Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Methodik der Fernerkundung

Particle Filters for Airborne Tracking and Lane-Level Map-Matching of Vehicles

Isabella Eckel

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt
der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr.-Ing. Uwe Stilla

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. habil. Richard H. G. Bamler
2. Univ.-Prof. Dr. Wolfram Burgard, Albert-Ludwigs-Universität Freiburg
3. Univ.-Prof. Dr.-Ing. habil. Thomas Wunderlich

Die Dissertation wurde am 17.06.2015 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 25.09.2015 angenommen.

Abstract

Particle filters have emerged as a powerful method for solving multi-hypothesis state estimation problems, which originate from non-linear, non-Gaussian dynamic systems. Two applications of particle filters in the fields of remote sensing and vehicle navigation are addressed in this work. The first application in remote sensing is given by tracking vehicles in airborne images for traffic monitoring and surveillance. The second application is related to vehicle ego-localization on a digital map for navigation and driving assistance systems.

For airborne vehicle tracking, the low image resolution and the small frame-rate of industrial airborne camera systems impede clear discrimination of individual vehicles and precise motion prediction. Existing particle filter approaches for airborne vehicle tracking lack advanced strategies to achieve robustness against errors from low-cost systems and fail to address issues in real-world situations. Previous work on airborne vehicle tracking assumes continuous vehicle motion and clear discrimination of individual vehicles with unvarying appearance, which is not given in urban traffic scenarios. In this work, adaptive particle sampling and weighting approaches are proposed to ensure that particles explore the search space more efficiently and appearance changes can be tolerated without target losses. An update strategy for the appearance template model of each tracked vehicle further increases the rate of successfully tracked vehicles. Information about the context of multiple tracked vehicles is integrated by an online map learning approach. Experiments were conducted on low frame-rate, low-resolution image sequences taken from an airplane and a UAV of dense urban traffic scenarios with up to 151 simultaneously tracked vehicles. The results demonstrate the robustness of the developed tracker against discontinuous vehicle motion, appearance changes and difficult discrimination of similar close vehicles.

For vehicle ego-localization, the standard automotive positioning sensors of series production vehicles and the abstracted representation of the road geometries in commercial digital maps complicate the computation of the correct map-matched position. Recent particle filter methods for map-matching rely on the quality of the map and the absolute positioning systems and ignore to examine the performance at ambiguous situations where correct map-matching is most critical for navigation functions. This work includes advanced models for lane-level particle sampling constrained to the digital map and particle weighting with measurements from the onboard lane detection camera system. A new medoid-shift clustering method is proposed to extract multiple hypotheses from the weighted particles in the road map space. A functional prototype of the map-matching system is implemented in a test vehicle using commercial map data and a low-cost absolute positioning system. The results of various tests including ambiguous situations like road bifurcations demonstrate the ability of the map-matcher to maintain multiple hypotheses until the situation becomes decidable. This investigation has revealed that the lane matching together with the lane topology-aware particle sampling supports the decision for the correct road segment. Overall the developed map-matcher proves its suitability for automotive applications, because of its good performance under the proposed quality criteria derived from requirements of navigation and advanced driving assistance systems.

Both particle filter methods achieve robust results on experimental real-world datasets, which were obtained from low-cost sensor systems. Therefore, they prove as suitable candidates for industrial applications.

Acknowledgements

I would like to express my gratitude to all the people who contributed to this thesis.

First of all, I would like to thank Prof. Richard Bamler for offering me the opportunity as a doctoral student to start work on this thesis at the chair of remote sensing technology. Furthermore, I am grateful to my co-advisor Prof. Wolfram Burgard for his interest in the topic.

Furthermore, I thank Matthias Butenuth, who supervised this work about airborne vehicle tracking. During the first two years of my thesis, he taught me structured paper writing and helped me to improve my presentation skills. I also thank all my former collegues at TUM for the friendly working atmosphere at the institute and fruitful discussions in workshops and meeting. My special thanks goes to Florian Burkert, Ke Zhu, Daniel Frey and Stefan Auer for the collaboration in our IGSSE project team and the great time also besides scientific topics. Many thanks to Stefan Gernhard for his support with the technical infrastructure.

In addition, my warm gratitude goes to Peter Reinartz, Franz Kurz and Dominik Rosenbaum for the good collaboration at DLR. I especially appreciate their effort to provide valuable data from flight campaigns.

For the second stage of my thesis, I would like to thank Robert Hein who gave me the chance to pursue my thesis at BMW. Many thanks also to my fellow collegues Dominik Gusenbauer and Klaas Klasing for their kind introduction into vehicle positioning and map topics. My special thanks to Marco Pfattner for his reliable work on the map-matcher implementation.

My deepest gratitude to my husband Marc for his honest feedback and good advices to keep me focused on the thesis.

I am most thankful to my parents for their support and encouragement whatever turns my life takes.

Contents

1 Introduction

1.1 Motivation

Particle filters have favorable properties for solving multi-hypotheses filtering problems. They provide a discrete approximation of the posterior density of the state of non-linear, non-Gaussian dynamical systems. Applications of particle filtering in the context of vehicle tracking and localization attracted much attention as the computational power increased over the years. Two filtering problems have emerged from this field that leverage the potential of a multi-hypothesis method. First, visual tracking of vehicles from airborne image sequences for estimating traffic parameters. Second, vehicle ego-localization on a digital map to obtain the vehicle position for navigation and driving assistance purposes.

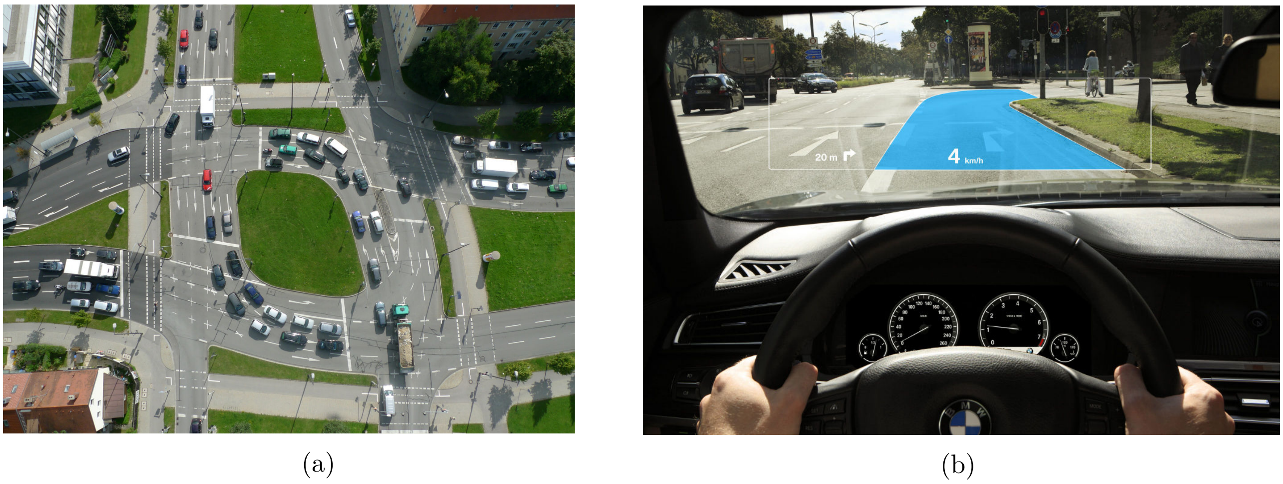
In the context of traffic monitoring, vehicle localization can be achieved with remote sensing technology. For example, when mass events, natural disasters or catastrophes occur, rescue teams require a fast overview over the present traffic situation. Ground-based sensors like induction loops and stationary traffic surveillance cameras support the estimation of macroscopic traffic parameters like density, flow or speed in a limited region. In contrast, airborne visual sensors provide a wide-area overlook but still with a sufficient spatial resolution to differentiate individual vehicles. Camera platforms mounted on airplanes and unmanned aerial vehicles (UAV) are employed to capture image sequences in real-time.

Estimating individual vehicle trajectories enables the analysis of driving maneuvers at the microscopic scale, which facilitates a detailed understanding of traffic. Such vehicle tracking systems are subject to challenges arising from properties of the employed camera system. Low-cost camera systems are characterized by a limited spatial and temporal resolution of the image sequences. The 3K airborne imaging system developed by the German Aerospace Center (DLR) captures image sequences at 2 Hz with a ground sample distance of 13 cm at a flight height of about 1000 m. Typical impediments to visual object tracking like occlusion, clutter and variations of the target appearance pose further challenges to robust trajectories estimation.

Vehicle ego-localization in a georeferenced system is a key technology for modern navigation and advanced driver assistant systems (ADAS) systems. The accurate position of a vehicle on a road network map enables onboard navigation units to provide reliable and well-timed turn-by-turn directions. Such guiding information will be extended to the lane-level in next generations of built-in navigation systems.

Fully integrated navigation systems calculate the vehicle position by an onboard fusion of measurements from inertial sensors and a global navigation satellite system receiver (GNSS). Navigation and ADAS require road features, which are stored within a digital map. Such features include for example the distance to the next crossing, curvatures, slopes and traffic signs. These data are accessed at the respective map-matched position of the vehicle, which is given by the corresponding position of the vehicle in the underlying road map. The process of linking the position of the vehicle in geographic coordinates to road map coordinates is referred to as “map-matching”.

The goal of map-matching is to determine the most probable position on a digital map considering uncertainties or ambiguities from inaccurate positioning systems or map digitalization errors. Precision and real-time requirements of the map-matched position depend on the application, which can



(a) (b)

Fig. 1.1 – (a) Traffic scene captured from camera mounted on a UAV. (b) Future navigation concept with contact analogue head-up display [16].

range from comfort driving assistance functions to fully automated driving. The accuracy of the map-matched position is influenced by sensor noise, environment conditions and the accuracy of the underlying road map. Regarding sensor noise, multi-path signal reception of GNSS receivers is one of the main sources of positioning errors. An inexact digitalization of the road geometry in the map causes ambiguities of the map-matched road segment especially at road bifurcations. The map creation and updating process of commercial map providers involves the semi-automatic analysis of field driven data. Thus, the validity of the map database depends on the time when mapping vehicles equipped with advanced measurement devices traveled on the respective road segment.

Particle filtering is well suited to applications of vehicle localization and tracking. For visual vehicle tracking, statistical inferences need to be deduced from a complex system with a highly nonlinear measurement model. Ambiguities in the map-matched position at road bifurcations demand for a method that makes multi-modal probabilities tractable. Both applications require the accurate estimate of a multi-hypothesis state estimate together with a measure of confidence.

Existing particle filter approaches for vehicle tracking problems assume image sequences with a high capturing frequency similar to a video stream. The camera system is usually in close range to the target, which allows for detailed measurements of its appearance. In contrast, low-cost airborne imaging systems mounted on aircrafts deliver image sequences of lower quality, which is characterized by a reduced spatial and temporal resolution, Figure 1.1a. Such imagery leads to an abrupt representation of the target motion and limits the target description consisting of only a few pixels. Thus, the design of a robust particle filter method, which successfully tracks multiple vehicles, is a challenging task.

Increasing the positioning accuracy for navigation applications could be achieved by improving the sensor quality. Since this would also increase the cost of series production vehicles, car manufacturers seek for alternative methods to maximize the potential of low-cost sensor systems. Particle filters integrating additional data from camera systems to compute a map-matched position on the lane-level are an emerging field of research to enable future innovative navigation functions, Figure 1.1b.

1.2 Contributions

The goal of this work is to develop particle filter methods for solving multi-hypothesis state estimation problems from imperfect and uncertain measurements. These methods are proven under real-world scenarios from airborne vehicle tracking and map-matching. The developed particle filters should meet high quality requirements, which means in this application to avoid mismatches and target losses especially under challenging conditions of the employed low-cost sensor systems and in ambiguous situations.

In the first part of this work, a particle filter method for tracking multiple vehicles in airborne image sequences is proposed that operates on low resolution and low frame-rate image sequences. The developed method integrates a vehicle motion model for particle sampling and suitable image features for particle weighting. The focus lies in advanced particle filter strategies, which address the specific challenges of airborne vehicle tracking. Novel adaptive approaches for the motion and observation model are presented to increase the robustness against discontinuous motion and appearance changes of the tracked vehicles. Furthermore, a template update strategy is proposed that adapts the target model to appearance changes in the image, while designed to avoid the template drift problem. The spatio-temporal context of the tracked vehicles is included into the motion model in an online road map learning approach. This method reduces the risk of target losses, because the particles of the target vehicle are guided in the travel direction of preceding and successfully tracked vehicles.

In the second part, a particle filter method for solving map-matching problems at lane-level is introduced. The developed method is designed to handle limitations from low-cost positioning and camera sensors and commercial map data. The main contributions are motion and observation models that include probabilistic models to spread particles along the road network and weight them according to absolute positioning measurements and road marking detections of the camera system. A detailed analysis of the probability parameters is described to identify parameter values that balance the trade-off between robustness and preciseness of the models. The mode seeking problem of finding the most dominant hypotheses for the estimated state is solved by a new medoid-shift clustering method constrained to the road network. A detailed investigation of the particle filter method at ambiguous situations is provided to prove the advantage of this multi-hypothesis method and demonstrate the ability of the map-matcher to resolve ambiguities fast and correctly. Suitable evaluation criteria are formulated to measure the quality of the results and show that the developed system meets quality requirements of ADAS applications.

Both particle filter methods are evaluated with data from real industrial applications of airborne vehicle tracking and navigation. The robustness of the developed methods is demonstrated on these data from low-cost sensor systems that operate also under adverse conditions.

1.3 Publications

Parts of this thesis have been published in an international journal, conference proceedings and patent applications:

- I. Szottka (Eckel), “Spurgenaue Positionierung eines Fahrzeugs”, Patent, DE102013217060A1, 2015
- I. Szottka (Eckel), “Spurgenaue Positionierung eines Fahrzeugs”, Patent, DE102013217061A1, 2015
- I. Szottka (Eckel), M. Butenuth. “Advanced particle filtering for airborne vehicle tracking in urban areas”, IEEE Geoscience and Remote Sensing Letters, vol. 11, no. 3, pp. 686-690, 2014

- I. Szottka (Eckel), “Particle filtering for lane-level map-matching at road bifurcations”, 16th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 154-159, 2013
- I. Szottka (Eckel), M. Butenuth. “An adaptive particle filter method for tracking multiple interacting targets”, IAPR Conference on Machine Vision Applications (MVA), pp. 6-9, 2011
- I. Szottka (Eckel), M. Butenuth. “Tracking multiple vehicles in airborne image sequences of complex urban environments”, Joint Urban Remote Sensing Event (JURSE), pp. 13-16, 2011

1.4 Outline

This work is organized as follows: In Chapter 2, the state of the art of systems and methods in the two fields of application for the developed particle filters is reviewed. First, to capture image sequences from traffic scenes, different options of airborne imaging systems are summarized and the resulting image resolutions are compared, as they have a major impact on the design and performance of the tracking application. Next, the challenges of onboard map-matching are described, which arise from the limitations of modern positioning and map technologies. Both applications can be addressed by single- or multiple-hypothesis methods, which are discussed in this chapter. In particular, different characteristics of particle filter methods for solving visual tracking and map-matching problems are described. At the end of each sub-chapter, specific challenges in these two applications are derived, which will be addressed in this work.

Chapter 3 introduces the basic concept of particle filters and provides the theoretical formulation in a Bayesian framework. Advanced techniques of particle filtering are presented that increase the computational efficiency. Furthermore, approaches for robustly estimating the state from the set of weighted particles are addressed.

In Chapter 4, the developed particle filter for airborne vehicle tracking is presented. The employed flight systems and the camera properties are described and typical challenges with these image sequences are identified. Suitable models for representation of the target motion and interpretation of image features in low resolution and low frame-rate image sequences are proposed. The most likely state for each vehicle is estimated by a mean shift method. To handle ambiguities from similar vehicles, a multiple target hypothesis association method is introduced. To achieve robustness against discontinuous motion and appearance changes, advanced techniques for adaptive models, target template updates and online map learning are presented. The results show the increased robustness of the tracker with these advanced techniques. A quantitative analysis of the tracking results in typical example scenarios demonstrates the quality of the developed method.

In Chapter 5, a map-matching system and its implementation is presented. Details of the prototype vehicle, the sensors and the map data is given. From the description of the Bayesian system model, the particle filter for lane-level map-matching is formulated. Probabilistic models for motion of the map-matched position along the road network and observations from the positioning and camera systems are introduced. A sensitivity analysis is conducted to determine the preferred parameter values for these models. Evaluations of test drives with the developed system at ambiguous situations demonstrate the performance of the map-matcher in the proposed categories accuracy, stability, confidence, robustness and performance.

Chapter 6 concludes this work and gives an outlook to future work in the context of methods for airborne vehicle tracking and lane-level map matching.

2 State of the art

This chapter presents state-of-the art systems for airborne vehicle tracking and onboard map-matching. It gives an overview of methods, which have been applied to these applications or which are relevant in the context of this work.

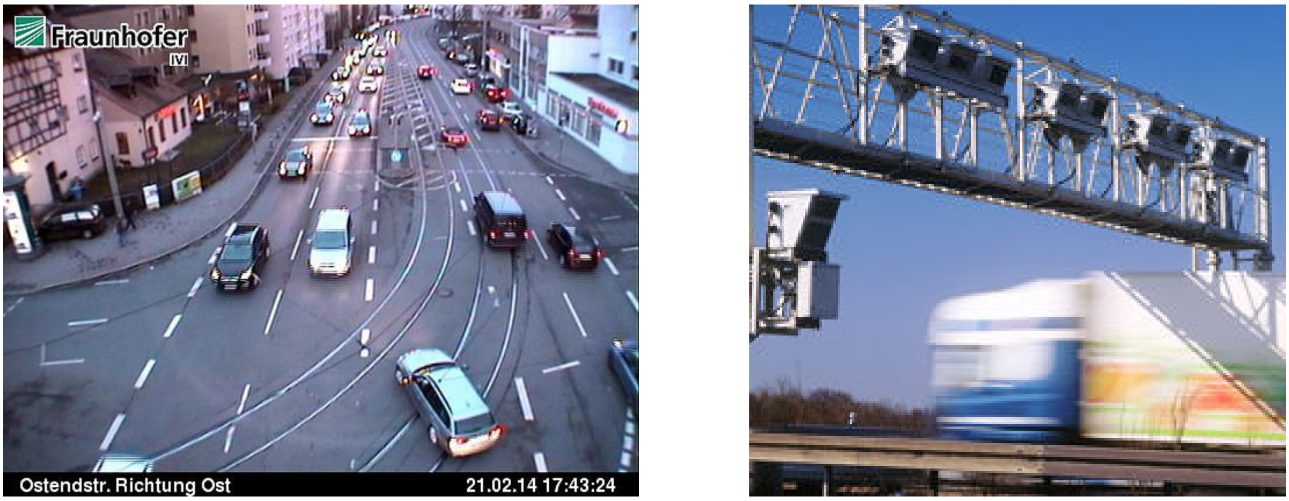
2.1 Airborne vehicle tracking

Capturing of image sequences from above ground offers possibilities to detect and track vehicles for different purposes. Depending on the application, the camera setup, image processing and tracking methods differ.

2.1.1 Applications

Camera systems mounted on poles or gantries at the road-side are a widespread approach to acquire sequences of traffic images from elevated positions. These stationary systems are mainly used for traffic monitoring, surveillance and tolling applications, Figure 2.1. Because of their limited field of view, they capture only stationary parts of the traffic situation. From the motion trajectories of individual vehicles, visual activity analysis can be performed to learn motion patterns and detect anomalies [122]. Since the objects under investigation are close to the camera, the object resolution in the images is large enough to detect detailed features for tracking. However, dense traffic and varying weather and lighting conditions require advanced tracking algorithms especially in urban scenarios [20].

A more flexible way to obtain imagery on traffic situations is provided by remote sensing systems on airborne or spaceborne platforms. Transportation management authorities seek current and complete



(a)

(b)

Fig. 2.1 – Stationary camera systems for (a) traffic monitoring [169] and (b) toll collection [165].

data on road traffic surveillance. Airborne systems have the potential to increase the coverage of road monitoring imagery complementary to stationary cameras. Large-scale coverage and flexible setting of remote sensing cameras allows a complete capturing of the area of interest from dynamic viewpoints. Such systems are suitable for monitoring specific areas on demand with flexible use of different imaging systems and aerial or spaceborne vehicles designed for the application.

Tracking of individual vehicles for a period of time allows to estimate their trajectories. From the collection of trajectories, traffic dynamics like flow, density and speed can be derived [5], [71]. Individual vehicle trajectories further allow detailed analysis of driving maneuvers and identification of driving patterns like merging from an entrance ramp. Tracking of multiple vehicles provides insights into interaction patterns. The interpretation of individual trajectories serves as valuable input to calibrate or verify microscopic driving models [74]. Challenges of vehicle tracking from remote sensed images are caused by a lower resolution over ground compared to statically mounted cameras. Detailed features of the vehicles cannot be detected, which complicates the distinction between vehicle and other objects or the background. Because of the increased distance to the vehicle, occlusions by buildings, bridges or vegetation occur and lead to partially or totally invisible vehicles. Also the camera system is moving, which adds additional dynamics to the captured traffic scene.

Applications of airborne vehicle tracking are for example to provide quick large area traffic situation overview at mass events or after natural disasters. The German Aerospace Center (DLR) aims to support emergency forces in rescue coordination with an airborne rapid mapping system [102]. A real-time processing chain has been installed to transfer data from the systems aboard an aircraft to a mobile ground station, Figure 2.2. This system allows to monitor actual traffic situations and react accordingly in the relevant region.

A similar approach to use mobile sensors for disaster management is proposed by the Fraunhofer Institute. Their project aims to support search operations with UAVs as part of a heterogeneous sensor network in combination with terrestrial robots [100].

In this work, tracking of vehicles from remote sensing imaging systems is considered. The following sections provide an overview of different imaging systems and methods for vehicle tracking.

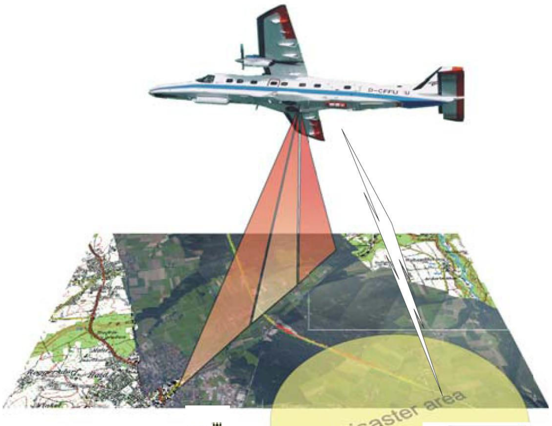


Fig. 2.2 – Airborne traffic monitoring and data processing [62].

2.1.2 Airborne platforms

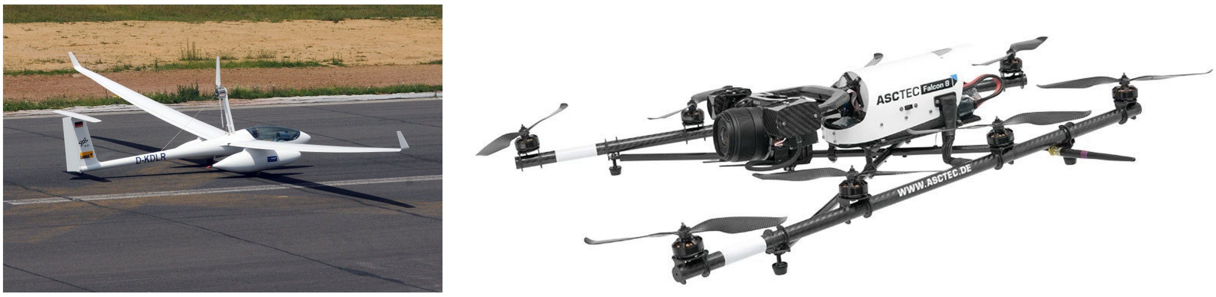
Air- or spaceborne imaging systems consist of several components. Depending on the application, a suitable platform for the imaging sensor is determined. Possible choices are given by satellites, airplanes, airships, helicopters or UAVs. The criteria for decision are cost, flexibility, spatial range and properties of the deployed camera system.

Satellite imagery has the largest field of view, but satellite missions are the most expensive and least flexible solution. Spaceborne optical sensors with a sub-meter resolution are state-of-the art. Examples for commercial satellite systems are the QuickBird and WorldView-2/-3 from DigitalGlobe [42]–[44], which operate at altitudes of 450 and 770/617 km, respectively. These systems provide optical imagery with panchromatic or multi-spectral bands. However, motion estimates from satellite imagery are difficult to obtain, because the revisit intervals in close nadir looking direction are longer than a day. Spaceborne traffic monitoring can also be achieved with Synthetic Aperture Radar (SAR) systems, which are independent of weather conditions and can be operated at any time of day [72]. For example, satellite imagery from the TerraSAR-X or RADARSAT-2 missions make traffic parameter extraction from space possible [27], [70].

Airborne platforms allow a flexible deployment of the camera system. Light aircrafts typically operate at flight heights between 1000 and 3000 m, which makes them ideal for a wide-area coverage of the traffic scene. They can travel long distances to reach the destination scene and they allow a flight duration of several hours. The disadvantage of small aircrafts is that they need to hold a minimum travel speed of approximately 100 km/h above around to safely avoid stalling. This constraint restricts traffic monitoring in urban areas, because a vehicle is visible in an image sequence only for a limited time. The onboard capacity of small aircrafts is large enough to accommodate camera systems and processing units. At DLR, the Cessna 208B Grand Caravan and the Dornier Do 228-212 belong to the research fleet for remote sensing applications [61].

Alternatively, helicopters and airships have been employed [18], [101]. Helicopters are even more flexible than small aircrafts, but have a lower range, less cruise time and less storage capacity. Further, camera systems mounted on helicopters require an effective stabilization effort. The advantage of helicopters is their variable flight speed and the lower flight altitude compared to airplanes. This also holds for airships, but flights are expensive and restricted to favorable weather conditions.

A very unobtrusive approach to monitor traffic is given by electric glider aircrafts. At DLR, a manned aircraft powered by fuel cells has been developed, Figure 2.3a. Due to its quietness and its maximum altitude of 4000 m, it is well suited for security applications [142].



(a) (b)

Fig. 2.3 – Examples of aircrafts for traffic monitoring. (a) Antares DLR-H2 airglider [60], (b) AscTec Falcon 8 UAV [4].

Another category of aircrafts are UAVs, also called drones. In recent years, their commercial availability immensely increased and the deployment of lightweight systems for civil applications became very popular. Examples include aerial photography, wildlife monitoring, industrial inspections and rescue operations. UAVs are very flexible, because they can be delivered to any location of operation. However, they allow only a small payload and the flight time is limited to the battery capacity. Maneuvering of UAVs is accomplished manually by a mobile ground station, which typically requires visual contact and a stable data link. These requirements limit the flight altitude to a few hundred meters. Modern UAVs feature fast reacting stabilizing systems, which make them robust to windy conditions. Commercial manufacturers of lightweight UAVs are for example microdrones and Ascending Technologies. The AscTec Falcon 8 UAV [4] is a multi-rotor system with eight propellers, Figure 2.3b. It is well suited for vehicle tracking applications, because it has a GNSS, altitude sensor, compass and a inertial measurement unit (IMU) onboard. This system provides an effective flight stabilization and allows a payload of up to 750 g. The camera mount accommodates different types of digital compact cameras, which can be tilted to a nadir viewing angle. Further, the camera mount is designed to compensate flight vibrations.

2.1.3 Airborne imaging systems

Tracking individual vehicles for traffic monitoring is best achieved when three properties of the imaging systems are fulfilled: First, coverage of the images should include the complete traffic scene. This requirement allows to estimate trajectories over a long timespan and put them into the context of the whole traffic dynamics. Second, the resolution and the ground sampling distance (GSD) of the images defines the level of detail at which features of the vehicles can be measured. The higher the resolution and the lower the GSD, the more precisely individual vehicles can be distinguished from the background and other vehicles. It is also preferable to capture images in nadir, because parallax effects caused by the moving platform are reduced during image registration. Third, the temporal resolution of the image sequences determines motion continuity of the vehicle trajectories. A high image acquisition rate ensures that fast, abrupt driving maneuvers can be captured by the tracker and motion models of the vehicles can be applied with higher certainty. The design of such an optimal imaging system is not always possible and the degree of realization depends on the airborne or spaceborne platform as well as it is also a question of pricing.

