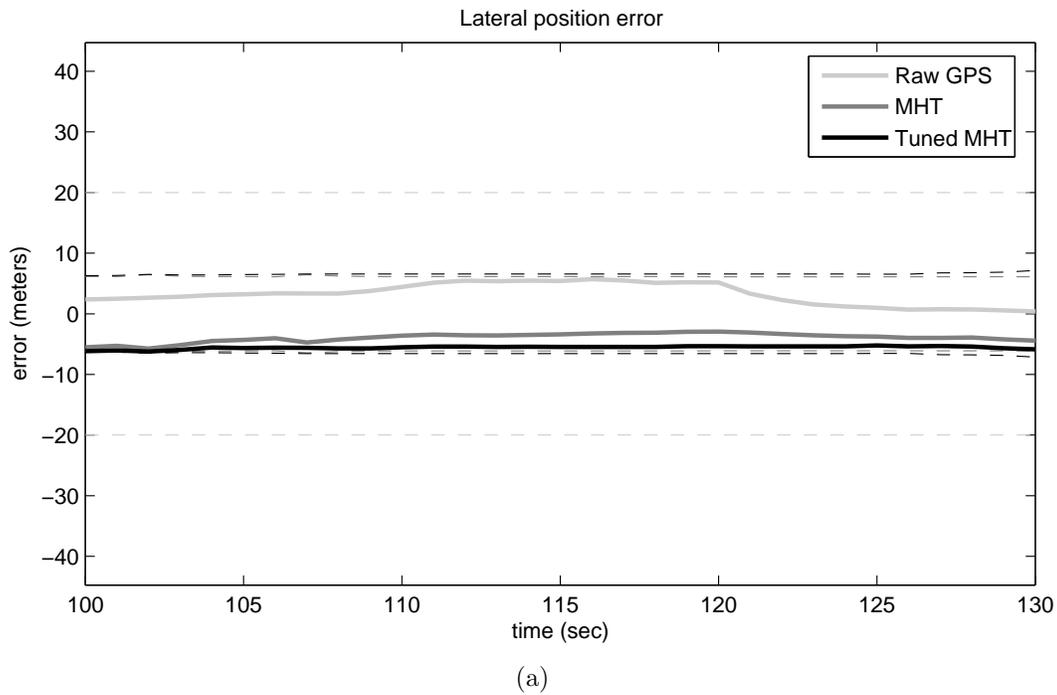


Figure 5.14: Average position error at each time instant using MHT localization in biased GPS conditions. The error in the raw GPS position measurements are shown, with the average error in MHT localization with and without tuning the filter. a) Error in the lateral direction. b) Error in the longitudinal direction; dotted lines show the beginning and ending of the time interval where a road marking is visible.

and robust to problems such as GPS measurement bias. The principal benefit of particle filtering is that it uses a random sampling of the state space rather than the closed form parameterization used by the Kalman filter, allowing multimodal distributions which may occur as a result of ambiguous data associations to be handled easily. A PF method for localization in known data association conditions was presented in Section 4.3 and found to give similar accuracy as the KF method. This method is now adapted to the case of unknown data associations.

A Monte Carlo approach is used to assign a random data association to each particle from a set of possible associations [46]. This selection method allows the consideration of relatively unlikely associations for some particles, which will improve the robustness of the data association should significant bias in the GPS measurements occur. The large number of hypotheses ($N = 200$ particles were used in Section 4.3) means that one can be more liberal in assigning data associations, and can consider unlikely hypotheses. With the MHT method there are a small number of hypotheses, and one must be conservative in selecting and pruning hypotheses so as to avoid incorrect data associations that could diverge the filter.

The drawback to a particle filtering approach is that it is not guaranteed to provide the optimal filtering solution. What it lacks in optimality, however, it makes up for with flexibility. It can be thought of as a higher-level approach; micro-management of hypotheses is not required since the randomness of the data association process should ensure that enough alternatives are considered in order to find the best association. Tuning of the filter is also a less demanding task.

5.2.2 Algorithm Description

Updating Particle Weights

The particle weights are updated as in the known data association case; see Section 4.3. This involves updating the weight of each particle using the measurement probability $p(z_t|x_t^i, c_t^i)$. The principal difference between the known and unknown association cases is that the associations are assigned on a per-particle basis, rather than each particle having the same association. This creates the potential for the same weight imbalance observed in MHT when a subset of particles have detected landmark associations and the rest do not. The former receive an additional measurement probability update, deemphasizing these particles since $p(\hat{f}_t|x_t^i, c_t^i) < 1$.

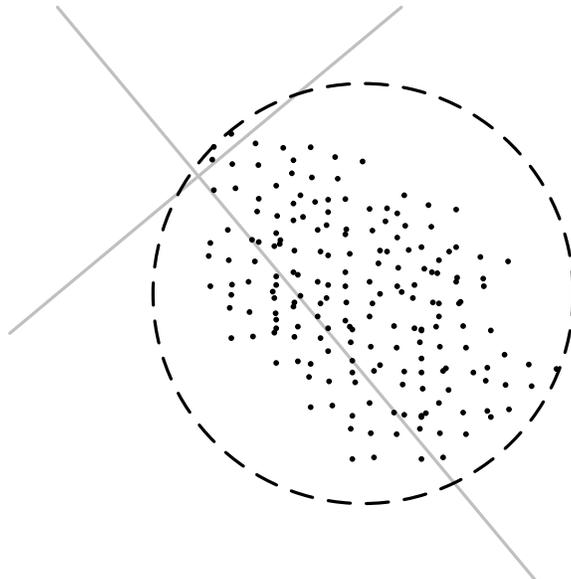


Figure 5.15: Road segment candidates are those within the space occupied by the particle set, represented by the dashed circle.

The imbalance is addressed using weighting terms as it was with MHT. Those particles without a landmark hypothesis are multiplied by α , the false positive rate of the landmark detector (5.29). Those particles with a landmark hypothesis are multiplied by $1 - \beta$, the “true positive” rate of the detector (5.30).

$$w_{t,-o}^i = w_{t-1}^i p(p_t | x_t^i) p(\zeta_t^i | x_t^i, c_t^i) \alpha \quad (5.29)$$

$$w_{t,o}^i = w_{t-1}^i p(p_t | x_t^i) p(\zeta_t^i | x_t^i, c_t^i) p(f_t | x_t^i, c_t^i) (1 - \beta) \quad (5.30)$$

Determining Possible Associations

At each time filter iteration, a list of possible data associations is created, from which the particular association for each particle is chosen randomly. First, the candidate road segments are defined as those lying within the space occupied by the set of particles. This can be represented by the smallest box or circle enclosing the particles in world coordinates (Figure 5.15). A search of the map database that returns road segments within this region provides the set of candidate road segments R_t .

Second, the orientation of the vehicle is constrained by the topology of the road network to be consistent with past orientation hypotheses. Three constraints are used to identify the orientation of the vehicle on a particular road segment. First,

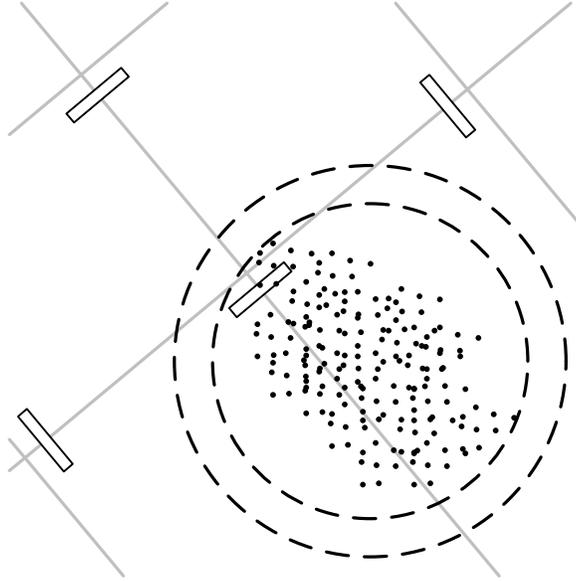


Figure 5.16: Visible road markings lie at the end of each road segment in the direction of the orientation hypothesis. Landmarks that lie outside the dilated validation region are not considered.

each candidate road segment $r_t \in R_t$ must be connected to one of the candidate road segments $r_{t-1} \in R_{t-1}$ from the previous iteration in order to maintain continuity. Second, any road segment that persists from the previous iteration automatically receives the same vehicle orientation. Third, a new road segment that was not present at the previous iteration is constrained to be oriented away from its intersection point with a previous road segment candidate, as previously illustrated in Figure 5.4. By these constraints, the number of data association candidates is at most $|R_t|$, and will be less if the connectedness constraint is violated for any $r_t \in R_t$.

Finally, the identity of detected landmarks are determined. The road marking landmarks used are only visible to a vehicle facing an intersection of roads. The orientation assigned to each segment indicates at which endpoint of the segment exists a potentially visible landmark. Thus, there is one possible landmark for each road segment. However, only those landmarks near enough to the particle set to be visible are considered. To remove those far away, a validation region is used, defined by dilating the road segment search region by $2\hat{f}_t$, twice the measured distance to a detected landmark. This is illustrated in Figure 5.16, where all but one landmark will be removed from consideration.

By this procedure a set D_t of possible data associations has been created. Each

element of D_t , $d_t^j \in D_t$, is one data association hypothesis consisting of a road segment, an orientation and any detected landmarks, $d_t^j = \{r_t^j \ \gamma_t^j \ \sigma_t^j\}$.

Per-Particle Monte-Carlo Data Association

The above procedure creates M_t data association hypotheses. Each particle $i \in [1, N]$ is assigned an association randomly from this set, where the probability of selecting hypothesis $j \in [1, M_t]$ is determined by the magnitude of its measurement probability relative to all M hypotheses for that particle. The data association c_t^i assigned to the particle is assigned randomly from the set D_t , based on the measurement probability for each hypothesis $d_t^j \in D_t$ (5.31). Thus, the measurement probability of each data association hypothesis must be computed for each particle.

$$p(c_t^i = d_t^j) \propto p(z_t^i | x_t^i, d_t^j) \quad (5.31)$$

Selecting the association randomly accounts for ambiguity among the hypotheses and in the measurements [46]. For example, bias in the GPS measurements may place most of the particles near an incorrect road segment, however with Monte-Carlo data association at least some of these particles should be matched with the correct segment. After randomly selecting a hypothesis for each particle, the weight of the particle is updated using the measurement probability for that hypothesis. This step is trivial because this probability has been previously calculated.

Resampling

A very simple resampling procedure is used to maintain diversity among the set of particles by pruning away those with low weight. This has the effect of removing unlikely data association hypotheses from consideration as well, since the likelihood of each data association is indicated by the weight of the particles it is assigned to.

It was found that periodically resampling the particle set was sufficient to maintain diversity among the particles while keeping the spread of the particle set relatively small. The prediction component of the filter spreads the particle set, while resampling contracts it to the highest weighted particles. After only a few iterations of the filter without resampling, much of the weight tends to become concentrated among a few particles [4]. A long resampling period therefore results in a highly

Algorithm 5.6 Pseudocode for particle filtering algorithm.

```

1. function ParticleFilteringLocalization(  $x_{t-1}^{1:N}, c_{t-1}^{1:N}, w_{t-1}^{1:N}, z_t$  )
2.
3. // State prediction and GPS update
4.  $x_t^{1:N} = \text{PFPredict}( x_{t-1}^{1:N} )$ 
5.  $w_t^{1:N} = \text{GPSWeightUpdate}( x_t^{1:N}, w_{t-1}^{1:N}, p_t )$ 
6.
7. // Find a set of associations
8.  $D_t = \text{FindPossibleCorrespondences}( x_t^{1:N}, c_{t-1}^{1:N} )$ 
9.
10. for  $i=1:N$ 
11. // Monte-Carlo data association
12.  $[c_t^i, \lambda_t^i] = \text{MonteCarloDataAssociation}( x_t^i, D_t, \hat{f}_t )$ 
13. // Update weight of particle
14.  $w_t^i = w_t^i \lambda_t^i$ 
15. end for
16.
17. if  $t\%T == 0$ 
18.  $x_t^{1:N}, c_t^{1:N} = \text{ResampleParticles}( x_t^{1:N}, c_t^{1:N}, w_t^{1:N} )$ 
19.  $w_t^i = 1/N \forall i \in [1, N]$ 
20. end if
21.
22. return  $x_t^{1:N}, c_t^{1:N}, w_t^{1:N}$ 

```

spread particle set where the weight is not widely distributed. Conversely, resampling at every iteration of the filter results in a very small number of unique particles which do not adequately cover the state space. Resampling every few iterations will contract the particles to an acceptably small region, and redistributing the weight evenly over all particles at this time will improve diversity. A resampling period of $T = 2$ was found to produce an acceptable tradeoff between state space exploration and weight distribution among the particles.

Pseudocode for the particle filtering localization algorithm with Monte Carlo data association is shown in Algorithm 5.6.

5.2.3 Results

Localization results are provided to compare the performance of particle filtering with Monte Carlo data association to the MHT method. Results are provided for the two data sets shown in the MHT section, the first without bias in the GPS

measurements, the second with bias.

Results are shown first for the case where Visual Context is not incorporated into the system. Figure 5.17 plots the particle cloud (in black) on the road network map, along with a particle cloud for the known data association case (in gray). It can be seen that particles are concentrated around the GPS position measurement (shown as *), but are also drawn to the road segments as a result of map matching. Ambiguity in the data association can be observed in Figure 5.17b, where the particles could be matched to any of the four road segments defining the intersection. The particles eventually converge to the road segment closest to the GPS measurement as shown in Figure 5.17c. Particles in this region will receive higher weight being closer to the GPS measurement, and therefore are more likely to be resampled. These results are similar to what was observed with MHT.

Figure 5.18 shows the localization result when Visual Context is included in the system. Performance of the filter is similar to the non-vision case, except that the particle cloud converges faster to the road segment nearest to the GPS measurement (Figure 5.18c) due to the presence of a detected landmark. The particle cloud is also more tightly concentrated, indicating a lower degree of uncertainty in the location of the vehicle.

It is of interest to note that the spread of particles is not significantly different than the known data association case, despite weight contributions from map matching to multiple roads. This result is surprising; it was expected that the particles would gather around the road segments – similar to MHT where the state estimate for each hypothesis is near its road segment – rather than remain in a single cloud. This result may indicate that the particles are primarily attracted by the GPS measurement, which would create a significant problem should the particles follow a biased GPS measurement.

The localization error with and without Visual Context is shown in Figure 5.19. The error in the longitudinal direction (Figure 5.19) tends to follow the GPS measurement. Data association ambiguity (between $t = 28$ sec and $t = 36$ sec) causes the error in both cases to increase, however through the observation of the intersection landmark this is resolved faster. Visual Context also reduces the localization uncertainty in the longitudinal direction. The localization errors are summarized in Table 5.4.

To evaluate the ambiguity associated with choosing among the data association alternatives, the entropy of the weight assigned to the various hypotheses is

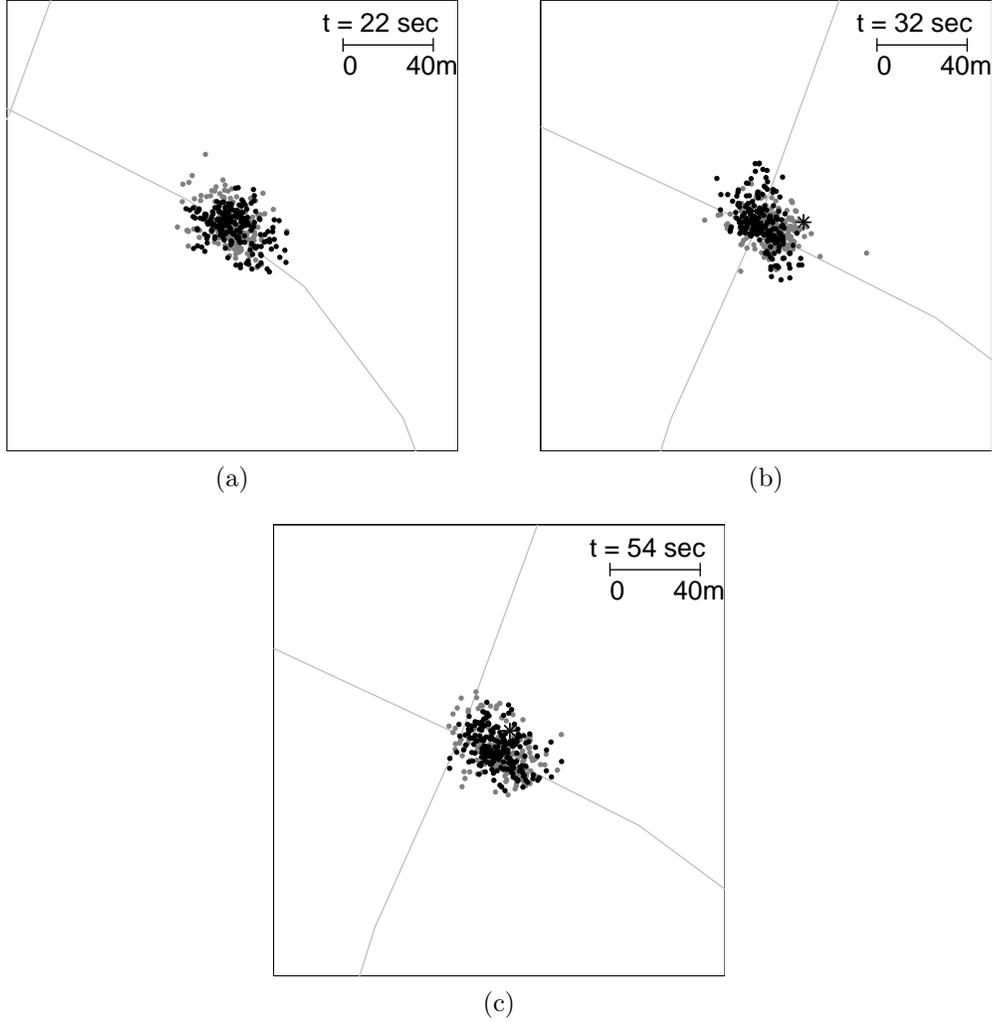


Figure 5.17: An illustration of PF localization using Monte Carlo data association without visual context. A known data association result is shown in gray for comparison, and the GPS measurement is shown as ‘*’.

calculated and shown in Figure 5.20a. This calculation requires first finding the total weight assigned to each data association hypothesis $d_t^j \in D_t$. I will denote by $N_t^j \subset [1, N]$ the subset of particles with data association d_t^j . I then find the total weight W_t^j assigned to this data association hypothesis as the sum of the weights of the particles in this subset (5.32). The entropy of $W_t = \{W_t^1, \dots, W_t^M\}$ is then calculated.

$$W_t^j = \sum_{k \in N_t^j} w_t^k \quad (5.32)$$

Figure 5.20a agrees with entropy among the MHT hypotheses in showing that the addition of Visual Context results in lower entropy of the weights, therefore in-