Spaceborne imaging systems cover a wide area, but have the largest distance to ground. Therefore, the GSD of optical images is relatively large compared to airborne camera systems. For example, QuickBird and WorldView-2 have a GSD of 0.61 m and 0.46 m for panchromatic and 2.44 m and 1.85 m for multi-spectral imagery [42], [43]. With an image swath width of above 16 km, satellite imagery is the most attractive option in terms of coverage. Vehicle tracking from optical spaceborne images is achieved by exploiting the time gap between the different bands of a single multi-spectral snapshot image [95], [105]. The WorldView-2 satellite also offers sequences of images in off-nadir, which allows vehicle tracking with a higher temporal resolution for a few minutes [113], [119].

The coverage and the spatial resolution of airborne imagery depends on the flight altitude, camera system and lens properties. The UltraCam family from Microsoft is a high performance, ultra-large sensor format (104 x 68 mm) camera system with 5.2 µm pixel size offering different lens options, which achieve 5 cm GSD at 2020 m altitude [120], [172], Figure 2.4. Their image processing chain produces orthophotos, which are orthographically rectified and radiometrically corrected. Despite the flight height and movement of the aerial platform, effective post-processing methods allow to reconstruct sharp images. The UltraCam systems are mainly deployed for producing imagery for mapping applications. For tracking vehicles, the system has some limitations: The temporal resolution of airborne image sequences does typically not achieve a full video sampling rate, because the storage

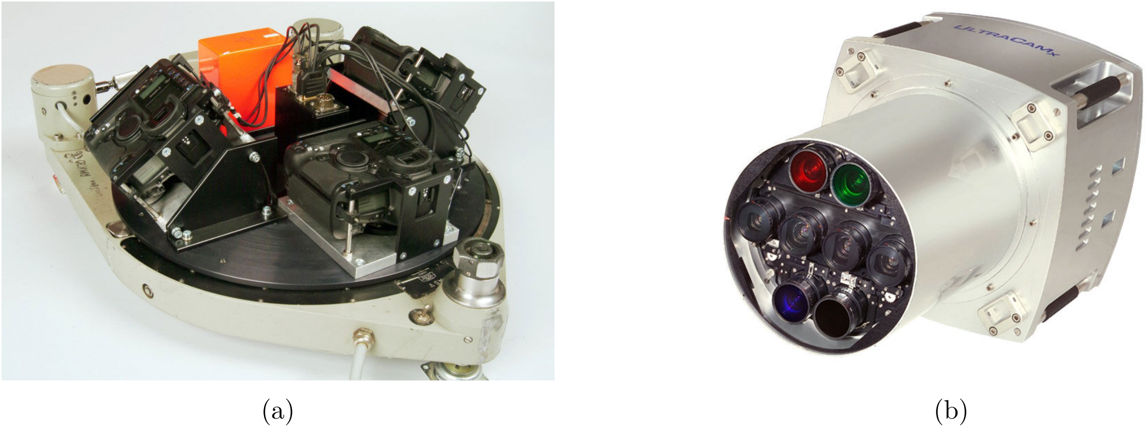


Fig. 2.4 – Camera systems for airborne flight systems. (a) 3K camera system [59], (b) UltraCam system [67].

time for the large-scale images with high resolution is limited to only a few frames per second (fps). As a consequence, highly dynamic motion of the objects appears discontinuous, which increases the uncertainty of motion prediction especially in dense traffic. Although fast electronics are used, the image acquisition of the UltraCam systems requires 1.8 seconds per image. To accomplish high-end aerial image acquisition, professional lenses, fast electronics and advanced software for high-quality and fast image post-processing are employed. These characteristics make the UltraCam system a high-budget option for aerial traffic monitoring.

An alternative aerial camera system, which is used onboard research aircrafts, has been developed at DLR. This multi-camera system unifies three off-the-shelf, single-lens reflex, small-format cameras mounted on a platform [103]. This imaging system was designed to meet low-cost requirements for real-time mapping and tracking applications. In the 3K system, three Canon EOS 1Ds Mark II cameras with Canon lenses are used. The successor system 3K+ consists of three Canon EOS 1Ds Mark III cameras with Zeiss lenses. The sensor size for both cameras is 36 × 24 mm with a pixel size of 7.21 µm (3K) and 6.41 µm (3K+). The GSD reaches 15 cm (3K) and 13 cm (3K+) at a flight height of 1000 m above ground. The cameras are arranged on the platform such that one camera is looking nadir, while the other two are side-looking with variable angle, Figure 2.4. Depending on the application, the platform can be turned along-track or across-track. In a real-time image processing chain, the images are orthorectified, registered and mosaicked by a stitching algorithm to form a continuous representation of the scene. In this way, an enlarged field of view for this multi-camera system is achieved, with coverage of 2560 × 480 m at 1000 m above ground. Although the GSD is lower than for the UltraCam system, the temporal resolution of the 3K systems is maximized to meet requirements from tracking applications and reaches 1-2 fps in the continuous shot mode.

Imaging systems on UAVs require lightweight cameras because of their limited loading capacity. For the AscTec Falcon 8 flight system, different options from the Panasonic Lumix and the Sony NEX compact camera series exist. They differ in the sensor size, pixel resolution and image acquisition rate in the continuous shooting mode. For example, the Lumix LX7 model has a 5.58 × 7.44 mm sensor size with 2.04 µm pixel size and 5 fps [41], while the NEX 7 has a 23.5 × 15.6 mm sensor size with 3.88 µm pixel size and 1.9 fps in practical tests [45], [156]. The GSD can be estimated by (from [145])

$$\text{GSD [cm]} = \text{pixel size [mm]}\frac{\text{flight altitude above ground [cm]}}{\text{focal length [mm]}}. \tag{2.1}$$

Assuming a flight altitude of 100 m above ground and a focal length of 17.7 mm, the GSD results in 1.15 cm for the Lumix LX7. The AscTec Falcon 8 has no onboard image post-processing, which requires image rectification and registration steps after transmission to a ground station.

The success of methods for tracking vehicles in optical aerial imagery depends on the spatial and temporal resolution of the image sequences. A comparison between images at different GSD for vehicle representation is shown in Figure 2.5. At a GSD of 1.6 m, the level of detail is too low to distinguish the vehicle from the background. It requires a spatial resolution of at least 0.2 m to identify individual features of a vehicle. With an decreasing GSD, it is more likely to correctly discriminate the target vehicle from other vehicles or clutter.

The required image acquisition rate to identify continuous motion depends on the vehicle velocity and the prediction accuracy of the motion model. For vehicles driving at 3 m/s, an image acquisition rate of 2 fps would capture the vehicles every 1.5 meters. To achieve the same result at a constant vehicle speed of 18 m/s a frame rate of 12 fps would be required. At 2 fps, the same vehicle speed would result in 9 meters of unobserved motion between consecutive frames. The higher the frame rate is, the more likely is the correct association between vehicles in consecutive frames. This is an important issue especially in dense urban traffic, where reliable motion prediction is difficult to obtain.

Illumination effects and their variation pose additional challenges to vehicle tracking in aerial imagery. The dynamics in illumination lead to a varying appearance of the vehicles, which complicates re-identification and the association of vehicles along time. Vehicles entering shadowed areas of the scene appear darker and are more difficult to be distinguished from the background. Reflection effects caused by sunlight produce a changed vehicle appearance, if the view or reflection angle changes. For vehicle tracking, the similarity or even identity of appearance causes so-called clutter, which generates ambiguity between the target vehicle and other vehicles or the background. Further, fog and dust between camera and ground cause blurring effects. This effect is more likely to occur, the higher the aerial platform flies. Also, stationary objects limit the visibility of the vehicles. Bridges, trees, gantries and buildings temporarily occlude the view, which makes it difficult to follow the motion of the vehicles.

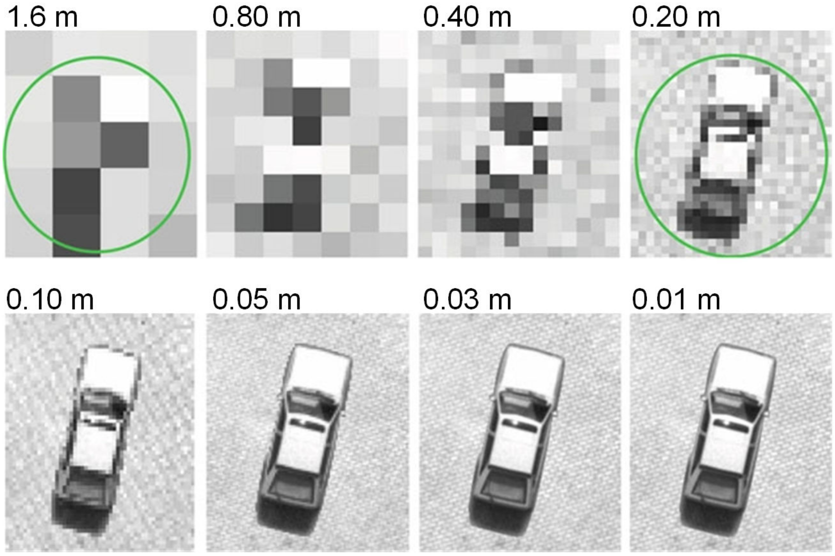


Fig. 2.5 – Comparison of level of detail for different simulated GSD in optical airborne imagery [144]. A GSD of 1.6 m is too large to discriminate vehicles from background or other objects. A GSD of at least 0.20 m is required to identify vehicles and discriminate similar vehicles by their individual structures.

2.1.4 Vehicle tracking

In general, visual object tracking is the task of localizing a target over time given its initial position. Tracking vehicles in airborne image sequences is a sub-problem belonging to the class of visual object tracking. For visual object tracking, a wide range of methods exists, which can be separated in single- and multiple-hypothesis tracking and mixtures thereof [115]. Single-hypothesis methods produce one candidate for the estimate of position over time. Multiple-hypothesis methods are computationally more expensive, but they maintain a set of candidates for the association of measurements over time.

Representative methods for single-hypothesis tracking are the Kalman Filter [56], Kanade-Lucas-Tomasi (KLT) [152] and mean-shift tracker [32]. The Kalman filter computes the optimal solution in terms of minimizing the posterior estimation error for a linear system with Gaussian noise. Kalman filtering for tracking applications is a common approach to estimate the vehicle state recursively [10]. Vehicle tracking with Kalman filtering has been demonstrated in the work of [8] and [139]. A feature-based tracker was developed, which estimates vehicle trajectories in video streams from stationary traffic cameras. Kalman filter based tracking approaches and their extensions perform well, if the condition of Gaussian noise is met and nonlinearity of the system is not severe. Otherwise, the Kalman filter is not able to represent multi-modal probabilities [148] and the target will get lost.

The KLT tracking method detects suitable visual features and tracks them with an optical flow algorithm. This method is suggested to produce stable results when the frame rate is high enough to produce only small changes in the object's motion. In the work of Cao et al. [24], KLT tracking has showed a good performance in airborne vehicle tracking, handling partial occlusion successfully. When objects are totally occluded, KLT tracking is not able to recover from temporary losses of the target and re-initialization is required. Another optical flow based method for airborne vehicle tracking working on whole vehicle regions is proposed in [124]. This approach tracks specific pixel in the vehicle region, but also requires partly visibility of the target.

Mean-shift based trackers find the location of the target by minimizing a similarity measure between the appearance model and candidate regions in the current frame. The candidate location is shifted iteratively by mean-shift moves until convergence. For aerial video tracking, mean-shift tracking performs well under certain tracking conditions [108]. Similar to the KLT tracker, total occlusion leads to loss of the target. Advanced strategies to recover from tracking failures are proposed in [175]. For predicting the target location in the current frame, a Kalman filter can be integrated into the mean-shift tracker [32], [140].

The vehicle tracking problem in this work has been treated before with methods operating on short image bursts [69], [106]. These approaches perform a template matching in candidate regions of the vehicle position. The matches are associated along image triplets to find the most likely vehicle position. This method is limited to short tracklets, which can be used for further estimation of traffic flow parameters. Another technique correlates image features across image pairs [94]. An advanced approach applies probabilistic graph matching after background subtraction across image triplets [173]. However, the presented methods are not robust against temporal occlusion of the vehicle.

In addition to occlusion, dense traffic is the another factor that poses great challenge to airborne vehicle tracking. Similar close vehicles are difficult to distinguish, which becomes more complicated when their appearance varies. In such situations, particle filters have advantages over the single-hypothesis trackers, because they are able to maintain multiple modes of the vehicle location probability density. Particle filters and variants thereof have proven to maintain correct target identification in challenging visual tracking applications [37], [143], [174].

For airborne vehicle tracking, particle filtering with suitable models for vehicle motion and appearance robustly estimates the vehicles' trajectories [23]. However, the performance of this method is only demonstrated for vehicle traveling along straight roads.

The combination of particle filtering with an offline-learned observers and online trained models of the target appearance is found in [110]. The application of this method to tracking vehicles in low-frame rate airborne images is demonstrated, but the learning strategy of the algorithm does not allow to discriminate between different objects. Advanced particle filter strategies exist that track close objects with similar appearance by incorporating the distance between objects into the likelihood [51]. Results of this method are given on image sequences with a high frame rate similar to video streams.

When tracking is extended to multiple objects, the complexity of the problem increases. Methods for multi-target tracking manage this complexity by including information of the mutual interaction between targets, motion constraints and motion coherence. Also the data association problem needs to be solved, which assigns measurements to the tracked objects. Tracking multiple objects in airborne video is demonstrated with a hierarchical system that tracks individual vehicles with a Kalman filter on a low-level and employs a high level tracker for groups of vehicles with coherent motion [25].

A deterministic way of handling the data association problem is given by the Multiple Hypothesis Tracker (MHT) [11]. However, MHT suffers from high computational complexity, because the number of hypothesis increases exponentially over time, which requires pruning of unlikely hypotheses.

Probabilistic techniques for multi-object tracking with particle filters are proposed, which rely on different approaches to incorporate the objects' context. The interaction between targets can be modeled with Markov random fields within an MCMC-based particle filter [88]. The results indicate that this method is more efficient than the conventional joint particle filter. Another graph-based interaction model is proposed in [129]. Application of this method to an airborne video example showed a more robust target identification compared to independent particle filters. Another approach tracks each target individually by an individual particle filter and solves the data association problem by a Joint Probabilistic Data Association Filter (JPDAF) [148]. The effectiveness of this method is demonstrated by tracking objects with a moving robot. The work of [127] suggests to design a mixture particle filter with one component for each target coupled with Adaboost detections of targets for the construction of the proposal distribution.

Besides algorithms for solving the intrinsic tracking problem, further strategies are proposed to increase the robustness of a tracker. Including the context of the target into the dynamic model has shown to increase the quality of the target location prediction [131]. This approach has been tested on people tracking, where the prediction of the motion direction has significant influence of the tracking performance. Another approach to include prior knowledge about the target motion into the tracker is to restrict its motion along a given map. For tracking vehicles with a particle filter, the road network gives valuable information to increase the robustness of the tracker against occlusions [68]. Typical motion patterns of the target can be segmented from vehicles in airborne videos [176]. The resulting motion flow is incorporated into the tracker, which allows to reacquire the targets in challenging situations with occlusion.

Successful tracking involves the design of suitable models to represent the motion and the appearance of the target. Since motion and appearance may change during tracking, adaptive strategies for model updates are considered. A particle filter tracker with adaptive motion and appearance models has shown to efficiently sample the particles and produce robust results under pose and illumination variations [177]. For color-based appearance models, an template update strategy is formulated that alleviates the template drift problem [99]. Examples demonstrate the capability of the tracker to adapt to illumination changes for people tracking applications. Describing the target with multiple features allows to combine complementary characteristics of the target appearance, which increases the reliability of tracking under failure of either feature. The adaptive combination of these features within a particle filter framework has demonstrated to outperform the non-adaptive tracker on face tracking examples [114].

2.1.5 Discussion

Tracking multiple vehicles in airborne images at urban scenarios poses three main challenges caused by the low spatial and temporal resolution of the image sequences:

1. discontinuous motion of the vehicles
2. fast appearance changes
3. difficult discrimination of similar close vehicles.

Related work on (airborne) tracking covers methods to solve the intrinsic (multi-target) tracking problem. However, no method is known that robustly tracks vehicles in airborne image sequences accounting for the three mentioned challenges. Further, performance measurements fail to include large-scale imagery with vehicles in dense urban traffic.

State-of-the art methods on robust trackers include approaches from other fields of applications to improve motion prediction, adapt to appearance changes and integrate the targets' context. Since the application of these strategies to the present application of vehicle tracking is not straightforward, a new method is required that considers typical characteristics of airborne image sequences.

Particle filters have emerged as a well understood and successful method to solve multi-hypothesis tracking problems. Their application to airborne tracking has been demonstrated in previous work. Their advantage compared to MHT is that they allow to naturally integrate adaptiveness of models in a probabilistic framework. However, experiments with abrupt motion and appearance changes on imagery comparable to the one of the present work are not accounted for in the evaluations.

The goals of the vehicle tracking part of this work are summarized as follows:

- Develop a particle filter method that tracks multiple vehicles in low frame rate airborne images of dense urban traffic.
- Present advanced strategies within the particle filter framework to handle discontinuous motion and appearance changes.

2.2 Map-referenced vehicle positioning

Digital maps for automotive applications are an enabler for modern navigation functions and driving assistance systems. To retrieve map information for the current driving situation, the correct position referenced to the digital map needs to be computed onboard in real-time. The process of localizing the vehicle position constrained to a map is called "map-matching".

Successful positioning referenced to a digital map depends on two major factors: First, the quality of onboard sensors, which determines the achievable accuracy of the absolute position in geographical coordinates. Second, the underlying map, which defines the search-space for the map-matched position. The confidence of the matching is higher, the better the map matches with the shape and connectivity relations of the real road network. Since neither positioning sensors nor map data give absolutely accurate estimates, automotive applications seek robust methods for map-matching.

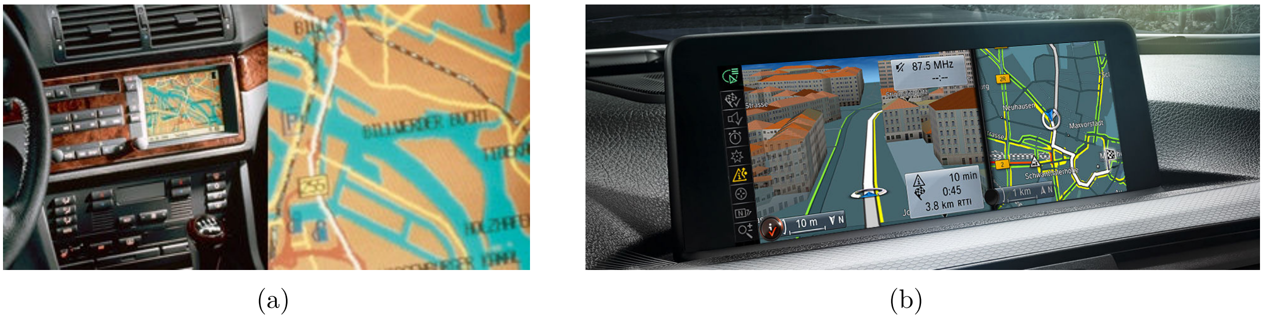
This chapter introduces automotive applications for map-matching and reviews methods and systems for map-matching in the context of vehicle navigation.

2.2.1 Applications

In the history of vehicle navigation, progress has been made towards systems, which provide reliable and precise positioning and route guiding information, Figure 2.6. Before global positioning services such as GNSS became available for civil applications, navigation was limited to built-in systems, which determine the vehicle position based on inertial sensors. Measurements of the angular velocity and acceleration were integrated to obtain a position estimate. Thus, route calculation and guiding was prone to slow recovering from erroneous positioning on the map. Also, updates of map data required a physical exchange of a CD-ROM as the employed storage medium.

Today, built-in as well as mobile navigation systems additionally contain a Global Navigation Satellite System (GNSS) receiver to determine the absolute vehicle position. Still, mismatches of the vehicle position occur, especially inside tunnels or in urban areas, where the direct satellite view is blocked. The map manufacturing process is expensive, which leads to sparsely periodic updates of the underlying map data.

Future navigation systems are aimed to guide the driver with precise information in every driving situation. Guiding will extend to the lane-level, which implies that map-matching needs to deliver the correct driven lane. Augmented reality systems are capable of projecting the route information with a large-scale head-up system on the windscreen to create the impression of an overlay onto the road surface for even more intuitive guiding, Figure 1.1b. In the field of connected cars, location-based-services



(a) (b)

Fig. 2.6 – History of navigation systems at BMW. (a) 1994: First integrated navigation system in 7 series vehicle [13]. (b) 2013: Navigation system Professional with 3-D map navigation and real-time traffic information (RTTI) [15].

offer selective information in the current context, which is retrieved via the location of the vehicle. Examples are emergency call services or a range map service, which displays the region of reachable destinations of an electric vehicle given the current battery charging level.

In the same time of line, driving assistance systems have strongly improved. Automatic control of certain driving manoeuvres started, when electronic stability control was introduced. Based on the fusion of multiple sensor information, braking is applied to individual wheels to stabilize the vehicle. The potential of driving assistance systems increases, when the foresight extends over the range of vehicle onboard sensors. Efficient driving assistants compute a preview of the road course to predict deceleration maneuvers and the optimal reduction of the driving speed. In this way, map data supports sensor fusion to compute relevant input information for advanced driving assistance systems (ADAS). Since ADAS road properties are stored with a map-reference, an accurate estimate of the map-matched position is required to retrieve actual information from the map.

The future of ADAS evolves towards highly automated driving (HAD), where vehicle longitudinal and lateral control is automated, but with the driver ready to take over control. HAD poses the highest requirements to positioning on the digital map, because situation interpretation and motion planning is automatically performed without driver interaction. The development of positioning for HAD involves further sensor information including camera and LiDAR. Thus, the complexity of map-matching will increase, because suitable features for these sensor types need to be stored in the map and matched to current observations of the vehicle's environment. As a consequence, the demand for instantaneous map updates over the air is apparent, because accurate map-matching relies on correct map information. These accuracy requirements will boost the quality of map data to a great level of detail.

The requirement of today and future navigation and driver assistance systems pose high challenges to reliable and precise map-matching. Given a limited sensor accuracy and imprecise map data, advanced map-matching methods need to be developed, which deliver the best achievable result.

2.2.2 Absolute vehicle positioning

The absolute reference of the vehicle location is determined in geographical coordinates. Several options of geographical reference systems exist, which approximate the nearly spherical shape of the earth surface. The reference ellipsoid defined by WGS84 is a common standard [84], which is also used internationally in automotive maps. Each horizontal position in the WGS84 is uniquely defined by a pair of numerical values, which identifies the longitudinal and the lateral degree. Alternative notations are given by Gauss-Krueger and UTM coordinate systems.

The accuracy of position estimates obtained from global satellite navigation systems (GNSS) is degraded by several factors. These sources of errors include irregularities in the ionospheric layer, which leads to a delay of the transmitted signals. Further atmospheric effects occur in the troposphere, where the signal delay depends on the temperature, pressure and humidity of the medium. When the receiver is located in an environment with obstructions in the direct line of sight to the satellite, reflections of the signals occurs. The reflected signal travels a longer path than the direct signal, which leads to signal distortion. For vehicle positioning, such multi-path effects are likely to occur when driving close to high buildings, trees or mountains. Further, the location of the reception antenna affects the performance of the receiver configuration. For vehicle navigation, an external antenna is typically mounted on the roof, Figure 2.7.

Modern GNSS receivers can mitigate some sources of errors. However, multi-path effects remain hardly predictable, because the magnitude of error depends on the environment of the receiver. Considering these different error causes, a horizontal positioning accuracy of 12 m in 95% of all

Fig. 2.7 – GPS antenna integrated in fin mounted on vehicle roof [14]

measurements is estimated for an industrial GNSS receiver [166]. It should be noted that performance measurements of GNSS depend on receiver configurations and local conditions of the environment.

The accuracy of GNSS navigation systems can be increased by differential GPS (DGPS) techniques. Compared to conventional GPS, the positioning accuracy can be improved to about 3.6 m in 95% of all measurements [166].

Even with the development of augmented GNSS systems, reliable positioning requires an unobstructed sky view to limit errors from multipath effects. For automotive applications, the integration of inertial measurement units (IMU) increases positioning accuracy and robustness. The vehicle motion can be predicted by the interpretation of inertial sensor signals, which is referred to as odometry. Inertial navigation systems (INS) contain an IMU and a processing unit to estimate the current vehicle position. Wheel encoders measure the rotation speed of the wheels, which is processed in a control unit to compute the distance traveled by the vehicle between two measurements. The accuracy of this estimate is affected by variations of the tire pressure and the road condition. Accelerometers and gyrometers are employed to determine the longitudinal and lateral acceleration and the yaw rate, which is defined by the angular velocity about the vehicle's vertical axis. Following models of vehicle kinematics, the relative position difference can be estimated with dead reckoning methods [87]. The drawback of dead reckoning lies in the accumulation of integration errors, which leads to a degrading positioning accuracy over time.

To overcome position drifts and deviates in adverse environment conditions, the advantages of absolute positioning from GNSS with relative positioning from dead reckoning can be combined [83]. A coupling of GNSS and INS smoothens the absolute position when multi-path effects occur and bounds the error accumulation. Loosely coupled architectures to integrate GNSS and INS correct the position update from dead reckoning with the absolute GNSS position. This integration can be performed by a Kalman filter approach [50], [160]. In a tightly coupled system, the integration of INS affects the signal interpretation within the GNSS receiver. Thus, knowledge about the internal receiver characteristics is required, which is only accessible for receiver manufacturers. The resulting system achieves an improved accuracy especially in situations with poor satellite visibility [135]. An example of performance with a commercial positioning system supplemented by vehicle sensors is shown in Figure 2.8. The accuracy of the resulting absolute position depends on the quality of the employed inertial sensors. In series production vehicles, low-cost sensors are unlikely to achieve the full potential of integrated positioning.

High performance positioning systems with tightly-coupled algorithms contain integrated high precision INS and GNSS receivers. For example, the accuracy of the RT3000 unit is specified as 1.8 m circular error probability (CEP) by the manufacturer [128]. However, the costs of such systems are beyond the limit for commercial employment in series production vehicles. Their field of application is rather found in highly automated driving (HAD) prototype vehicles or in vehicles operated by map manufacturers for raw data acquisition during the mobile mapping process.

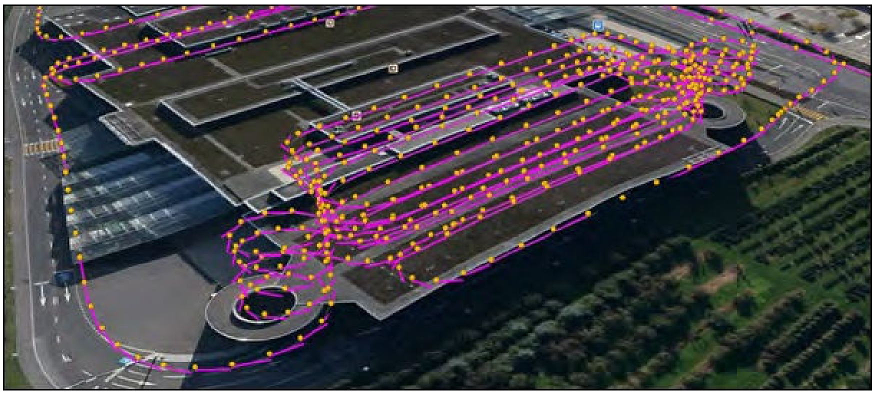


Fig. 2.8 – Positioning quality of u-blox 3D Automotive Dead Reckoning solution [135]. Vehicle trajectories on this test drive inside a park garage are computed by dead reckoning only. The positioning error at the exit of the park garage (upper-right) is smaller than one lane width.

The output of GNSS and INS measurement fusion contains the absolute position of the vehicle in lateral and longitudinal coordinates. Further, the velocity and the driving direction of the vehicle, also referred to as the heading, are returned.

In addition to INS, relative positioning of the vehicle can be enhanced with onboard vision systems. Modern vehicles are equipped with cameras capturing image streams of the environment. Examples of such video systems are given by stereo front cameras, which monitor the lane course ahead of the vehicle to supply data for driving assistance functions, Figure 2.9. Detections of the type and the distance to road markings allows relative positioning within the lane. When detected lane markings are georeferenced and stored in a map, absolute positioning errors can be corrected [162].

The captured video streams can further be processed by using visual odometry [55]. In each image frame of the vehicle camera, features of the captured scene are extracted. The vehicle egomotion is estimated by matching features of the reference frame to detected features of consecutive frames. Similar as with inertial navigation, visual odometry suffers from position drifts after some time. Also, large variations between consecutive image frames deteriorate position estimates.

In most cases, absolute vehicle positioning for navigation and ADAS applications relies on GNSS systems supplemented by INS or camera measurements. Thus, the accuracy of the absolute vehicle position depends highly on the environment and the integration method of GNSS, INS and camera measurements.

(a) (b) (c)

Fig. 2.9 – Camera systems for ADAS applications. (a) Stereo camera system [34]. (b) Visible road markings in image from front-facing camera [12]. (c) Lane departure warning system monitoring the road surface markings [35].

2.2.3 Automotive maps

Maps for automotive applications serve two purposes: First, they contain valuable information for predictive ADAS systems and navigation functions. Second, they provide a visual reference system for displaying the current position and the route for vehicle navigation.

The latter motivates maps, which can be easily visualized. This type of maps is stored in a raster format. However, the requirements of raster maps on storage capacity are high.

For a more compact storage solution, vector maps are considered. This format is defined by points, lines and polygons, which describe map features. Geographical coordinates specify the location of these geometrical primitives. Map attributes with additional information about the entity can be defined and attached to each geometrical primitive. Vector format maps meet requirements for navigation and ADAS well, because they produce less data volume than raster data and computations for map updates usually show better performance. This generalization ability qualifies vector maps as the preferable map format for automotive applications. Depending on the application, the trade-off between generalization and accuracy, which is limited by the storage capacity, needs to be balanced.

The map-referenced position defines the look-up position in the map to retrieve relevant and mostly predictive information about the current driving situation. Examples include guidance information for the following driving maneuver along a route or maximum drivable curve speeds and the distance to the following curve apex.

For positioning, maps should contain features that can be matched with measurements from vehicle onboard sensors or fused sensor signals. As described in the previous section, the absolute vehicle position together with the heading and velocity are estimated by the fusion of GNSS and INS measurements. Thus, a map containing absolute reference positions and the direction and expected speed of the road segments is suitable for comparison of map-matched candidate positions and the output of absolute vehicle positioning.

Vector maps for automotive applications follow the GDF format standard, which has been created to exchange road networks stored in map databases [78]. The GDF data model describes the road network as a non-planar graph with nodes and edges. Edges are also referred to as road segments or links. Intermediate nodes, also denoted as shape points, define the shape of an edge. The road geometry is approximated by a suitable number of shape points, which define a piecewise linear curve. Each node of the graph is associated with geographical coordinates defined by a pair of longitudinal and lateral numerical values. Additional features of a node or link are specified by attributes, e.g. driving direction, number of lanes or turn restrictions. The position of a link is intended to represent the road centerline. Opposite driving directions of a road are either specified by an attribute on the respective link or determined by individual links, if the driving directions are physically separated. An example of a map representation is shown in Figure 2.10.

To account for ambiguities in the definition of a “map”, the following terminology is applied throughout this work: A map denotes a digital representation of the road network as a graph consisting of links and nodes, which represent the topological structure. Geometric properties are defined by shape points of the graph, which are at least given for each node. Additional attributes contain further information about the link or node and relations between them.

Maps for vehicle navigation need to meet high requirements. Their content should be accurate, up-to-date and have a global coverage. Map building includes raw data acquisition with specially equipped vehicles with high-end measurement systems. Efficient storage and access to map database is guaranteed only by an advanced map compilation process. Therefore, a commercial business around map-making, efficient storage and distribution of digital map data exists. Among the established

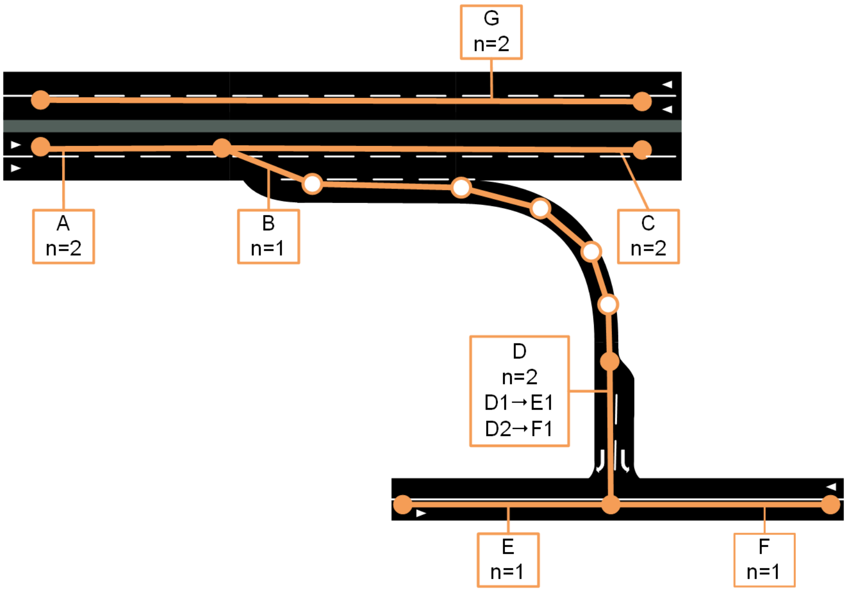


Fig. 2.10 – Example of a map representation in commercial map products based on the GDF format. Orange dots correspond to nodes, orange circles to shape points and orange lines to edges of the road graph. The number of lanes and the lane connectivities are given in the orange boxes. Links A,C,G represent separated driving lanes of a highway with two lanes for each driving direction. Link B represents an exit ramp, which splits into two lanes at link D. Lane connectivities at link D denote the permitted driving maneuvers to reach link E or F.

manufacturers for commercial navigation map data are TomTom and Nokia HERE. The community-based OpenStreetMap project provides free map data in a specific XML file format. Since maps are a generalized representation of the real word, absolute errors between the real road geometry and the map digitalization occur. This displacement can reach several meters and even more, if new roads have not been captured in the latest map version. Each map manufacturer has its own flavor of map generalization, which leads to inconsistencies between different maps.

The increasing significance of extended road preview for ADAS caused a demand for additional features. For lane-level positioning, the number of lanes is stored as a link attribute. In case of existing lane numbers, also the topology of connected lanes may be defined. Such lane connectivities describe the transition of lanes on incident links at the respective node according to present driving rules.

It is expected that in the next decade also commercial road map models will follow the trend towards a more detailed and accurate modeling of individual lanes. This is because future ADAS functions and automated driving require maps that allow matching of features detected by camera and LiDAR measurements. A comparison between standard polyline maps and the HD map model proposed by Nokia HERE is shown in Figure 2.11. However, the problem of up-to-dateness of map data is still unresolved, which limits their reliability to some extent.

2.2.4 Map-matching systems and algorithms

Since both available map data and absolute position estimates contain errors, robust map-matching methods are required to produce reliable results for vehicle navigation and ADAS.

Similar to object tracking methods, map-matching algorithms can be classified into single-hypothesis and multi-hypothesis methods. The first category consists of methods that match the estimate of the absolute vehicle position to the 'closest' position on the map given a suitable distance measure.

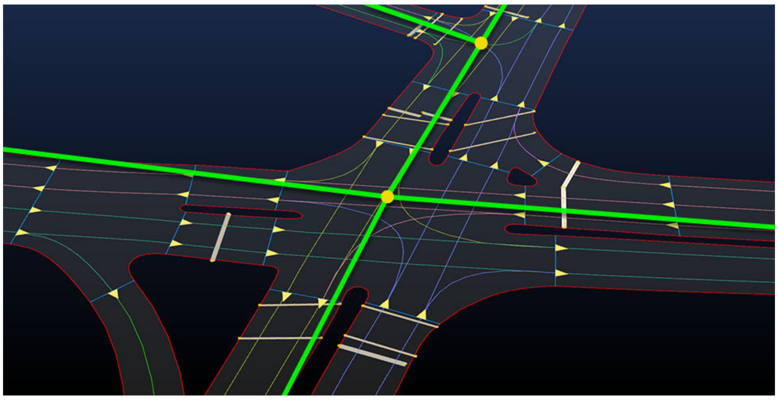


Fig. 2.11 – Comparison of standard map and next generation road map model from Nokia HERE [57]. The road geometry in the standard map is defined by polylines (green). The HD map format underneath uses more complex geometrical primitives for a precise representation of the geometry at the lane-level than the standard map.

Basic map-matching methods in this category consider only geometric properties of the road. The simplest distance measure computes the point-to-point distance between the estimate of the vehicle position and the candidate position on the map. The map-matched result is determined by a search for the node or shape point in the map, which minimizes this point-to-point metric [6]. Although this method is fast and easy to implement, its results are ruled by the resolution and accuracy of the shape point representation of the map. A second approach is given by point-to-curve matching. It computes candidate points along close road segments with the minimum distance to the estimated vehicle position. The candidate point with the smallest distance to the estimated vehicle position is determined as the map-matched result. In practice, this method often fails, when the correct road segment is not perfectly aligned with the trajectory of the vehicle, Figure 2.12a. As an extension of point-to-curve matching, the heading of the candidate road segments and the estimated heading of the vehicle are considered [171]. Due to the abstracted representation of the road geometry in the map, mismatches especially at road bifurcations cannot be avoided.

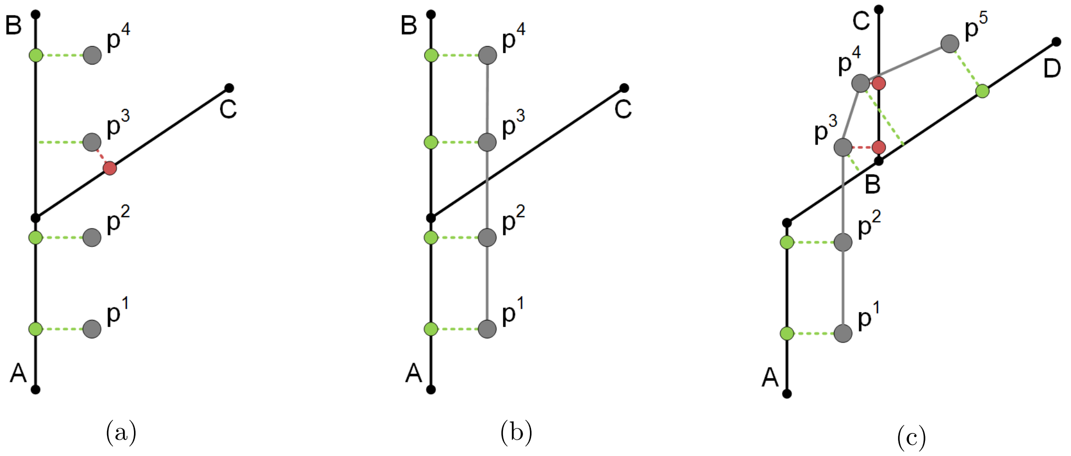


Fig. 2.12 – (a) Point-to-curve matching of vehicle trajectory points p^1, p^2, p^3, p^4 to a map consisting of three road segments A, B, C. For p^1, p^2 and p^4, the correct road segments A, B are identified by the nearest distance measure. The trajectory point p^3 is not correctly matched to B, because road segment C is closer. (b) Curve-to-curve matching of the same vehicle trajectory produces the correct result. (c) Curve-to-curve matching of vehicle trajectory points p^1, p^2, p^3, p^4 to road segments A, B, C, D. For p^1, p^2 and p^5, the correct road segments are identified. A mismatch occurs for p^3 and p^4, because the vehicle trajectory is more similar to the curve along the path $A - B - C$ than to the path $A - B - D$.

A more robust method of geometric analysis is proposed by curve-to-curve matching [171]. It starts by finding close nodes in the road network with the point-to-point matching. From these nodes, possible paths along the road network are constructed. The history of estimated absolute vehicle positions defines a piecewise linear curve. The similarity between the vehicle trajectory and the candidate paths on the map is computed by a suitable metric. Finally, the map-matched position is determined by the projection of the latest absolute vehicle position on the path along the road segment that is most similar. This method is more likely to produce a correct estimate of the traveled road segment, but finding a robust metric with an appropriate handling of map inaccuracies is difficult, Figures 2.12b and 2.12c. Comparing curves of different lengths is also a particular issue.

In addition to geometrical information, the integration of topological information from the map can exclude invalid matches. From the connectivity of links and lanes, only reachable road segments need to be considered. Turn restrictions can also serve as criteria for selecting the successor link at intersections. Advanced topological map-matching algorithms fuse multiple criteria into a weight-based evaluation of possible successor links [167]. In the example of Figure 2.12c, mismatches could be avoided, if the map contains connectivity information that road segment C is not reachable from B. However, examples exist where the topology allows multiple connected links and inaccurate position estimates and map data lead to ambiguous situations, Figure 2.13.

The geometrical and topological analysis of road features allows to select candidates for the map-matched position and to evaluate them to identify the most probable one. Related algorithms of such single-hypothesis methods operate in a heuristic manner, which prevents a straightforward integration of additional criteria. As a consequence, limitations of geometrical and topological algorithms are encountered, especially in ambiguous situations [138].

Multiple-hypothesis methods are considered as more advanced, because they produce more stable results in ambiguous situations. Ambiguities are likely to arise in dense urban areas, where multiple closely spaced roads are possible candidates for the map-matched position. The decision for the correct road segment is critical, if the road splits into more than one slowly diverging successor segments and the estimate of the absolute vehicle position lies between them. A synthetic example is shown in Figure 2.14, although real situations with not decidable link assignments likewise exist. In such situations, maintaining multiple hypotheses for the correct link would allow to defer the decision to a later point in time, when the confidence for either hypothesis is significant enough. In case the wrong hypothesis is selected as the map-matched position, multiple hypothesis methods can recover quickly from a wrong decision.

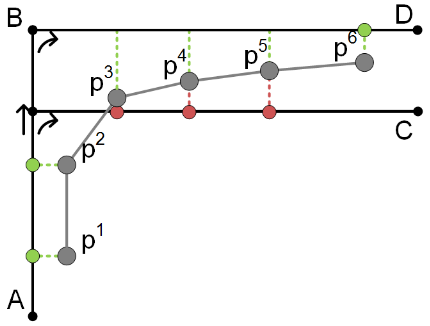


Fig. 2.13 – Example of mismatching with geometrical and topological information at a turn situation where two possible roads are close and run parallel. The vehicle trajectory points p^1 and p^2 are correctly matched. When the vehicle starts to turn, road segment C is likely to be assigned to p^3 and p^4. At position p^5 the situation is ambiguous, because C and D have equal distance. The map-matched position switches to D at position p^6, but the ambiguity is still not reliably resolved. The ground truth solution is required to decide if the matches were correctly assigned.

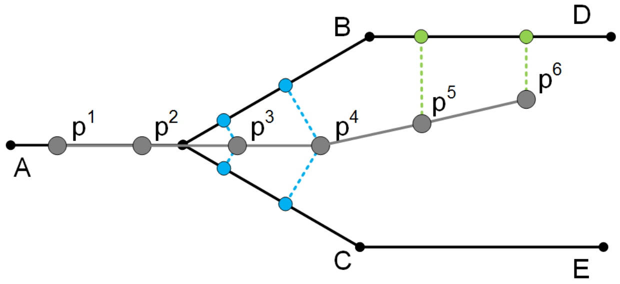


Fig. 2.14 – Map matching of a vehicle trajectory at road bifurcation. The first two trajectory points p^1, p^2 are correctly matched to link A. For p^3, p^4, the situation becomes totally ambiguous, because both links B and C match equally well. A multiple hypothesis method would track candidates on both links (blue). The matching of p^5, p^6 can then correctly be resolved to link D.

One method for solving the map-matching problem with multiple hypotheses is given by Multi-Hypothesis Tracking (MHT) [11]. In this context, map-matching is a single target tracking problem, where the map is interpreted as a sensor [136]. Observations from the map are associated with the predicted absolute vehicle position from the fusion of GNSS and dead reckoning. The probability of each hypothesis includes topological information of the map and the estimated position and heading of the vehicle. Because the number of hypotheses grows exponentially over time, pruning the hypotheses tree is necessary. Heuristic strategies to remove hypotheses for the map-matched position are suggested [80]. An MHT-related method proposes to replace topological map information with estimated probabilities for link switches to account for incomplete map data [118]. However, this approach heavily relies on the performance of GPS positioning.

Multiple road segments can also be handled with Belief Theory methods [50]. The absolute vehicle position is first estimated with an Extended Kalman Filter. Then map data is considered in several similarity criteria to measure the degree of belief in each candidate road segment. A similar approach can be combined with a bounded-error estimate of the absolute vehicle position [1].

In a probabilistic Bayesian framework, the concept of multiple hypotheses can be implemented with Markov localization methods. The idea is to estimate the vehicle position by Kalman filtering, while the estimate of the road segment is performed by Markov localization. A discrete Hidden Markov Model (HMM) includes topological information about the road network [154]. Still, ambiguous situations remain, where such methods cannot avoid mismatches [76]. An explicit map-matching step can be avoided, when Kalman filtering is constrained to the road network [163].

The integration of Monte Carlo localization approaches also known as particle filters allows to control the computational demand of probabilistic localization methods [54]. This development led to particle filter methods for vehicle localization and map-matching, which describe recursive algorithms for solving the Bayesian filtering problem with Monte Carlo techniques.

The map has a twofold role in vehicle localization methods: First, it acts as an observation additional to GNSS, INS and vision systems. Second, it can be interpreted as prior information, which constrains the estimated position to that map. From these aspects, two design principles for particle filter methods emerged.

The first class of particle filters estimates the vehicle position in geographical coordinates taking into account map data. The motion model of the system is often chosen as the sampling function for the particles. Particle weighting involves the observation model, which also includes map data. In its simplest form, the motion model calculates the vehicle position given the measured velocity and some error model, whereas the observation model computes the distance of the particle to the nearest road [68]. In are more complex form, the motion model can be determined by dead reckoning of measured accelerations with loosely or tightly coupled GNSS measurements. Efficiency of the filtering algorithm is increased if linear parts of the system are estimated with a Kalman filter and nonlinear parts are estimated with a particle filter [19]. Especially for the loosely coupled integration of INS and GNSS measurements, an extended Kalman filter can be employed. When positioning is not constrained to the map, map-aided positioning is a suitable description for the related vehicle localization approach. Results indicate that integrating map information into particle filtering produces more accurate results than without map data [58]. This effect demonstrates that GNSS outages can be mitigated by map data. This positioning technique is also referred to as “snapping” the vehicle position to the road network. Well-designed observation models include additional map attributes to provide a precise and reliable weighting of the particles as candidates of the map-matched position. Especially the topology of the road network is a strong criterion to rule out misleading candidates [130].

The lateral positioning of the vehicle on a digital map can be improved by visual features [29]. A lane detection module using the visual system of the vehicle computes its relative position within the lane, which is used during the particle weighting stage of the filter. This approach is limited to a position correction within the lane, but it is not suited to determine the driven lane on a multi-lane road. Measurements of the visual systems can also enhance the precision of the longitudinal vehicle position. The relative position to a detected traffic sign is compared with the mapped sign to assign high weights to particles which match this distance [107]. However, such approaches to incorporate visual information require precise maps of the road environment, which limit their application for industrial systems.

The second class of particle filters assumes that the vehicle position lies on the map, which circumvents explicit map-matching. Instead, the evolution of the position along the road network is calculated. The particles are propagated along the road network by the distance traveled since the last time step, while their weight is determined by a comparison of height and direction information [116]. Another hybrid approach of map-aided and map-constrained particle filters is proposed, which propagates particles along straight roads and switches to absolute positioning for curved road segments [39]. Results with the OpenStreetMap road network and a dead-reckoning positioning system indicate robustness of this approach. The strict constraint of the particles to the road network leads to a one-dimensional estimation problem. The traveled distance is proposed to be computed by the fusion of GNSS and IMU with an EKF. The ambiguity in the state estimate can be measured by an integrity monitoring strategy developed for particle filter map-matching methods [17]. Further development of this approach suggests to employ GNSS raw measurements directly in the observation model without the need to compute an absolute position estimate [52]. Results with geometrical and topological features of an industrial map product show the effectiveness of this approach.

Although not explicitly designed for solving the map-matching problem, approaches for road target tracking share similar concepts for map-constrained motion models [153]. Particle filters for combining on-road and off-road motion models prove the efficiency of this approach [96].

The first class of map-aided particle filters requires more computational capacity, because a larger state space needs to be explored. The second class of map-constrained particle filters is less flexible for handling map errors, but provides a more efficient solution for the map-matching problem in terms of particles sampling in the search space. The evaluation of particle filter concepts for map-matching is steered by the application. Map-matching for vehicle navigation is an enabler to retrieve map-referenced data. Therefore, vehicle localization in this context must produce a result constrained to the map, even if the geometry of road segments in the map is not accurate. Methods belonging to the second class of particle filters that meet this assumption are the preferred choice.

Increasing functionality of modern navigation systems requires map-matching to produce results with lateral accuracy to the lane-level. A simple curve-to-curve map-matcher demonstrates lane-level positioning, but it relies on the positioning accuracy of DGPS and specific map data with lane-based attributes [49]. To compensate GNSS outages, a pre-built map of detected lane boundaries is employed and fused during map-matching with lane marking detections from a series vision system [161]. Particle filter approaches for lane-level map-matching that are based on very detailed maps, achieve competitive results with GNSS and INS measurements [149], [164]. Mobile mapping data allows to extract clothoid segments for a precise approximation of the road geometry at the lane-level [7]. However, because such specific maps with decimeter accuracy are not available at an industrial scale, these methods have a limited field of applications.

Commercial vendors of built-in navigation systems depend on map data, which is delivered by map manufacturers as described in Section 2.2.3. Their developed map-matching algorithms are concealed within the embedded system, which exclude them for comparison with state-of-the art methods. Practical testing of onboard navigation systems revealed that mismatches of the current link occur at bifurcations and exit links or when GNSS is blocked by vegetation or inside tunnels. As of the publication date of this thesis, no commercial vehicle navigation system was available on the market, which demonstrates lane-level map-matching.

2.2.5 Discussion

Commercial navigation systems motivate the need for robust map-matching. Mismatches are more likely to occur the more ambiguous the situation is. Although research is focused on improving the quality of map-matching, evaluations particularly on ambiguous situations are not covered thoroughly. Multi-hypothesis methods have produced promising results on field test, but evidence is needed that they also perform well at challenging situations.

Precision and correctness of the map and the geographical positioning system are the two main factors that bound the accuracy of the map-matcher. Hence, considerable work has been spent to improve the quality of the map or the positioning system. These developments encouraged researchers to rely on enhanced maps or high-accuracy positioning. However, methods subject to these costly conditions have little chance to be applied at large scale. The development of map-matching methods with standard maps and low-cost positioning becomes challenging, because robustness needs to be ensured.

These practical considerations become even more relevant when accuracy at the lane-level is aimed at. It has not been addressed yet how already mapped lane features such as lane connectivities can be exploited to increase the accuracy of map-matching at the lane-level.

The goals of this work considering particle filtering for map-matching are formulated as follows:

- Develop a particle filter method and evaluate its quality at ambiguous situations.
- Employ lane topology information for lane-level map-matching.
- Achieve robustness of the method to tolerate deficiencies from standard map and low-cost positioning.

3 Particle filters for localization and tracking

3.1 Bayesian filtering

Vehicle localization and tracking problems can be formulated as estimating the states of a dynamical system from sensor measurements. Dynamical systems describe the evolution of time series in a state space. They are modeled in a discrete state-space representation, assuming that measurements arrive sequentially at discrete time-steps.

The state vector $x_t \in \mathbb{R}^n, t \in \mathbb{N}$ contains all information about the target that are relevant for the application or define internal parameters of the system. For tracking applications, the state vector contains the vehicle position and the time series of state vectors $\{x_t, t = 0, \ldots, T\}$ defines the trajectory of the target. The evolution of the state vector over time is given by the motion model

$$x_t = f_t(x_{t-1}, o_t), \tag{3.1}$$

where $f_t : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$ is the state transition function and $o_t \in \mathbb{R}^p$ is a stochastic noise process characterizing uncertainties in the motion of the target.

Sensor measurements are represented by the measurement vector $y_t \in \mathbb{R}^m$. They are related to the current state by the observation model

$$y_t = h(x_t, e_t), \tag{3.2}$$

where $h_t : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^m$ is the measurement function and $e_t \in \mathbb{R}^r$ denotes the measurement noise.

The Bayes filtering approach proposes to estimate the posterior distribution $p(x_t|y_{1:t})$ of the state given all measurements up to the current time, $y_{1:t} = \{y_1, \ldots, y_t\}$. The posterior holds the complete information about the estimate, because it allows to derive any statistics.

To design a recursive Bayesian algorithm, it is assumed that the state sequence is a first order Markov process: The past and future is conditionally independent given the current state. In terms of probabilities, the motion model defines the prior probability $p(x_t|x_{t-1})$ and the observation model defines the likelihood $p(y_t|x_t)$. The Markovian assumption implies $p(x_t|x_{t-1}, y_{1:t-1}) = p(x_t|x_{t-1})$ and $p(y_t|x_t, y_{1:t-1}) = p(y_t|x_t)$.

The posterior can be transformed using Bayes' rule and the Markov property

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})}. \tag{3.3}$$

The marginal distribution $p(x_t|y_{1:t-1})$ can be derived as

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}. \tag{3.4}$$

These equations define recursive computations with a prediction (3.4) and update (3.3) step. For complex problems, a direct computation is not feasible, because the evaluation of complex integrals prevents an efficient implementation. A closed form solution exists only for restrictive cases. If the system is linear with Gaussian process noise, then the Kalman filter [56], [82] provides the optimal Bayesian solution to this problem.

If an analytical solution does not exist, approximations for the probability distributions are required to make the problem tractable. The extended Kalman filter (EKF) is based on a local linearization of the system [2]. Since the noise model is assumed Gaussian, the EKF will produce poor results if the true posterior is non-Gaussian. Further, the EKF is not robust against multi-modal probability distributions. A numerical approximation of the integrals is provided by grid-based methods [21]. The idea is to find a discrete representation of the state-space using a regular grid. This approach has limited flexibility, because the posterior is approximated in a fixed resolution. The computational cost of grid-based methods is high and the complexity increases drastically for an increasing dimension of the state space. The recursive Bayesian estimation for the nonlinear, non-Gaussian case can be computed with numerical integration [90]. However, this method has high requirements of memory and computational time for high dimensional state spaces.

3.2 Particle filter

3.2.1 Standard particle filter

The discussed approaches in the previous section are little attractive for an efficient solution of the general nonlinear, non-Gaussian filtering problem. With the availability of modern computers, Monte Carlo techniques emerged as an alternative approach. The idea of particle filters is to approximate the integrals in recursive Bayesian filtering (3.4), (3.3) by Monte Carlo sampling methods. From the principle of importance sampling, a recursive formulation can be obtained, which describes the general class of Sequential Monte Carlo methods [47].