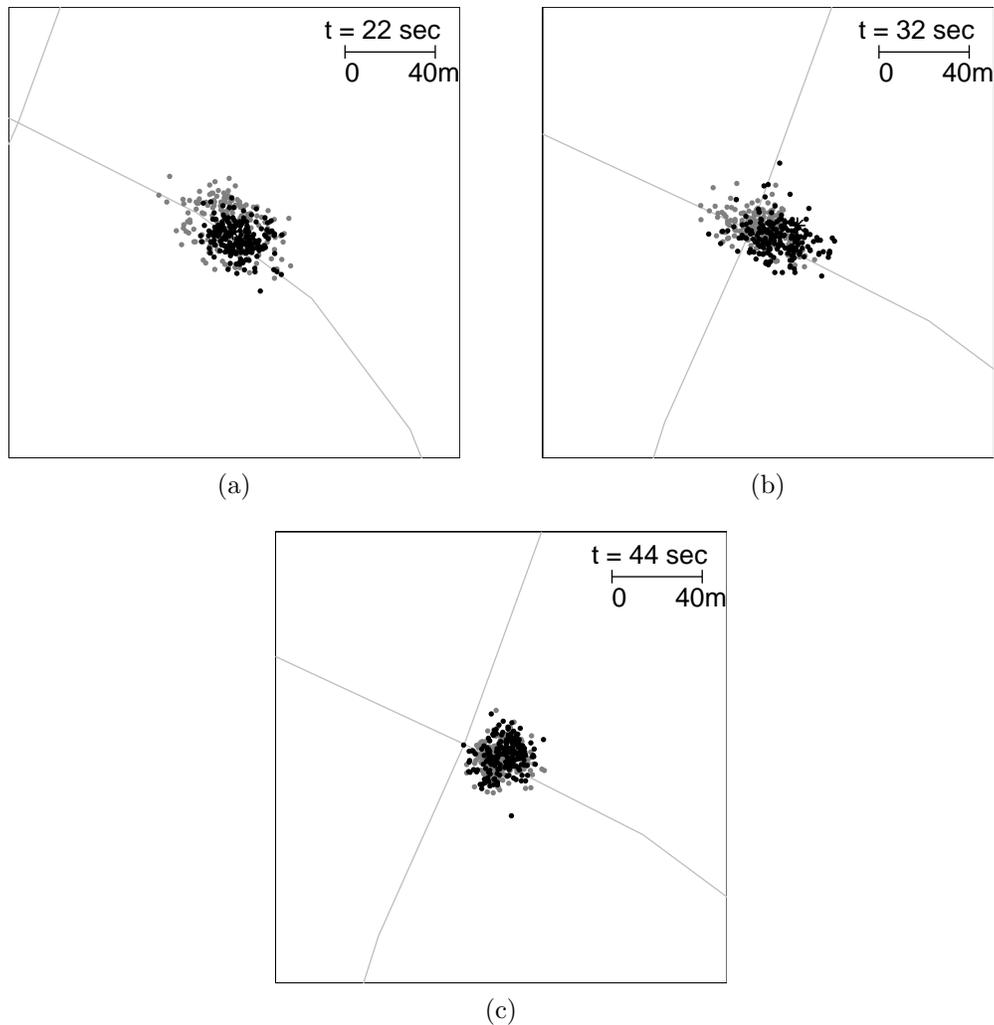
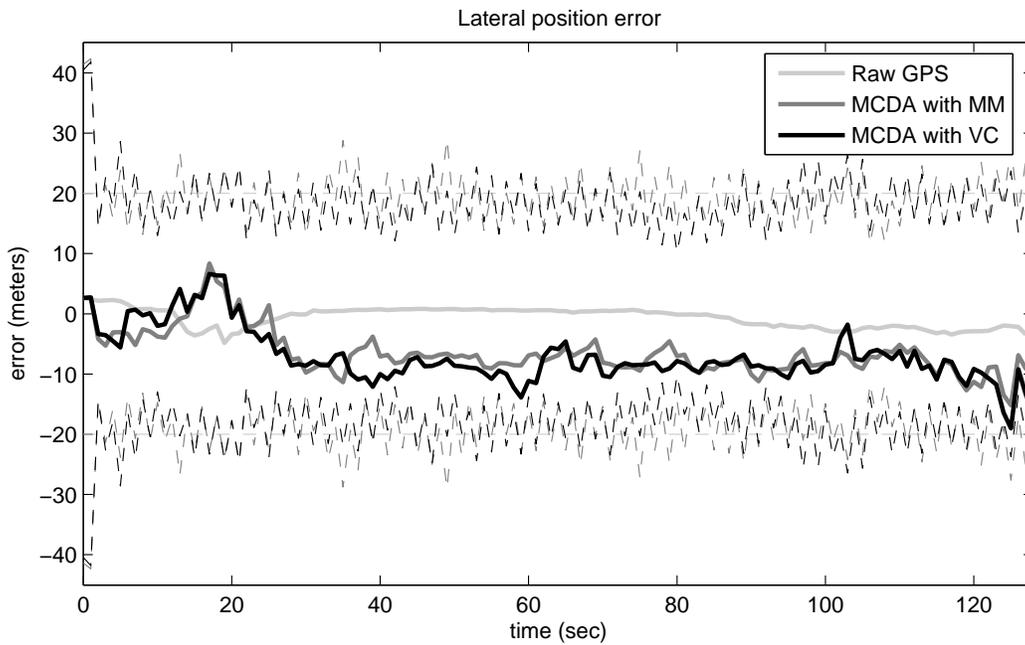


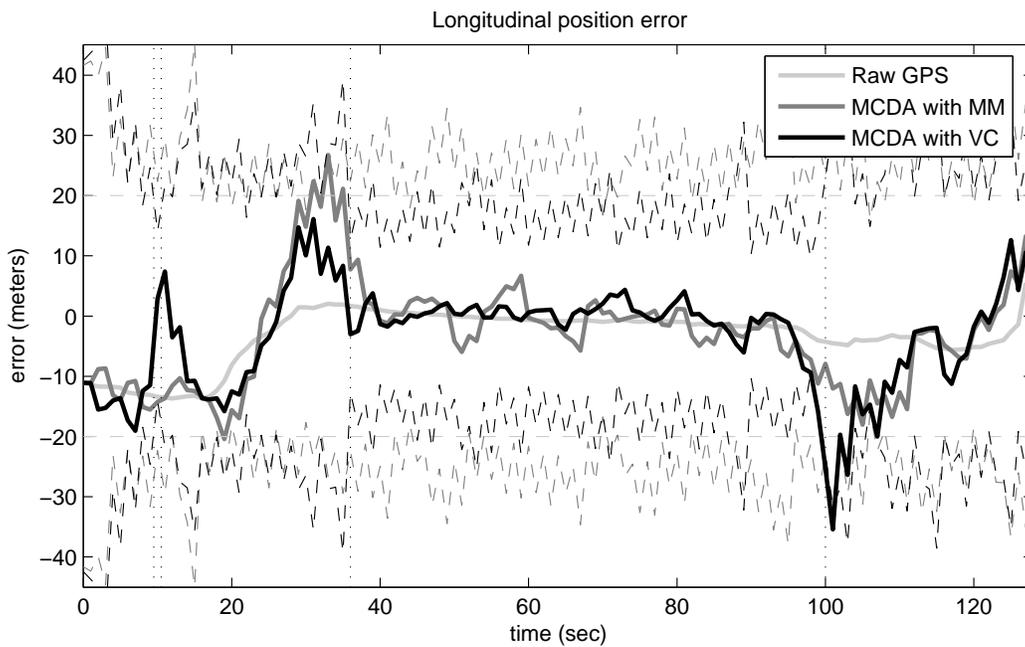
Figure 5.18: An illustration of PF localization using Monte Carlo data association with visual context. A known data association result is shown in gray for comparison, and the GPS measurement is shown as ‘*’.

dicating that the weight is concentrated on fewer hypotheses. Therefore, it appears to be the case that ambiguity is removed from the association process by adding new information in the form of a landmark detection.

The total weight assigned specifically to particles with the correct data association hypothesis is shown in Figure 5.20b. This figure shows that the correct hypothesis is receiving nearly all of the weight when a landmark is detected. This is not the case when Visual Context is not used. This shows a significant advantage of adding vision to the localization system; while there is little to separate the two in terms of accuracy for this data set, as can be seen in Figure 5.19, with the use of vision there is an improved certainty of which data association hypothesis is

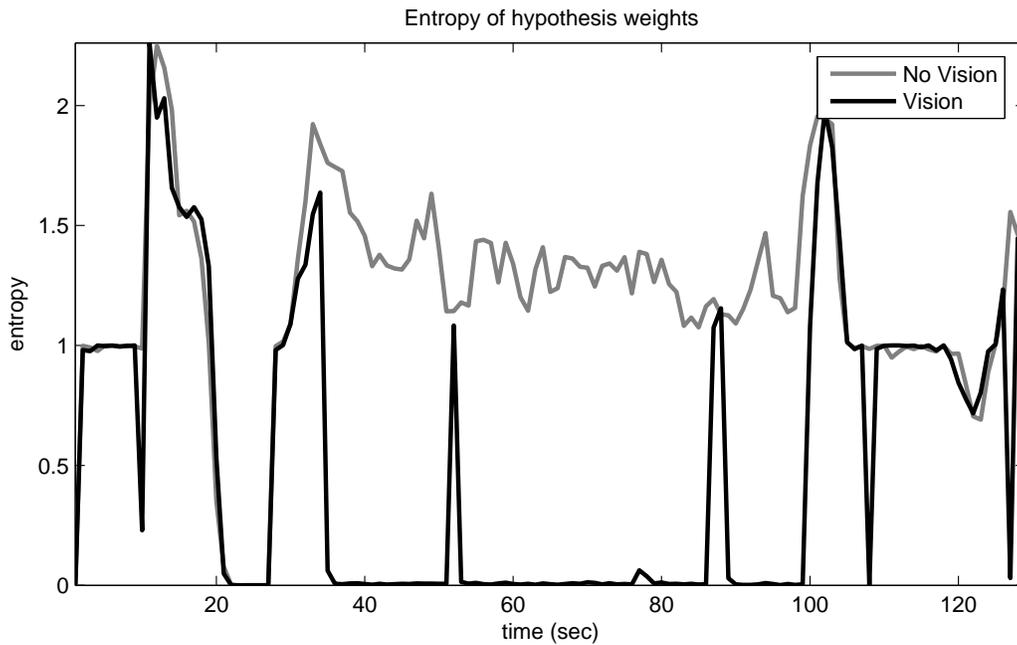


(a)

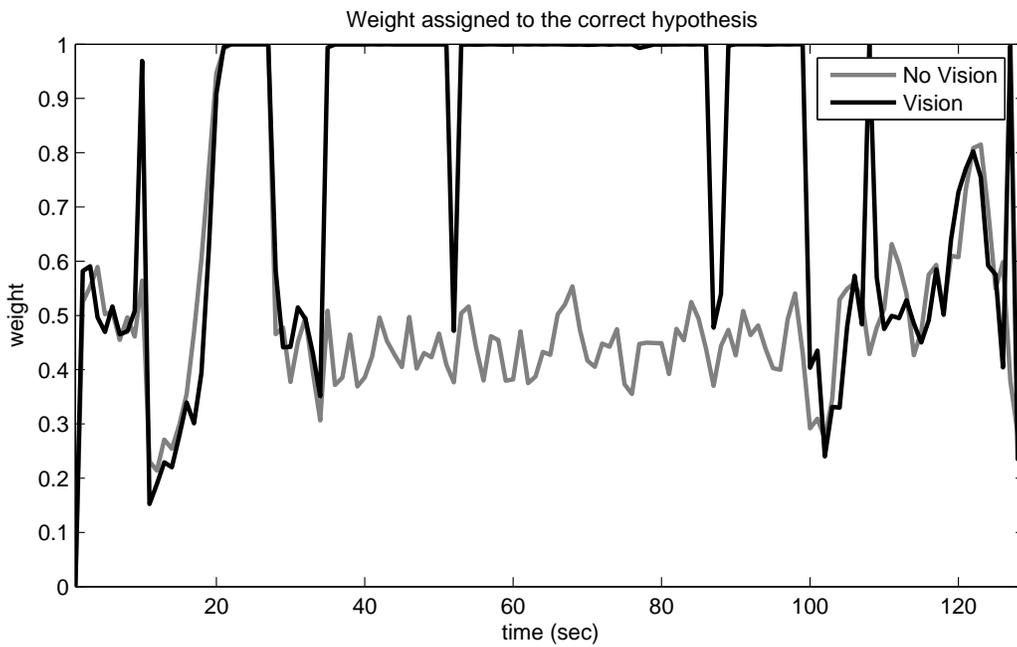


(b)

Figure 5.19: Average position error at each time instant for PF localization using Monte Carlo data association. The error in the raw GPS position measurements are shown, with the average error in MCDA using map matching (MM) and using map matching and visual context (VC). a) Error in the lateral direction. b) Error in the longitudinal direction; dotted lines show the beginning and ending of time intervals where a road marking is visible.



(a)



(b)

Figure 5.20: A plot of a) the entropy among the hypotheses in the particle set, and b) the total weight assigned to the correct hypothesis, for PF localization using Monte Carlo data association.

	$\bar{\varepsilon}$ (m)	s_{ε} (m)	$\bar{\sigma}$ (m)	s_{σ} (m)
$p_{\perp,MM}$	-7.6	1.7	9.5	2.2
$p_{\parallel,MM}$	-0.23	5.1	12.9	2.5
$p_{\perp,VC}$	-8.8	1.9	8.7	2.1
$p_{\parallel,VC}$	-0.13	3.2	8.7	2.8

Table 5.4: Lateral and longitudinal position error and uncertainty for particle filtering with Monte Carlo data association. Results are shown using map matching (MM) and using map matching and visual context (VC). These results are for the time instants when road markings are visible.

correct. Without vision, there is still significant uncertainty in this regard despite localization error being near zero while the vehicle is stopped at the intersection.

These results are not surprising since they echo those found when using MHT. Thus, both appear to be viable approaches for maintaining localization performance in unknown data association conditions with an unbiased GPS. What would separate the PF approach from MHT would be to succeed in correcting GPS measurement bias, which MHT was unable to accomplish without special tuning. For the same data set containing GPS measurement bias used to test MHT, PF results are shown in Figure 5.21. This shows the particle cloud plotted near the intersection where the bias occurred, with known data association results shown for comparison in gray. These results show that even with significant measurement bias, many of the particles are clustered near the correct road segment (Figure 5.21b). After 10 sec of observing the landmark, the particle cloud has converged (Figure 5.21d). The localization error for this data set during this period is shown in Figure 5.22.

It is important to note that this correction occurred because the observed landmark was correctly identified, and that it was identified without having to perform any tuning of the filter in addition to that done for the unbiased case. This can be attributed to the randomness of the particle filter that enables exploration of regions of the state space further away from the GPS measurement. Thus, it can not be guaranteed that this bias correction will occur every time this algorithm is run on the same data set, since a different area of the state space may be explored, in which case it would be less likely that the correct hypothesis would be identified. One way to improve the robustness of particle filtering to this problem are to increase the number of particles to increase the chance of exploring the correct

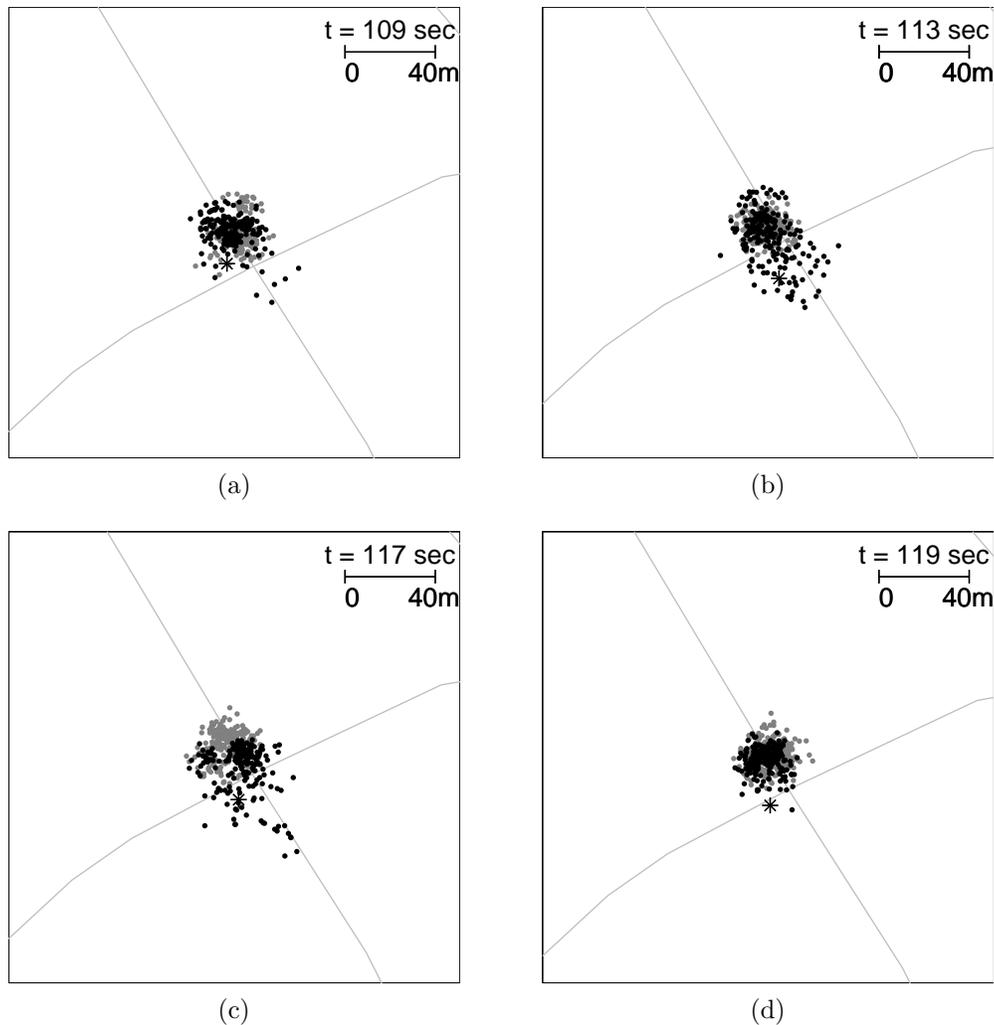
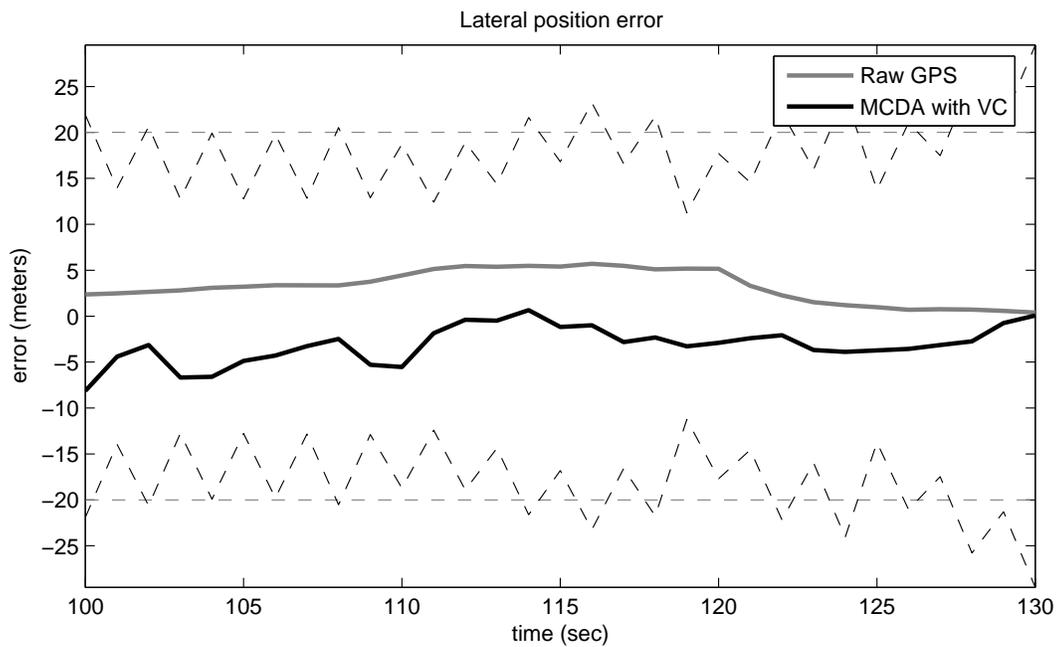


Figure 5.21: An illustration of PF localization with visual context in biased GPS conditions. A known data association result is shown in gray for comparison, and the GPS measurement is shown as ‘*’.

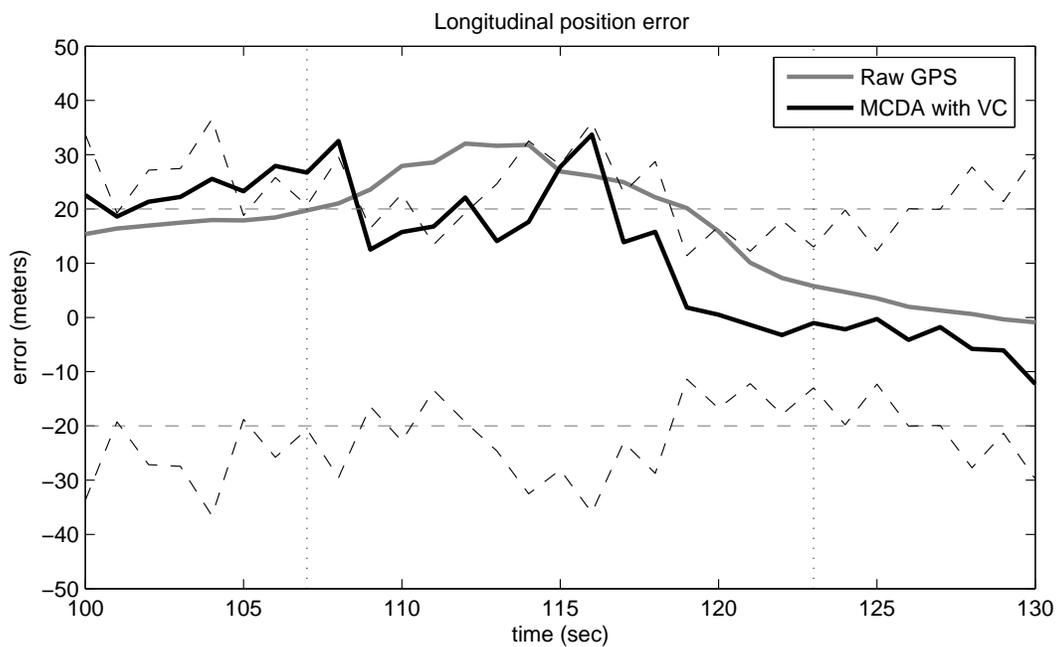
region of the state space. This increases the computational burden of the filter, and still does not guarantee convergence. Choosing an appropriate proposal density to sample particles from is a more intelligent solution, whereby particles are guided to the correct region of the state space while maintaining a reasonable number of particles.

5.2.4 Discussion

As with MHT, the results indicate that through the addition of vision, choosing among a set of data association hypotheses becomes less ambiguous. The reason



(a)



(b)

Figure 5.22: Average position error at each time instant using PF localization with Monte Carlo data association, in biased GPS conditions. The results show a reduction in longitudinal position error compared to the GPS position measurements.

for this is similar to the discussion of MHT results; a higher weight assigned to particles with landmark hypotheses versus those without results in the former group of particles receiving a higher weight than the other. This distribution of weight in favor of one group of hypotheses results in that group being the more probable choice, therefore reducing entropy among all hypotheses.

How this distribution of weight occurs depends on the parameters α and β used to balance the weight applied to these two groups of hypotheses. These parameters are set to static values in this investigation, however they could be dynamic; their values could be adjusted to reflect the probability of detecting a landmark at the current vehicle position or depending on weather or illumination conditions. Also, this type of imbalance would be made more complicated with more sophisticated visual capability if more than one type of landmark is detected at a time. There is therefore much that can be done within this localization framework in terms of adding more sophisticated visual capability, including:

- detecting more types of landmarks in images from vehicle-mounted cameras,
- modeling the location of these landmarks as functions of vehicle location, and
- adaptively weighting hypotheses to reflect the likelihood of landmark detections, particularly when multiple landmark types are detected.

The incorporation of vision into the framework for particle filtering localization under unknown data association conditions creates a feedback mechanism. The added emphasis placed on particles with landmark detections results in a higher probability of those particles being resampled. Consequently, these hypotheses are more highly represented in the particle set after resampling. Repeated observation of the same landmark results in the convergence of the particle cloud to the location(s) from which that landmark is visible. Again, this result assumes these particles are more highly weighted than those from which landmarks are not visible, and that the correct landmark is always present among the hypotheses.

If landmarks can be identified correctly, the use of vision can counteract the effect of GPS drift. However, GPS has a significant advantage over vision in that no data association is required in order to integrate it into the state estimate. With vision, some particles may not be affected by the landmark detection because those particles cannot be associated with any nearby landmark. Conversely, the GPS measurement affects all particles and thus is a more ubiquitous, and therefore powerful, attractor.

Attribute	MHT	MCDA
Localization accuracy	+	-
Ambiguity of hypothesis selection	+	-
Execution time	+	-
Simplicity of approach	-	+
Exploration of search space	-	+
Robustness to GPS bias	-	+

Table 5.5: Comparison of MHT and PF data association techniques.

5.3 Summary

This chapter proposes two methods to achieve vehicle localization incorporating visual context in the case of unknown data associations. These methods are Multiple Hypothesis Tracking (MHT) based on a Kalman filtering approach, and Monte Carlo Data Association (MCDA) based on a particle filtering approach. A comparison of MHT and MCDA approaches to highlight their relative advantages is summarized in Table 5.5. Based on the results from applying both MHT and MCDA to the same data sets, a number of observations can be made. First, it was found that MHT provides a more consistent and accurate localization result with a lower degree of ambiguity among hypotheses when there is no bias in the GPS measurements (see Figures 5.10 and 5.11 versus Figures 5.19 and 5.20). Also, the MHT algorithm executes faster than the MCDA algorithm, which is expected due to the high number of particles ($N = 200$), however both are capable of executing one iteration within the one second update period of the GPS receiver.

Second, the MCDA approach is in many respects a simpler algorithm than MHT, due to the lack of hypothesis management methods needed to keep the number of hypotheses low. These would include the merging and pruning of hypotheses; with MCDA, unlikely hypotheses become removed after resampling. The randomness of the MCDA approach is its strength, enabling greater exploration of the search space and consideration of unlikely data associations which allows for correction of the GPS measurement bias without requiring special tuning of its parameters to do so.

Chapter 6

Experimental Investigations

This chapter elaborates on the experimental work performed as part of the research for this thesis. Experimental results are presented in the preceding chapters as concepts are presented in order to show the impact of these concepts. The results presented in this chapter summarize the experimental results presented previously. The contributions of this chapter are: descriptions of the experimental setup and the components that make up the vehicle localization system, and; a comparison of the experimental findings for the various localization system implementations presented.

6.1 Objectives

The purpose of this experimental investigation is to demonstrate the practical merit of the proposed localization system. The common theme of the investigation is to evaluate the effect of adding visual context to the localization system. In particular, this investigation has the following objectives:

1. To compare the performance of GPS-based localization to the performance when map matching and visual context are successively added to the localization system. Data associations are assumed known to demonstrate the potential of the proposed system.
2. To evaluate the performance of the localization system when data associations are unknown, and to determine the ability of the system to identify the correct data association. This evaluation is performed with and without visual

context to assess the benefit of adding vision to the system in the unknown data association case.

3. To evaluate the capability of the proposed system to correct GPS measurement bias in unknown data association conditions. This capability would demonstrate the practical potential of using visual context to reduce the effect of GPS measurement errors on vehicle localization accuracy.

Each of these objectives is the subject of an experimental investigation discussed in this chapter. Each are also explored within the context of both Kalman filtering and particle filtering.

6.2 Experimental Setup

This section describes the setup of the experiments, including the hardware used, how data was acquired and the preprocessing that was performed to create data sets used for testing.

6.2.1 Hardware and Installation

Three pieces of sensory hardware were used: an inexpensive GPS receiver to provide experimental vehicle position data; a camera to collect experimental images, and; a highly-accurate GPS receiver to provide ground truth vehicle position data. The vehicle used to acquire this data was a 2001 Volkswagen Golf.

A GlobalSat BU-353 WAAS-capable GPS receiver with an integrated antenna and a USB interface, pictured in Figure 6.1a, was used to provide the experimental GPS position measurements. This receiver has a horizontal position accuracy of 10m RMS, which is reduced to 5m RMS with WAAS enabled. It was decided to operate the receiver with WAAS capability disabled, to make it more likely that the receiver would provide inaccurate measurements exhibiting bias, so that this could be corrected using vision. This receiver was installed on the interior dash along the midline of the vehicle directly below the windshield (see Figure 6.1d).

The camera used was a Unibrain Fire-i board-level camera with an IEEE 1394 (Firewire) interface, pictured in Figure 6.1b. This camera has an RGB image sensor with 640×480 pixel resolution. The camera was configured to provide RGB color

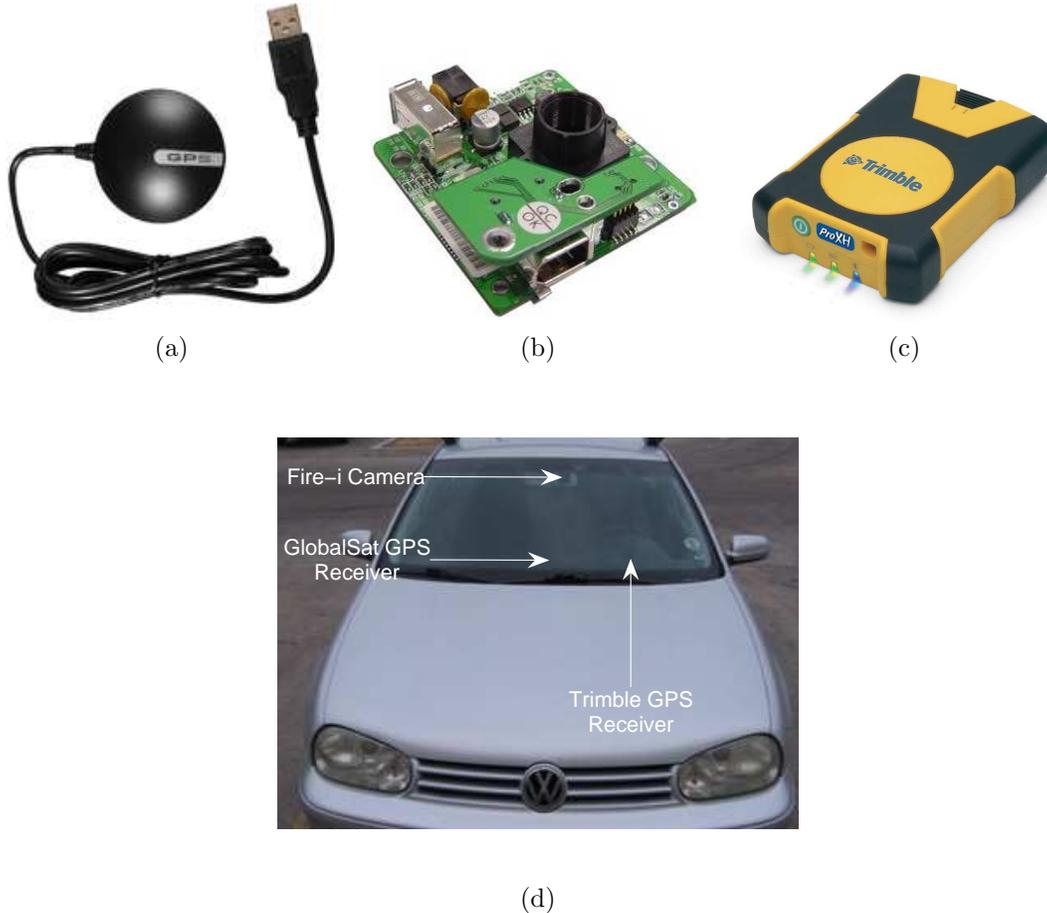


Figure 6.1: Hardware used in data acquisition. a) GlobalSat BU-353 GPS receiver, used to provide experimental data. b) Unibrain Fire-i camera, used to acquire experimental images. c) Trimble Pro-XH GPS receiver, used to provide ground truth data. d) Hardware installation locations in the test vehicle.

images in 320×240 pixel resolution, to reduce memory consumption requirements. A lens with a 2.0mm focal length and infrared filter coating was used. Despite this lens coating, direct sunlight caused significant saturation in the acquired images. This camera was mounted to the back of the rearview mirror of the vehicle, as this provided a convenience way of adjusting the view of the camera by adjusting the mirror position (see Figure 6.1d). The camera axis was oriented along the vehicle midline at an angle of approximately 20° below horizontal.

Ground truth vehicle positions were provided by a Trimble Pro-XH GPS receiver, pictured in Figure 6.1c with an RS-232 serial interface. This receiver has submeter horizontal position accuracy. This is not a Differential GPS receiver, thus no extra hardware was required to receive correction messages. The receiver monitors the satellite configuration, and only provides position measurements when the

configuration is favorable for providing accurate measurements. As a result, this receiver often does not provide a position measurement in the presence of foliage or tall buildings which obstruct satellite signal transmission. A light-emitting diode (LED) on the receiver indicates when it has acquired a valid position measurement; the receiver was installed on the interior vehicle dash beside the experimental GPS receiver (see Figure 6.1d), and oriented such that the driver had view of this status LED. This provided the driver with insight into the suitability of the environment for acquiring reliable ground truth data.

6.2.2 Data Acquisition

The above three pieces of hardware were connected to a notebook computer running the Microsoft Windows XP operating system. A data acquisition utility was written in the C programming language to interface to the GPS receivers and the camera. The GPS receivers output ASCII sentences every second using the NMEA 0183 serial communications standard. A number of standard NMEA sentences exist for GPS receivers. The specific sentence recorded each second from both receivers is the GPRMC sentence, and contains the time, date, receiver position in latitude and longitude, speed over ground in knots, and bearing relative to North in degrees. The GPRMC sentences are stored in a text file for each receiver by the data acquisition utility. The successful reception of a GPRMC sentence from the experimental receiver is used to trigger the acquisition of an image from the camera. The camera is capable of much higher frame rates (up to 30 frames per second for the chosen resolution), however even with one frame per second the memory requirements to store these images are considerable. Images were stored using the lossless Tagged Image File Format, and given a name containing the time stamp of the GPS measurement triggering acquisition of the image.

6.2.3 Data Collection and Preprocessing

Data was collected by running the acquisition utility while driving the vehicle around the City of Waterloo, Ontario. The city is mostly suburban, thus there is little opportunity to observe extreme multipath events which normally occur in urban canyon environments. Data was collected during a weekday morning when traffic was not heavy. This was a necessary precaution to create favorable conditions for the vision system; it was found in previous testing that other vehicles often

confused the road marking detection algorithm.

In all, approximately 40 minutes of driving data were collected over four separate trips. Prior to using this data for experimental testing, preprocessing was needed to generate data sets containing the necessary information for the system. The criteria used to create data sets for testing were the following.

1. The presence of continuous ground truth data; the ground truth receiver often would not provide any data for minutes at a time as a result of poor satellite reception in certain areas. Areas known to have poor reception were avoided, however the ground truth receiver is extremely sensitive. “Continuous ground truth data” was defined as a series of consecutive GPRMC strings with temporary outages lasting at most 3 seconds.
2. Accurate representation of the road network; the road network database used was approximately 7 years old at the time of data collection, meaning that changes to the road network in the intervening period were not accounted for. As an accurate representation of the road network is assumed by the localization system, any data set used for testing needed to be in an area where the road network was accurately represented. As the city of Waterloo has grown significantly since 2001, this was found to be a very restrictive requirement.
3. Favorable conditions for the vision system; as vision was used to detect road markings at intersections, any data set must have the vehicle pass at least one intersection that was detected. Ideally, the vehicle would also stop at the intersection to enable the road marking to be continually observed. In a moving vehicle, a marking is typically observable in one or two frames only, which is not sufficient to determine if the use of vision provides any significant benefit to localization. Furthermore, there could not be any major image saturation resulting from direct sunlight while the vehicle was near an intersection, as this would make it more difficult to detect the road markings. Also, intersections where the paint of the road markings had faded significantly were avoided.
4. Availability of experimental GPS data; this receiver was less sensitive to its environment than the ground truth receiver, however it was still prone to occasional outages.

Of the 40 minutes of driving data acquired, only six minutes of data met these conditions. These six minutes were split amongst two trips: one containing 129 seconds of data; the other 230 seconds of data. The former shows the vehicle stopping at an intersection for approximately 60 seconds, during which time the experimental GPS receiver provided position measurements with minimal error. The latter shows the vehicle stopping at two intersections for approximately 10 and 17 seconds, respectively. During the second of these stops, the experimental GPS receiver showed significant measurement bias of up to 30 meters. These two data sets are used to evaluate the proposed system.

Having identified usable sets of data, it was necessary to determine what the ground truth data associations were for these two data sets. The ground truth road segment was found by matching the ground truth vehicle location with the closest road segment; any mistakes made by this process were corrected manually. Vehicle orientation was established knowing the sequence of road segments traveled by the vehicle. Detected landmarks were manually identified by the road segment they are located on, and by which of the two endpoints of that segment is the one detected; therefore a set $o_t = \{r_{t,o}, \gamma_{t,o}\}$ is used to identify the landmark. False detections and images without landmarks are assigned an empty set association, $o_t = \emptyset$.

GPS position measurements typically have a latency period between when signals are sent by the satellites to when the computed position is available. It was necessary to account for the latency between the GPS position measurement and the corresponding image acquired from the camera, so that fusion occurs only between data acquired at the same time instant. This was found by manually adjusting the amount of latency until the ground truth vehicle position and its corresponding image were consistent with each other. For example, both sources would be considered consistent when they indicate that the vehicle comes to a stop at the same time. The latency period between the time stamp on an image and its corresponding GPS measurement was approximately 7 seconds for both data sets. No latency was observed between the position measurements provided by the two GPS receivers.

6.3 System Components

Data sets having been created and annotated with ground truth data associations, testing can now be performed on this data. The following discusses the components

of the system architecture (Figure 3.7) used in testing.

6.3.1 Road Network Database

The map database used for localization is a single-line road network layer of the Region of Waterloo. The Region has made this data available to the University of Waterloo community for research purposes. It contains municipal and private roads within the Region as of 2001. The road network database is stored in a Shapefile format, developed by the Environmental Systems Research Institute.

The road network is represented by a database of road segments. Each entry in the database is one segment, representing the centerline of a road between its two endpoints defined by intersections with other road segments or by dead-ends. Each road segment is a piecewise-linear curve defined by a series of coordinates in the Universal Transverse Mercator coordinate system (see Figure 4.1).

The information contained in the road network database is used to model the location of the intersection road markings detected using vision. As road segments are modeled by their centerlines, where two such centerlines meet defines the center of an intersection. Knowing that intersections are designed and built to specified dimensions, including the location of road markings (given in [45] for the Province of Ontario, Canada), the approximate locations for these markings on individual road segments can be determined knowing the center point of the intersection.

The road marking detection algorithm does not distinguish between the three horizontal lines typically present at the entrance to an intersection (two crosswalk lines and one stop bar). Instead it returns an estimate of the distance to the marking nearest to the vehicle. A single point is used to represent the average location of these markings. These points are located on the centerlines of the intersecting road segments, indicating the average distance from the center point of the intersection to the markings. This distance is modeled as (6.1), where $W_l = 4\text{m}$ is the width of one lane, $W_c = 2.5\text{m}$ is the width of the crosswalk, $W_b = 1.2\text{m}$ is the width of the bike lane, and $W_x = 0.6\text{m}$ is “extra” width representing the gap between the lane boundary and the crosswalk or between the middle of the intersection and the first lane (for example, when there is a boulevard in the middle of the road).

$$D_o = 2W_l + W_c + W_b + W_x \quad (6.1)$$

The final result is that the locations of intersection road markings are approximated as shown in Figure 4.4. This is kept constant throughout the investigation, as all of the intersecting roads observed in the data sets contain two lanes in each direction, bike lanes and crosswalks.

6.3.2 Vision Module

The Vision module analyzes the images acquired from the camera to detect the horizontal road markings at intersections. These features were chosen because they are easy to detect and have a fixed position in the world, meaning they can be used as landmarks to localize the vehicle in an absolute sense. The following describes how these markings are detected, and how the distance from the camera to the detected marking is estimated.

Road Markings Detector

The following simple procedure is used to detect intersection road markings. It takes advantage of their horizontal orientation, uses the contrast of the light paint on the dark road surface to remove false candidates, and gives preference to lines detected closer to the vehicle.

1. A region of interest (ROI) is established limiting the search to the area of the image containing the road. This is established *a priori* since the camera is assumed to be rigidly mounted to the vehicle and does not have moving parts.
2. A Canny edge detector produces a binary image of the significant gray-level transitions within the ROI.
3. A Hough transform is used to detect straight lines in the edge image. Significant peaks in the Hough space are considered candidates for road markings provided the angular orientation is within a predefined range, nominally $\pm 5^\circ$ from horizontal.
4. Candidate lines are examined systematically to determine if a horizontal road marking is present.

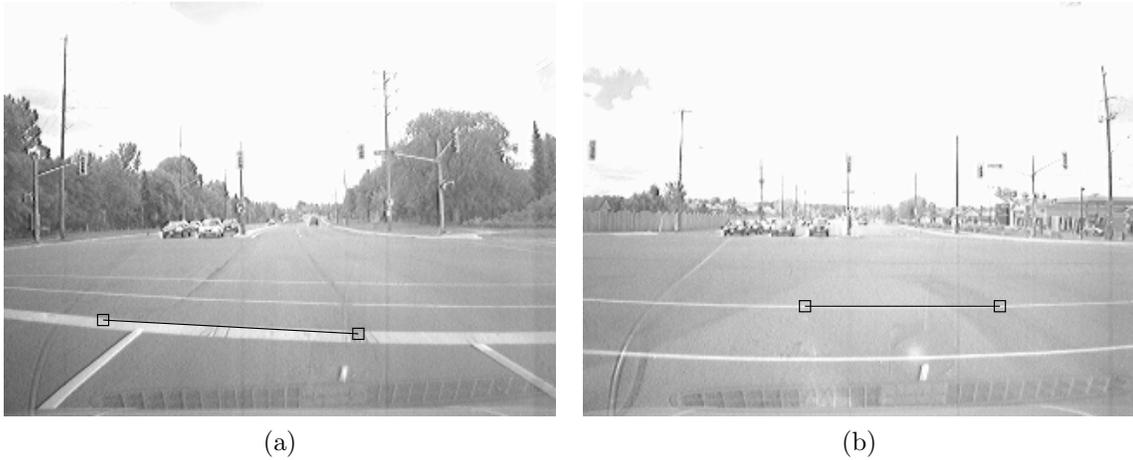


Figure 6.2: Examples of successful road marking detections in overcast conditions.