An empirical representation of the posterior $p(x_{0:t}|y_{1:t})$ is obtained by a set of random samples $\{x_{0:t}^{(i)}, i = 1, \ldots, N\}$ drawn independently from this distribution:

$$\hat{p}(x_{0:t}|y_{1:t}) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_{0:t}^{(i)}}(x_{0:t}), \tag{3.5}$$

where $\delta_{x_{0:t}^{(i)}}$ is the Dirac delta mass at $x_{0:t}^{(i)}$. Since sampling from the posterior is only possible for restricted simple cases, this approach is not practical. For the general case, the importance sampling method [63] provides an approximation. It introduces an arbitrary distribution $q(x_{0:t}|y_{1:t})$, called the importance function or proposal distribution, which allows to generate samples from it. If the support of $q(x_{0:t}|y_{1:t})$ contains the support of $p(x_{0:t}|y_{1:t})$, then the approximation of the posterior is given by

$$\hat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_{0:t}^{(i)}}(x_{0:t}), \tag{3.6}$$

where $w_t^{(i)}, i = 1, \ldots, N$ are defined as the normalized importance weights following

$$w_t^{(i)} \propto \frac{p\left(x_{0:t}^{(i)}|y_{1:t}\right)}{q\left(x_{0:t}^{(i)}|y_{1:t}\right)} \tag{3.7}$$

and $\sum_{i=1}^{N} w_t^{(i)} = 1$. The drawback of this integration method is that it cannot be applied recursively. However, if the importance function allows a factorized formulation, then a sequential version can be derived [48]. It can be shown that an importance function which follows

$$q(x_{0:t}|y_{1:t}) = q(x_{0:t-1}|y_{1:t-1})q(x_t|x_{0:t-1}, y_{1:t}) \tag{3.8}$$

enables a recursive computation of the importance weights [3]

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(y_t|x_t^{(i)}\right) p\left(x_t^{(i)}|x_{t-1}^{(i)}\right)}{q\left(x_t(i)|x_{0:t-1}^{(i)}, y_{1:t}\right)} \tag{3.9}$$

and the extension of the sample sequence by drawing new samples from $q\left(x_t|x_{0:t-1}^{(i)}, y_{1:t}\right)$. These recursive computations are known as the sequential importance sampling (SIS) method.

After SIS is applied for some iterations, the so-called degeneracy problem occurs. It can be proven that the variance of the importance weights will increase over time [92]. This behavior of SIS can be practically observed, since all but one importance weight will be close to zero. The degeneracy leads to a poor estimate of the posterior and causes the importance weights to be updated, although their contribution to the approximation is negligible.

This problem can be addressed by selecting an importance function, which minimizes the variance of the importance weights. The optimal importance function in this sense is given by

$$q\left(x_t|x_{0:t-1}^{(i)}, y_{1:t}\right) = p\left(x_t|x_{t-1}^{(i)}, y_t\right), \tag{3.10}$$

as shown in [48], [92]. In general, the practical use of the optimal importance function is limited, because the resulting computations cannot be obtained in closed form. For most applications, suboptimal importance functions need to be designed.

The severity of degeneracy can be estimated by evaluating the effective sample size [92], [112], which is approximately given by

$$N_{\text{eff}} = \frac{N}{1 + \text{Var}(w_t^\star)}, \tag{3.11}$$

where $w_t^\star$ is the unnormalized importance ratio from (3.7). A computable approximation is provided by

$$\widehat{N_{\text{eff}}} = \frac{1}{\sum_{i=1}^{N} {w_t^{(i)}}^2}. \tag{3.12}$$

The effective sample size N_{eff} measures the efficiency of the importance sampling. It provides an estimate of the degeneracy of the algorithm, as small values of N_{eff} indicate that the sample mass is concentrated at few samples and thus degeneracy is severe.

To mitigate the degeneracy problem, a resampling step was proposed by [141] and first introduced in the context of nonlinear filtering [65]. The idea is to eliminate samples with negligible importance weight and duplicate samples with significant weight. The resulting set of samples replaces the original samples and the new weights are uniformly distributed. In this way, the weighted representation of the posterior (3.6) is transformed to an unweighted representation with the new drawn samples $\tilde{x}_{0:t}^{(i)}$,

$$\tilde{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N} \delta_{\tilde{x}_{0:t}^{(i)}}(x_{0:t}). \tag{3.13}$$

A new set of samples is generated by drawing with replacement from the approximation of the posterior (3.6). Thus the probability of the new samples is proportional to the normalized importance weights. This resampling scheme corresponds to the basic *multinomial resampling* [65], [141]. Since resampling increases the variance of the Monte Carlo approximation, alternative resampling methods should be considered. Among these methods are *stratified resampling* [89], *residual resampling* [111] and *systematic resampling* [89], which have different properties in terms of variance reduction and computational complexity [46], [73]. The SIS method with resampling is referred to as sequential importance resampling (SIR) or sequential Monte Carlo (SMC). The SIR method is also known as particle filters, where samples correspond to "particles". In the computer vision community, SIR is also known as the "CONDENSATION" algorithm [79].

Resampling introduces some undesirable effects, among these are "sample impoverishment" meaning that the diversity of the samples decreases. Therefore, a resampling schedule is preferred rather than performing the resampling step at each iteration. A suitable trigger for resampling is the efficient sample size (3.12). When it drops below a certain threshold, resampling is initiated.

In the context of filtering, only the marginal posterior $p(x_t|y_{1:t})$ is required for estimation. Thus, a favorable importance function for filtering applications depends only on the previous state and the current observation, $q(x_t|x_{0:t-1}, y_{1:t}) = q(x_t|x_{t-1}, y_t)$. The weights computation is then simplified to

$$w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p\left(y_t|x_t^{(i)}\right) p\left(x_t^{(i)}|x_{t-1}^{(i)}\right)}{q\left(x_t^{(i)}|x_{t-1}^{(i)}, y_t\right)}. \tag{3.14}$$

A generic algorithm for SMC for filtering is summarized as

SMC for filtering

```
Initialization
  0 :  For $i = 1, \dots N$, sample $\bar{x}_0^{(i)} \sim p(x_0)$ and set $w_0^{(i)} = 1/N$.
For $t = 1, \dots, T$
  1 :  For $i = 1, \dots N$, sample $x_t^{(i)} \sim q(x_t|\bar{x}_{t-1}^{(i)}, y_t)$.
  2 :  For $i = 1, \dots N$, calculate the importance weights
       $w_t^{\star(i)} = w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)})p(x_t^{(i)}|\bar{x}_{t-1}^{(i)})}{q(x_t^{(i)}|\bar{x}_{t-1}^{(i)}, y_t)}$ and normalize them $w_t^{(i)} = \frac{w_t^{\star(i)}}{\sum_{i=1}^{N} w_t^{\star(i)}}$.
  3 :  If $N_{\text{eff}}$ is below a threshold:
             Resample $\{x_t^{(i)}, i = 1, \dots N\}$ to obtain the new sample set $\left\{\bar{x}_t^{(i)}, i = 1, \dots N\right\}$.
             Assign equal weights $w_t^{(i)} = 1/N$.
       Else:
             For $i = 1, \dots N$, $\bar{x}_t^{(i)} = x_t^{(i)}$.
```

At time t the approximation of the filtering density is given by

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t^{(i)}}(x_t). \tag{3.15}$$

The first practical realization of SIR in the context of nonlinear filtering was introduced by the *bootstrap filter* [65]. Here, the prior $p(x_{0:t}|x_{0:t-1})$ is chosen as the importance function, which leads to a simplified computation of the importance weights

$$w_t^{(i)} \propto w_{t-1}^{(i)} p\left(y_t|x_t^{(i)}\right). \tag{3.16}$$

Because multinomial resampling is applied at each iteration, the weights satisfy

$$w_t^{(i)} \propto p\left(y_t | x_t^{(i)}\right). \tag{3.17}$$

The calculation of the importance weights now requires only evaluation of the likelihood at the sample position. Although this configuration of SIR leads to a simple implementation, it has some disadvantages. The choice of the prior as the importance function causes a poor approximation of the posterior, if the likelihood is peaked compared to a more uniformly shaped prior. In this situation, most samples will be placed in regions of low likelihood density, which leads to small importance weights. The resulting set of weighted particles will not approximate the true posterior until their number is chosen sufficiently high. This type of importance sampling is inefficient, because a large number of samples is required until samples coincide with regions of high posterior density.

3.2.2 Advanced particle filters

The SIR algorithm provides a generic approach for filtering, but the previous section showed that the choice of the importance function and the design of the resampling step is crucial for the quality and performance of the approximation. In literature, several variants of particle filtering or SIR are proposed.

The auxiliary particle filter [134] aims to predict promising locations for the samples by propagating only particles selected according to an auxiliary index variable, which is drawn from a distribution proportional to some intermediate weights. The computation of these intermediate weights involves the evaluation of the likelihood at intermediate sample locations. These intermediate points are defined as representatives of the states x_t given $x_{t-1}^{(i)}$, for example the mean. The advantage of the auxiliary particle filter lies in its ability to consider current measurements. This strategy leads to a more efficient sampling than SIR, if the noise of state transition function is large. However, the computational effort is increased, because the weights computation needs to be performed twice.

To improve the performance by variance reduction, marginalized particle filters also known as Rao-Blackwellized particle filters have been proposed [47]. The idea of marginalized particle filters is to approximate linear, Gaussian substructures of the model with optimal filtering. The application of a Kalman Filter for estimating the linear state variables reduces the computational cost and decreases the variance of the estimate. The underlying assumption is that the system model can be separated in linear and nonlinear parts [146]. The states vector follows this structure and is partitioned in linear and nonlinear states, $x_t = (x_t^l, x_t^n)$. From the definition of conditional probability it follows

$$p(x_t|y_t) = p(x_t^n|y_t)p(x_t^l|x_t^n, y_t). \tag{3.18}$$

If $p(x_t^l|x_t^n, y_t)$ can be formalized as a linear, Gaussian system, then a Kalman filter can provide the estimate. The nonlinear part described by $p(x_t^n|y_t)$ is then estimated by a particle filter.

Further strategies for advanced particle filtering include adaptive importance sampling (AIS) schemes. Early work on this topic in the context of Monte Carlo integration is found in [126]. The idea of AIS is to improve the allocation of the particles by adjusting parameters of the importance function automatically. At each stage of AIS, a number of particles is sampled and the weights of the particles are used to update parameters of the importance function for the next stage until convergence. The application of AIS on importance functions that are represented by mixture densities is demonstrated in [26]. Adaptive importance sampling in the particle filter framework is shown in [155] and [157]. In both approaches, the parameters of the importance function are updated online during each iteration of the particle filter. In each update stage, current measurements are incorporated to bias the particle sampling towards the optimal importance function. In the context of object tracking, adaptive sampling is shown in [178]. A density grid is constructed and updated to estimate promising regions and produce an efficient sampling of the particles.

Alternatively, kernel particle filters are proposed to increase the sampling efficiency [28]. This technique shifts particles towards dominant modes of the posterior, which are found by a kernel density estimate. However, this method requires additional weight computations.

The computational efficiency of particle filtering can be improved by adapting the number of particles. The KLD sampling approach measures the approximation error of the particle representation by the Kullback-Leibler divergence and generates a number of particles, such that this measure is bounded [53]. The resulting particle set size is small if the posterior is concentrated at a small portion of the search space and it is large if it spreads out widely.

3.2.3 Multiple target filtering

Tracking multiple objects relates to the problem of simultaneously estimating states of the respective objects. In the particle filter framework, the natural extension would be to concatenate the states of all objects into a single state vector. However, this joint particle filter approach causes the state space to grow exponentially with the number of objects, which affects likewise the number of required particles. Further, the number of targets is often not known and must therefore be estimated as an additional parameter. The increased computational cost makes this approach not attractive for a large number of objects. Tractable implementation strategies for estimating the joint multi-target probability density are proposed in [97].

Alternatively, multiple objects are tracked by assigning an individual particle filter to each object. The dependencies of these single instances of the particle filter need to be modeled explicitly. In particular, the data association problem needs to be resolved to find the correct match between measurements and objects. A solution to this problem in the context of particle filtering is proposed by [75]. This method models the data association problem with assignment probabilities and estimates them with a Markov Chain Monte Carlo technique. Another approach uses Joint Probabilistic Data Association [36] to estimate the unique correspondences between measurements and objects with particle filtering [148]. Further Monte Carlo techniques to apply data association strategies exist, which enable an efficient computation for nonlinear, non-Gaussian problems [168].

3.3 Robust state estimation

Object localization and tracking systems should produce a reliable and stable output, which holds all relevant information about the object state. For vehicle tracking, this includes the unique position of the vehicle at each time step. It is reasonable that the result should follow the same dynamics as the vehicle, which implies spatial smoothness of the estimated trajectory. Further, a suitable confidence measure of the result needs to be computed.

The strength of the particle filter is to approximate the whole posterior density of the state. From this discrete representation, a point-estimate of the state is obtained as the requested tracking output. It can be assumed that dominant hypotheses of the vehicle state correspond to modes of the posterior.

Simple features of the posterior density include the weighted mean, which is derived from the weighted particles by

$$x_{\text{mean}} = \sum_{i=1}^{N} w_t^{(i)} x_t^i \tag{3.19}$$

or the maximum-a-posteriori estimate, which is given by the particle with the highest weight. However, neither estimate is suitable for obtaining a robust, reliable and stable hypothesis for the vehicle position from a multi-modal posterior. A combination of these two estimates is the robust mean, which is computed by the weighted mean within a window around the particle with the highest weight. Although this estimate is less sensitive to outliers, it needs the definition of an appropriate window size. The weighted median as another alternative is a robust estimator, but also fails to compute multiple modes of the posterior. A solution to this mode seeking problem is proposed by clustering techniques, such as k-means or mean-shift [30]. The application of these methods to vehicle tracking problems requires the definition of a suitable distance measure in the state space.

In general, the choice of an appropriate estimation method depends on the application and on the characteristics of the prior and the likelihood of the underlying system. A comparison of k-means, maximum-a-posteriori, weighted mean and a new clustering algorithm for robot self-localization in shown in [104]. This work concludes that further work is required to overcome limitations of the developed method for navigation application along branching paths.

The question remains, at which step during particle filtering the estimates should be computed. In the work of [111], it is suggested to perform state estimation before resampling to avoid additional variation in the particles.

3.4 Discussion

Particle filtering is a recursive approach to solve nonlinear, non-Gaussian state estimation problems. It provides a flexible, easy to implement method to approximate multi-modal probability densities. For a fixed size of the state vector, the computational complexity depends linearly on the number of particles, which can be used to control the precision of the approximation. The structure of the particle filter enables a partly parallel implementation on multiple processors, which makes this method highly scalable on modern distributed computing systems.

The design of a particle filter method depends on the application and poses three main challenges:

First, the choice of an appropriate importance function is crucial for an efficient sampling process. The simple choice of the prior probability might produce samples with low weights, which leads to inefficiency of the method. Thus, advanced strategies are required to design an importance function, which approaches the optimal importance function. Adaptive importance sampling techniques have been demonstrated to produce efficiently sampled particles. However, the potential of adaptive strategies for particle weighting has not been studied in great detail yet. Here, it is desirable to obtain a precise representation of the posterior with a low number of particles and weight computations, while dealing with the trade-off between performance and computation cost.

Second, the data association problem in multi target systems needs to be resolved efficiently. Tracking applications require advanced strategies to deal with occlusion and clutter. Specific knowledge of object dependencies are useful to prune the hypotheses tree for correspondences between measurements and objects. In the same context, suitable features need to be extracted from measurements to model a likelihood function that discriminates similar candidates for the targets' states but also tolerates measurement errors.

Third, vehicle tracking applications require not only the estimation of the multi-modal probability distribution, but also inference of the states at each time step. Robust state estimation techniques need to be developed to derive suitable statistics from the set of weighted particles. Often, ambiguities in the best hypothesis occur, which require suitable treatment. This point of often neglected in literature, but needs to be considered for the development of a comprehensive tracking system.

4 Airborne vehicle tracking

This section introduces a particle filter method for tracking multiple vehicles in a sequence of aerial images. Three sources of images and their individual properties are described, two systems are mounted on an airplane, the other one is mounted on a UAV. The models for particle propagation and weighting allow to capture the vehicles under abrupt motion and appearance changes caused by the low spatial and temporal resolution of the image sequences. Novel strategies for adaptive motion and observation models are proposed to combine the desired precision and robustness of the tracker. To avoid target losses, an observation model update approach is given. Tracking multiple vehicles is handled by individual particle filters for each target. A graph-based method is suggested to find the global best assignment between vehicles and hypotheses of their position. The results in this chapter demonstrate the increased quality of the tracking results for the advanced strategies compared to standard models in the particle filter.

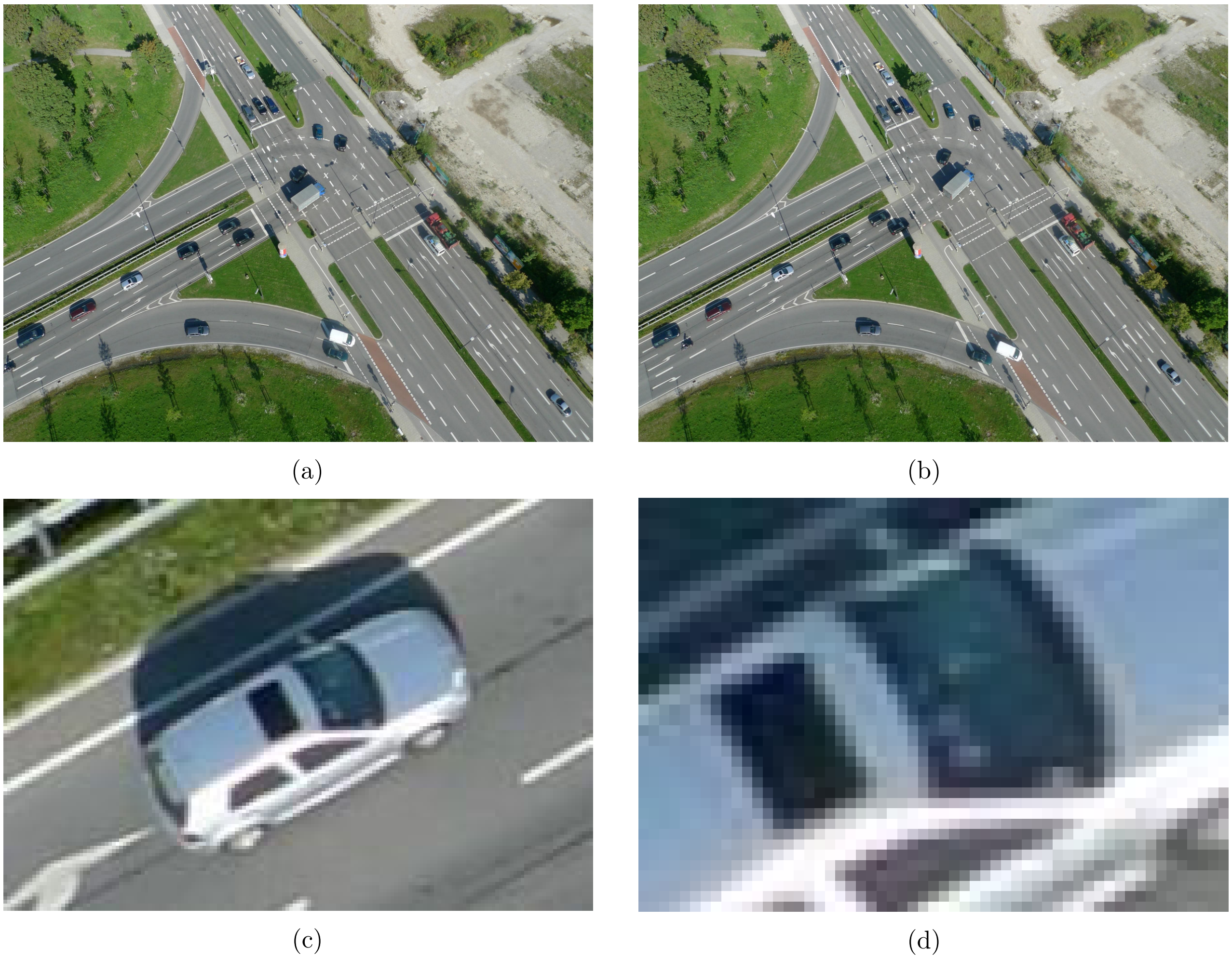
4.1 Platforms and imaging systems

The image sequences are taken from two types of flight systems, the DLR Cessna aircraft camera system and the AscTec Falcon 8 UAV, as described in Section 2.1.2. The properties of the mounted camera systems when operated in the continuous image acquisition mode are summarized in Table 4.1.

flight system	camera type	GSD (per pixel)	average frequency
DLR Cessna aircraft 3K	3 x Canon EOS 1Ds Mark II	15 cm	2 Hz
DLR Cessna aircraft 3K+	3 x Canon EOS 1Ds Mark III	13 cm	2 Hz
AscTec Falcon 8 UAV	Panasonic DMC Lumix LX3	$\approx$ 5 cm	2 Hz

Tab. 4.1 – Properties of the employed camera systems on two different flight systems

The camera system mounted on the airplane is embedded into an onboard image processing chain for georeferencing and orthorectification of the images [103]. The results of the rectification process are quite exact, because GPS/IMU measurements are considered. For the UAV images, orthorectification of the images was done in an offline postprocessesing step without information about the geographical position of the camera. The rectification errors for both image processing systems are observed to range below 1 meter . Since the flight height of the UAV was not measured, the GSD must be estimated by visual inspection of the images. The camera position can be assumed to be stationary for the UAV system. In contrast, the movement of the 3K/3K+ camera system is at least the minimum speed of the aircraft, which leads to parallax effects. The desired camera view angle is in nadir direction, because appearance changes of the vehicles due to perspective change are minimized in this way. This view angle could not be achieved with the UAV system because of state takeoff regulations, which required to choose the viewpoint a few meters beside the actual scene. The images from the aircraft are aligned in nadir direction with small deviations at the image boundaries due to the flight height of around 1000 meters. The three images of the 3K/3K+ system are stitched into a single panoramic image. The image quality from the UAV camera depends on the motion of the system. Although the camera



(a) (b)

(c) (d)

Fig. 4.1 – Images from UAV camera system, (a), (b) consecutive images of overlooked area show temporal resolution of vehicle motion, (c) perspective view of vehicle from non-nadir direction, (d) detailed view of image resolution at estimated GSD of 5 cm.

platform is stabilized, shaking of the camera cannot be completely avoided in windy conditions, which leads to blurring of the images.

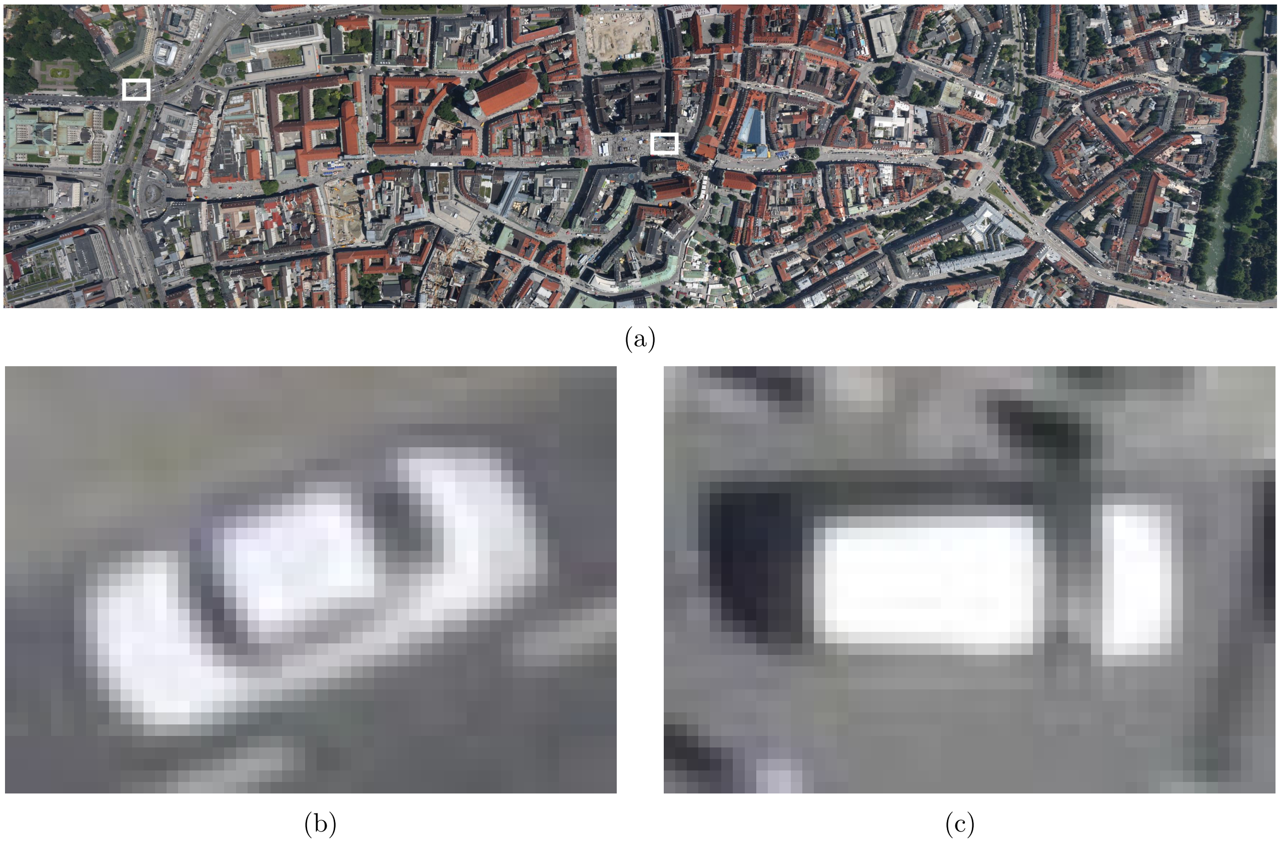
The flight campaigns to obtain representative image sequences for the tracking application were conducted in urban areas with dense traffic. Due to limitations of the 3K/3K+ camera data storage system in the continuous image acquisition mode, the image sequences are limited in time and typically terminated after less than 23 seconds. For the UAV camera system, the sequences are only limited by the storage capacity. However, since the UAV is set stationary, the image sequence can be terminated once all vehicles have left the scene.

4.2 Bayesian system model

In the context of Bayesian filtering, the tracking problem is defined as estimating the state vector X_t given the history of measurements $Y_{1:t}$. The state vector contains the position of a single vehicle in image coordinates and its velocity as the first derivatives in both directions,

$$X_t = (x_t, y_t, \dot{x}_t, \dot{y}_t) \in \mathbb{R}^4. \tag{4.1}$$

The extension of the system for tracking multiple vehicles is explained in Section 4.5.



(a)

(b)

(c)

Fig. 4.2 – Images from aircraft camera system 3K+. (a) Image of overlooked area with selected vehicles for detailed view (white rectangles). Detailed view of vehicles from (b) almost nadir direction and (c) nadir direction with GSD of 13 cm.

Under the Markovian assumption and following the notation from section 3.1, the state model is described by a function of the previous state and a noise variable o_t,

$$X_t = f(X_{t-1}, o_t). \tag{4.2}$$

The measurements Y_t in this system are based on the image data at the current time step. Feature extraction computations are used to obtain observations about the location and appearance of the target. The observation model, which relates the measurements to the states, is given by the measurements function of the current state and a noise variable r_t,

$$Y_t = h(X_t, r_t). \tag{4.3}$$

4.3 Particle filter for airborne vehicle tracking

Feature extraction typically involves image processing operations that lead to non-linearity of the observation model. The high GSD of airborne images makes it difficult to distinguish similar vehicles by their shape and appearance. This ambiguity leads to multi-modality of the posterior in dense urban traffic. Because of this multi-modality, particle filtering is suitable to track multiple hypotheses about the vehicle position. The particle filter method for vehicle tracking is designed such that the importance sampling function is defined by the motion model of the target. In this application, the targets are given by individual vehicles.

4.3.1 Target dynamics

The particles are sampled from the motion model, which predicts the motion of the vehicle. Because of the absence of vehicle dynamics measurements, a straight motion model at constant velocity is the preferred choice [109]. The evolution of the state vector is given by

$$X_t = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} X_{t-1} + \begin{pmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} o_t, \tag{4.4}$$

where Δt denotes the interval between consecutive time steps. The parameter $o_t \in \mathbb{R}^2$ follows a white Gaussian noise process and models random acceleration of the vehicle in image coordinates. The low sampling rate of the image sequence causes large uncertainties about the target motion compared to classical video tracking applications. Errors in the motion prediction can occur when the vehicle turns, which is beyond the limitations of the straight motion model. Robustness against such motion errors needs to be ensured by a suitable choice of the noise parameter, which controls the spreading of the particles.

The maximum acceleration in lateral and longitudinal direction of the vehicle follows the physical rules of transfer of forces from the tires of the vehicles to the road. Under stable driving conditions of the vehicle, the lateral and longitudinal forces F_x, F_y and the tire load F_z are subject to the circle of friction, which is defined by

$$\sqrt{F_x^2 + F_y^2} \leq \mu F_z, \tag{4.5}$$

where μ is the friction coefficient [121]. The friction coefficient depends on the road condition and the tire characteristics. Under optimal friction conditions, a value of $\mu = 1$ can be assumed [93]. The equation (4.5) can then be approximated by the lateral and longitudinal acceleration of the vehicle as

$$\sqrt{a_{\text{lat}}^2 + a_{\text{lon}}^2} \leq g, \tag{4.6}$$

where $g = 9.8\, m/s^2$ defines the acceleration due to gravity. This formulation does not take the dynamic tire forces into account.

The standard deviations of the noise distribution of o_t are determined experimentally. It is suggested to set the parameters to $\sigma_{\text{lon}} = 15\ m/s^2$ for longitudinal and $\sigma_{\text{lat}} = 7\ m/s^2$ for lateral direction of the vehicle movement to obtain the noise from

$$o_t^{\text{lon,lat}} = \mathcal{N}((0,0), \text{diag}(\sigma_{\text{lon}}^2, \sigma_{\text{lat}}^2)) \tag{4.7}$$

before transformation to image coordinates.

Note that the resulting lateral and longitudinal accelerations after sampling from (4.7) can violate the inequality (4.6), because image rectification errors need to be considered.

4.3.2 Observation model

Extracting robust features of shape and appearance of the vehicles allows to distinguish a target vehicle from other vehicles and background. Variations of illumination due to sun light reflections or shadows from vegetation and buildings lead to appearance changes of the vehicles. Because of the low sampling rate of the image sequences, these changes can appear abruptly. When the view angle of the camera is in nadir, the shape of the vehicle remains fairly constant. Otherwise, the shape of the vehicle includes structures from the vehicle sides, which complicate re-identification under vehicle turn maneuvers. Both shape and appearance are affected, when the vehicle is partly occluded by vegetation, buildings or bridges. Scaling of the vehicles can be neglected assuming a flat road level, because the distance between the camera and the scene is normalized during the image rectification process. The main source of ambiguity for vehicle tracking in airborne images is caused by the high GSD, which degrades the discriminative power of extracted image features. Tracking errors are almost inevitable when identical vehicles are close to each other.

For each vehicle, a template is created that defines the shape and appearance of the target. During particle weighting, this template is matched against the candidate region around the current particle. The resulting matching score is employed in the weighting function, which is given by the measurement function. The initial template is computed by feature extraction in a rectangular region of the image around the initial vehicle position X_0. The dimensions of the rectangle are computed by generic values for the outer dimensions of a typical mid-size car and transformed to image coordinates by the (estimated) GSD. The templates are encoded in high-level variables Ω_t, Γ_t, which contain all parameters for the shape and color matching. The Bayesian system model for the single vehicle tracker with template matching is shown in Figure 4.3.

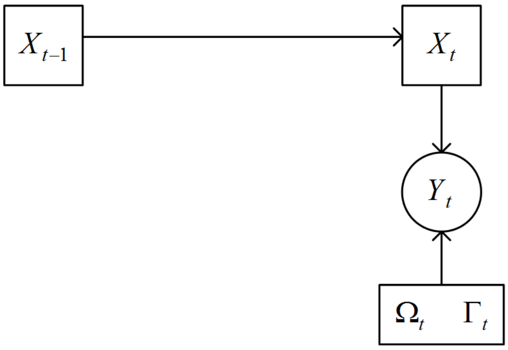


Fig. 4.3 – Schematic illustration of the Bayesian network modeling with state vector X_t, measurements Y_t and template matching parameters Ω_t, Γ_t.

For particle weighting, the scores from shape and appearance matching are evaluated separately. Therefore, the measurement function is combined of two components, $h^{\mathrm{c}}(X_t, e_t^{\mathrm{c}}, \Omega_t)$, $h^{\mathrm{s}}(X_t, e_t^{\mathrm{c}}, \Gamma_t)$ for weighting the color matching score and for weighting of the shape matching score. The fusion of these two components is defined as the linear combination of the individual measurements functions, because statistical evaluation showed that sum rule outperforms product rules in typical classification applications, [91], [115]. The final likelihood function that determines the particle weight w_t is given as

$$w_t = h(X_t, r_t) = \alpha^{\mathrm{c}} h^{\mathrm{c}}(X_t, e_t^{\mathrm{c}}, \Omega_t) + \alpha^{\mathrm{s}} h^{\mathrm{s}}(X_t, e_t^{\mathrm{s}}, \Gamma_t), \tag{4.8}$$

where $r_t = (e_t^{\mathrm{c}}, e_t^{\mathrm{s}}) \in \mathbb{R}^2$ and the mixture coefficients satisfy $\alpha^{\mathrm{c}} + \alpha^{\mathrm{s}} = 1$.

Established techniques to represent a target by means of shape and color features are known from the field of image processing and computer vision. Their application for representing and identifying vehicles in low resolution airborne images is described in the following sections.

4.3.2.1 Shape based matching

The characteristic shape of a vehicle is a suitable feature to recognize an individual vehicle and discriminate it from other objects. The vehicle's shape is specified by the contour of the outer boundary between the vehicle and the road and other structures within the vehicle's region, e.g. the sunroof or the windscreen. Extracting such structures allows to distinguish an individual vehicle from similar vehicles.

An effective method for shape based matching for machine vision applications as proposed in [158] and [159] is employed. It has proven to recognize objects under occlusion, clutter and non-linear illumination changes. The suitability of a shape-based description of the target for airborne vehicle tracking has been demonstrated in [69]. The template is created by extracting edges of the rectangular region in the first image around the initial vehicle position X_0. These edges define direction vectors at characteristic points within the template region. Standard image processing algorithms like the Canny edge detector can be used to obtain the gradient vectors [22]. For finding the template at the candidate position X_t in subsequent images, the same method for computing the direction vectors are applied to the candidate region. A similarity measure $S(X_t, \Gamma_t)$ is defined to compare the template with the candidate region and compute a matching score. The shape template model parameter Γ_t contains all extracted contours and their associated direction vectors. If the direction vectors of template and candidate region align, a high score is obtained.

The shape of the extracted contours of the template depends on the contrast of vehicle structures and the outer boundary of the vehicle. If only few contours are found, the template model is less discriminative. The examples of extracted contours from an airplane image in Figure 4.4b indicate that dark cars hold less contours than bright cars. The decreased GSD of the UAV image in Figure 4.4c reveals detailed structures of the vehicles. The templates of the UAV image exhibit more contours making them more specific for an individual vehicle, Figure 4.4d. This property compromises target recognition under pose and appearance changes of the vehicle. These templates also show that the correct identification of the rectangular reference region is critical. The region should be large enough to capture all characteristic structures and the outer vehicle boundary, but it should not include background or other vehicles in the template.

The matching score between template and candidate ranges between 0 and 1. This similarity score can also be understood as the portion of the target that is not occluded. Because the matching score rapidly drops around the local maximum of a target candidate, only few candidate positions will not be discarded. If the matching score was directly employed for particle weighing, the corresponding likelihood would become extremely peaked, which leads to a fast degradation of the particles, Figure 4.5. To mitigate this problem, the maximum shape matching score is computed in a local search region around the particle. This search region is suggested to have the same dimensions as the template region and includes template angle variations in the range of $[-\pi/8, \pi/8]$ around the particle orientation.

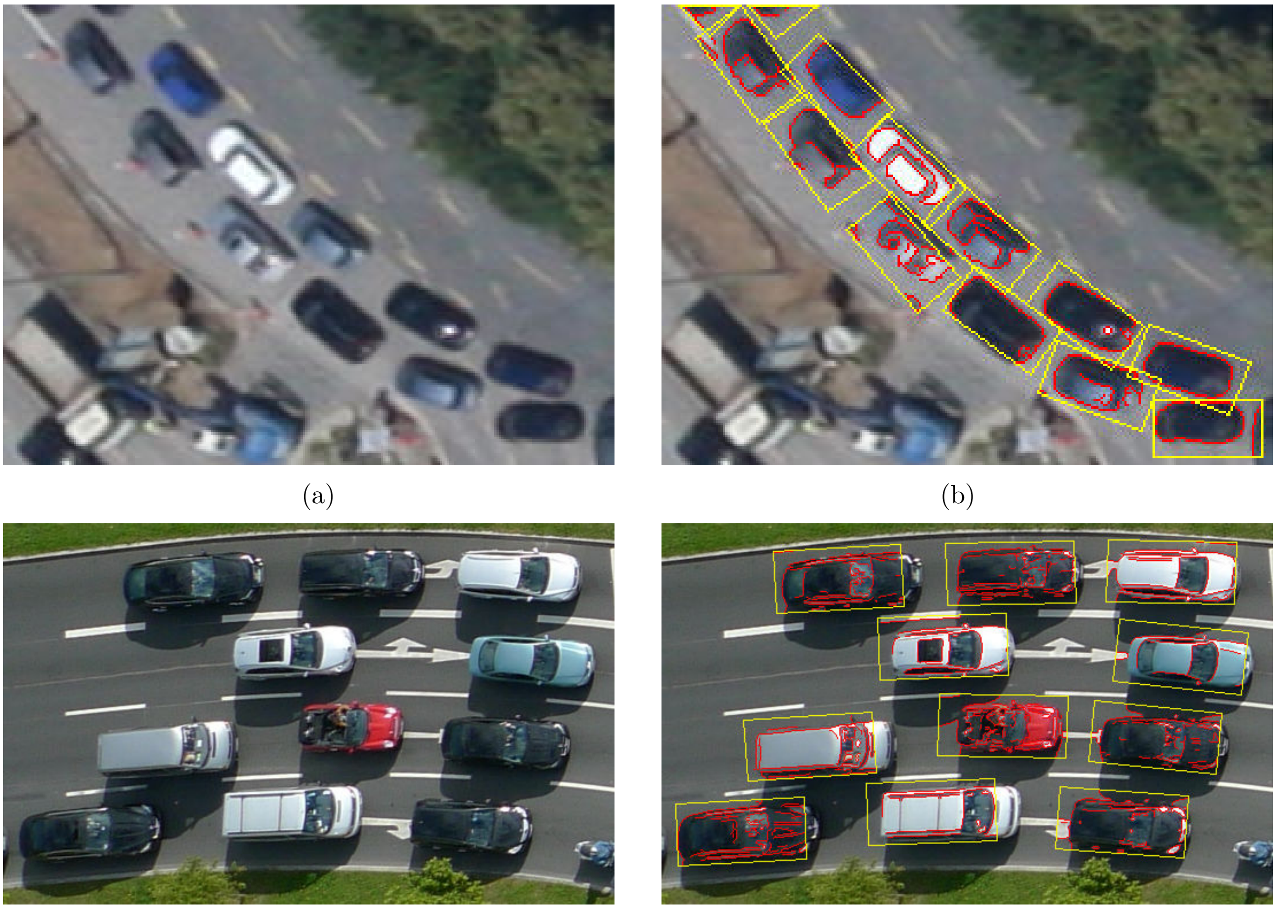
Smoothing of the likelihood is achieved by evaluating the matching score with an exponential function. The measurement function for the shape based particle weights w_s is defined as

$$w_\mathrm{s} = h^\mathrm{s}(X_t, e_t^\mathrm{s}, \Gamma_t) = \exp(-e_t^\mathrm{s}(1 - S(X_t, \Gamma_t))^2), \tag{4.9}$$

where the shape matching score $S(X_t, \Gamma_t)$ is computed at the maximum within the local region around X_t. The higher the value of the noise parameter e_t^s is, the less tolerant the measurement function becomes. A typical range of values that have shown to be effective for the vehicle tracking application and the resulting shape based particle weights are shown in Figure 4.6.

The information about the local best shape based match around a particle is used to correct the particle position. If a valid match is found, the particle position is snapped to this matching position. A similar technique is found in [66] to incorporate measurements at the current time step into the motion model.

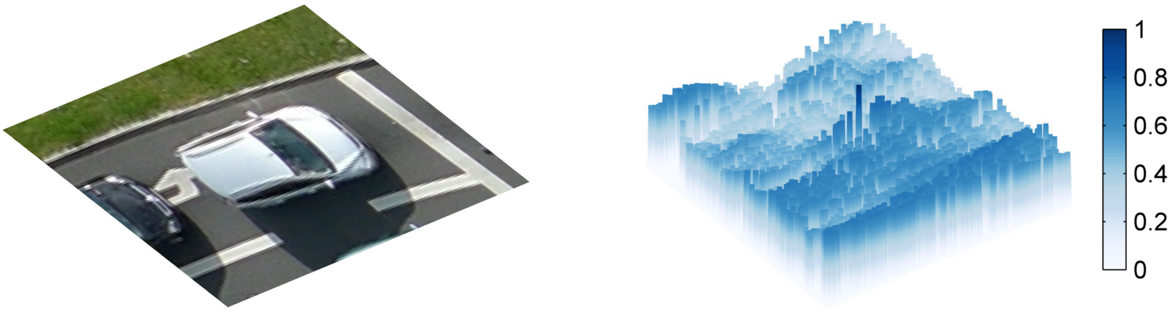
The contribution of the shape based matching weight to the total particle weight is determined experimentally as 50 %, which is modelled by the mixture coefficient $\alpha^\mathrm{s} = 0.5$ in the measurement function (4.8).



(a) (b)

(c) (d)

Fig. 4.4 – Template model generation for shape based matching at initial vehicle positions X_0. Original reference image from (a) airplane, (c) UAV. (b), (d) Extracted contour shapes (red) within template region (yellow). The GSD limits the level of detail of the shape based model.



(a) (b)

Fig. 4.5 – Shape matching scores evaluated at the reference image (a) where the template has been created. (b) The matching scores are computed for each pixel position with the same orientation as the template.

4.3.2.2 Color histogram matching

The vehicle's color is another distinctive feature to recognize the target in subsequent images. Complementary to the shape based matching, the appearance of the vehicle contains information to mitigate ambiguity from similarly shaped vehicles. Color matching can increase the robustness of the tracker, if the shape based matching fails.

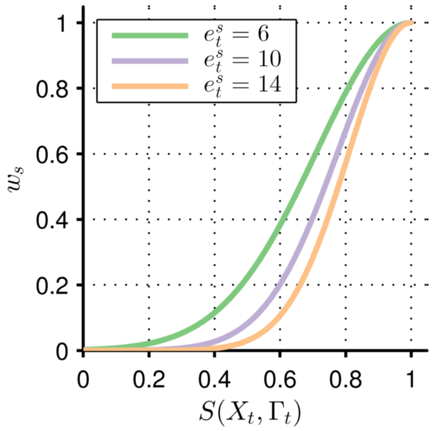


Fig. 4.6 – Shape based matching score measurement function for variation of e_t^{s} from (4.9)

Again, a template based approach is used to determine the color particle weight. The frequency of pixels with a similar color is a suitable feature to describe the color based appearance of the vehicle. The well-known concept of comparing color histograms has been successfully applied in particle filter tracking [132]. The proposed color based matching for vehicle tracking follows this approach. The same template region as for the shape based matching is used to compute a color model of the target. The RGB color space is transformed to the HSV space, because it has demonstrated robustness against illumination changes in computer vision applications. The three dimensions of the HSV color space are discretized in $N_{hsv} = N_h N_s + N_v$ bins, where N_h, N_s, N_v denote the number of bins for the hue, saturation and value channel. The first $N_h N_s$ bins are used to represent color values that have a saturation above 10 % and a value above 20 %. Colors below these thresholds are similar to black or white and are separately discretized in the N_v value bins. The suggested number of bins is $N_h = N_s = N_v = 10$ to encode the color information in a compact and computable form. The bins together with their values represent the color histogram in a defined region. The value of a given bin is defined as the sum of all pixels whose color falls in that bin. The color histogram is normalized such that the bin values sum up to one. Hence, the color histogram fulfills the definition of a probability density function. The reference histogram is computed from the first image in a template region around the initial vehicle position, which encloses only the vehicle itself without background. The template model parameter Ω_t is defined by a vector of dimension N_{hsv} containing the normalized bin values.

A suitable metric to compute the similarity between two color histograms is proposed by the Bhattacharyya coefficient [81]. Originally designed for measuring the distance between two probability density functions $p(x), q(x)$, this measure is defined as

$$\rho = \int \sqrt{p(x)q(x)}dx. \tag{4.10}$$

From the discrete counterpart of this distance, a similarity measure is derived to compare the template Ω_t with the extracted color histogram H at X_t [33]. The color matching score is defined as

$$C(X_t, \Omega_t) = 1 - \sqrt{\left(1 - \sum_{n=1}^{N_{hsv}} \sqrt{\Omega_t(n)H(n)}\right)}. \tag{4.11}$$

Examples of the color matching score for one template compared with three target candidate histograms are shown in Figure 4.7. Since the color histogram is invariant to the spatial ordering of the pixels, the inverse heading of the candidates would lead to the same matching score. Therefore, the

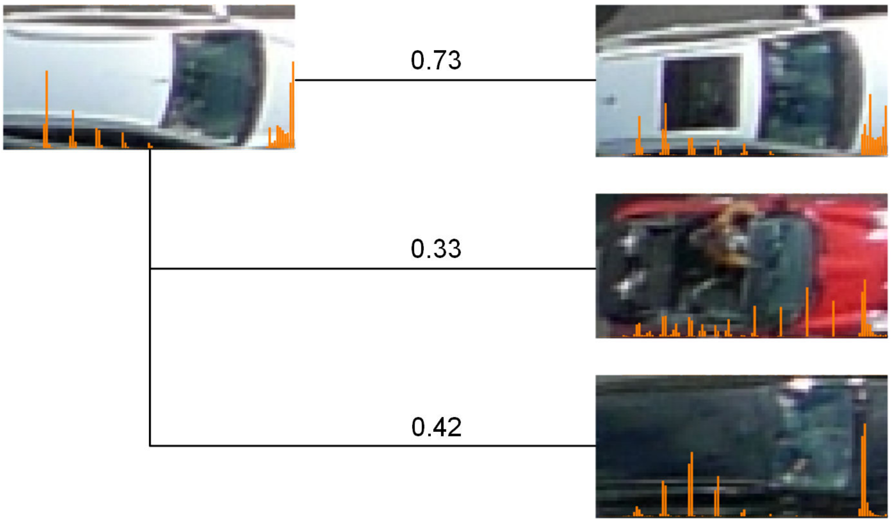


Fig. 4.7 – Color histogram matching scores evaluated for different target candidates. The more similar the colors between template (left) and candidates (right) are, the higher the matching score.

color histogram matching allows only limited evaluation of the correctness of the particle orientation.

The same strategy as for the shape based matching is used to create a smooth likelihood function from the matching score. The measurement function for the color histogram matching that determines the color weights is given as

$$w_{\mathrm{c}} = h^{\mathrm{c}}(X_t, e_t^{\mathrm{c}}, \Omega_t) = \exp(-e_t^{\mathrm{c}}(1 - C(X_t, \Omega_t))^2). \tag{4.12}$$

Since the contribution of the shape based matching weight to the total particle weight is set to 50%, the mixture coefficient of the color measurement function in (4.8) is defined as $\alpha^{\mathrm{c}} = 0.5$.

Only particles with a color matching score of above 0.4 are considered as candidates for the target. When the color matching score is below 0.4, the total particle weight is set to zero regardless of the shape based matching weight. This approach eliminates false positives from the shape based matching.

4.3.3 Resampling

A resampling step is performed to mitigate particle degeneracy. Residual resampling is applied, because it has proven to be superior over multinomial resampling in terms of variance reduction [46]. Since resampling decreases the particle diversity, it is only performed when the efficient sample size (3.12) drops under half of the total number of particles. After resampling, all particles have uniform weights and the particle weighting process in the subsequent time step reduces to the evaluation of the measurement function,

$$w_{t+1} \propto h(X_{t+1}, r_{t+1}). \tag{4.13}$$

4.4 Mode seeking

The output of the particle filter is a discrete approximation of the posterior density of the state. Because of the multi-modality of the measurement function, multiple modes of the posterior exist.

The goal of mode seeking is to estimate the true state from the weighted particles that approximate the posterior.

A weighted mean-shift clustering method is formulated to find the modes by the cluster centers [31]. The advantage of mean-shift clustering is that no prior knowledge about the number of clusters is required. The mean-shift algorithm iterates over all particles and replaces the particle with its local weighted mean. This procedure is repeated until convergence. The weighted sample mean [30] is defined by its kernel K at $\mathrm{x} = (x, y)$ and the weights from the measurement function (4.8),

$$m(\mathrm{x}) = \frac{\sum_{\mathrm{s}\in\mathcal{N}(\mathrm{x})} K(\mathrm{s}-\mathrm{x})w(\mathrm{s})\mathrm{s}}{\sum_{\mathrm{s}\in\mathcal{N}(\mathrm{x})} K(s-\mathrm{x})w(s)}, \tag{4.14}$$

where the local neighborhood $\mathcal{N}(\mathrm{x})$ around x is defined by all particles that are closer to x than the bandwidth parameter d. The Gaussian kernel is given as

$$K(\mathrm{x}) = \exp\left(-0.5\frac{\|\mathrm{x}\|^2}{d^2}\right), \tag{4.15}$$

where $\|.\|$ denotes the Euclidean norm.

Particles that converge to the same pixel position are merged into the same cluster. The cluster weight is determined by the sum of particle weights that belong to this cluster.

The state estimation process includes two instances of the mean-shift clustering algorithm. For the first stage, the bandwidth parameter d is chosen as the generic car length. The second stage proceeds the weighted clusters of the first stage to merge closely spaced clusters. The bandwidth for the second stage is chosen as the generic car width.

The output of the second stage defines hypotheses for the estimated state. In the single vehicle case, the hypothesis with the highest weight determines the final state estimate. For multiple vehicle tracking, the global best hypothesis is assigned to the state estimate. Details of this globally best estimation are given in Section 4.5. The confidence of the hypothesis is defined as its cluster weight.

4.5 Tracking multiple vehicles

For tracking multiple vehicles, two extensions of the method are possible: First, a joint particle filter would estimate a state vector, which consists of all states of all vehicles in the scene. However, this approach is not feasible for a large number of vehicles, because the computational complexity increases exponentially with the number of vehicles. Second, each vehicle is tracked independently by a single particle filter. This approach is more suitable for the given tracking problem, because mutual vehicle occlusion is unlikely to occur, but requires the interrelation between adjacent vehicles to be modeled explicitly. In this work, the latter approach is employed by running multiple instances of the particle filter, one for each vehicle. For finding the globally best assignment of the hypotheses for the position of each vehicle and the estimated position, a maximum weighted matching in a bipartite graph is computed.

It can be assumed that a position in the image cannot be occupied by more than one vehicle at a time, because mutual occlusion of vehicles does not occur when they travel on the same road level. Further, the modes of the posterior density are most likely to correspond to vehicles than to clutter. The mode seeking method from Section 4.4 computes multiple weighted hypotheses for each vehicle. Now the task of data association is to decide which hypothesis belongs to which vehicle.

If similar vehicles are close, the modes of true posterior would overlap around the true vehicle position. This would cause the particles to mutually capture the other vehicle position and particles on the wrong vehicle obtain relatively high weights. If the motion models and the observation models fit well, the clustered hypotheses of each vehicle will be located close to each other, Figure 4.8. A consistent assignment of hypotheses is possible, if each vehicle is assigned its best hypothesis and the sum of assigned hypotheses weights cannot be increased by any different assignment, Figure 4.8a. The hypotheses are conflicting, if the "greedy" assignment of the best hypothesis leads to close hypotheses that are assigned to different vehicles, Figure 4.8b. Therefore, close hypotheses are merged to resolve conflicting situations by the globally best assignment. Possible assignments are defined by edges between the vehicle and the hypotheses. The edge weights are determined from the weight of the hypotheses that is assigned to the vehicle, Figures 4.8d, 4.8e. The optimal association is defined by the set of edges that maximize the sum of edge weights.

In the general case of multiple vehicles, the assignment between hypotheses and vehicles can be formalized as a matching problem in a bipartite graph. One set of nodes denotes the vehicles, the other disjoint set denotes the hypotheses. The optimal assignment is then given by the maximum weighted matching. The solution of this matching problem is found with the Hungarian algorithm [98].

The resulting hypotheses define the final output of the multiple vehicle particle filter tracker. Since the hypotheses association step runs separated from the actual particle filter procedure, the information of the assigned hypotheses is not directly fed back into the particle filter. However, the particle filter takes advantage of this knowledge, because a template update step is proposed in Section 4.6.2, which takes the estimated state as input parameter. This strategy mitigates the risk of hijacking of the particle cloud to another similar vehicle, because the updated template biases the observation model in subsequent time steps to the best hypothesis.

4.6 Robust techniques

The presented models for particle sampling and weighting have demonstrated to produce stable results under certain conditions. However, if abrupt motion of the vehicles occurs or their appearance changes, target losses reduce the quality of the tracking result. To address these typical challenges in airborne vehicle tracking, advanced strategies for particle sampling and weighting are introduced.

4.6.1 Adaptive models

The ability to handle variation of motion and appearance are desired properties of the motion and observation models. Thus, these models should be generic enough to ensure robustness of the tracker. On the other hand, these models need to be specific for the tracked vehicle to avoid confusion of the tracker with close similar vehicles. To meet these contradicting requirements, a careful design of the model parameters is necessary. However, the chosen parameters will always be a compromise between robustness and precision of the tracker. In this work, adaptive models are presented to satisfy both conditions at the same time.

The discriminative power of the models is defined by the noise parameters of their associated functions (4.4), (4.9) and (4.12). The proposed strategy aims to learn these noise parameters online during particle filtering at the current time step. The learning process incorporates measurements at the current time step without additional computational effort for evaluation of the measurement function. The implementation of the learning steps requires reordering of the particle filter process. The particle sampling and weighting stages are merged into a single stage, where each particle is directly weighted

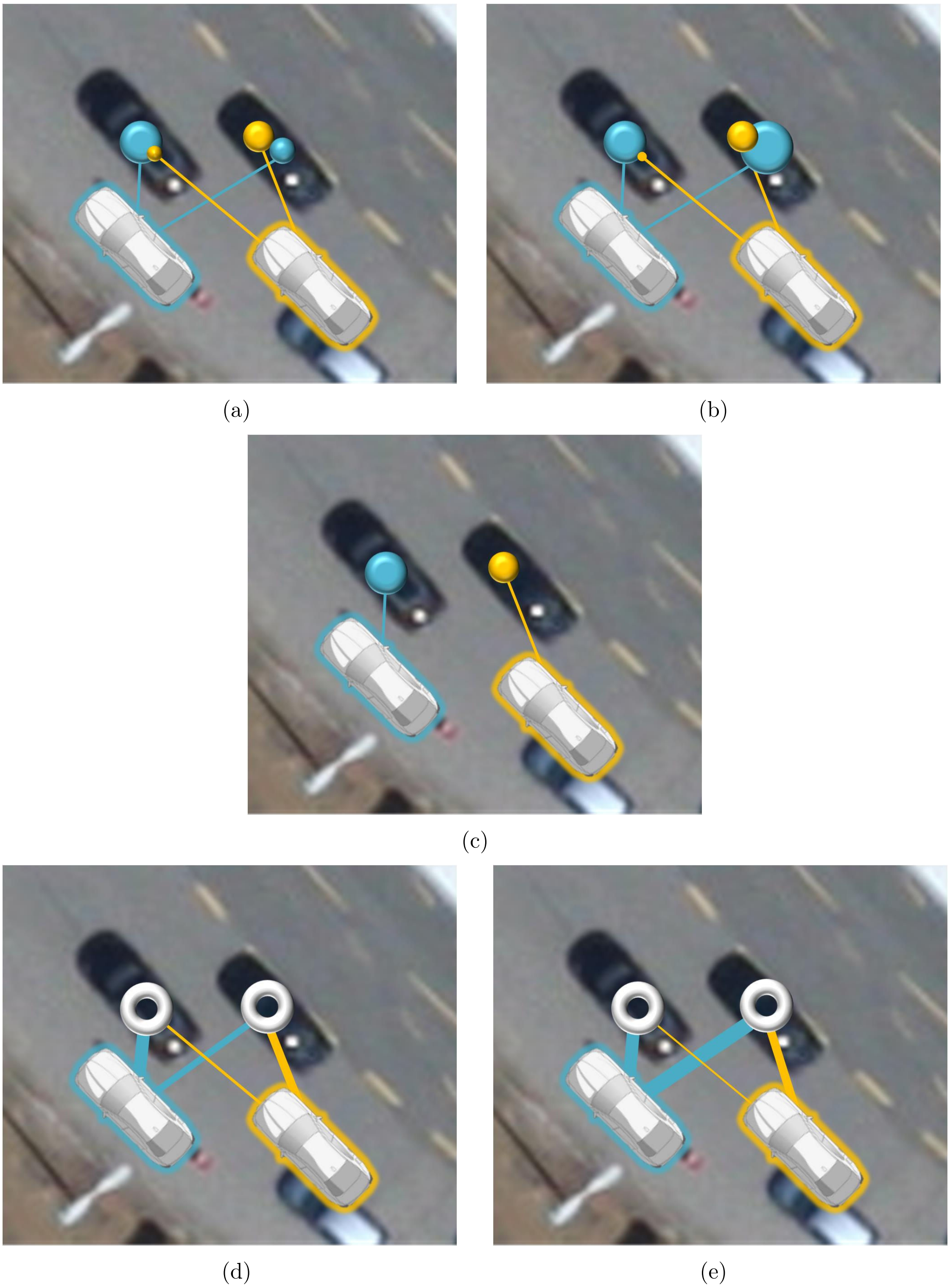

(a)
(b)
(c)
(d)
(e)

Fig. 4.8 – Two possible configurations of weighted hypotheses for two similar vehicles after mode seeking. (a) The hypotheses are not conflicting, because each vehicle is assigned the hypothesis with the highest weight. (b) The hypotheses are conflicting, because the assignment of the hypothesis with the highest weight of the blue vehicle rules out the hypothesis with the highest weight of the yellow vehicle. (c) The optimal assignment in terms of total weights maximization is the same for both configurations. (d), (e) To construct a bipartite graph, close hypotheses are merged and the hypotheses weights are transformed into edge weights. The line thickness corresponds to the hypothesis weight. The maximum weighted matching in this graph defines the optimal hypotheses association.

after it has been sampled. This procedure allows to take into account recent measurements for particle sampling.