- (a) Candidates near the bottom of the image (which would correspond to markings closer to the vehicle) are examined first, by sorting the list of candidates by vertical image location.
- (b) A series of vertical image gradient profiles are examined at regular intervals along each candidate line to detect dark-light-dark transitions characteristic of a bright marking on dark pavement.
- (c) A majority voting scheme is used to confirm the presence of a road marking; if the majority of tested profiles have the expected transitions, a line is present. If none of the candidate lines pass this test, the algorithm terminates without a detection.

The road marking detector is most successful in overcast conditions, where sunlight does not saturate the image. Successful road marking detections in these conditions are shown in Figure 6.2. The performance of the detector in overcast conditions was evaluated on images belonging to the longer of the two data sets used for testing. A 150 second contiguous portion of this data set was isolated which did not show significant image saturation due to sunlight. Of these 150 images, 30 contained road markings which should be found by the detector. Table 6.1 shows a confusion matrix for this detector. From this matrix, the false positive rate $\alpha = \frac{2}{120}$ and false negative rate $\beta = \frac{1}{30}$ are determined, which are used by the Data Association module to weight competing hypotheses.

Examples of incorrect detections are shown in Figure 6.3. The previously-mentioned effect of image saturation by sunlight is illustrated in Figure 6.3a, where

		Actual	
		Present	Not Present
Detected	Present	29	2
	Not Present	1	118

Table 6.1: Confusion matrix for road marking detector.

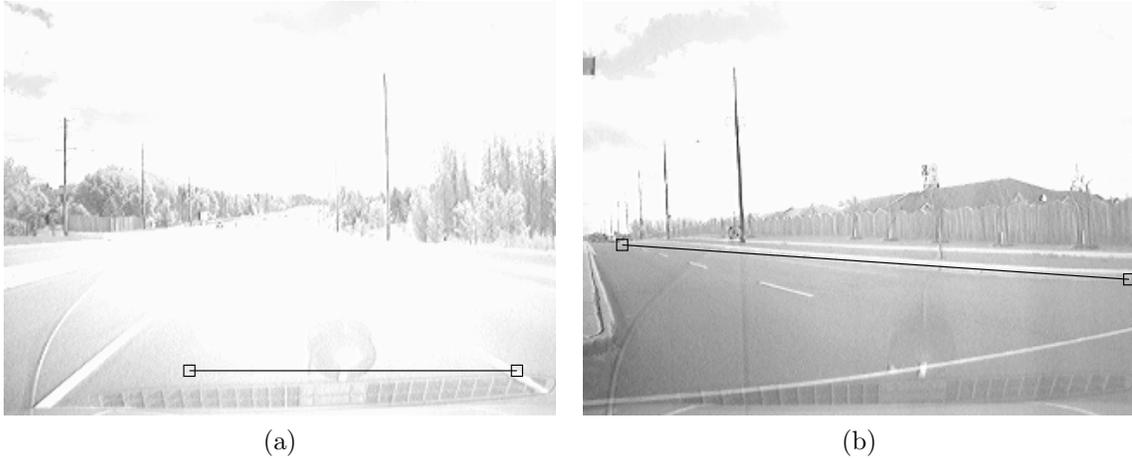


Figure 6.3: Examples of incorrect road marking detections due to a) image saturation by sunlight, and b) the vehicle heading.

a false detection occurs near the bottom of the image. Another false detection occurred as a result of the instantaneous vehicle heading, placing the road curb in a horizontal orientation in the image as illustrated in Figure 6.3b.

Distance to Road Markings

To localize the vehicle with respect to the detected intersection marking, an estimate of the distance between the vehicle and the marking is required. This will be used in conjunction with a model of the road marking locations to update the vehicle state.

The distance estimate is essentially a transformation from image coordinates to world coordinates. Examples of this type of transformation for monocular systems can be found in [8, 15]. By assuming that the road surface is planar, this transformation is simplified.

We use a method derived from [15]. The camera is mounted to the vehicle z_0

meters above the road’s surface. The camera’s optic axis is at an angle θ below the parallel with the road’s surface. The vertical image location of the detected road marking is transformed using (6.2) into an estimate of distance along the road from the camera to the marking, \hat{f}_t . In this expression, v represents the vertical image location of the detected road marking (in pixels, relative to the top of the image) and $e_v = \frac{f}{d_v}$, where f is the camera focal length and d_v is the height of a pixel.

$$\hat{f}_t = \frac{z_0 e_v (1 - \theta^2)}{v - e_v \theta} \quad (6.2)$$

For our installation of the camera, $z_0 = 1.1\text{m}$, $\theta = 20^\circ$, $f = 2.0\text{mm}$, and $d_v = 5.6\mu\text{m}$.

6.3.3 Measurement Models

The measurement models used to update the localization state are described here. These models are used for both Kalman filtering and particle filtering localization. The state vector of the system is defined as the vehicle position and velocity in the East-North plane: $x_t = \{p_{t,e} \ p_{t,n} \ v_{t,e} \ v_{t,n}\}$. When using the road coordinate system defined in Section 4.1.1 the state is expressed in the lateral and longitudinal directions relative to the road segment: $x_t^r = \{p_{t,\perp}^r \ p_{t,\parallel}^r \ v_{t,\perp}^r \ v_{t,\parallel}^r\}$. The covariance matrix is denoted P_t and P_t^r in world coordinates and road coordinates, respectively.

GPS Measurement Model

A linear-Gaussian measurement model is used for the GPS position measurement. GPS provides a direct measurement of the vehicle position in latitude and longitude coordinates. This measurement is first converted to the Universal Transverse Mercator coordinate system – the same used by the road network database – before it is used to update the vehicle position. Because the state vector explicitly contains the vehicle position in this coordinate system, this measurement model is extremely simple (6.3). $w_t \sim \mathcal{N}(w_t; 0, R_t)$ is the additive Gaussian noise expressing uncertainty in this model, where R_t is the measurement covariance matrix defined by (6.4). The GPGST sentence is one of the NMEA sentences supported by GPS receivers; this sentence provides the covariance matrix of the GPS position measurement in the horizontal plane, and can therefore be used to dynamically adapt the measurement uncertainty. Unfortunately, this information is not provided by

$p_{t,\parallel}^r$	ζ_t	M_t	Z_t
$0 \leq p_{t,\parallel}^r \leq L_r$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1000 \\ 0010 \end{pmatrix}$	$\begin{pmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix}$
$p_{t,\parallel}^r < 0$	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1000 \\ 0100 \\ 0010 \end{pmatrix}$	$\begin{pmatrix} \sigma_p^2 & 0 & 0 \\ 0 & \sigma_p^2 & 0 \\ 0 & 0 & \sigma_v^2 \end{pmatrix}$
$p_{t,\parallel}^r > L_r$	$\begin{pmatrix} 0 \\ L_r \\ 0 \end{pmatrix}$		

Table 6.2: Pseudomeasurement parameters ζ_t , M_t and Z_t as a function of longitudinal vehicle position $p_{t,\parallel}^r$.

the GlobalSat BU-353 receiver, and therefore the constant value of $\sigma_{gps} = 10\text{m}$ is set based on the receiver's specified RMS horizontal position accuracy.

$$\begin{aligned} \hat{p}_t &= H_t x_t + w_t \\ &= \begin{pmatrix} I_2 & 0_2 \end{pmatrix} x_t + w_t \end{aligned} \quad (6.3)$$

$$R_t = \begin{pmatrix} \sigma_{gps}^2 & 0 \\ 0 & \sigma_{gps}^2 \end{pmatrix} \quad (6.4)$$

Map Matching Measurement Model

A pseudomeasurement ζ_t is used to perform map matching. The purpose of this measurement update is to adjust the localization estimate to be closer to the road segment being driven, and to align the heading of the vehicle with the road segment. Map matching is performed in road coordinates, therefore the measurement model is a function of x_t^r (6.5). $v_t \sim \mathcal{N}(v_t; 0, Z_t)$ is the additive Gaussian noise expressing uncertainty in this model.

$$\zeta_t = M_t x_t^r + v_t \quad (6.5)$$

The value of this pseudomeasurement and the form of its model depend on the longitudinal position $p_{t,\parallel}^r$. The reasons for this are given in Section 4.2.2 and not repeated here. Table 6.2 provides the value of ζ_t , M_t and Z_t as a result of the value of $p_{t,\parallel}^r$. L_r is the length of the road segment.

Vision Measurement Model

Vision provides a measurement of distance to the detected road markings. This measurement is modeled as a function of the longitudinal vehicle position $p_{t,\parallel}^r$ and

the location of the detected landmark $p_{o,\parallel}^r$ (6.6), as the difference between the two is equal to the distance from the landmark to the vehicle. The orientation γ_t ensures that this predicted distance is always positive whenever the landmark is in front of the vehicle.

$$\hat{f}_t = (p_{o,\parallel}^r - p_{t,\parallel}^r) \gamma_t + \omega_t \quad (6.6)$$

The error in this model is given by an additive Gaussian noise term $\omega_t \sim \mathcal{N}(\omega_t; 0, \sigma_f^2)$. The value of $\sigma_f = 2\text{m}$ that is used is larger than the value found experimentally for this distance estimator of 1.1m. This is chosen to account for the uncertainty in which road marking has been detected (an intersection normally has three such lines, two crosswalk lines and one stop bar, within a distance of approximately 3.5m), and the approximate location of the landmark $p_{o,\parallel}^r$ for which a standard model has been used for all intersections.

6.4 Experiment 1: Best-case localization performance

The objective of this experiment is to evaluate the localization performance of the proposed system. In particular, I would like to determine if an improvement in localization can be expected when vision is added to the system, in the best-case scenario that data associations are known *a priori*.

Localization performance is evaluated using the following two criteria.

- Localization accuracy is determined by the mean localization error.
- Localization consistency is determined by the standard deviation of the localization error: the lower the standard deviation, the more consistent is the localization error.

This experiment is performed using what will be called Data Set 1 from now on. This is the longer of the two data sets, containing an instance of GPS measurement bias. The path taken by the vehicle in this data set is shown in Figure 6.4.

The mean localization error for Data Set 1 is shown in Figure 6.5 when using only Passive Localization (PL), Passive Localization and Map Matching (MM) and finally the full measurement update using Passive Localization, Map Matching and



Figure 6.4: Path driven during collection of Data Set 1.

Visual Context (VC). Error bars of length one standard deviation are used to show how consistent the localization estimates are. The error is divided into the lateral and longitudinal directions to better illustrate the effect of adding one module to the system.

Using only Passive Localization gives similar error in both directions as the raw GPS measurements (GPS measurement error is shown as the yellow line). This is expected since the PL module uses these measurements to track the vehicle location.

Adding the Map Matching module to the system maintained a similar level of error in the longitudinal direction, however lateral error was increased. The fact that lateral error is affected the most can be attributed to map matching primarily adjusting the lateral position of the vehicle to bring it closer to the centerline. The increase in error is due to the fact that the vehicle does not drive directly on the road centerline, but rather offset from the centerline within a lane.

Adding the Visual Context module reduces the error in the longitudinal direction compared with PL, MM and the raw GPS measurements; lateral error was maintained at the same level as MM. This result can be explained by a similar argument as above; since visual context is used to adjust the longitudinal position of the vehicle with respect to a landmark, lateral position is unaffected by this

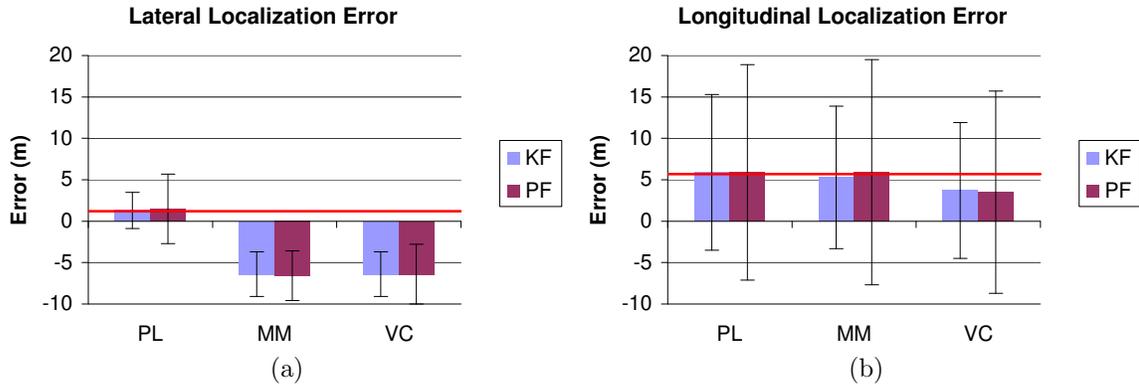


Figure 6.5: Localization accuracy for known data association case, using Data Set 1.

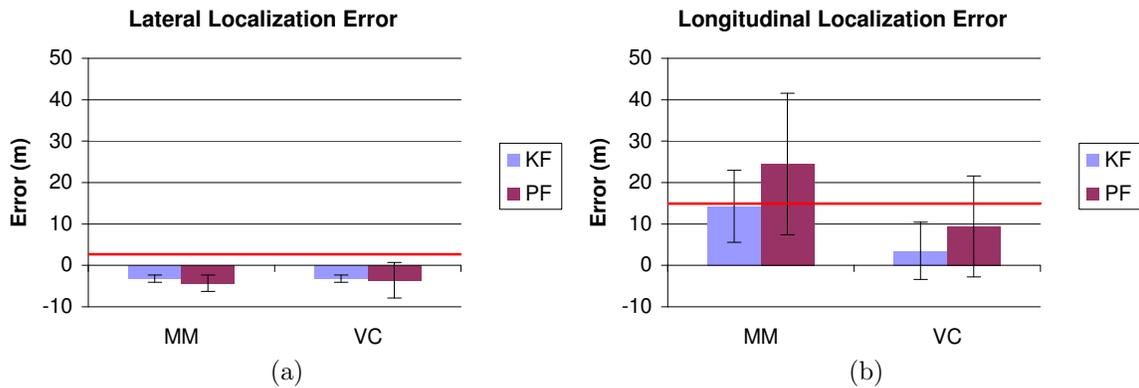


Figure 6.6: Localization accuracy for known data association case while road markings are visible, using Data Set 1.

update.

One interesting observation of these results is that similar error is achieved using both Kalman filtering and particle filtering. The difference between the two is the consistency of this error, as indicated by the standard deviations. In all cases shown in Figure 6.5, particle filtering errors were less consistent than Kalman filtering. This can be attributed to the periodic resampling of the particle filter, which allows the particle cloud to grow for two iterations before it is contracted by resampling.

The results shown in Figure 6.5 are for the entire data set, however vision is only able to detect landmarks when the vehicle is near an intersection. Figure 6.6 shows the error in the lateral and longitudinal directions during times when landmarks were visible. These results show that the addition of vision dramatically reduces

the error in the longitudinal direction, using either Kalman or particle filtering. Localization is also more consistent using vision, as seen by the smaller standard deviation on the longitudinal errors. During these times, the GPS measurements also contained a large bias, as shown by the average longitudinal GPS error being 14.9m. To reiterate, it is important to keep in mind the assumption that data associations are known.

Based on these results, I am confident in saying that adding vision to a GPS-based localization system has the capability to improve localization accuracy, provided data association parameters are known.

6.5 Experiment 2: Data Association in Unbiased GPS Conditions

This experiment aims to evaluate the localization performance of the proposed system in unknown data association conditions, and the success of identifying the correct data association hypothesis. In particular, I am interested in the less challenging situation where the GPS measurements are not affected by significant bias.

Localization performance is evaluated as before using the mean and standard deviation of the position error. Data association success is evaluated by calculating the mean and standard deviation of the weight assigned to the correct data association hypothesis. A correct hypothesis is one that has the same road segment and orientation as the ground truth values. By not considering landmark identity, it is possible to compare data association with and without vision. This could therefore be considered map matching.

This experiment is performed using what will be called Data Set 2 from now on. This is the shorter of the two data sets, containing unbiased GPS position measurements. The path taken by the vehicle in this data set is shown in Figure 6.7.

Figure 6.8 shows the localization error and data association success for Data Set 2. The mean localization error is shown using Passive Localization and Map Matching (MM) system modules, and the full measurement update using Passive Localization, Map Matching and Visual Context (VC). Error bars of length one standard deviation are again used to show the consistency of the localization estimates. The error is divided into the lateral and longitudinal directions to better illustrate the effect of adding the Visual Context module to the system. The mean

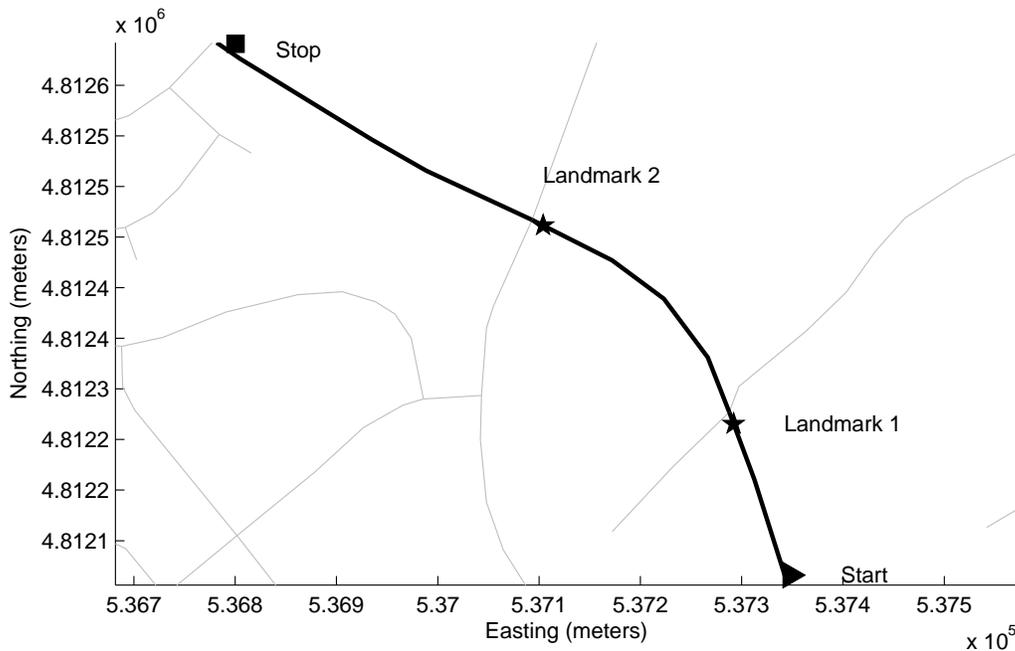


Figure 6.7: Path driven during collection of Data Set 2.

weight assigned to the correct hypothesis when using the MM and VC filter implementations is shown in Figure 6.8c.

These results show that localization accuracy is consistent with that found using known data associations. The lateral position error (Figure 6.8a) shows the introduction of a bias due to matching to the road centerlines. The longitudinal position error (Figure 6.8b) is not significantly different from the GPS measurement error in this case, however this is unsurprising as these measurements were unbiased. Results using Kalman filtering with Multiple Hypothesis Tracking (MHT) and particle filtering with Monte Carlo Data Association (MCDA) are similar in terms of mean position error. MHT shows more consistent position error than MCDA, having smaller standard deviations in almost all cases. This agrees with the known data association results as well, where KF was more consistent than PF.