4.6.1.1 Adaptive particle sampling

The noise parameter o_t of the motion model controls the spreading of the particles. Small values of the standard deviations σ_{lon},σ_{lat} result in precise estimates, whereas large values ensure robustness against abrupt motion. The adjustment of the noise parameter starts with a small value which increases during iterative particle sampling. The rate of increase is computed recursively and takes into account the unnormalized weight of the previously sampled particle. The smaller the particle weight, the larger the noise value. This strategy ensures that the particles explore a larger search space only if the situation requires. It leads to an efficient exploration of the search space with the given number of particles.

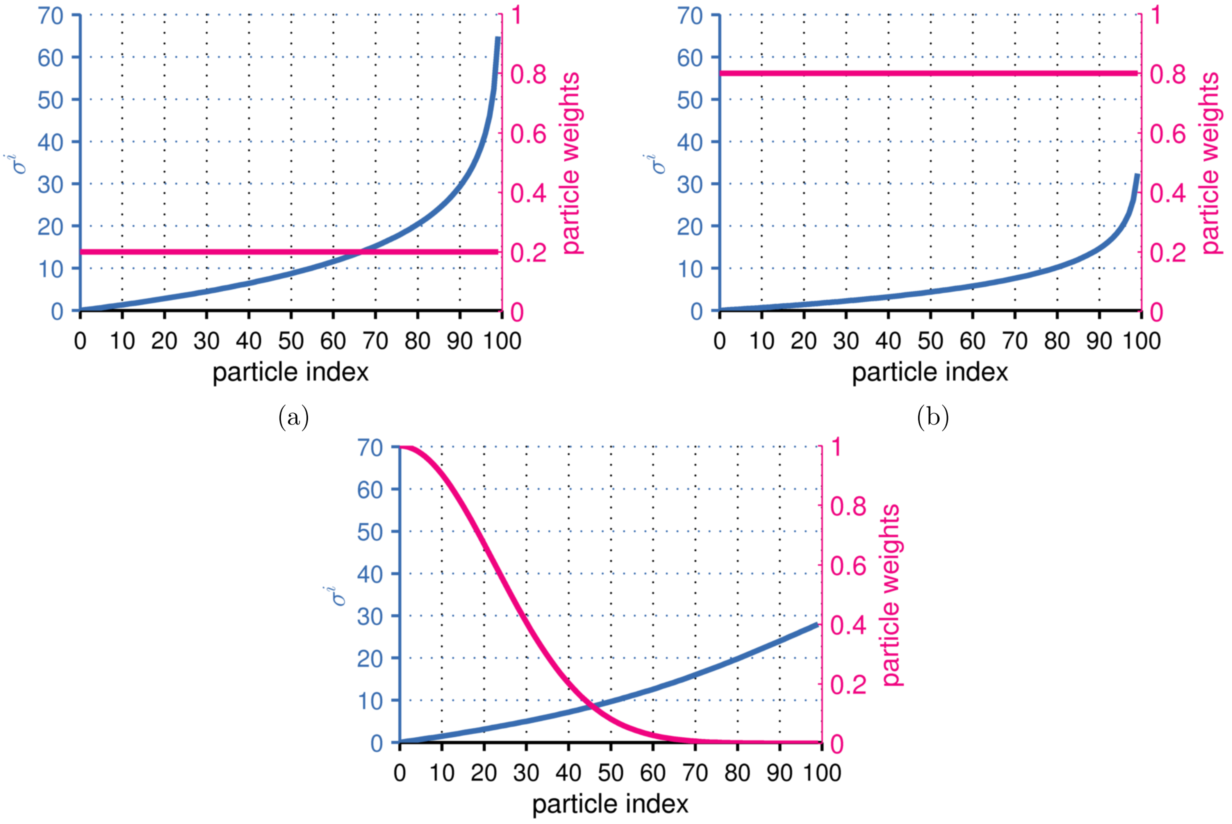
This adaptive particle filter method can be formalized as a variant of adaptive importance sampling (AIS) [126]. The importance sampling function (4.4) represents a family of probability distributions parametrized by σ_{lon}, σ_{lat}. The parameters are updated recursively after each stage, which consists of sampling exactly one particle. Without loss of generality, it is assumed that the particle weights range between 0 and 1 without normalization. The update scheme is generally described for one dimension of the noise, because the principle for updating σ_{lon} and σ_{lat} is the same.

Starting with zero noise, the noise increases monotonically. The amount by which the noise increases depends on the current particle index i, the weight w^{i-1} of the previously sampled particle and the previous noise standard deviation σ^{i-1},

$$\sigma^i = \sigma^{i-1} + \frac{\Psi}{N-i}\sqrt{1-w^{i-1}},\ i = 1, \dots, N-1, \tag{4.16}$$

where $\Psi > 0$ denotes a constant increase rate parameter. The value of Ψ is determined experimentally as $\Psi = 14$. The factor $\frac{1}{N-i}$ ensures that σ^i increases slowly in the beginning of the particle sampling to concentrate the first sampled particles around the noise-free prediction position. As long as the particles obtain a high weight, the noise increases moderately, because it suggests that the vehicle has been captured correctly. If the particle weights are low, the noise increases strongly to allow the particles to spread out widely across the feasible search space and capture the vehicle. In this way, the noise adapts to the quality of the sampled particles, which leads to a more efficiently explored search space. Other than AIS, the proposed adaptive scheme always updates the full particle set without using a stopping criterion. The particles with the highest weights from the previous time step are more likely to be propagated to the correct vehicle position by the noise-free model. Therefore, the particle set is proceeded in descending order according to their weight (before the resampling process) in the previous time step.

The evolution of the noise parameter for different sets of synthetic particle weights is presented in Figure 4.9. In all three examples, the noise increases monotonically. The increase rate differs depending on the distribution of particle weights. The differences can be observed when the noise value is compared at half of the particle set size. The particle set with low particle weights in Figure 4.9a is spread with $\sigma = 10$, whereas the particle set with high particle weights in Figure 4.9b is spread with $\sigma = 5$ at half of the particles. The weights of the particle set in Figure 4.9c follow the measurement function from (4.8). When the noise value for this particle distribution is compared to Figure 4.9a, it shows that the noise value at half of the particle set size is almost the same. The influence of the particle index to the noise update function causes both noise curves to have a similar increase rate in the first half of the particle sampling process.



(a)

(b)

(c)

Fig. 4.9 – Evolution of the noise parameter together with examples of synthetic unnormalized particle weights in descending order with $\Psi = 14$. (a) Constant low particle weights lead to a moderate increase of the noise in the beginning of particle sampling. The noise increase rate grows rapidly towards the end of the sampling process. At half of the sampled particles, the noise is $\sigma^i = 10$. (b) Constant high particle weights lead to slowly increasing noise in the beginning. At half of the sampled particles, the noise has a value of $\sigma^i = 5$. (c) Exponential distribution of the particle weights produces a moderate increase of the noise in the beginning. At half of the sampled particles, the noise has a similar value as in (a), but the noise increase rate grows only slightly towards the end of the sampling process.

The adaptive sampling strategy increases the ranges for sampling the lateral and longitudinal accelerations. Because of the same considerations as in Section 4.3.1, the ranges of the longitudinal and the lateral accelerations a_{lon} and a_{lat} of a vehicle depend on each other.

This physical constraint from (4.6) is considered when sampling the noise vector $o_t^{\text{lon,lat}}$ by limiting the noise σ_{lat} for sampling the lateral acceleration after σ_{lon} has been computed. The same constraint as in (4.6) is applied after replacing a_{lon} with σ_{lon} and a_{lat} with σ_{lat}.

The application of the noise update strategy to vehicle tracking is demonstrated in Figure 4.10. The sampled particles spread out more widely than in the non-adaptive case when the particles obtain low weights, Figures 4.10a, 4.10b. This strategy increases the chance to capture the vehicle, because the particles explore a larger search space. In contrast, when the particles obtain high weights, the noise increases slower and the density of the sampled particles is comparable to the non-adaptive case, Figures 4.10c, 4.10d. The evolution of σ_{lon} in Figure 4.10e shows the different noise increases for the cases when the particles obtain high or low weights compared to the constant noise value for non-adaptive sampling. The higher the particle weights, the larger the portion of particles that are sampled with a noise value below the constant noise. If the noise of the non-adaptive sampling was increased to capture the vehicle in Figure 4.10a, the constant noise would lead to more widely spread particles in Figure 4.10c, which increases the risk of mismatches. However, for the adaptive sampling strategy the tracker is less likely to be confused with false positive alarms, because only a small amount of particles has a larger spreading as for sampling with constant noise.

4.6.1.2 Adaptive particle weighting

A similar strategy as for the motion model noise update is used to adapt the noise parameters of the likelihood function. The weight of a particle is determined by evaluating the likelihood function (4.8), which depends on the noise parameters e_t^{s} and e_t^{c} for shape and color based matchings. Since the noise update method is the same for both components of the observation model, the formalization is given only for the shaped based component, the color based component follows analogously.

For large e_t^{s}, the likelihood function appears peaked and the observation model will produce significant values for the particle weights only if the template fits very well. Thus, the observation model can be regarded as a specific one. If the candidate matches the template well, then the vehicle can be discriminated easily from other non-vehicle locations. This property is especially useful when dealing with clutter. Also, the risk of false positive detections is quite low. If e_t^{s} is small, the observation model becomes more generic. In this case, matches with a poor quality will still receive moderate weights. The strength of such an observation model lies in its ability to deal with temporal changes of the vehicle's appearance. This generic characteristic of the observation model ensures robustness against illumination changes or pose variations of the vehicle.

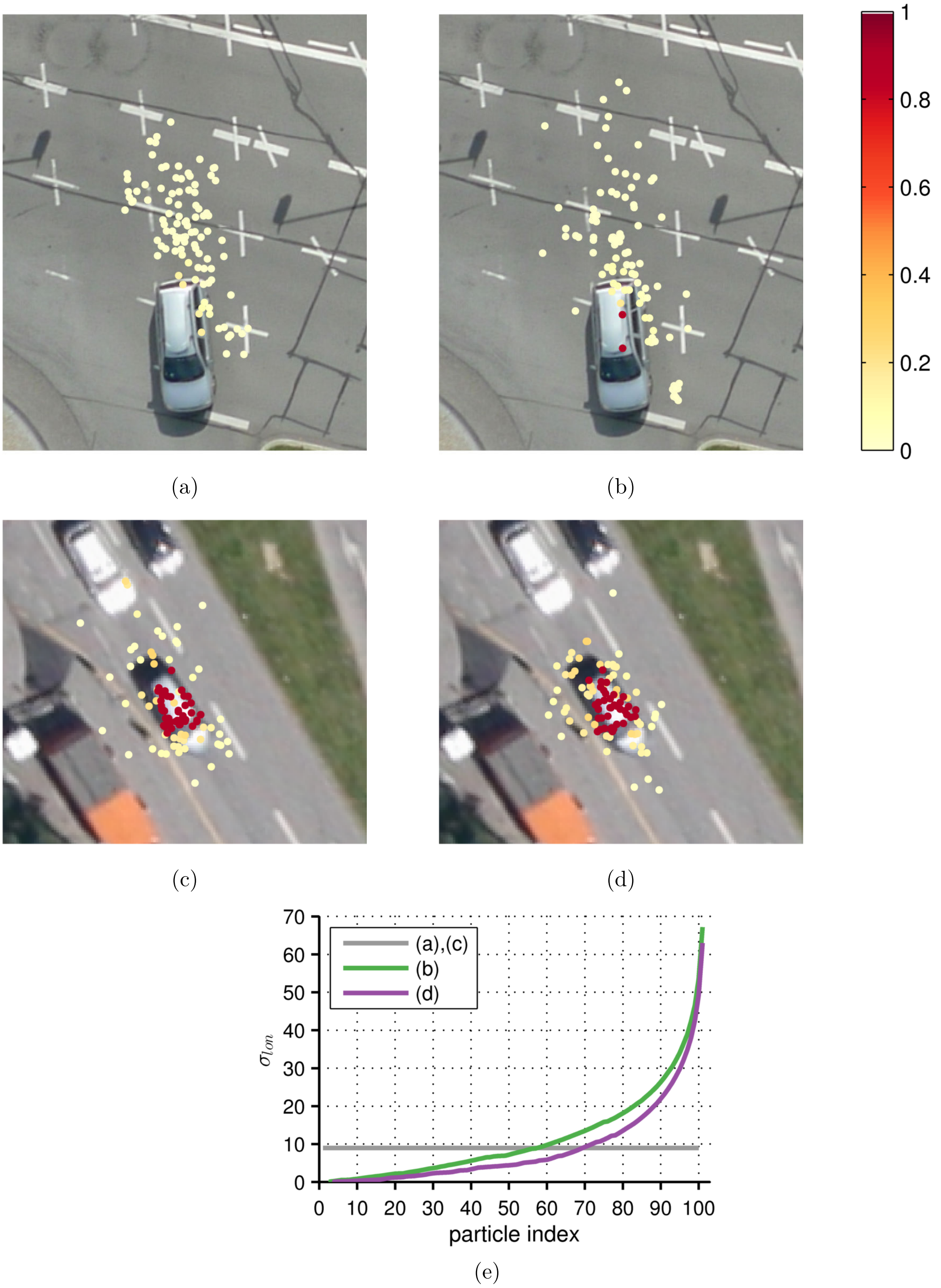
The proposed update strategy combines both properties of the observation model, the discriminative power and the robustness against temporal appearance changes. Starting with a low noise of the observation model, the noise increases and successively adapts to the quality of the sampled particles. This strategy is formulated as an update rule for the noise parameter. For better readability, a short notation for the noise parameter is introduced, $e = e_t^{\mathrm{s}}$ and the time index is neglected. At each iteration step i in the particle weighting, the noise parameter is updated with

$$\begin{aligned} e^i &= e^{i-1} - \frac{\Lambda}{i}(1 - S(X,\Gamma))^2, \; i = 1, \ldots, N-1 \\ e^0 &= e_{\max}, \end{aligned} \tag{4.17}$$

where $\Lambda > 0$ denotes a constant decrease rate parameter. Depending on the matching score $S(X,\Gamma)$ and the current particle index i, the noise parameter e decreases monotonically. The starting value $e_{\max}$ is chosen to define a specific model. Suggested values for the constant parameters are given by $\Lambda = 4$ and $e_{\max} = 12$.

The evolution of the noise parameter for different sets of synthetic matching scores is presented in Figure 4.11. In all three examples, the noise decreases monotonically. The decrease rate differs depending on the distribution of matching scores. Figure 4.11a shows that low matching scores lead to a fast drop of e, which corresponds to an increase of the model noise. In contrast, Figure 4.11b indicates that good matchings with high scores cause e to remain on a high level. A linearly decreasing distribution of matching scores between 0 and 1 produces slowly decreasing values of e, which is visible in Figure 4.11c.

The advantage of the adaptive likelihood function is demonstrated in Figure 4.12. When the appearance of the vehicle changes, the adaptive likelihood is more robust than the constant noise model, Figure 4.12a, Figure 4.12b. In the adaptive case, the observation model noise increases because of the low template matching scores. As a consequence, the particle weights increase already for moderate matching scores, which awards more discriminative power from poor matches to these particles. Yet the adaptive likelihood preserves the precision of the particle weights in case of an unchanged



(a)

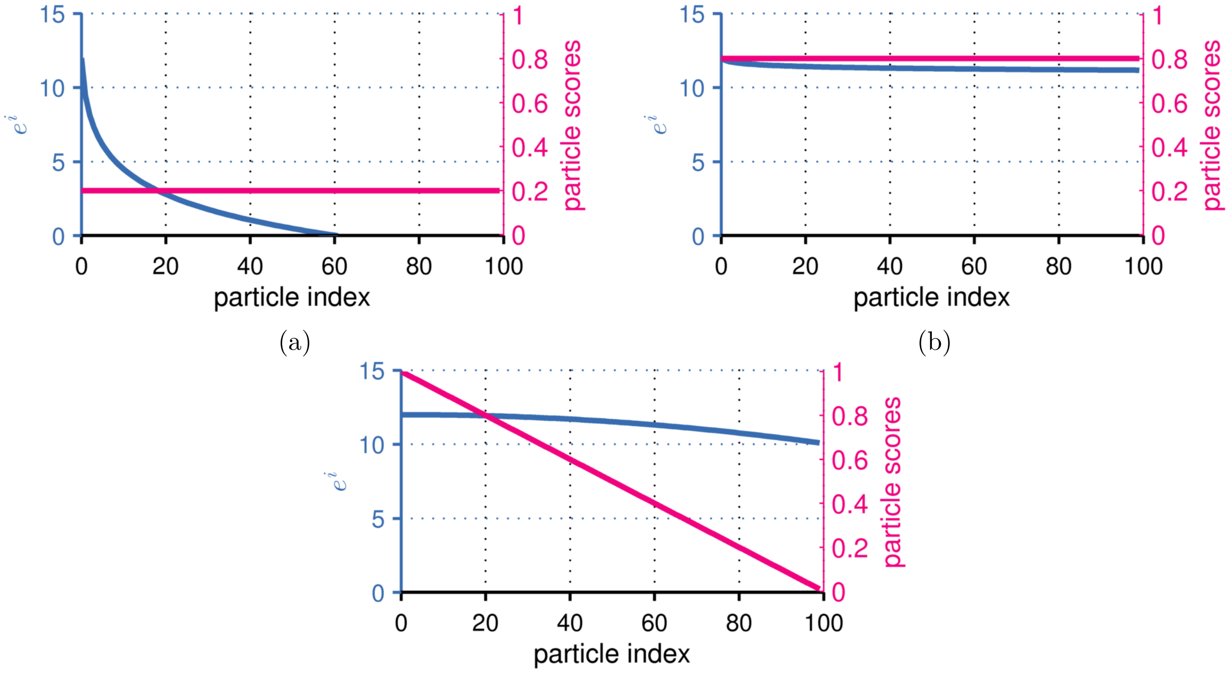
(b)

(c)

(d)

(e)

Fig. 4.10 – Unnormalized weighted particles for the non-adaptive and adaptive sampling strategy. (a) The motion model does not match well the true target motion, which leads to misplaced particles for the non-adaptive sampling strategy. The constant noise value is set to $o_t = (9, 5)$. (b) The same particles from the previous time step are sampled adaptively. Some particles capture the true vehicle position, because the noise increases, which spreads the particle cloud widely. (c) The motion model matches well the true target motion, which results in high particle weights for with the non-adaptive sampling around the true vehicle position. (d) The adaptive sampling strategy produces a very similar particle spreading compared to the non-adaptive sampling, because the noise increase is slower. (e) Comparison of noise parameter σ_{lon} for the above examples. For (b), 58% of the particles are sampled with a noise value smaller than in the non-adaptive case. For (d), this rate is 70%.



(a)

(b)

(c)

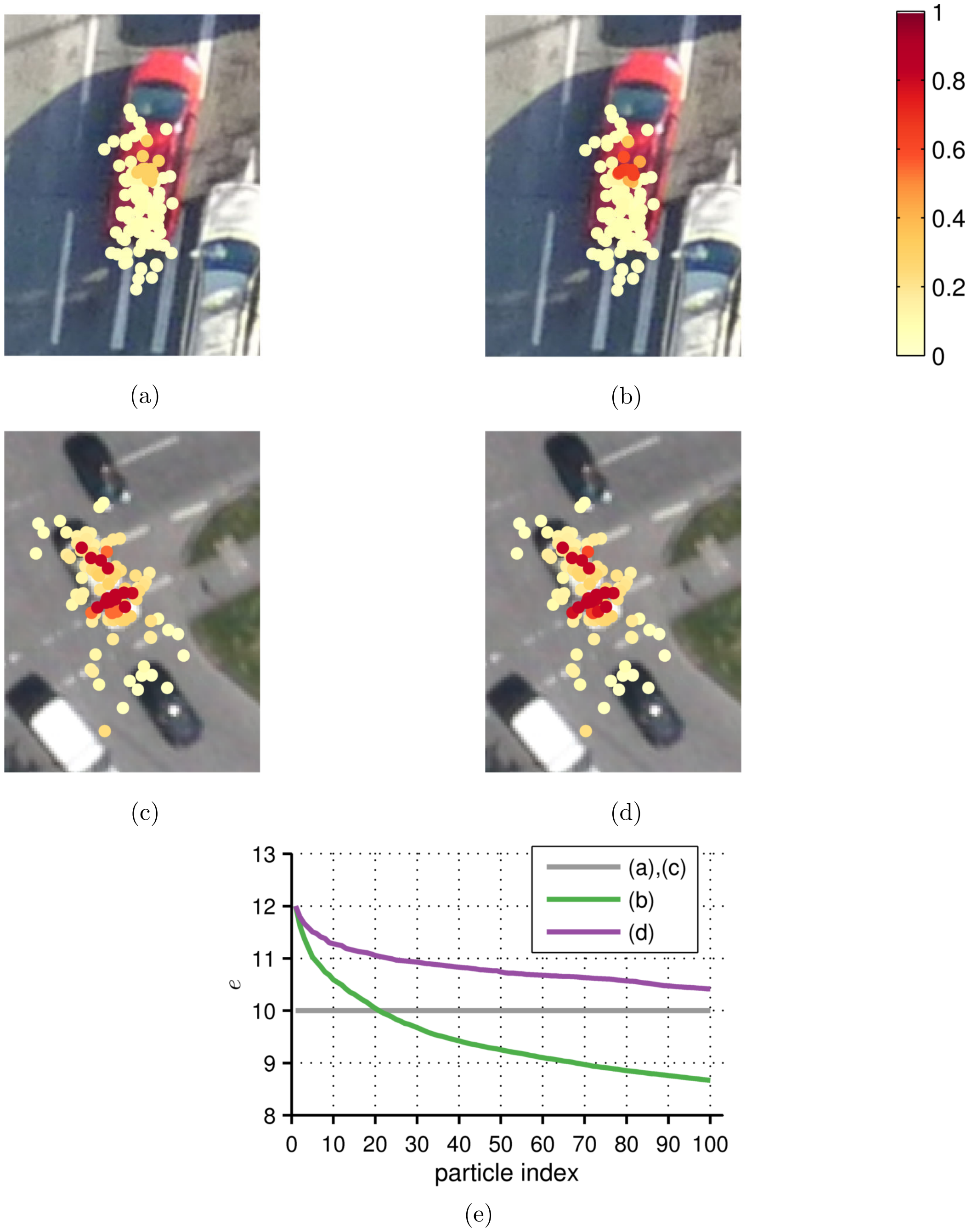
Fig. 4.11 – Evolution of the noise parameter together with examples of synthetic matching scores in descending order with $\Lambda = 4$ and $e_{\max} = 12$. (a) Constant low matching scores lead to a fast decrease of the noise parameter. At half of the sampled particles, the noise as low value of 0.5. (b) Constant high matching scores lead to an almost constant value of the noise parameter. At half of the sampled particles, this value is 11.3. (c) Linearly decreasing matching scores lead to a moderate decrease of the noise. The noise value at half of the sampled particles is 11.5, which is very similar to the noise in case of constant high matching scores.

appearance of the vehicle, which matches the template well, Figures 4.12c, 4.12d. The evolution of e in Figure 4.12e shows the different noise parameter decreases for the cases when the particles obtain high or low weights compared to the constant noise value for non-adaptive sampling. The higher the particle weights, the larger the portion of particles that are sampled with a noise parameter value above the constant noise.

4.6.2 Target template update

The particle weights are computed from the color and shape based matching, which use a pre-generated template model from the first image of the sequence. If the appearance of the vehicles changes, the matching scores will degenerate, because the template does not match well with the true vehicle position. Because of the low frame-rate in the given application and the camera movement, illumination changes and shadows cause abrupt appearance changes. Thus, the risk of mismatches or target losses increases with time. To mitigate this problem, a template update method is proposed. Shape and color information are combined to update the template in a way that it adapts to appearance changes, but avoids the template drift problem. Template drift occurs when the location of the new template contains errors, which are accumulated over time and cause the template to drift away from the true target position [117].

The goal of the proposed template update is to create new templates from the current image that represent the true vehicle appearance best. From the set of weighed particles, the estimate $\tilde{X}_t$ of the most likely vehicle position and orientation is computed by the mode seeking method from Section 4.4. The estimated vehicle position defines the reference point for a template, which is generated around



(a) (b) (c) (d) (e)

Fig. 4.12 – Unnormalized weighted particles for the non-adaptive and adaptive weighting strategy. (a) The template does not match well to the particles, which obtain small weights with the non-adaptive weighting strategy. The constant noise value is set to $e = 10$. (b) The same particles are weighted adaptively, which produces significantly higher weights around the true vehicle position. (c) The template matching generates high scores, which results in high particle weights with the non-adaptive weighting around the true vehicle position. (d) The adaptive weighting strategy produces very similar weights compared to the non-adaptive weighting, because the noise decrease is slower. (e) Comparison of noise parameter e for the above examples. For (b), 79%=100-21 of the particles are sampled with a noise value smaller than in the non-adaptive case. For (d), the noise does not drop below the constant noise level.

this position. This template acts as a candidate to replace the old template. To avoid assimilation of the template to a mismatched vehicle, the new template should not be too different from the old one. The candidate template is matched against the old template to obtain the correlated inter-template matching score $S(\tilde{X}_t, \Gamma_t)$, which measures the similarity of both templates. The same procedure is performed to compute the color based matching score $C(\tilde{X}_t, \Omega_t)$. The candidate template is accepted as the new template, if either of the inter-template matching score is above a threshold for a very good match and the other score is above another smaller threshold that corresponds to a still tolerable match. These two thresholds are denoted by $T1$ and $T2$ and their suggested values are determined experimentally as $T1 = 0.8$, $T2 = 0.6$. Considering both the shape based matching and the color histogram matching provides a robust template while avoiding the template drift problem. If only the color histogram matching is considered, the rotational invariance of the color histogram matching would cause drifts of the template regarding the heading of the vehicle. Thus, the shape based matching anchors the model to the contours of the vehicle. On the other side, the shape based matching alone would not have the discriminative power to avoid false alarms from similarly shaped vehicles. The following criteria are checked to decide if the candidate template replaces the old template:

$$S(\tilde{X}_t, \Gamma_t) > T1 \wedge C(\tilde{X}_t, \Omega_t) > T2, \tag{4.18}$$

$$S(\tilde{X}_t, \Gamma_t) > T2 \wedge C(\tilde{X}_t, \Omega_t) > T1. \tag{4.19}$$

If either (4.18) or (4.19) applies, then the template is updated with the candidate template.