The benefit of adding vision in this case is seen in the success of identifying the correct data association. Figure 6.8c shows that the weight assigned to the correct data association hypothesis is higher when Visual Context was used compared to Map Matching. This can be attributed to higher measurement likelihood being assigned to the correct data association hypothesis, because a valid landmark was only identified for this hypothesis. As a result, the use of vision provides evidence of the vehicle’s position on the road network, even if in this case it does not improve

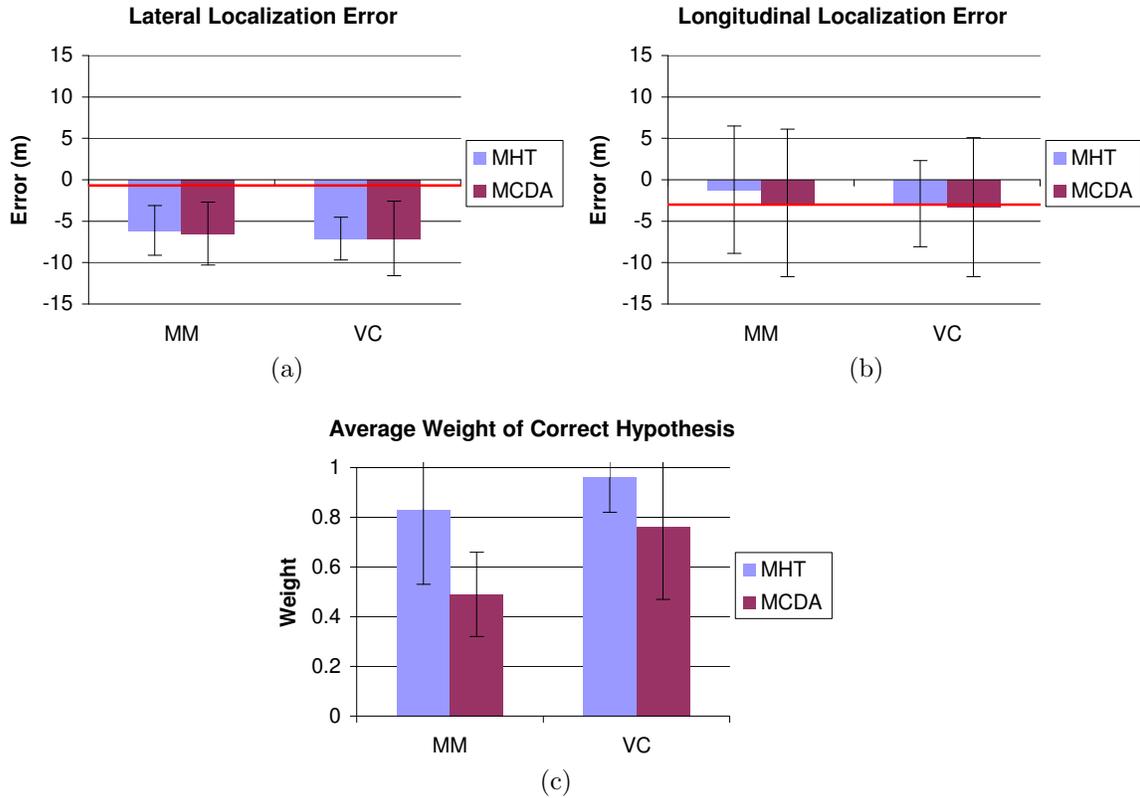


Figure 6.8: Localization accuracy and data association success in unbiased GPS conditions, using Data Set 2.

upon the localization accuracy of GPS.

MHT consistently places a higher weight on the correct hypothesis than MCDA in these results. The cause of this is that MCDA will always assign unlikely data associations to some of its particles. Conversely, MHT prunes unlikely hypotheses to maintain filter tractability. The ability to consider unlikely hypothesis is a benefit of MCDA because it adds to the robustness of the filter – as will be examined next – however the cost of this robustness is that these particles take weight away from the correct hypothesis in trivial data association situations.

6.6 Experiment 3: Data Association in Biased GPS Conditions

This experiment seeks to evaluate the localization performance of the proposed system when data associations are unknown and the GPS position measurements

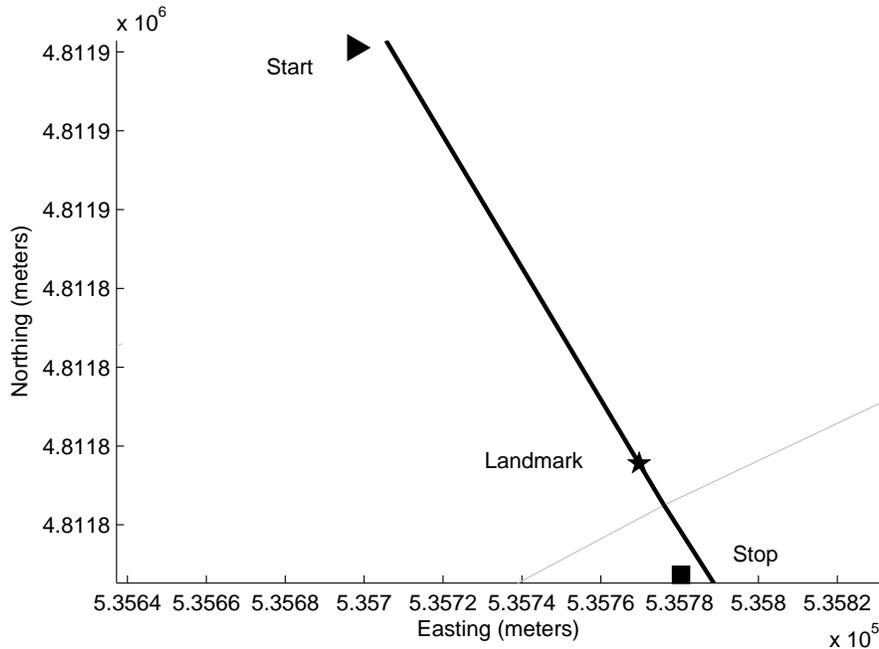


Figure 6.9: Path driven during the portion of Data Set 1 exhibiting GPS measurement bias.

exhibit significant bias. I am interested in determining how robust the proposed filters are to the biased measurements, and whether the same data association success can be achieved as in the unbiased case.

Three filters are used in this evaluation: the standard MHT formulation proposed in Section 5.1.2, the tuned MHT filter proposed in Section 5.1.4 to improve robustness to GPS measurement bias, and the MCDA method based on particle filtering proposed in Section 5.2.2. All filters use the complete system architecture, meaning that Visual Context is always included. Localization performance is evaluated as before using the mean and standard deviation of the position error. Data association success is again evaluated by calculating the mean and standard deviation of the weight assigned to the correct data association hypothesis.

This experiment is performed using the portion of Data Set 1 with significant GPS measurement bias; the vehicle path for this portion of Data Set 1 is shown in Figure 6.9. Figure 6.10 shows the localization error and data association success during this time period. The standard MHT is identified by ‘MHT-1’ and the tuned MHT by ‘MHT-2.’

These results show that the tuned MHT filter is superior to the standard MHT and MCDA filters in terms of longitudinal position error (Figure 6.10b), and in

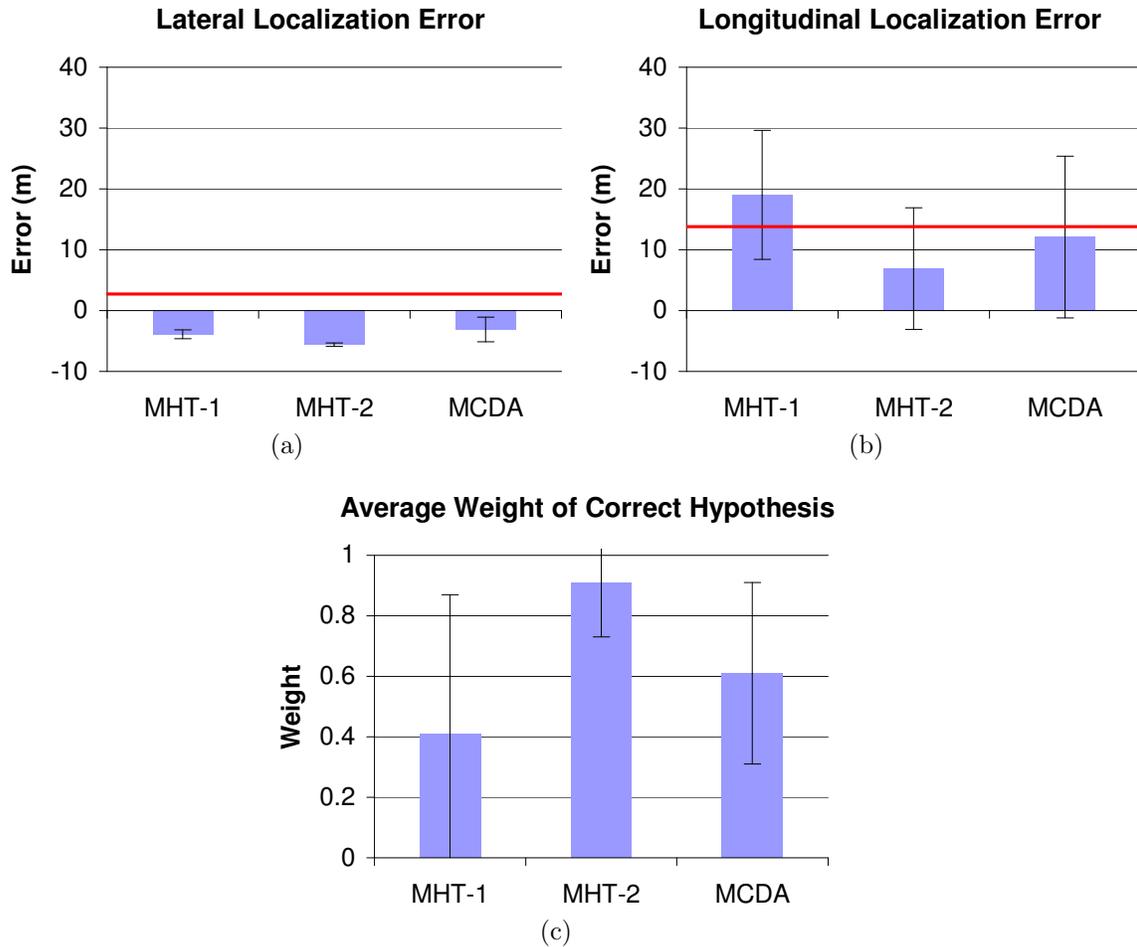


Figure 6.10: Localization accuracy and data association success in biased GPS conditions, using the portion of Data Set 1 exhibiting GPS measurement bias.

identifying the correct hypothesis (Figure 6.10c). All three filters show similar lateral position error (Figure 6.10a).

This result should not be surprising since the MHT was tuned to work for this specific data set. What is more interesting is how the MCDA filter, with no extra tuning, performs better than the standard MHT in terms of longitudinal error and in identifying the correct hypothesis. This confirms that the MCDA method is a more robust localization approach than MHT to GPS measurement biases.

More testing is needed to determine the best data association approach for vehicle localization. The MHT method has advantages in that it can use constraints of the road network to keep the number of hypotheses low, and methods of tuning have already proven effective against GPS measurement bias. The MCDA approach shows more robustness to GPS measurement bias without the need for filter tuning

and is in many ways a simpler approach. Using more intelligent proposal distributions and resampling methods to guide the particle set to the correct region of the state space may prove promising for achieving a robust vehicle localization system.

6.7 Summary

This chapter presents the results of the experimental investigations performed within this thesis. The experimental setup is described, the data acquisition procedure explained, and a list of criteria given that were used to select appropriate data sets for testing.

Two data sets are used to test the proposed system. Data Set 1 exhibits significant GPS measurement bias, while Data Set 2 is without bias. Testing on Data Set 1 shows that the addition of vision to a localization system enables this bias to be corrected, provided that data association parameters are known. Kalman filtering and particle filtering methods gave similar results in this evaluation. These are best-case results and would not be applicable in most situations, as it will not be known ahead of time what objects are detected by the vision system.

Testing on Data Set 2 shows that vision can help to position the vehicle on the road network, by improving the ability to determine which road the vehicle is driving on. Data associations were unknown in this case, and GPS measurements unbiased. The ambiguity created by considering multiple data association hypotheses was lessened by the addition of vision, and localization accuracy was maintained.

Data Set 1 was used to evaluate the ability to use vision to correct for GPS measurement bias, this time with unknown data associations. A multiple hypothesis tracking (MHT) approach tuned for this purpose gave accuracy similar to the known data association case, and was successful in identifying the correct road segment. However, a Monte Carlo Data Association (MCDA) method based on particle filtering showed an inherent robustness to GPS measurement bias, and is a promising avenue for further research.

Chapter 7

Mutual Influence Between Localization and Visual Feature Estimation

Until now this thesis has focused on the use of vision to influence the result of GPS-based vehicle localization. This chapter looks at the complementary (and, as it happens, related) problem of using position knowledge to influence the visual feature measurements. By using the map database and the visual feature measurement model, the vehicle position estimated using GPS can be used to predict the value of the measured visual features. Fusing this with the measured values provides a means for adding position context to the visual feature estimates, in an effort to make them consistent with the vehicle position.

This chapter specifically examines how to add this capability to the developed localization architecture, to enable the simultaneous adjustment of position using visual features and *vice versa*. This is analogous to the idea of arc consistency [57]. Arc consistency ensures that two variables connected by a constraint have domains consistent with that constraint. This involves pruning values separately from each variable's domain to ensure that the constraint between the two variables is always satisfied. A simple example is shown in Figure 7.1. The analogy with arc consistency comes from the separate consideration of each variable's domain in light of the other's.

The motivation behind this investigation is to provide vehicle systems with improved environmental awareness in addition to improved location awareness. Systems where this may be applicable would include:

- Autonomous driving systems, which visually measure and follow the curvature

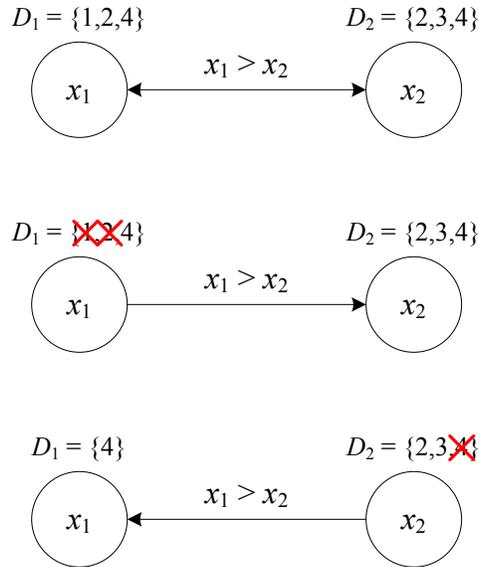


Figure 7.1: Example of arc consistency. The domains of x_1 and x_2 are pruned separately of values that do not satisfy the constraint $x_1 > x_2$.

of the road [20, 15].

- Vehicle rollover warning systems, which use factors such as the lateral road curvature and road cross-elevation to generate an indication of vehicle rollover potential [55].
- Navigation systems capable of visually identifying their destination.

The remainder of this chapter develops an EKF formulation for mutual influence and integrates it within the developed architecture. Experimental results and discussion follow.

7.1 A Mutual Influence Formulation using Kalman Filtering

7.1.1 A Dynamic Bayesian Network Model

The mutual influence problem is modeled using the DBN in Figure 7.2a, with that for the localization problem shown in Figure 7.2b for comparison. The difference between these two models is the addition of a feature state f_t , upon which the feature measurements \hat{f}_t are now dependent, as opposed to being dependent on the

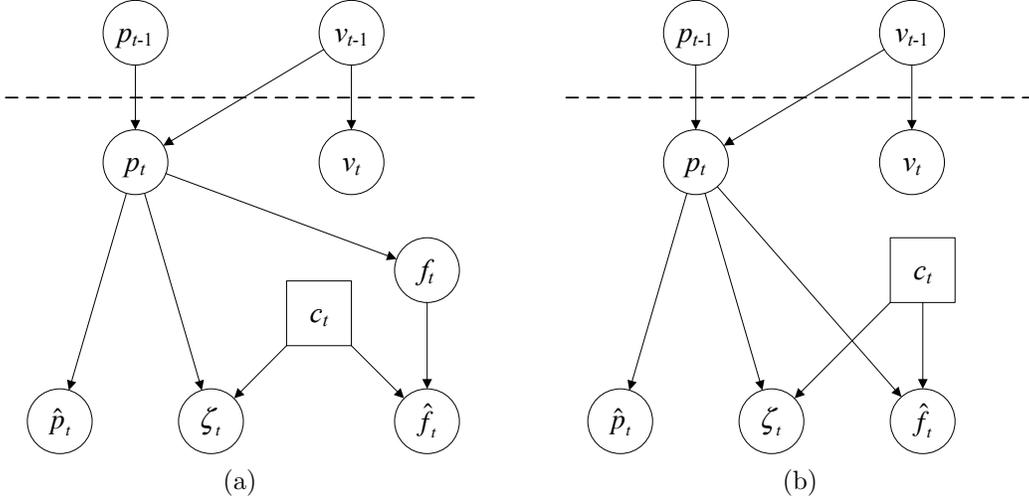


Figure 7.2: Dynamic Bayesian Network models for a) the mutual constraint problem, and b) the localization problem.

position state p_t . It was also decided to have the feature state f_t be dependent on the position state p_t in order to simplify the prediction of feature values by using available feature models. The principal reason for this is that features may cease to be visible or become visible between time steps, making it difficult to use previous feature values to predict new feature values. It is therefore being assumed that what features are visible is a direct consequence of where the vehicle is located.

The connections within the DBN indicate how observation of certain variables (through measurements) will affect other variables in the network [49]. For example, the measured visual features \hat{f}_t affect the feature state f_t and the position state p_t through the serial connection $p_t \rightarrow f_t \rightarrow \hat{f}_t$. Evidence for \hat{f}_t propagates up to the other variables in the serial connection, as they all can provide an explanation for the observed value. Similarly, the measured vehicle position \hat{p}_t affects the position state p_t through the serial connection $p_t \rightarrow \hat{p}_t$. The feature state f_t is also affected by this measurement through the diverging connection $\hat{p}_t \leftarrow p_t \rightarrow f_t$; since \hat{p}_t provides knowledge of p_t , this knowledge propagates to f_t due to its dependence on p_t [49].

Therefore, it is known from the network topology that measured features \hat{f}_t will affect the position state p_t , and that measured position \hat{p}_t will affect the feature state f_t . This makes using DBNs attractive for modeling, since without making assumptions about the distribution of variables or the inference method, it can be determined which variables will have influence on others from the network topology.

7.1.2 Integration into Modular System Architecture

We can write without loss of generality the filtering posterior for the above DBN. We wish to infer the values of vehicle position p_t , vehicle speed v_t and visual features f_t based on the measurements $\hat{p}_{1:t}$, $\zeta_{1:t}$ and $\hat{f}_{1:t}$ and any data associations $c_{1:t}$. This is identical to the localization posterior except for the addition of f_t to the state. The expanded state vector will be denoted x'_t to distinguish it from the localization state which will remain x_t (7.1).

$$\begin{aligned} x'_t &= \{p_t \ v_t \ f_t\} \\ &= \{x_t \ f_t\} \end{aligned} \tag{7.1}$$

$$z_t = \{\hat{p}_t \ \zeta_t \ \hat{f}_t\} \tag{7.2}$$

The posterior for the filtering problem is written in (7.3), dropping the association $c_{1:t}$ for the time being. η represents a normalization term.

$$p(x'_t|z_{1:t}) = \eta p(z_t|x'_t)p(x'_t|z_{1:t-1}) \tag{7.3}$$

The predicted posterior $p(x'_t|z_{1:t-1})$ is found using the Chapman-Kolmogorov equation in (7.4) [4].

$$p(x'_t|z_{1:t-1}) = \int p(x'_t|x'_{t-1})p(x'_{t-1}|z_{1:t-1})dx'_{t-1} \tag{7.4}$$

Recognizing from the DBN in Figure 7.2a that the feature state f_t is dependent on the position state p_t , the state evolution probability $p(x'_t|x'_{t-1})$ can be rewritten as in (7.5). This shows the state evolution probability for the localization state x_t and an extra term $p(f_t|x_t)$ denoting the dependence of f_t on p_t .

$$\begin{aligned} p(x'_t|x'_{t-1}) &= p(x_t, f_t|x_{t-1}f_{t-1}) \\ &= p(f_t|x_t)p(x_t|x_{t-1}) \end{aligned} \tag{7.5}$$

The Chapman-Kolmogorov equation (7.4) can now be rewritten as (7.6). Integrating over f_{t-1} and factoring out $p(f_t|x_t)$ reduces this expression to (7.7).

$$p(x'_t|z_{1:t-1}) = \int p(f_t|x_t)p(x_t|x_{t-1})p(x_t, f_{t-1}|z_{1:t-1})dx_{t-1}df_{t-1} \tag{7.6}$$

$$= p(f_t|x_t)p(x_t|z_{1:t-1}) \tag{7.7}$$

7.1.3 Extended Kalman Filter Inference

For simplicity and to be consistent with previous work on localization, an EKF is used to perform inference on the DBN. This requires assumptions of Gaussianity and (approximate) linearity in the system dynamics and measurements.