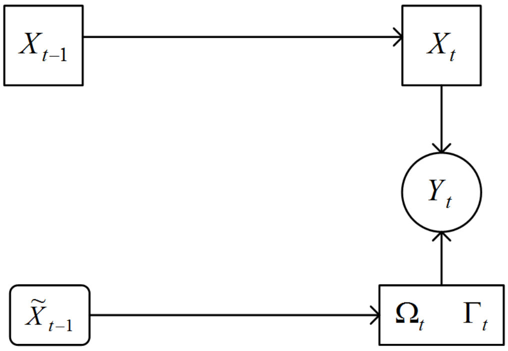


Fig. 4.13 – Schematic illustration of the template update strategy in the Bayesian network. The state estimate $\tilde{X}_t$ from the previous time step defines the candidate template region for the template update method.

The proposed template update method assumes that a reliable and precise estimate of the state is found. If the template region around the estimated position contains a large portion of background, the risk of template drifts occurs.

The template update method is limited to situations with moderate appearance changes. In situations with strong or abrupt appearance changes, the similarity between the candidate template and the old template is too large to fulfill the template acceptance criteria. In situations where only temporal occlusions occur, this property is desired, because the tracker can recover quickly once the vehicle is visible again under its original appearance. However, adoption to abruptly occurring and persistent illumination or perspective changes are not addressed by this approach.

4.6.3 Online map learning

The accurate and precise prediction of the true vehicle motion is particularly important for this tracking application, because it is likely that identical vehicles are traveling nearby and can easily be confused by the observation model. The trajectories of preceding or close vehicles around the current location of a vehicle contain information about the roadway, which can be fused directly with the prediction from the motion model. Under the assumption, that the following vehicle takes the same path as its predecessor or close neighbor, the particles can be guided by predecessing or neighboring vehicles. This procedure can be considered as learning a local map by extracting map features from

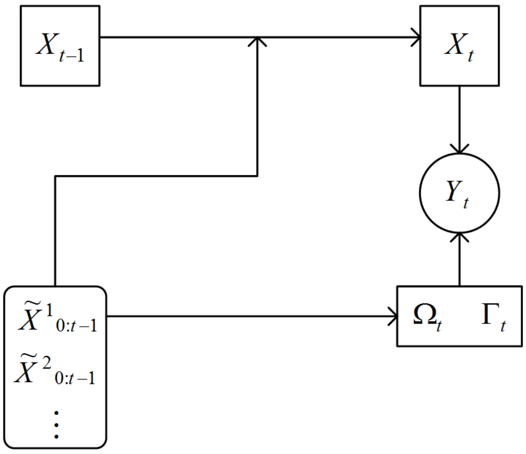


Fig. 4.14 – Schematic illustration of the online map learning strategy in the Bayesian network. The state estimates $\tilde{X}^1_{0:t}, \tilde{X}^2_{0:t}, \ldots$ from all vehicles at previous time steps define trajectories, which are fused with the motion model of the current vehicle.

already computed trajectories. This map is learned online during tracking, because it does not require other prior knowledge. The integration of the learning approach into the dynamic model of the tracker is shown in Figure 4.14.

The learned map contains all trajectories of the so far tracked vehicles. Before the sampling stage, a fraction of particles is assigned the direction of the vehicle that was closest to the current position of the vehicle at any point of time. The fraction of particles whose direction will be replaced is determined by the confidence of the last state estimate $\tilde{X}_{t-1}$ of the current vehicle and the confidence of the state estimate of the vehicle, which was closest. This means that the share of particles increases with the loss of reliability of the last state estimate and the gain of reliability of the closest vehicle. The confidences are denoted by q^j for the closest vehicle j and q_{t-1} for the current vehicle. The confidence of an estimate is defined by the weight of the corresponding cluster from the mean shift clustering algorithm from Section 4.4. The fraction P of N particles that is replaced with the learned direction is computed as

$$P = \min(0.8, q^j(N - q_{t-1}))N. \tag{4.20}$$

The direction replacement is performed for a random set of P particles just before the new sampling process of the next time step starts.

This particle following method may mislead particles at junctions or bifurcations, if the succeeding vehicle takes another route than its predecessor. However, the fraction P rarely reaches the total number of particles, which ensures the robustness of the tracker.

4.7 Results

The proposed tracking method is tested on airborne image sequences taken at different urban scenes in Munich. These images are characterized by dense traffic and large-scale coverage of the scene. Table 4.2 gives an overview of the scenes that are used for performance measurements of the tracker. The selected test scenes include discontinuous vehicle motion, illumination variations and temporal occlusions of the vehicles. All scenes are characterized by vehicles driving on multi-lane roads, which challenges the tracker to discriminate similar vehicles.

The experimental setup contains a manual detection of the vehicle positions and orientations in the first image to prevent misinterpretation of the tracking performance evaluation from incorrect initialization of the target templates. The vehicle velocity has been roughly guessed from the vehicle density

flight system	scene name	number of frames	number of detected vehicles
3K	3K-A	26	13
	3K-B	16	30
	3K-C	10	22
	3K-D	20	151
3K+	3K+A	25	14
	3K+B	19	12
	3K+C	46	89
UAV	UAV-A	71	11
	UAV-B	103	21

Tab. 4.2 – Overview of scenes for tracking performance evaluation. The flight system properties are described in Table 4.1.

and the traffic situation. Since the UAV camera system has not been calibrated, the GSD of the images is estimated manually for each dataset. Also the time difference between consecutive images of the UAV system was not measured and has been determined manually assuming a constant frame-rate of 2 Hz. For the 3K/3K+ system, the frame-rate is known from measurements of the camera trigger.

The implementation of the shape based matching is provided in the HALCON 11.0.3 machine vision software library [123]. To avoid mismatches when the vehicle is partly occluded, shape matching scores below 0.6 and color matching scores below 0.4 are set to zero.

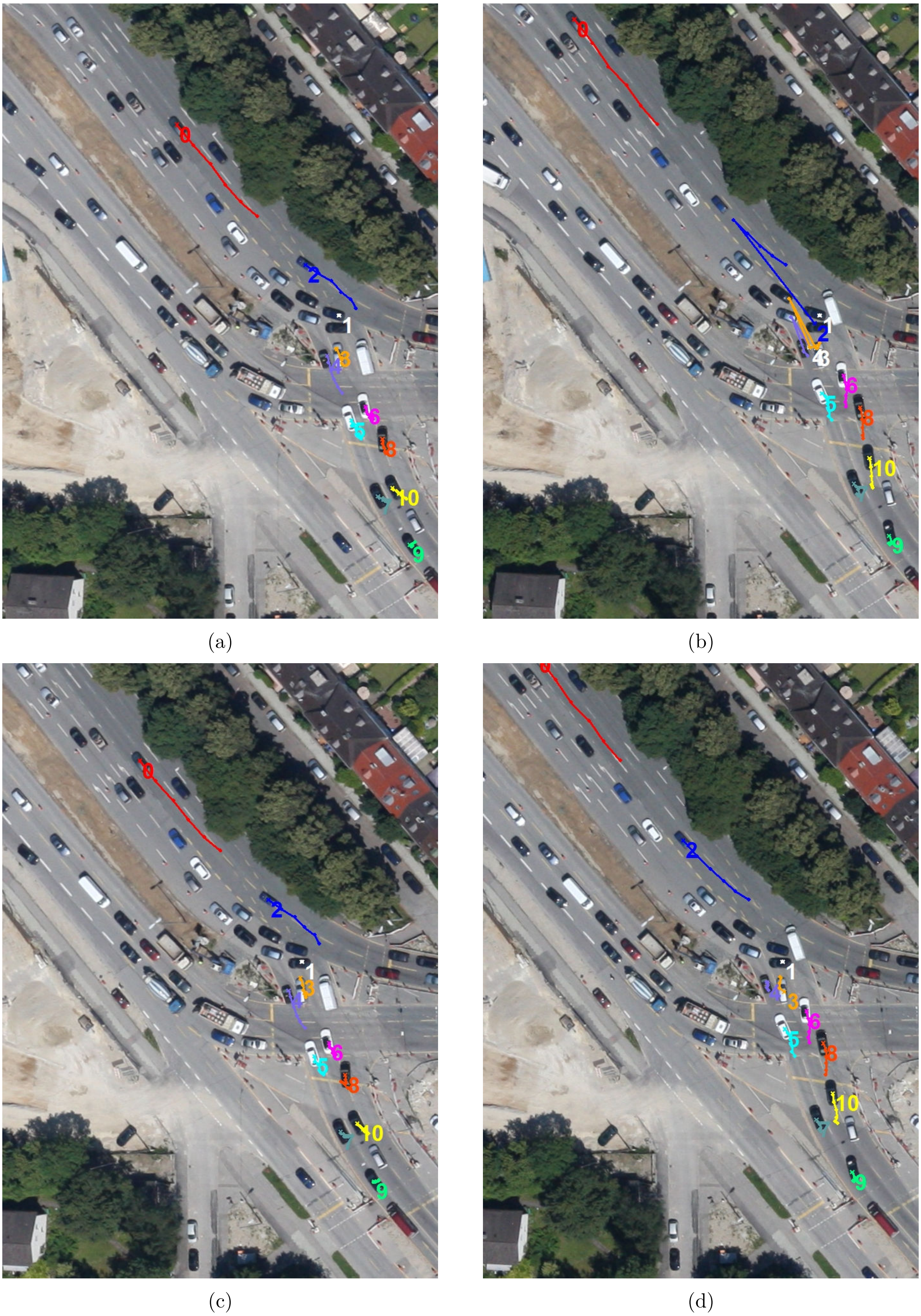
4.7.1 Tracking robustness

In this section, the effectiveness of the robust techniques introduced in Section 4.6 is demonstrated. The evaluation is performed by comparing the tracking results of the conventional particle filter as described in Section 4.3 with the particle filter where one of the robust techniques is applied.

First, the impact of the adaptive sampling strategy from Section 4.6.1.1 is presented. A small group of vehicles is tracked in the *3K+C* scene. The motion of these vehicles is difficult to predict, because they accelerate fast after passing the intersection. Figure 4.15 shows the results as trajectories of the tracked vehicles with the non-adaptive and the adaptive sampling strategy. The estimated vehicle positions of the last 6 time steps are visualized. The particle filter with the non-adaptive sampling looses vehicle 2 at frame 22, whereas the adaptive sampling strategy tracks this vehicle correctly throughout the rest of the scene.

Second, the adaptive weighting strategy from Section 4.6.1.2 is evaluated with a group of vehicles from the *UAV-B* scene. Figure 4.16 shows the trajectories of these vehicles with non-adaptive and adaptive particle weighting. In the non-adaptive case, two vehicles are lost early during tracking, after frame 4 of the sequence, Figure 4.16b. By using the adaptive weighting method the appearance changes are tolerated by the observation model and also vehicles that enter a shadow or are partly occluded are tracked successfully, Figure 4.16d.

Third, tracking results with the target template update method from section 4.6.2 are presented. The selected vehicles from the *3K-D* scene drive along a curved road while their appearance changes due to the perspective change and their own shadows, Figure 4.17. Without the template update method, two vehicles of this group are lost after the sharp turn, Figure 4.17b. With the template update method, all vehicles are tracked successfully.



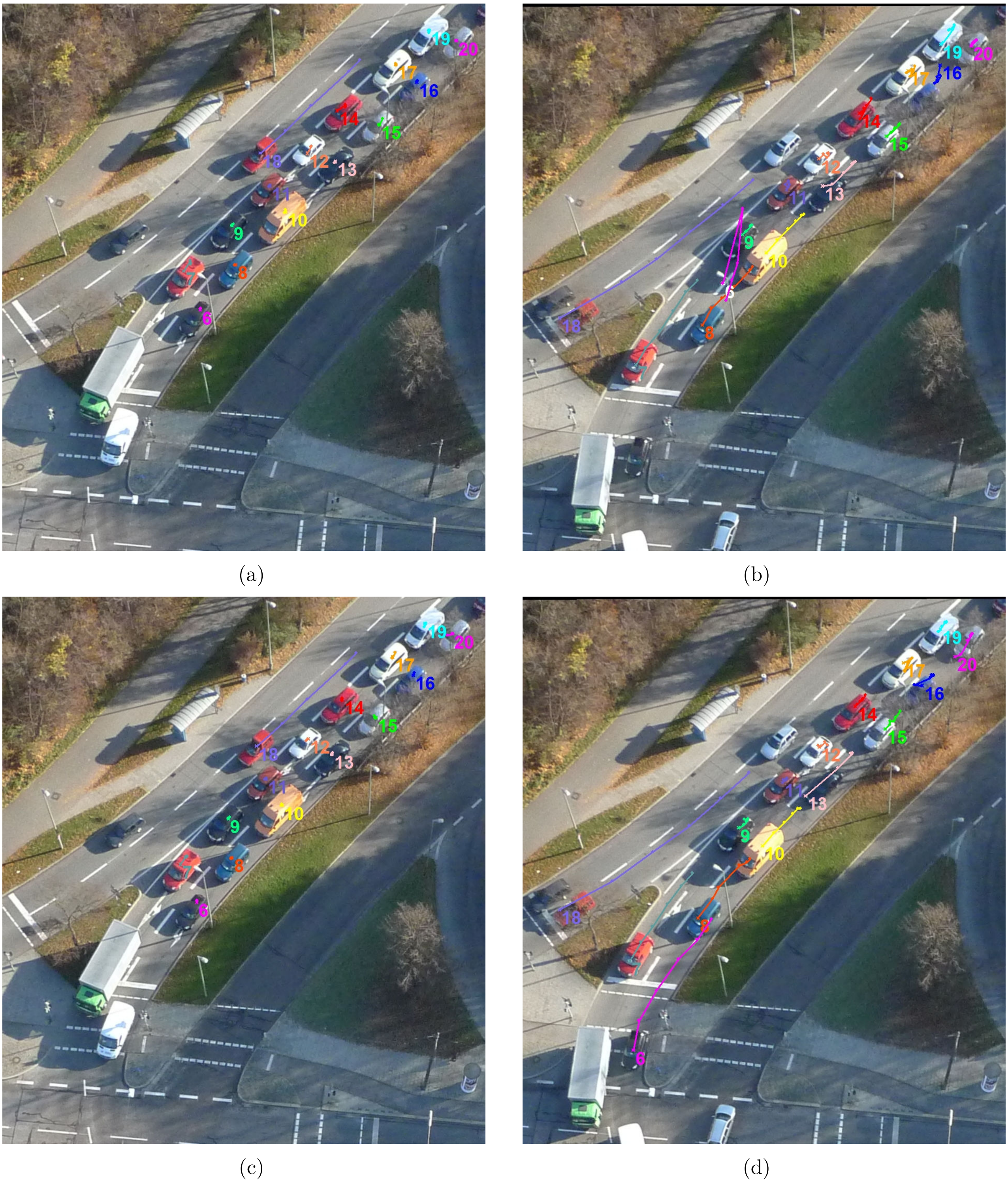
(a)

(b)

(c)

(d)

Fig. 4.15 – Tracking results for small vehicle group from the *3K+C* scene. Frames 17 and 22 are shown for (a), (b) non-adaptive sampling and (c), (d) adaptive sampling strategy. Vehicle 2 is lost with the non-adaptive sampling, but tracked successfully with the adaptive sampling strategy.



(a)

(b)

(c)

(d)

Fig. 4.16 – Tracking results for group of vehicles from the *UAV-B* scene. Frames 4 and 10 are shown for (a), (b) with non-adaptive and (c), (d) adaptive weighting strategy. Vehicles 6 and 16 are lost with non-adaptive weighting, because their appearance changes after vehicle 6 enters the shadow and vehicle 16 is partly occluded by a tree. The same vehicles are tracked successfully when the adaptive weighting strategy is enabled.

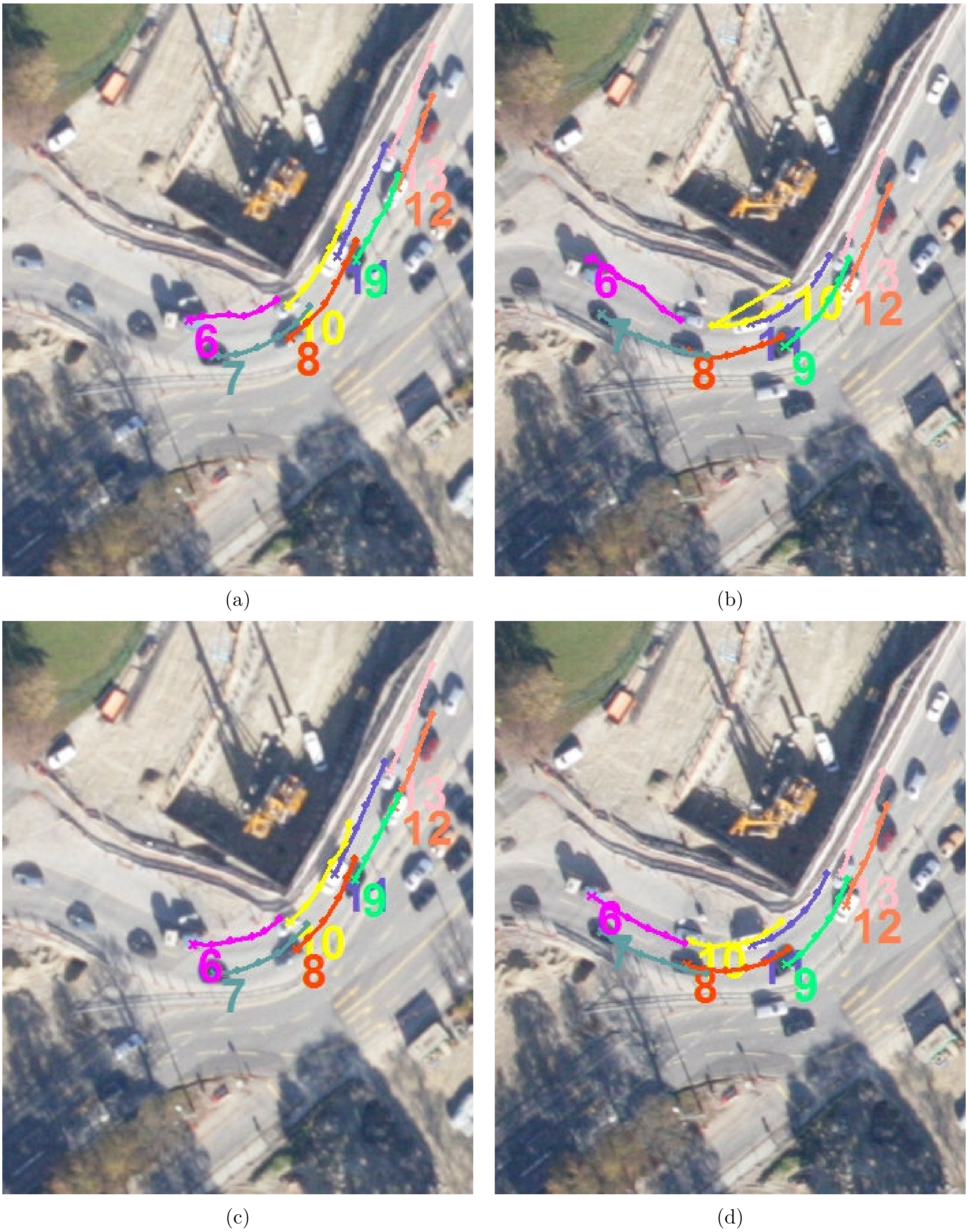


Fig. 4.17 – Tracking results for group of vehicles from the *3K-D* scene. Frames 10 and 15 are shown for (a), (b) without template update and (c), (d) adaptive with template update method. Vehicles 6 and 10 are lost without updating the template, because their appearance changes after the sharp turn. Their shadows and the perspective change from the non-nadir view angle cause the template to not match well without template updates. The same vehicles are tracked successfully when the template update method is enabled.

Finally, the effectiveness of including the online map learning approach from Section 4.6.3 is demonstrated. A group of three vehicles turning left one after the other in the *3K-A* scene is tracked, Figure 4.18. The turning maneuvers and the illumination changes impose uncertainty on the motion and observation model. The results show that the particle filter of the third vehicle successfully integrates the learned directions from the leading vehicles. Without the map learning method, the third vehicle 2 is mismatched to another close similar vehicle.

4.7.2 Tracking quality

A quantitative analysis of the tracking quality is presented for input data of test sequences from Table 4.2. The quality of the tracking results is evaluated with the Multiple Object Tracking Accuracy (MOTA) metric following the definition from [86]:

$$\text{MOTA} = 1 - \frac{\sum_{t=1}^{N_{\text{frames}}} \left(c_m(m_t) + c_f(fp_t)\right)}{\sum_{t=1}^{N_{\text{frames}}} N_G^{(t)}}, \tag{4.21}$$

where m_t is the number of misses, fp_t is the number of false positives and $N_G^{(t)}$ is the number of detected vehicles completely visible in the image at time t. The weighting functions are given by $c_m(m_t) = m_t$ and $c_f(fp_t) = fp_t$. Note that the term for counting track identity switches has been omitted from the original definition of MOTA, because the developed particle filter tracker does not perform re-initialization of the tracked vehicles when they are lost. A vehicle is missed if the estimated vehicle position is not assigned, but the vehicle is visible in the image. A false positive occurs if the estimated vehicle position exists, but is outside the ground truth vehicle region.

To account for lost vehicles, a second metric suggested by [9] is used, which defines the tracking success rate (TSR) at time step t:

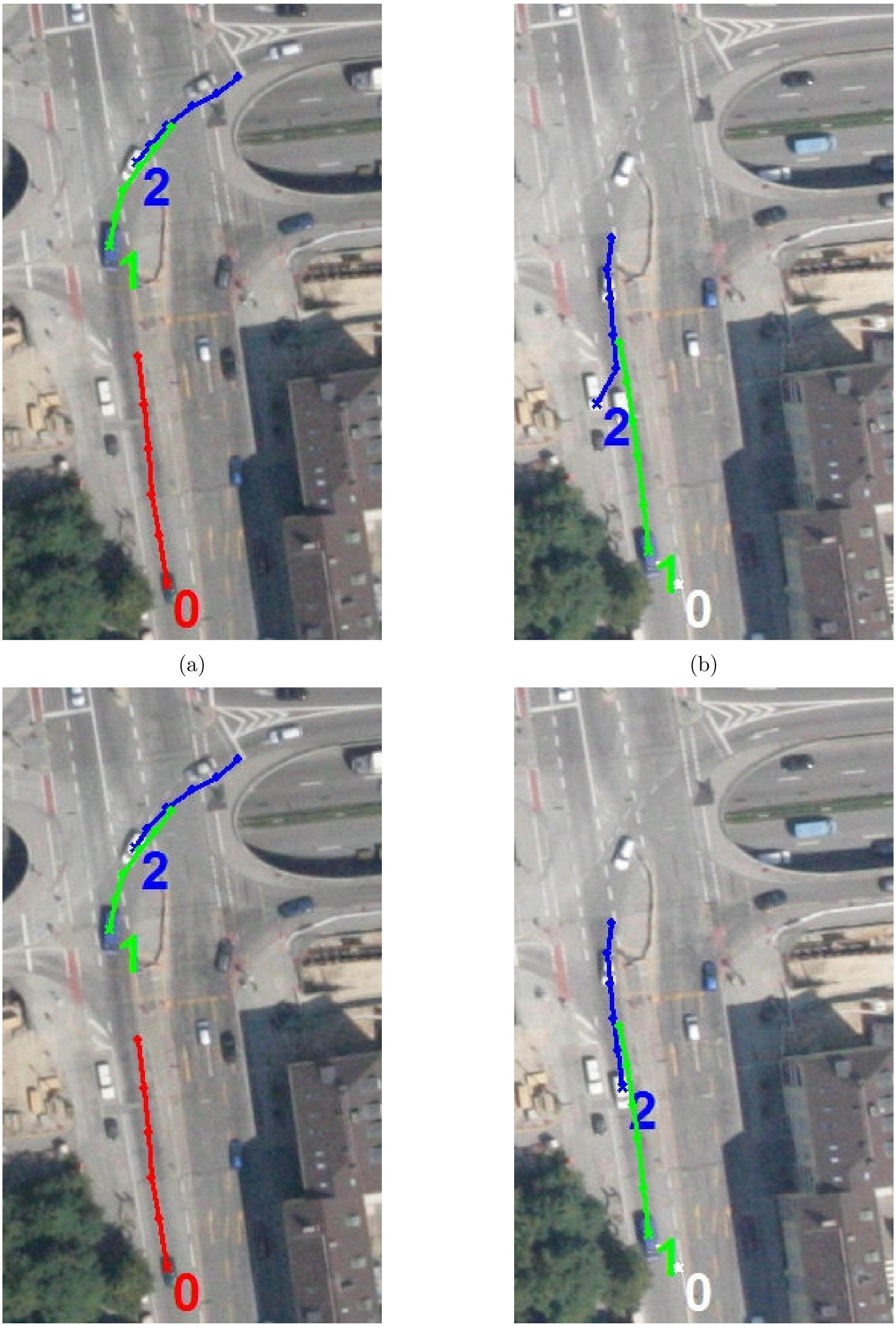
$$\text{TSR}_t = 1 - \frac{l_t}{N_G^{(t)}}, \tag{4.22}$$

where l_t denotes the number of lost vehicles. Note that the original definition of TSR counts the number of non-fragmented tracked objects, which is replaced by the number of lost vehicles, because the track label will change to undefined when the vehicle is lost. A vehicle is defined as lost from time step t onwards, if it is missed in the image t and the tracker never recovers from the mismatch until the end of the image sequence. If the tracker manages to recover from mismatches, the track label is maintained and the the number of images where the mismatch occurred are counted in the MOTA metric.

The first evaluated scene from the 3K system contains 13 detected vehicles and 26 frames. It shows a typical small urban intersection with turning vehicles, Figure 4.19. All vehicles with except from one mismatch in one frame are tracked correctly, which results in a MOTA of 0.99. Note that also the vehicle that was partly in shadow in the first couple of frames is successfully tracked. Since all vehicles are tracked until the last frame, the TSR is one.

The second sequence of the 3K system shows a similar scene with 30 vehicles and 16 frames, but the vehicles are more densely spaced and some vehicles are temporarily occluded by trees, Figure 4.20. These occlusions cause some of the vehicles to get lost during tracking. The MOTA still achieves a high value of 0.92 and the TSR in the last frame of this sequence is 0.85.

A similar situation as in Figure 4.20 with the 3K system is presented with a sequence from the 3K+ system, Figure 4.21. In this sequence, vehicles are partly occluded from trees. During 25 frames, all 14 vehicles are tracked correctly with a MOTA and TSR of 1. The particles are able to capture the vehicles when and after they are occluded, because the vehicles move straight with constant velocity.