Keeping the notation the same, x_t will denote the localization state (vehicle position and speed) and x'_t will denote the full state (localization state plus feature state). The localization state prior to the visual update will be denoted $x_{t,-}$. A similar notation is used to represent the state covariance P .

Expanding the State

Expanding the state from the localization state x_t to the full state x'_t is a mapping from 4 dimensions to $4 + N$ dimensions, where N is the number of features. This mapping is denoted by the function $b_t(\cdot)$ (7.10), and uses feature models $F_1 \dots F_N$ for each of the feature measurements, similar to the distance feature used in Section 4.1.

$$\begin{aligned} x'_t &= b_t(x_t) \\ &= \begin{bmatrix} x_t \\ F_1(x_t) \\ \vdots \\ F_N(x_t) \end{bmatrix} \end{aligned} \quad (7.10)$$

Similarly, the covariance matrix P_t is mapped to the full state covariance P'_t . This is achieved by first linearizing the mapping $b_t(\cdot)$ using the Jacobian evaluated at x_t (7.11). The linearization of the feature models F_i results in a matrix C_t that is analogous to the Jacobian G_t from (4.29) in Section 4.2.3. The Jacobian B_t is then used to expand the covariance matrix (7.12).

$$\begin{aligned} B &= \frac{\partial b_t}{\partial x_t} \\ &= \begin{bmatrix} I_{4 \times 4} \\ \frac{\partial F_1}{\partial p_t} & \frac{\partial F_1}{\partial v_t} \\ \vdots & \vdots \\ \frac{\partial F_N}{\partial p_t} & \frac{\partial F_N}{\partial v_t} \end{bmatrix} = \begin{bmatrix} I_{4 \times 4} \\ C_{t, N \times 4} \end{bmatrix} \end{aligned} \quad (7.11)$$

$$\begin{aligned}
 P'_t &= B_t P_t B_t^T \\
 &= \begin{bmatrix} P_t & P_t C_t^T \\ C_t P_t & C_t P_t C_t^T \end{bmatrix}
 \end{aligned} \tag{7.12}$$

Updating the Full State using Feature Measurements

From the expanded state, the modeling of feature measurements is quite straightforward and can be achieved using a linear model (7.13), where $\omega_t \sim \mathcal{N}(0, \Omega_t)$.

$$\begin{aligned}
 \hat{f}_t &= H x'_t + \omega_t \\
 &= \begin{bmatrix} 0_{N \times 4} & I_{N \times N} \end{bmatrix}
 \end{aligned} \tag{7.13}$$

Using this model H , the Kalman gain K_t can be computed for the visual update (7.14). The matrix S_t is the innovation covariance matrix (7.15), where Ω_t is the measurement uncertainty. K_t can be seen as two distinct gain terms: $K_{t,L}$ that affects the localization state, and $K_{t,F}$ that affects the feature state. The feature gain is also related to the localization gain as $K_{t,F} = C_t K_{t,L}$. The localization gain $K_{t,L}$ is identical to the Kalman gain that would be computed for a localization-only filter.

$$\begin{aligned}
 K_t &= P'_t H^T S_t^{-1} \\
 &= \begin{bmatrix} P_t & C_t^T & S_t^{-1} \\ C_t & P_t & C_t^T & S_t^{-1} \end{bmatrix} = \begin{bmatrix} K_{t,L} \\ K_{t,F} \end{bmatrix}
 \end{aligned} \tag{7.14}$$

$$S_t = H P'_t H^T + \Omega_t = C_t P_t C_t^T + \Omega_t \tag{7.15}$$

The updated full state vector using this Kalman gain is shown in (7.16). This equation indicates that the feature state update can be derived from the localization state update knowing the predicted feature values $F(x_{t,-})$ and the Jacobian C_t .

$$\begin{aligned}
 x'_t &= x'_{t,-} + K_t \left(\hat{f}_t - H x'_{t,-} \right) \\
 &= \begin{bmatrix} x_{t,-} + K_{t,L} \left(\hat{f}_t - F(x_{t,-}) \right) \\ F(x_{t,-}) + C_t K_{t,L} \left(\hat{f}_t - F(x_{t,-}) \right) \end{bmatrix}
 \end{aligned} \tag{7.16}$$

Similarly, the covariance update for the feature state can be found knowing the localization gain $K_{t,L}$, the Jacobian C_t and the prior localization covariance

Algorithm 7.1 Pseudocode for adding mutual influence to an existing routine.

```

1. function KFVisionUpdateMutualInfluence(  $x_{t,-}, P_{t,-}, \hat{f}_t, c_t$  )
2.
3.  $f_{t,-} = \text{FeatureModel}( x_{t,-}, c_t )$ 
4.  $C_t = \text{Jacobian}( \text{FeatureModel}, x_{t,-}, c_t )$ 
5.
6.  $S_t = C_t P_{t,-} C_t^T + \Omega_t$  // Innovation covariance
7.  $K_t = P_{t,-} C_t^T S_t^{-1}$  // Kalman gain
8.
9. // Localization update
10.  $x_t = x_{t,-} + K_t (\hat{f}_t - f_{t,-})$ 
11.  $P_t = (I - K_t C_t) P_{t,-}$ 
12.
13. // Feature update
14.  $f_t = f_{t,-} + C_t K_t (\hat{f}_t - f_{t,-})$ 
15.  $P_{F,t} = (I - C_t K_t) C_t P_{t,-} C_t^T$ 
16.
17. return  $x_t, P_t, f_t, P_{F,t}$ 

```

$P_{t,-}$. This is seen in (7.17) which shows the covariance update for the full state (the correlation terms between the localization state and feature state have been omitted for clarity).

$$\begin{aligned}
P'_t &= P'_{t,-} - K_t H P'_{t,-} \\
&= \begin{bmatrix} (I - K_{t,L} C_t) P_{t,-} & \\ & (I - C_t K_{t,L}) C_t P_{t,-} C_t^T \end{bmatrix} \quad (7.17)
\end{aligned}$$

Implementation Within an Existing Routine

The significance of the feature update being so closely related to the localization update is that it becomes much simpler to implement mutual influence within an existing localization system. Algorithm 7.1 shows in pseudocode how this can be added with minimal effort to a routine designed to perform visual update to the localization state. Lines 14-15 are the only additions to this existing routine required to perform the feature update.

7.2 Experimental Investigation

The mutual influence code was added to the routine used for Kalman filter localization under known data association conditions, given in Section 4.2. The primary reason for adding it to this routine versus the MHT routine in Section 5.1 is to avoid ambiguity associated with unknown data association. It will be easier to see the effect of updating feature estimates using position without the ambiguity associated with multiple vehicle location or landmark hypotheses.

The localization routine with mutual influence is applied to Data Set 2, used previously to test data association performance in unbiased GPS conditions. A GPS receiver with a horizontal accuracy of 10 m RMS provides the estimates of vehicle position. A camera system is used to detect horizontal road markings at intersections, and to estimate the distance to these detected markings from the camera, as described in Section 6.2. The uncertainty in this distance estimate is $\sigma_f = 2\text{m}$.

One objective of this experiment was to observe if the addition of position context would improve the accuracy and consistency of the estimate of distance to a road marking. The average estimation error and its standard deviation were used to determine if estimates were more accurate and consistent, respectively. In both cases, values closer to zero would represent an improvement.

Another objective was to observe how the uncertainty in the feature estimate changed as a result of adding position context. This was achieved by comparing the difference between the measurement uncertainty $\Omega_t = \sigma_f$ and the posterior variance of the feature state $P_{t,F}$. Because the Kalman filter update contracts the posterior distribution, it should be true that $P_{t,F} < \Omega_t$, however I am interested in the magnitude of this reduction compared with how much the uncertainty in vehicle position P_t is reduced by the addition of vision.

7.2.1 Experimental Results

Position error is shown in Figure 7.4 for both lateral and longitudinal directions. Error in the visual feature estimate is shown in Figure 7.5. As the ground truth position p_t^* is used to calculate the error in the position estimate (7.18), so it is used to calculate error in the feature estimate by using the feature model F and

	$\bar{\varepsilon}$ (m)	s_{ε} (m)	$\bar{\sigma}$ (m)	s_{σ} (m)
f_t	0.029	1.4	1.8	0.024
\hat{f}_t	-0.0051	1.5	2.0	0
$p_{\parallel,VC}$	-0.029	1.4	1.8	0.024
$p_{\parallel,MM}$	-0.68	2.2	7.5	0.13
$p_{\perp,VC}$	-8.2	0.75	3.1	0.0060
$p_{\perp,MM}$	-8.2	0.75	3.1	0.0060

Table 7.1: Visual feature and localization errors showing the effect of mutual influence filtering. Comparison of filtered feature estimate using position context f_t and the raw measurement \hat{f}_t , and comparison of longitudinal (\parallel) and lateral (\perp) position errors using visual context (VC) and map matching (MM).

known data association c_t^* (7.19).

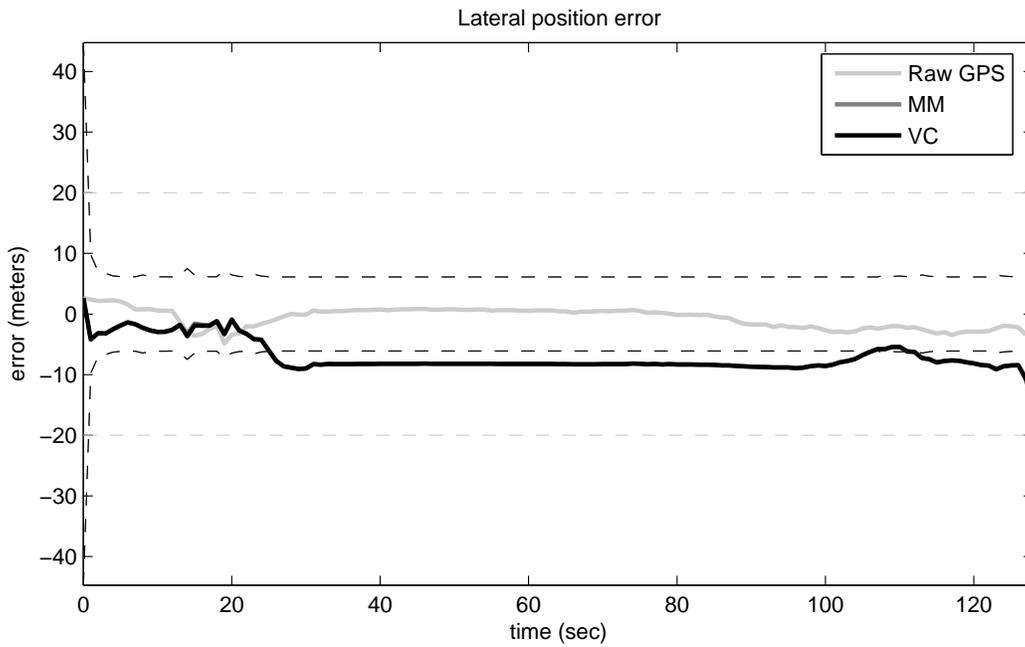
$$\epsilon_{p,t} = p_t - p_t^* \quad (7.18)$$

$$\epsilon_{f,t} = f_t - F(p_t^*, c_t^*) \quad (7.19)$$

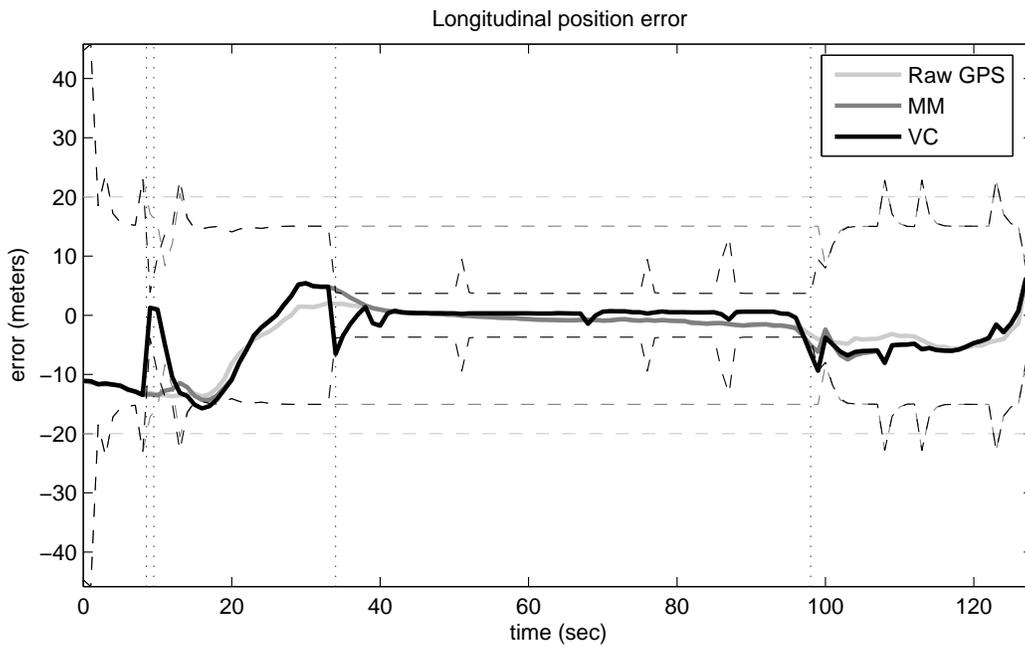
In calculating the error in the visual feature estimate this way, the assumption is being made that the feature model F is errorless and simply a mapping from the position space to the feature space. This may not be true in general as the accuracy of the model depends on the geometry of the intersection and the specific road marking being detected. However, this assumption is necessary because it is not known what the true distance from the camera to the road marking is, because no sensor was available to provide ground truth nor was it feasible to manually measure this distance while in traffic.

Table 7.1 summarizes the results of the mutual influence filter **for those time instants where vision correctly detected a landmark**. The columns marked $\bar{\varepsilon}$ and s_{ε} are the mean and standard deviation of the error in each estimated variable, and are used to illustrate the accuracy and consistency of these estimates, respectively. While the accuracy of feature estimates and longitudinal position estimates is not changed significantly, these estimates are made more consistent by the filter as shown by smaller standard deviation of the estimation error.

Another observation that can be made is that the amount of reduction in the uncertainty of the feature estimate and localization estimate is not equal; this can

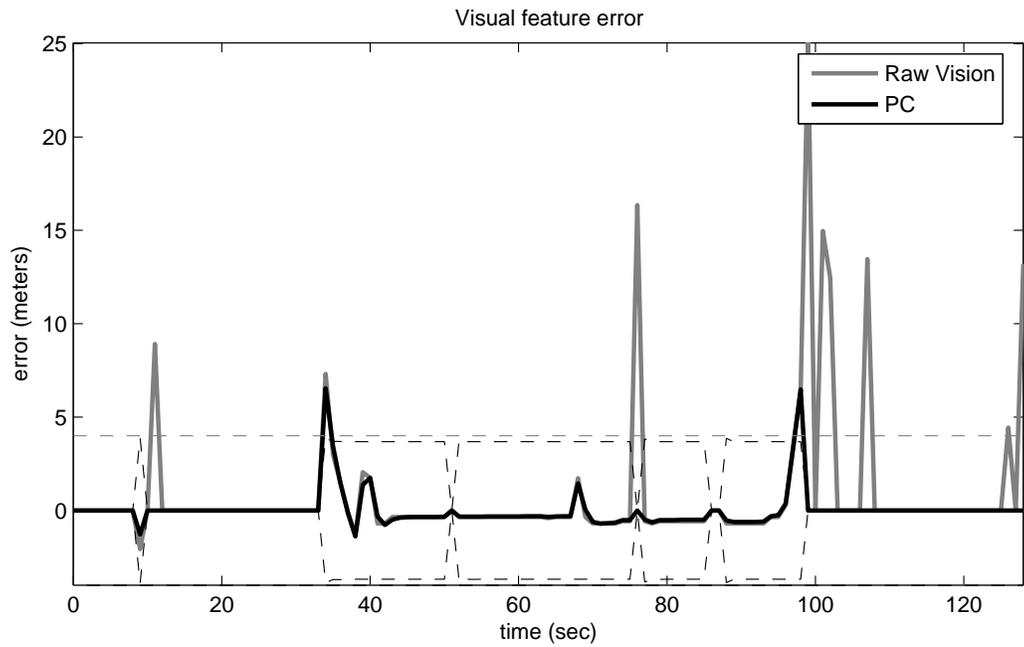


(a)

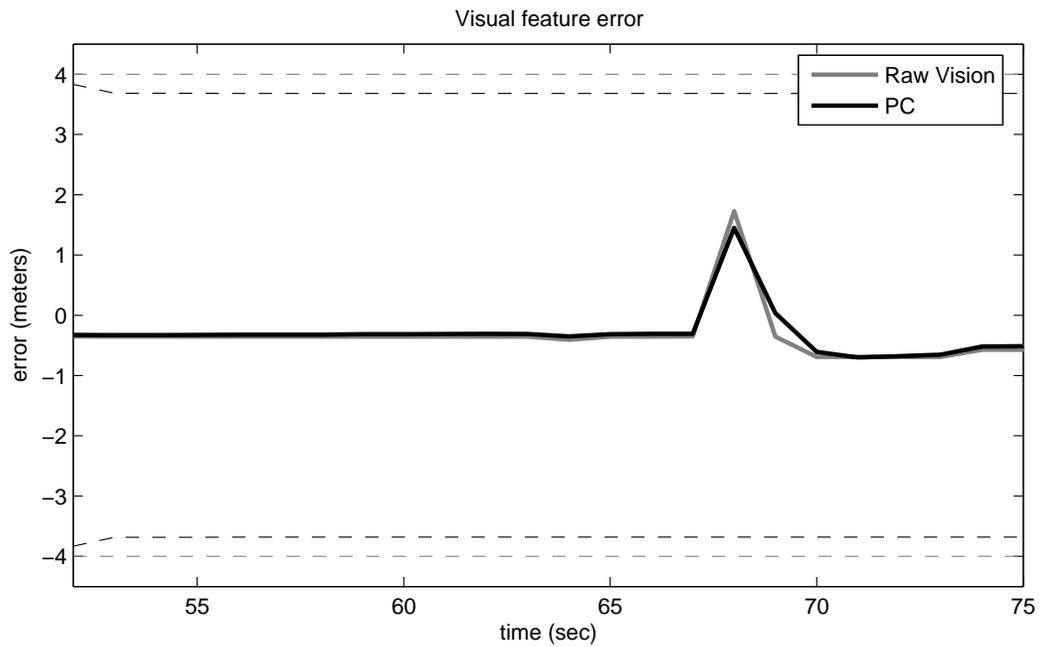


(b)

Figure 7.4: Position error in KF localization with known data associations. This plot shows the result of the KF incorporating GPS and visual feature measurements compared with the raw GPS measurements. a) Error in the lateral vehicle position. b) Error in the longitudinal vehicle position.



(a)



(b)

Figure 7.5: Error in the visual estimate of the distance to an intersection road marking. a) The result of the KF incorporating GPS and vision measurements compared with the raw vision measurements. b) A magnified portion of (a) to illustrate the reduced error and uncertainty.

be seen in the column marked $\bar{\sigma}$, denoting the mean uncertainty. Uncertainty in longitudinal position is decreased more by the addition of visual context than is uncertainty in the feature estimate by the addition of position context. This can be observed in Figure 7.4b, where uncertainty is reduced significantly, and in Figure 7.5b, where the reduction is small even when magnified. As I will discuss in the next section, there is a tradeoff in the reduction of uncertainty in this mutual influence formulation.

7.3 Discussion

The experimental results point toward two related phenomena. First, that large errors in the feature estimates caused by false detections (the result of the landmark detector identifying other objects as road markings) are not corrected by adding position context; these spurious measurement errors can be observed in Figure 7.5. Second, that the uncertainty in the vehicle position is reduced by more than the uncertainty in the feature estimate is reduced. Both of these observations are related to the amount of emphasis given to the feature estimate \hat{f}_t by the Kalman filter.

7.3.1 Correcting Detection Errors

The estimated value of the feature state f_t by the Kalman filter is a weighted average of the feature measurement \hat{f}_t and its predicted value $F(x_{t,-})$, which is a function of vehicle position. The Kalman filter update equation (7.16) can be rearranged into this weighted average form, shown in (7.20). The innovation covariance term S_t representing the common denominator of this expression is shown in (7.15).

$$\begin{aligned} f_t &= F(x_{t,-}) + C_t K_{t,L} \left(\hat{f}_t - F(x_{t,-}) \right) \\ &= C_t K_{t,L} \hat{f}_t + (I - C_t K_{t,L}) F(x_{t,-}) \\ &= (C_t P_{t,-} C_t^T S_t^{-1}) \hat{f}_t + (\Omega_t S_t^{-1}) F(x_{t,-}) \end{aligned} \quad (7.20)$$

The weights in this weighted average are given by covariance matrices, which express uncertainty in the measurement and prediction. The measurement uncertainty is given by Ω_t , and that of the predicted measurement is given by $C_t P_{t,-} C_t^T$, which represents a projection of uncertainty in the vehicle state $P_{t,-}$ into the visual

feature space. When $\Omega_t < C_t P_{t,-} C_t^T$, more weight is given to the raw measurement \hat{f}_t than its prediction $F(x_{t,-})$ in this weighted average. Experimentally, the measurement uncertainty was $\Omega_t = \sigma_f^2 = 4\text{m}^2$, while $C_t P_{t,-} C_t^T = \sigma_{p,\parallel,-}^2$ converged to a value of 22.2m^2 while a landmark was visible between $t = 34$ sec and $t = 98$ sec. These numbers indicate that approximately 85% of the weight in the estimate of f_t was given to the raw measurement \hat{f}_t .

The visual measurement model used was static; a value of $\Omega_t = \sigma_f^2 = 4\text{m}^2$ was always used. This model does not consider that the feature measurement (which is a distance to a detected road marking) makes the assumption that the road marking is successfully detected. False landmark detections occur occasionally, and produce biased distance measurements as can be observed in Figure 7.5. These erroneous measurements are given as much weight as valid measurements because of the static measurement model.

A dynamic measurement model that assigns higher uncertainty to erroneous measurements caused by false landmark detections would enable more consistent feature estimates by placing more weight on the predicted value $F(x_{t,-})$. Furthermore, it would improve the robustness of localization to false landmark detections, thereby maintaining integrity of the localization system.

7.3.2 Uncertainty Tradeoff

The uncertainty in both position estimates and feature estimates is observed to decrease as a result of mutual influence filtering. This in itself is not a surprising result, as it is known that the Kalman filter update contracts the PDF of the state, thereby reducing the variance. What is interesting to observe is that the reduction in position uncertainty is greater than the reduction in feature uncertainty. In fact, an inverse relationship exists between the reduction in uncertainty of the localization state x_t and that of the feature state f_t . To make this relation evident, one must rearrange the posterior feature covariance matrix $P_{F,t}$ which is given in (7.17) into the form of (7.21). This form shows a gain term $C_t K_{t,L}$ applied to the feature measurement covariance Ω_t .

$$\begin{aligned} P_{t,F} &= (I - C_t K_{t,L}) C_t P_{t,-} C_t^T \\ &= (I - C_t P_{t,-} C_t^T S_t^{-1}) C_t P_{t,-} C_t^T \\ &= (S_t S_t^{-1} - C_t P_{t,-} C_t^T S_t^{-1}) C_t P_{t,-} C_t^T \end{aligned}$$

$$\begin{aligned}
&= ([C_t P_{t,-} C_t^T + \Omega_t] S_t^{-1} - C_t P_{t,-} C_t^T S_t^{-1}) C_t P_{t,-} C_t^T \\
&= \Omega_t (S_t^{-1} C_t P_{t,-} C_t^T) \\
&= (S_t^{-1} C_t P_{t,-} C_t^T)^T \Omega_t = (C_t P_{t,-} C_t^T S_t^{-1}) \Omega_t \\
&= C_t K_{t,L} \Omega_t
\end{aligned} \tag{7.21}$$

$$P_{t,L} = (I - K_{t,L} C_t) P_{t,-} \tag{7.22}$$

Compare this expression to the posterior localization covariance matrix (7.22), which has a gain term $I - K_{t,L} C_t$ applied to the prior localization covariance $P_{t,-}$. These gain terms are related by the presence of the terms $K_{t,L}$ and C_t . Thus, for large $K_{t,L}$ and/or large C_t , the uncertainty in the localization state $P_{t,L}$ will be small compared to its prior uncertainty $P_{t,-}$. The reduction in uncertainty of the feature state $P_{t,F}$ over its measurement uncertainty Ω_t will be small by comparison.

The principal factor affecting the Kalman gain $K_{t,L}$ and in turn the posterior uncertainties $P_{t,L}$ and $P_{t,F}$ is again the balance between $C_t P_{t,-} C_t^T$ and Ω_t within the innovation covariance matrix S_t . An uncertain localization estimate such that $C_t P_{t,-} C_t^T > \Omega_t$ places more emphasis on the measured features \hat{f}_t in the state estimate, as discussed in Section 7.3.1. This will reduce significantly the uncertainty in the localization state due to the emphasis given to the more reliable feature measurements. However, the uncertainty in the feature state will not be reduced significantly since the feature measurements are considered more reliable than the localization estimate. This explains the results shown in Figure 7.5, in which feature uncertainty is not reduced significantly due to the fact that it is already significantly lower than localization uncertainty. However, Figure 7.4 shows a significant reduction in localization uncertainty in the longitudinal direction as a result of incorporating the more reliable feature measurement.

Conversely, an uncertain feature measurement such that $\Omega_t > C_t P_{t,-} C_t^T$ will place more emphasis on the prior localization estimate $x_{t,-}$ in the state estimate. This will reduce the uncertainty in the feature state $P_{t,F}$ to reflect the contribution of the more reliable localization estimate. However, the uncertainty in the localization state $P_{t,L}$ will be less affected.

Therefore, it can be said that there is mutual influence between the localization state and the feature state. The more uncertain of the two will receive a significant contribution from the other, improving confidence in that estimate without the other changing significantly. There is thus a selfless and cooperative aspect to this

integration of localization with visual features.

An Illustrative Example

A simplified example demonstrates the tradeoff in uncertainty between the localization estimate and the measured features quite well. This example uses the following notation:

- $x_t = p$ is the prior position estimate, which is univariate.
- $P_{t,-} = \sigma_{p,-}^2$ is its uncertainty.
- $f = F(x_t)$ is the observed feature, modeled as a function of position p . The Jacobian is simply the rate of change of f with respect to p , given by $C_t = \alpha$.
- The measurement of this feature is \hat{f} . $\Omega_t = \sigma_{f,-}^2$ is its measurement uncertainty.

The expressions for posterior uncertainty (7.21) and (7.22) are reduced to the following. The symbol β is used to represent the uncertainty gain and is present in both expressions.

$$\sigma_f^2 = \frac{\alpha^2 \sigma_{p,-}^2}{\alpha^2 \sigma_{p,-}^2 + \sigma_{f,-}^2} \sigma_{f,-}^2 \quad (7.23)$$

$$= \beta \sigma_{f,-}^2 \quad (7.24)$$

$$\sigma_p^2 = \left(1 - \frac{\alpha^2 \sigma_{p,-}^2}{\alpha^2 \sigma_{p,-}^2 + \sigma_{f,-}^2} \right) \sigma_{p,-}^2 \quad (7.25)$$

$$= (1 - \beta) \sigma_{p,-}^2 \quad (7.26)$$

It is apparent from these expressions that the uncertainty gain for the two terms are inversely related through the parameter β . Thus a small gain for one (indicating a large reduction in uncertainty) is accompanied by a large gain (small reduction) for the other. The term β is dependent on the relative magnitudes of $\sigma_{f,-}^2$ and $\sigma_{p,-}^2$, and the derivative α , and can be written as follows where $r = \frac{\alpha^2 \sigma_{p,-}^2}{\sigma_{f,-}^2}$.

$$\beta = \frac{r}{r + 1} \quad (7.27)$$

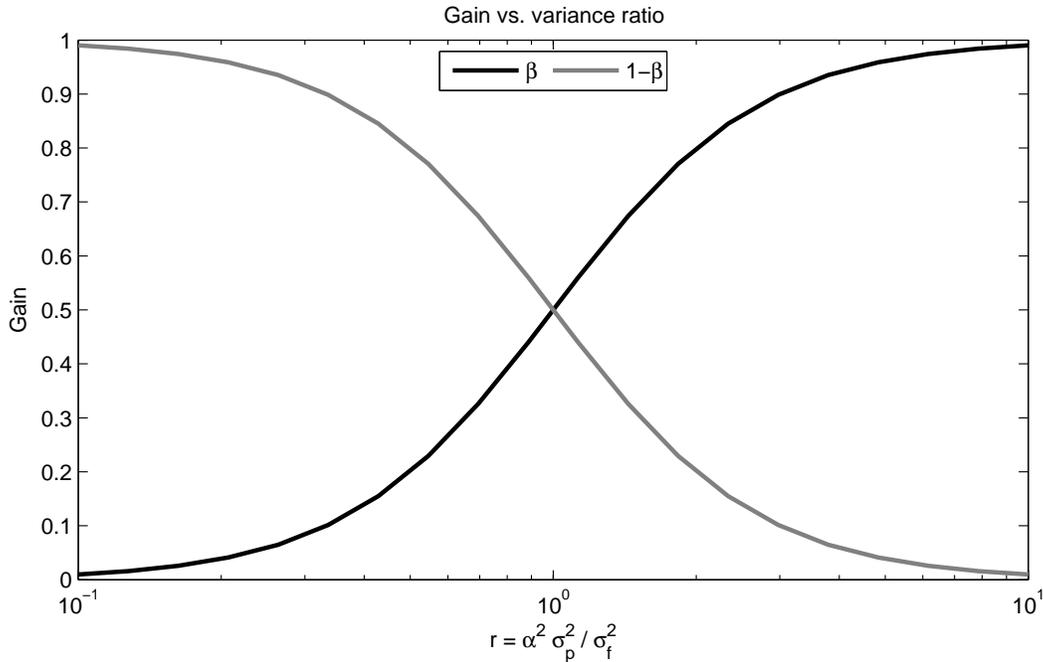


Figure 7.6: Tradeoff in gain applied to position and feature uncertainty as a function of the prior-variance ratio r .

The gain is plotted against this ratio r in Figure 7.6 to show the effect of varying the relative magnitudes of $\sigma_{f,-}^2$ and $\alpha^2 \sigma_{p,-}^2$ on the posterior estimates of σ_f^2 and σ_p^2 . From this figure it should be apparent that there is a tradeoff in the uncertainty given to the localization state and the feature state. Due to their inverse relation, one will always have a lower gain than the other unless $\sigma_{f,-} = |\alpha| \sigma_{p,-}$; in this case both the position estimate and the feature measurement contribute equally to both posteriors.

7.4 Summary

A Kalman filter formulation is developed in this chapter that performs simultaneous estimation of vehicle localization and visual features. This formulation is termed “Mutual Influence” because it is found that vision and localization systems can share their information to achieve a common gain. In particular, each system contributes to the other in accordance with its relative level of uncertainty, resulting in a decrease in uncertainty for both systems. The end result is that the knowledge contained in each system is made consistent with the other, so that subsequent

systems can use this information without concern for potential conflicts.

Implementing a mutual influence filter was also found to require minimal effort in addition to that required for adding visual context to localization. Results for an implementation of the filter are shown in Section 7.2, and demonstrate the sharing of information between vision and localization systems according to their respective levels of uncertainty. These results assume known data associations and as a result further work is required to demonstrate the feasibility of a mutual influence filter in the general case. Improvements in modeling the visual feature measurements to handle false landmark detections is also important to maintain system integrity.

Chapter 8

Concluding Remarks

A number of contributions to the field of vehicle localization have been made by this thesis research. I provide a summary of these contributions within this chapter, discuss the significance of each to the future of intelligent vehicles and intelligent transportation systems, and present conclusions reached and questions raised. I also propose future research opportunities that extend the investigations performed in this thesis research or that lie in related areas.

8.1 Summary of Contributions

Contributions arising from research presented in Chapters 3 through 7 are described in the following. A summary of each contribution is given, its significance discussed, and any conclusions reached or questions raised presented.

8.1.1 Chapter 3: System Architecture

Chapter 3 presents a modular system architecture for multi-source vehicle localization incorporating visual context, in accordance with the first thesis objective given in Section 1.3. The system architecture is built on a theoretical Bayesian filtering foundation. Assumptions concerning the posterior distributions are made, making this architecture applicable for localization in continuous domains using either Kalman filtering or particle filtering, or in discrete domains using grid-based filters.

The modularity of the architecture is due to the assumption of conditional independence between passive measurements of vehicle position, measurements of

visual features, and map information. It is this assumption that enables an existing passive localization system to be augmented using visual context or map matching.

The significance of this system architecture is that it builds on the use of Bayesian filtering by the majority of current vehicle localization systems. Many of these systems fuse GPS with dead reckoning for the purpose of mitigating the satellite visibility and multipath issues that are the pitfalls of using GPS technology in urban canyon environments. Being modular in nature, the developed system architecture enables the current capabilities of such systems to be maintained, while augmenting their localization results by the addition of visual context and map matching in an effort to further improve localization accuracy and reliability.

8.1.2 Chapter 4: Visual Context in Localization

The focus of Chapter 4 is to provide an implementation of the proposed architecture within a Kalman filtering framework. Its purpose is to propose a method of data fusion between vehicle localization estimates and visual feature data, as described by the second thesis objective in Section 1.3. A passive localization system using a GPS receiver to provide vehicle position measurements is augmented using visual context provided by observing intersection road markings and by map matching which pulls the position estimate closer to the road the vehicle is driving on. These augmentations are performed in a road-based coordinate system, which expresses vehicle position and speed relative to the road rather than in world coordinates. Data association parameters – including the identity of the road, the orientation of the vehicle on the road, and the identity of detected landmarks – are assumed known at this time to avoid a significant source of ambiguity and illustrate “best-case” localization results. A particle filtering implementation is also briefly described, however it does not improve upon Kalman filter localization in the known data association case.

Augmenting Vehicle Localization by Visual Context

Visual context is added to an initial vehicle localization estimate using an extended Kalman filter update. Vision provides an estimate of the distance from the camera to a detected intersection road marking. Modeling this measurement as a function of the longitudinal vehicle position, and knowing the location of the road marking

from the road network database, a correction is made to the initial localization estimate.

Experimental results show that, when landmarks are successfully detected and identified, the average error in the longitudinal vehicle position estimate is decreased, giving improved accuracy compared to a filter without visual context. Furthermore, the standard deviation of this error is decreased giving more consistent localization estimates, and the uncertainty of the longitudinal position is also reduced showing a higher level of confidence in the estimate. When landmarks are not present or not detected, the position estimate converges to that provided by the filter without visual context.

It can be concluded therefore that visual features modeled as functions of vehicle position on a known road segment can provide sufficient context to correct localization estimates potentially corrupted by GPS multipath or accumulated error from dead reckoning. However, it is only possible to do so when such features are present and successfully detected.

The significance of this finding is that systems with the capability to use visual context to help localize the vehicle can provide a more robust localization result. This will improve the performance of vehicle localization in urban canyon environments. It also indicates that using multiple visual features would be beneficial to help ensure that a high degree of accuracy can be continually maintained.

What is not addressed is whether or not vision can be used to maintain localization integrity when GPS position measurements are not available. Provided the environment is densely populated with detectable landmarks, can visual context provide a consistent level of localization accuracy without the accumulated error issues of dead reckoning?

In addition, can the use of visual context for localization be as successful when data association parameters are not known *a priori*? This question is addressed in Chapter 5.

Augmenting Vehicle Localization by Map Matching

Map matching is used to pull an initial vehicle localization estimate closer to the road segment using a pseudomeasurement vector and Kalman filter update. This pseudomeasurement is modeled as a function of lateral vehicle position and speed and is created on the assumption that the vehicle is restricted to travel on the

centerline of the road. The location of the road centerline is provided by the road network database.

Experimental results show that map matching in this way creates a bias in lateral position error. This is due to the above assumption, since the vehicle does not travel on the centerline of the road, but rather within a lane offset from the centerline. Unless the lateral error in the initial localization estimate is greater than the magnitude of this bias, lateral position accuracy is degraded by this method.

Despite this issue map matching as implemented is shown to be complementary to visual context since it affects lateral position whereas visual context is used to affect longitudinal position. Map matching also has an advantage over visual context in that it can always be used provided the identity and location of the road being driven by the vehicle is known; it is not dependent on an environment densely populated with features.

The significance of this map matching method is its simplicity, and implementation within a Kalman filter update aided by the use of the road-based coordinate system. It could also be used with more detailed road network databases to match the vehicle to the location of its lane.

As with visual context, it was assumed that the road driven by the vehicle was known making map matching trivial. How will this extend to the unknown data association case? In addition, how would map matching be complicated by more detailed map databases with lane-level accuracy? The former question is addressed in Chapter 5 along with landmark identification.

A Road-based Coordinate System

A coordinate system is defined expressing vehicle position and speed in the lateral direction (perpendicular to the road) and longitudinal direction (parallel to the road) relative to the current road segment. Conversions for the localization state and covariance matrix are defined between East-North world coordinates and the road-based coordinate system for a particular road. This coordinate system is used to update the localization estimate using visual context and map matching.

The benefit of this coordinate system is it enables a natural modeling of visual features as a function of vehicle position on a specific road. It takes advantage of the fact that vehicles are restricted to drive on the road network, and are constrained to follow the direction of the road. Map matching is also simplified, as the vehicle

position estimate can be pulled closer to the road segment without affecting the longitudinal position.

This coordinate system and the conversions to and from world coordinates are based on an assumption of the road being locally-straight in the region near the localization estimate. In situations where the road bends or the vehicle turns onto a different road, a Gaussian distribution may not be adequate to define the vehicle state in road coordinates. This issue is alluded to in [59], upon which much of my work in this area is based, however it is not explained in depth. The impact of the straight-road assumption should be studied to determine if it significantly affects localization accuracy.

8.1.3 Chapter 5: Data Association

Chapter 5 is concerned with relaxing the known data association assumption made in Chapter 4 to enable visual context to be used for vehicle localization in a general context. Maintaining the Kalman filtering framework, a Multiple Hypothesis Tracking (MHT) approach is proposed to simultaneously consider multiple data association hypotheses. For comparison, a Monte Carlo Data Association (MCDA) method based on a particle filtering framework is proposed as well. In both cases, constraints imposed by the road network are used to limit the number of hypotheses and thereby maintain filter tractability. Experimental results seek to evaluate the impact of visual context on identifying the road segment being driven, and to determine how well GPS measurement biases can be corrected by using visual context in the unknown data association case. The former addresses the third thesis objective given in Section 1.3.

Multiple Hypothesis Tracking

The MHT method is used to consider multiple data association hypotheses and to track each using its own Kalman filter. The filtering posterior thus becomes a mixture of Gaussians, one for each hypothesis. Each hypothesis has a weight defining its relative importance to the mixture, calculated using measurement likelihoods. Those hypotheses that provide better explanations for the measurements from GPS, vision, and map matching have higher likelihoods and receive higher weight relative to other hypotheses. Merging redundant hypotheses and pruning unlikely ones keep the number of hypotheses low and maintain filter tractability.

Experimental investigations were performed using two data sets: one without GPS measurement bias, and one with significant bias. In the unbiased case, it was found that the use of vision results in a higher weight being given to the correct road segment hypothesis on average, however localization accuracy is unchanged. In this biased case, visual context was unable to correct for GPS measurement bias because the detected road marking was not present among the hypothesis set due to the magnitude of this bias. By tuning the filter to place less emphasis on GPS measurements, the detected road marking was identified successfully and the measurement bias was corrected.

From these investigations it can be concluded that using visual context within an MHT framework can improve the success of identifying the correct road segment being driven by the vehicle and can be used to correct for GPS measurement biases, provided the correct landmark identity is among the set of hypotheses. It can also be concluded that tuning of the MHT filter may be required to ensure that this occurs.

This is significant because it implies that visual context can serve two purposes: improving robustness to localization issues such as GPS multipath, and improving the ability to locate the vehicle on the road network. The latter especially is relevant because matching the vehicle to the road network is needed for applications such as vehicle navigation, emergency vehicle dispatching, and monitoring road usage.

Further study is needed to determine how MHT would be complicated using multiple visual features. Tuning the filter to operate reliably in biased GPS conditions is also required, and may be aided by fault detection capabilities to detect significant bias events indicative of multipath.

Monte Carlo Data Association

MCDA is proposed as an alternative method to MHT based on particle filtering localization. This generalizes the particle filter presented in Chapter 4 to the unknown data association case. Each particle is randomly assigned a data association from a set of hypotheses. Coupled with the ability of the particle filter to randomly explore the state space, this allows unlikely data associations to be considered. As with MHT, each particle has an associated weight indicating how well it explains the current set of measurements. This weight also implicitly indicates the likelihood of the data association hypothesis assigned to that particle. Resampling

removes particles with low weight, representing unlikely locations for the vehicle and unlikely data association hypotheses.

Experimentally, this filter shows similar accuracy results to MHT for the unbiased data set, however by considering more hypotheses it takes weight away from the true hypothesis even in situations where the correct data association is obvious. For biased data, MCDA shows an improved robustness to GPS measurement bias due to its ability to consider unlikely data association hypotheses in more areas of the state space. Localization accuracy is improved over the untuned MHT filter and partially corrects biased GPS measurements.

It can be concluded that MCDA is a viable alternative to MHT that has the natural flexibility needed to correct for localization inaccuracies such as GPS multipath and dead reckoning error. More study is needed to determine the full extent of its potential, including how using multiple visual features will complicate the data association assignment, and how the number of possible data associations will affect the number of particles required. Also, other proposal distributions and resampling schemes can be implemented to better guide the particle set to the correct area of the state space.

Use of Road Network Topology Constraints

Both MHT and MCDA take advantage of constraints imposed by the road network topology to restrict the number of road segment hypotheses considered. Principal among these is a connectedness constraint, being that any hypothesized road segment must be connected to a road segment considered at a previous time instant. Also, a constraint is used to restrict the orientation of the vehicle on the road to help define the view of the camera. This assumes that the vehicle does not make unexpected maneuvers such as U-turns.

While these constraints are useful for keeping the number of hypotheses low in most situations, they may restrict the flexibility of the filter to U-turns or when driving on unmapped roads. Turning information provided by inertial sensors, wheel speed sensors or visual odometry may be useful here, as well as a heading sensor to better fix the view of the camera. A condition whereby the map matching is reinitialized in situations where the road traveled is not connected to the rest of the road network would also improve robustness to mapping inaccuracies.

8.1.4 Chapter 6: Experimental Investigations

Chapter 6 is the fulfillment of the fourth thesis objective given in Section 1.3: to demonstrate with a real system implementation the improvements in localization accuracy and map matching that result from the addition of visual context. This chapter describes the experimental testbed, which uses an inexpensive GPS receiver and camera to gather experimental data, and a second GPS receiver with submeter accuracy to provide ground truth vehicle positions. Installation locations within a passenger vehicle for these sensors are described. Custom software is used to acquire sensory and ground truth data at a frequency of 1 Hz. A description is given of the machine vision system used to detect intersection road markings and to estimate the distance to the detected marking from the camera.

Results from a series of experiments are shown which summarize the findings from Chapters 4 and 5. It can be concluded from these findings that the addition of visual context to a GPS-based vehicle localization system:

- Improves vehicle position accuracy in the known data association case, and in the unknown data association case provided the uncertainty in GPS position measurements is not underestimated.
- Provides evidence regarding the road driven by the vehicle, thereby assisting map matching.

8.1.5 Chapter 7: Mutual Influence

Chapter 7 proposes a reformulation of the Kalman filter localization framework to estimate the value of visual features as well as the vehicle localization state. This enables position to provide context to augment the feature measurements by the vision system, in addition to the existing capability of using vision to provide context to vehicle position. This sharing of information is termed “mutual influence,” and is found to be easily implementable in the Visual Context module of the system architecture. This addresses the fifth objective given in Section 1.3.

Using the unbiased data set and assuming known data associations, experimental observations show that that longitudinal vehicle position is significantly affected by visual context, while the estimate of distance to a road marking is largely unchanged from the raw measurement. The reason for this observation is that the

feature measurement has less uncertainty than the position estimate, and so the position estimate receives the most benefit from the sharing of information, rather than *vice versa*.

It can be concluded that the contribution provided by the localization system and visual system is in accordance with the uncertainty of each system; thus, the most uncertain system receives the most benefit. The end result is that the vehicle localization estimate and visual feature estimate are made consistent with each another.

The significance of mutual influence is that position information can be used as a second visual sensor to more reliably sense the environment. It also provides a level of robustness to visual detection errors for systems using visual information, such as lane keeping systems. Also, by comparing what is seen by vision with what is expected based on position, it may be possible to detect inaccuracies in the map database.

It must be determined how the mutual influence filter will perform in unknown data association conditions, in order for it to be useful in a general case. Improved measurement models for visual features are also needed to detect false landmark detections that may create a bias in the filter as a result of placing too much emphasis on erroneous measurements.

8.2 Extensions to Thesis Research

A number of extensions to the research presented in this thesis can be undertaken which will serve to improve localization results within the system's current capabilities. For example, improved measurement models for GPS and visual features can be used to help identify measurement errors such as multipath or false landmark detections that will improve the integrity of the system. Increasing the frame rate of the vision system would allow visual information to be integrated into the localization estimate more often than once per second. In addition, by relaxing topological constraints used for data association, unexpected driver behavior such as U-turns can be handled. Also, by using visual information such as the location of the vehicle within its lane, it will be possible to improve localization accuracy in the lateral direction and address the bias introduced by matching to road centerlines.

The following are more ambitious extensions to the thesis research. I consider

these to be within the same vein as the research presented, yet would be by themselves significant undertakings.

8.2.1 Maintaining Accurate Localization Without GPS

Using visual features for localization ultimately depends on the availability of visual features; vehicle localization is unaffected when there are no features present, or when features do not change with vehicle position. The presence of GPS, despite the bias that is often present in its position estimates, controls the accumulation of uncertainty and error that would otherwise occur in the absence of visual landmarks. Maintaining localization integrity over long periods when GPS is unavailable is important for localization in urban areas, particularly in tunnels. This will involve continual updates to the localization state using visual information to control the accumulation of error in the absence of GPS. This therefore requires continuous availability of visual information, necessitating improvements in machine vision capability to extract features and the use of more complete maps of the environment.

One can choose to investigate this problem in one of two ways: as a relative localization problem, where an initial GPS position is provided and subsequent positions must be determined using vision; or as an absolute localization problem, where no initial position is provided. The latter is a much more difficult problem necessitating comparison of detected features with all such features in a global map.

8.2.2 Generalizing Data Association to Improved Visual Capability

It was shown that by continually observing the same features from a static position eventually all but one data association hypothesis would be pruned. This was a by-product of using intersection road markings as landmarks for localization. In a moving vehicle features will play a much more transient role, possibly being visible for only a few seconds. Is it possible to guarantee convergence to one hypothesis for a moving vehicle? How will this be affected by new features becoming visible and others no longer visible? Improved visual capability will result in the ability to detect more features, therefore providing more context for localization. This comes at a cost of increased data association ambiguity.

8.2.3 Incorporating Uncertainty in Map Information

It has been assumed throughout this thesis that the map provides a perfect representation of the world, that it is without error. This is not true in practice. Localizing the vehicle with respect to a landmark or road at an incorrect location in the map creates a biased result. By associating a degree of uncertainty with the location of roads and landmarks in the map perhaps this type of bias can be avoided. A natural extension of this approach would be to correct the location of roads and landmarks in the map, and to add missing ones that are not present because the map is out of date or incomplete. This problem is known as Simultaneous Localization and Mapping (SLAM).

8.2.4 Improved Visual Capability and Environmental Awareness

In particular, the ability to recognize static objects is required for localization since such objects can be mapped and their location known *a priori*. In complex environments, it may be useful to use location knowledge to restrict what is expected to be seen. For example, position-based priors can be used for object recognition [68].

Static object detection must also be robust to the presence of moving objects such as other vehicles or pedestrians. Having the ability to detect and track moving objects would therefore greatly improve the use of vision for localization, despite such objects not being directly used for that purpose. Therefore, systems designed for pedestrian and/or vehicle detection can be coupled with localization systems to improve performance.

8.2.5 Using Shared Sensory Capability from Other Vehicles

It may not always be possible for a vehicle to generate a reliable, standalone localization estimate. When satellite visibility is low and landmarks cannot be detected, sensory data acquired by other vehicles may be the only way of maintaining an accurate location. This is known as multi-robot or multi-vehicle localization, and is possible to implement in a distributed manner [22, 56], with each vehicle computing its location based on its own sensory data and the relative location of other vehicles.

8.2.6 Building and Updating a Map of the Environment

One problem with using existing maps is that they may not be current or complete. Having the vehicle update the map with detected features will aid future localization efforts, and reduce the amount of manual mapping that needs to be performed. Also, it will save time and money required to obtain updated maps. This is an example application of SLAM.

Implementing SLAM for multiple vehicles that update a common map will have even more benefits, since multiple areas of the map will be updated simultaneously [23, 74]. In particular, areas that are traveled most often by vehicles should then have the most accurate maps, further assisting the use of vision for localization.

8.2.7 Impact of Vision on Other Localization Criteria

The research performed in this thesis has been concerned with improving the accuracy of GPS-based vehicle localization systems using visual context. However, accuracy is only one criterion by which a localization system can be evaluated. Other performance criteria are integrity, availability and continuity of service [18]. Studying the effect of visual context on these performance criteria would be an interesting endeavor.

Bibliography

- [1] Eric Abbott and David Powell. Land-vehicle navigation using gps. *Proceedings of the IEEE*, 87(1):145–162, 1999.
- [2] Morton N. Andersen, Rasmus O. Andersen, and Kevin Wheeler. Filtering in hybrid dynamic bayesian networks. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, volume V, pages 773–776, May 2004.
- [3] K. O. Arras, J. A. Castellanos, and R. Siegwart. Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints. In *Proceedings of the International Conference on Robotics and Automation*, pages 1371–1377, 2002.
- [4] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [5] S. Atiya and Gregory D. Hager. Real-time vision-based robot localization. *IEEE Trans. Robotics and Automation*, 9(6):785–800, 1993.
- [6] D. J. Austin and P. Jensfelt. Using multiple gaussian hypotheses to represent probability distributions for mobile robot localization. In *Proceedings of the International Conference on Robotics and Automation*, pages 1036–1041, 2000.
- [7] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press, Boston, MA, 1988.
- [8] M. Bertozzi and A. Broggi. GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. *IEEE Trans. Signal Processing*, 7(1):62–81, 1998.
- [9] Richard Bishop. *Intelligent Vehicle Technology and Trends*. Artech House ITS Series. Artech House, Boston, MA, 2005.
- [10] Philippe Bonnifait, Pascal Bouron, Paul Crubille, and Dominique Meizel. Data Fusion of Four ABS Sensors and GPS for an Enhanced Localization of Car-like Vehicles. In *Proc. 2001 IEEE Int. Conf. Robotics and Automation*, pages 1597–1602, 2001.

- [11] Wolfram Burgard, Dieter Fox, Daniel Hennig, and Timo Schmidt. Estimating the Absolute Position of a Mobile Robot Using Position Probability Grids. In *Proc. 14th AAAI National Conference on Artificial Intelligence*, 1996.
- [12] Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Active mobile robot localization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997.
- [13] Francois Caron, Emmanuel Duflos, Denis Pomorski, and Phillipe Vanheeghe. Gps/imu data fusion using multisensor kalman filtering: introduction of contextual aspects. *Information Fusion*, 7:221–230, 2006.
- [14] H. Carvalho, P. Del Moral, A. Monin, and G. Salut. Optimal Nonlinear Filtering in GPS/INS Integration. *IEEE Trans. Aerospace and Electronic Systems*, 33(3):835–850, 1997.
- [15] Roland Chapuis, Romauld Aufrère, and Frédéric Chausse. Accurate road following and reconstruction by computer vision. *IEEE Trans. Intelligent Transportation Systems*, 3(4):261–270, 2002.
- [16] Frédéric Chausse, Jean Laneurit, and Roland Chapuis. Vehicle localization on a digital map using particles filtering. In *Proceedings of the 2005 IEEE Intelligent Vehicles Symposium*, pages 243–248, June 2005.
- [17] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Hahnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pages 225–242. Springer, 2006.
- [18] United States National Research Council. *The Global Positioning System: A Shared National Asset: Recommendations for Technical Improvements and Enhancements*. National Academies Press, 3 edition, 1995.
- [19] Andrew J. Davision and David W. Murray. Mobile robot localization using active vision. *Lecture Notes in Computer Science*, 1407:809–825, 1998.
- [20] Ernst D. Dickmanns and Birger D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):199–213, 1992.
- [21] Chris Drane and Chris Rizos. *Positioning Systems in Intelligent Transportation Systems*. Artech House ITS Series. Artech House, Boston, MA, 1998.
- [22] Nabil Drawil and Otman Basir. Vehicular collaborative technique for location estimate correction. In *Proc. 68th IEEE Vehicular Technology Conference*, pages 1–5, September 2008.

- [23] John W. Fenwick, Paul M. Newman, and John J. Leonard. Cooperative concurrent mapping and localization. In *Proc. 2002 IEEE Int. Conf. Robotics and Automation*, pages 1810–1817, May 2002.
- [24] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte Carlo Localization: Efficient Position Estimation for Mobile Robots. In *Proc. 1999 AAAI National Conference on Artificial Intelligence*, 1999.
- [25] Atanas Georgiev and Peter K. Allen. Localization methods for a mobile robot in urban environments. *IEEE Trans. Robotics*, 20(5):851–864, 2004.
- [26] Amadou Gning and Phillipe Bonnifait. Dynamic vehicle localization using constraints propagation techniques on intervals: A comparison with kalman filtering. In *Proc. 2005 IEEE Int. Conf. Robotics and Automation*, pages 4144–4149, April 2005.
- [27] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2007.
- [28] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forsell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Trans. on Signal Processing*, 50(2):425–437, 2002.
- [29] D. Hahnel, S. Thrun, B. Wegbreit, and W. Burgard. Towards lazy data association in slam. In P. Dario and R. Chatila, editors, *Robotics Research*, number 15 in Springer Tracts in Advanced Robotics, pages 421–431. Springer-Verlag, Berlin, 2005.
- [30] Ian Heywood, Sarah Cornelius, and Steve Carver. *An Introduction to Geographic Information Systems*. Pearson Education, Harlow, England, 2002.
- [31] Zhencheng Hu, Uchimura Keiichi, Hanqing Lu, and Francisco Lamosa. Fusion of vision, 3d gyro and gps for camera dynamic registration. In *Proc. 17th Int. Conf. on Pattern Recognition*, volume 3, pages 351–354, August 2004.
- [32] Patric Jensfelt and Steen Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Trans. on Robotics and Automation*, 17(5):748–760, 2001.
- [33] Derek Johns and Gregory Dudek. Urban position estimation from one-dimensional visual cues. In *Proc. 3rd Canadian Conf. Computer and Robot Vision*, 2006.
- [34] Simon J. Julier and Jeffrey K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls*, 1997.

- [35] Mikael Kais, Stephane Dauviller, and Arnaud De La Fortelle. Towards outdoor localization using gps, vision system and stochastic error propagation. In *Proc. 2nd International Conference on Autonomous Robots and Agents*, pages 198–205, Palmerston North, New Zealand, December 2004.
- [36] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.
- [37] Seong-Baek Kim, Seung-Yong Lee, Ji-Hoon Choi, Kyoung-Ho Choi, and Byung-Tae Jang. A bimodal approach for gps and imu integration for land vehicle applications. In *Proc. 58th IEEE Vehicular Technology Conference*, pages 138–143, June 2003.
- [38] Akio Kosaka and Avinash Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. In *Proc. 1992 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 2177–2186, July 1992.
- [39] Jean Laneurit, Roland Chapuis, and Frédéric Chausse. Accurate vehicle positioning on a numerical map. *Int. J. Control, Automation, and Systems*, 3(1):15–31, 2005.
- [40] Richard B. Langley. The mathematics of gps. *GPS World*, 2(7):45–50, 1991.
- [41] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, 1991.
- [42] Uri Lerner and Ronald Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proc. Conf. Uncertainty in Artificial Intelligence*, pages 310–318, August 2001.
- [43] David Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. 7th Int. Conf. Computer Vision*, pages 1150–1157, 1994.
- [44] James Manyika and Hugh Durrant-Whyte. *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Ellis Horwood, New York, 1994.
- [45] Government of Ontario Ministry of Transportation. *Pavement, Hazard and Delineation Markings*. Ministry of Transportation, Government of Ontario, March 2000.
- [46] Michael Montemerlo and Sebastian Thrun. Simultaneous localization and mapping with unknown data association using fastslam. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 1985–1991, September 2003.

- [47] Kevin P. Murphy. Switching kalman filters. Technical report, UC Berkeley, 1998.
- [48] Kevin P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.
- [49] Judea Pearl. Fusion, propagation and structuring in belief networks. *Artificial Intelligence*, 29:241–288, 1986.
- [50] J. A. Perez, J. A. Castellanos, J. M. M. Montiel, J. Neira, and J. D. Tardos. Continuous mobile-robot localization: Vision vs. laser. In *Proc. 1999 International Conference on Robotics and Automation*, pages 2917–2923, May 1999.
- [51] Jon-Sun Pyo, Dong-Ho Shin, and Tae-Kyung Sun. Development of a map matching method using the multiple hypothesis technique. In *Proceedings of the 2001 IEEE Intelligent Transportation Systems Conference*, pages 23–27, August 2001.
- [52] Honghui Qi and J.B. Moore. Direct kalman filtering approach for gps/ins integration. *IEEE Transactions on Aerospace and Electronic Systems*, 38(2):687–693, 2002.
- [53] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [54] James Rankin. Gps and differential gps: An error model for sensor simulation. In *Proceedings of the 1994 IEEE Position Location and Navigation Symposium*, pages 260–266, April 1994.
- [55] Seth Rogers and Wenbing Zhang. Development and evaluation of a curve rollover warning systems for trucks. In *Proc. 2003 IEEE Intelligent Vehicles Symposium*, pages 294–297, June 2003.
- [56] Stergios I. Roumeliotis and George A. Bekey. Distributed multirobot localization. *IEEE Trans. Robotics and Automation*, 18(5):781–795, 2002.
- [57] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 2003.
- [58] J. Z. Sasiadek and Q. Wang. Low cost automation using ins/gps data fusion for accurate positioning. *Robotica*, 21:255–260, 2003.
- [59] Craig Scott and Chris Drane. Increased accuracy of motor vehicle position estimation by utilising map data, vehicle dynamics, and other information sources. In *Proceedings of the 1994 Vehicle Navigation and Information Systems Conference*, pages 585–590, 1994.

- [60] Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–759, 2002.
- [61] Sylvie Servigne, Thierry Ubeda, Alain Puricelli, and Robert Laurini. A methodology for spatial consistency improvement of geographic databases. *GeoInformatica*, 4(1):7–34, 2000.
- [62] Jianbo Shi and Carlo Tomasi. Good Features to Track. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 1994.
- [63] Padhraic Smyth. Belief networks, hidden markov models, and markov random fields: A unifying view. *Pattern Recognition Letters*, 18:1261–1268, 1997.
- [64] Salah Sukkarieh, Eduardo M. Nebot, and Hugh F. Durrant-Whyte. A high-integrity ins/gps navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*, 15(3):572–578, 1999.
- [65] George Taylor and Geoff Blewitt. *Intelligent Positioning: GIS-GPS Unification*. Wiley, Chichester, England, 2006.
- [66] George Taylor and Geoffrey Blewitt. Virtual differential gps & road reduction filtering by map matching. In *Proceedings of the ION GPS 1999*, pages 1675–1684, September 1999.
- [67] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, Cambridge, MA, 2005.
- [68] Antonio Torralba, Kevin P. Murphy, William T. Freeman, and Mark A. Rubin. Context-based vision system for place and object recognition. In *Proc. 9th IEEE Int. Conf. Computer Vision*, pages 273–280, October 2003.
- [69] Jeffrey K. Uhlmann. *Dynamic map building and localization: New theoretical foundations*. PhD thesis, University of Oxford, 1995.
- [70] S.S. van Leeuwen. Gps point position calculation. *GPS Solutions*, 6:115–117, 2002.
- [71] Eric A. Wan and Rudolph van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *Proc. IEEE 2000 Adaptive Systems for Signal Processing, Communications and Control Symposium*, pages 153–158, October 2000.
- [72] Christopher E. White, David Bernstein, and Alain L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C*, 8:91–108, 2000.
- [73] W.S. Wijesoma, L.D.L. Perera, and M.D. Adams. Toward multidimensional assignment data association in robot localization and mapping. *IEEE Transactions on Robotics*, 22(2):350–365, 2006.

- [74] Stefan B. Williams, Gamini Dissanayake, and Hugh Durrant-Whyte. Towards multi-vehicle simultaneous localization and mapping. In *Proc. 2002 IEEE Int. Conf. Robotics and Automation*, pages 2743–2748, May 2002.
- [75] Wei Zhang and Jana Kosecka. Localization based on building recognition. In *Proc. 2005 IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, volume 3, pages 21–28, June 2005.