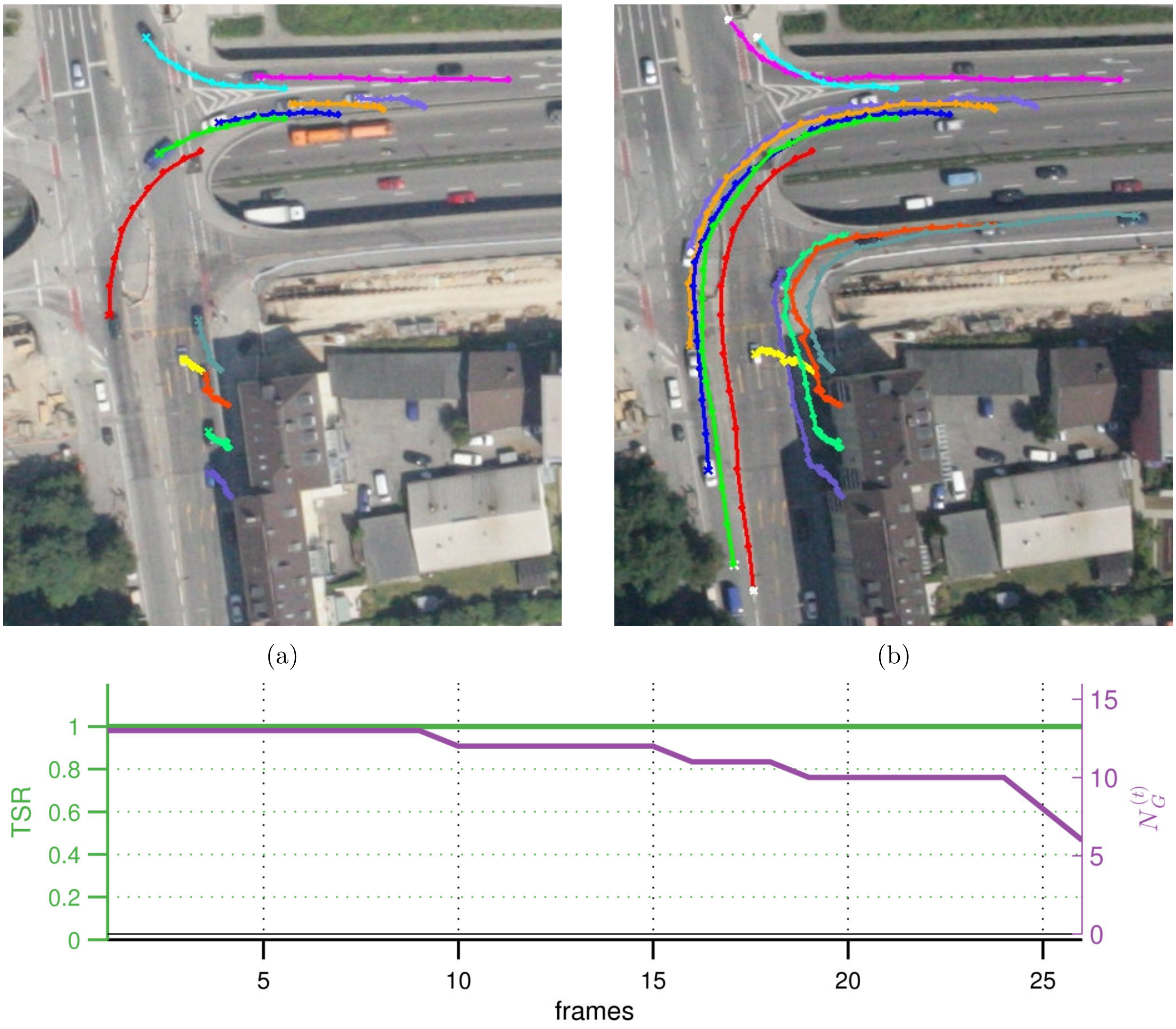
(a)

(b)

(c)

(d)

Fig. 4.18 – Tracking results for three vehicles from the *3K-A* scene. Frames 14 and 22 are shown for (a), (b) without map learning and (c), (d) map learning method. Vehicle 2 is mismatched to another vehicle without map learning, but tracked successfully with the map learning method.



(a) (b)

(c)

Fig. 4.19 – Trajectories of tracked vehicles in the *3K-A* scene, (a) frame 8, (b) frame 24. All vehicles are tracked successfully, also the bottom right vehicle (purple), which shows a different appearance after it has left the shadow. (c) The TSR is one until the end of this sequence.

The *3K+B* test scene from the 3K+ system with 12 vehicles in 19 frames is another example for a typical urban scenario. The challenge in this sequence lies in the illumination change, which produces abruptly darker images and different reflection patterns especially on dark vehicles, Figure 4.22. The high MOTA of 0.99 and the high TSR of 0.91 in the last frame demonstrate that the tracker is invariant against such illumination changes.

A shorter sequence of a group of 22 vehicles in 10 frames is presented, which suffers from poor illumination because of fog, which reduces the contrast of the images, Figure 4.23. Additionally, the low sun produces long shadows and occlusions from a small bridge and a gantry occur. Apart from some outliers, all vehicles are tracked successfully with a MOTA of 0.98. Also the TSR is very high, with a value of 0.94 in the last frame.

The UAV system has a longer recording time and a higher resolution of the images, but it suffers from a deviate from the optimal nadir view angle. Two example sequences from the UAV are evaluated to compare the tracking quality between the UAV and the 3K/3K+ system.

The first sequence *UAV-A* shows an intersection with 11 vehicles turning left, right and going straight, Figure 4.24. Since this sequence is very long with 71 frames, it captures multiple traffic light phases, which cause the vehicles to accelerate and decelerate. The non-nadir view angle lets the vehicles appear under different perspectives during the turn maneuvers. The fast acceleration maneuvers and shadows from trees pose challenging conditions to the tracker. The resulting MOTA is still very high with a value of 0.92, which indicates successful tracking. Target losses are caused by fast accelerations of the vehicles and appearance changes caused by perspective variation. The TSR at the last frame is lower compared with the last frames of the 3K/3K+ examples, because this sequence has considerably more frames than the previous 3K/3K+ examples. Before the TSR decreases after frame 37, it is 1, which is better than in the 3K/3K+ examples, because the 3K/3K+ sequences are shorter than 37 frames or their TSR at frame 37 is smaller.

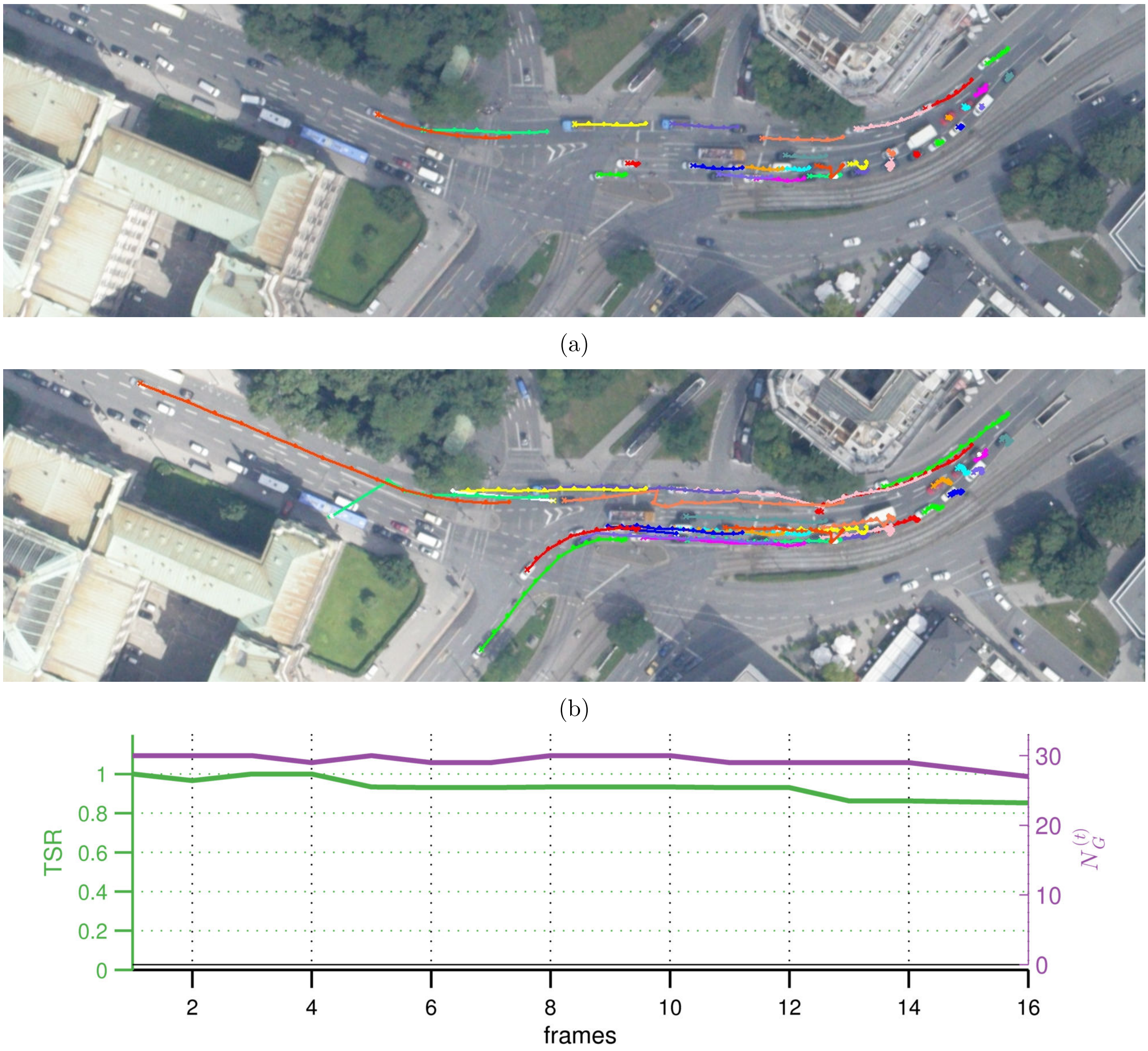
Another larger and longer sequence from the UAV, which has 103 frames and 21 vehicles, shows that a higher resolution does not correlate with a better tracking quality considering the peculiarities of the UAV sequences as mentioned in the previous example. From the results of the *UAV-B* sequence shown in Figure 4.25 it is apparent that shadows and perspective changes cause difficulties for the tracker. In this example, the sun is very low and the shadows from high trees and large vehicles lead to severe illumination changes of the vehicles. Although many vehicles are tracked correctly, which is indicated by a MOTA value of 0.69, a considerable number of them is lost. In the middle of tracking, when still 14 vehicles are visible, the TSR has decreased to 0.57.

Since the 3K+ images have a higher resolution than the 3K images and the other properties of the flight system are comparable, a correlation between resolution and quality might be assumed. Two sequences from the same scene taken from the 3K system and at another day from the 3K+ system are compared. This traffic scene is characterized by a large intersection, where the road course has been relocated for construction purposes, Figures 4.26 and 4.27.

With the 3K system, the *3K-D* scene contains 151 vehicles and it is captured with 20 frames. Since this scene is very large, only one image is presented with the tracked trajectories at frame 17, where most of the vehicles are still visible, Figure 4.26. The scene features vehicles in dense traffic, acceleration and deceleration maneuvers, sharp turns and shadows mainly from other vehicles. The evaluation of this sequence finds a MOTA rate of 0.89, which is lowest from all 3K examples, but still a very good result regarding the adverse tracking conditions. In the last frame, the TSR is 0.76, which is little lower than in the previous 3K examples at frame 20.

The same scene is evaluated with images from the 3K+ system. The *3K+C* sequence contains 89 vehicles in 46 frames and is similarly characterized as its 3K counterpart, Figure 4.27. Since this sequence is longer than *3K-D*, additionally merging and pulling out vehicles can be observed, which are difficult to predict in their motion. Also, the risk of mismatches or target losses increases with each frame. For the total sequence, the MOTA is 0.80, which is lower than in the 3K example. However, when the MOTA of *3K+C* is computed only until frame 20, which is the last frame of *3K-D*, then the resulting value of 0.94 is higher than for the 3K sequence where the MOTA is 0.89. The TSR at frame 20 is 0.89, which is also higher than the TSR of 0.76 of the 3K sequence. Thus, the comparison of *3K-D* and *3K+C* until frame 20 suggests, that tracking vehicles with the 3K+ system produces better results than with the 3K system.

The MOTA is a common measure to evaluate the tracking quality, which makes the results comparable to other work. A similar application of airborne vehicle tracking reports an average MOTA of 0.85 [173]. The analyzed test sequences have similar characteristics as the images in this work, because they contain wide-area traffic scenes with dense traffic and they are captured with 1 Hz. In comparison to that tracker, the average MOTA rate for the test sequences of this work is 0.91. The MOTA for all test sequences is summarized in Figure 4.28. It should be noted that this comparison lacks normalization to equal sequence lengths, but it indicates that the developed particle filter tracker in this work produces competitive results.

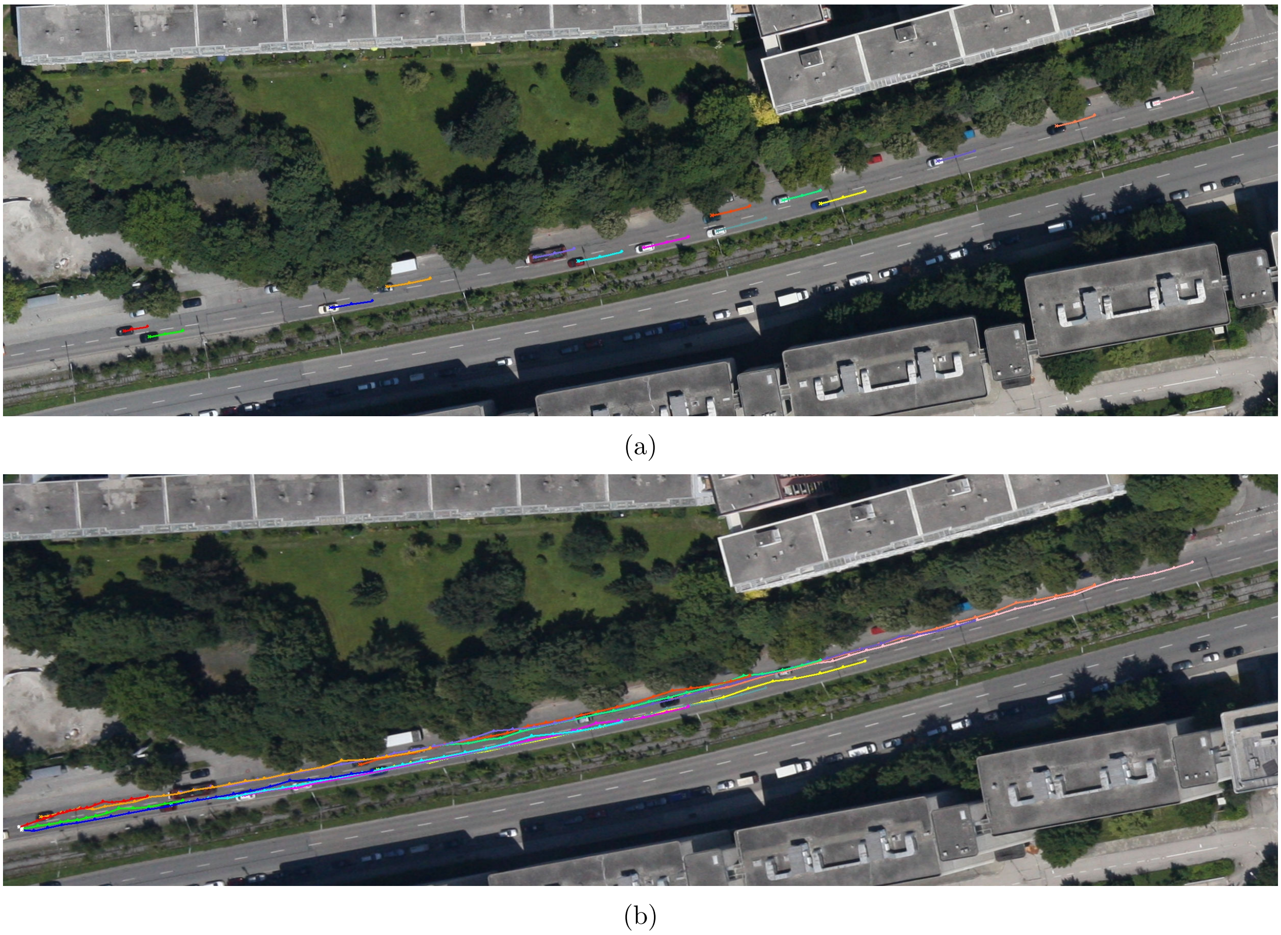


(a)

(b)

(c)

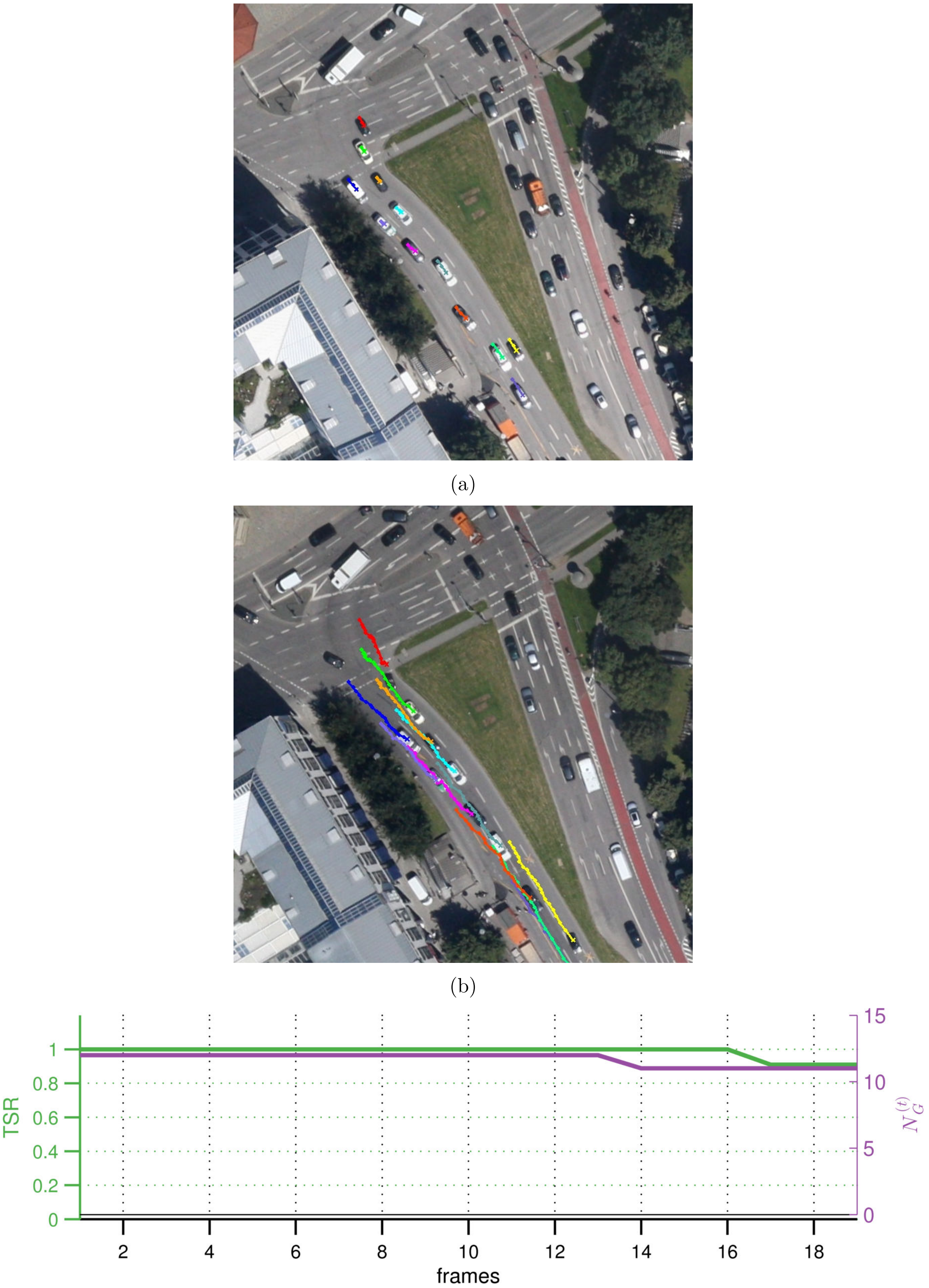
Fig. 4.20 – Trajectories of tracked vehicles in the *3K-B* scene, (a) frame 5, (b) frame 14. Occlusions from trees occur in this scene, which causes four vehicles to get lost. (c) The TSR is remains at a high level until the end of the sequence.



(a)

(b)

Fig. 4.21 – Trajectories of tracked vehicles in the *3K+A* scene, (a) frame 2, (b) frame 22. Vehicles are partly occluded by trees, but all vehicles are completely tracked correctly, which results in a constant TSR of 1.

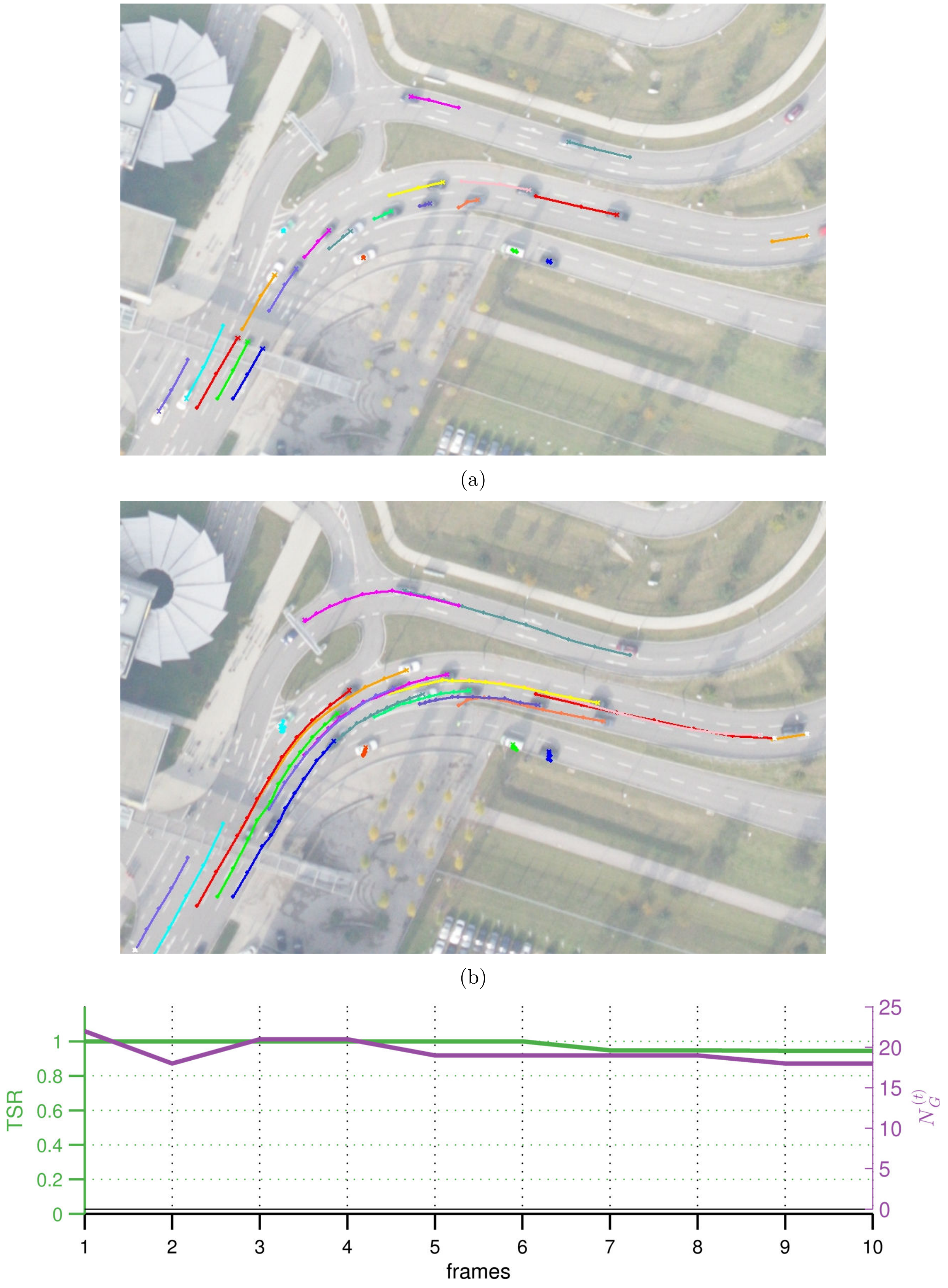


(a)

(b)

(c)

Fig. 4.22 – Trajectories of tracked vehicles in the *3K+B* scene, (a) frame 2, (b) frame 19. The image from (b) appears darker than (a) because of illumination changes, which also affects the reflection of sunlight on the vehicles. (c) The TSR is remains at a high level until the end of the sequence.

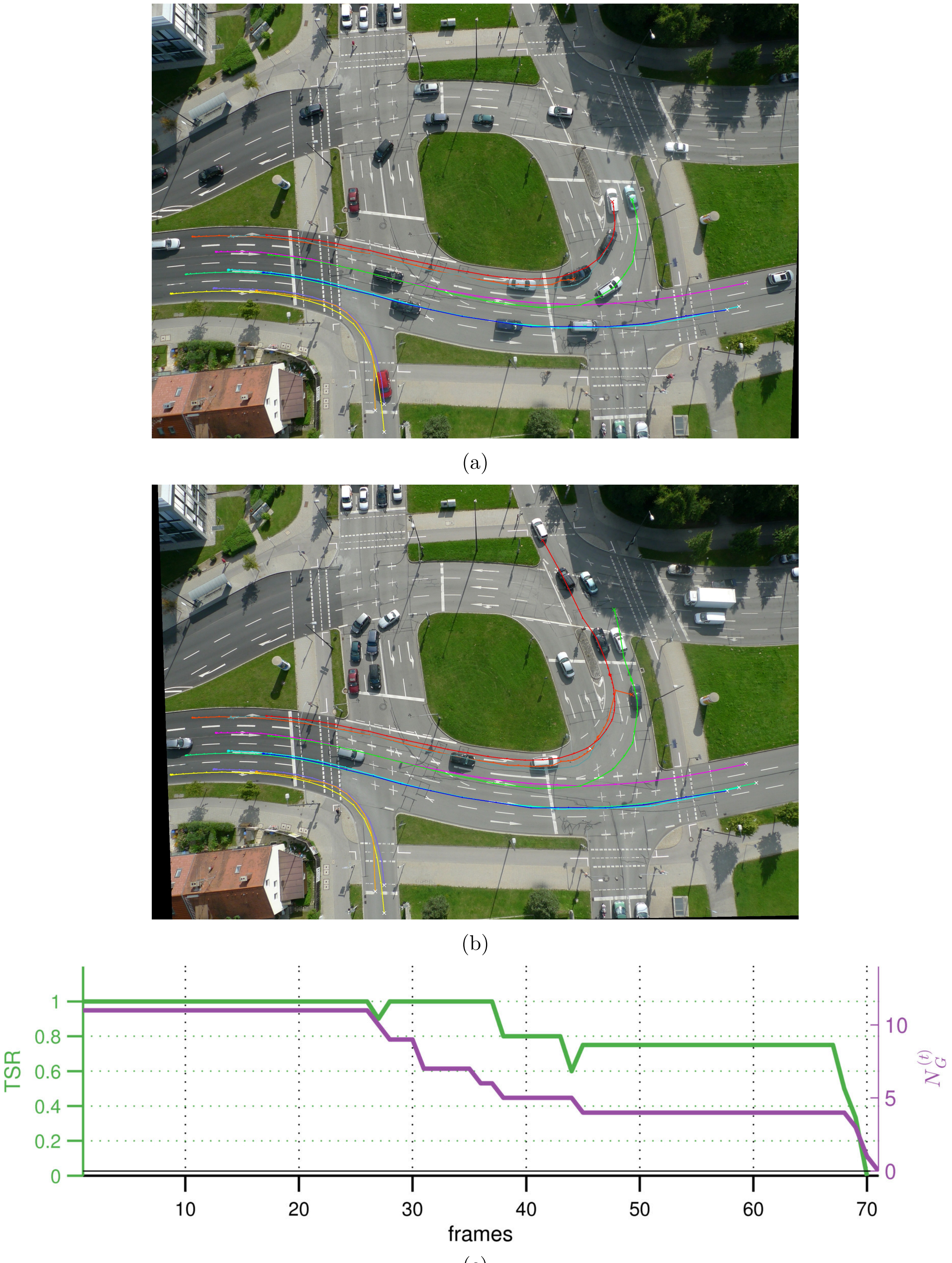


(a)

(b)

(c)

Fig. 4.23 – Trajectories of tracked vehicles in the *3K-C* scene, (a) frame 2, (b) frame 10. Although three vehicles are almost totally occluded by a bridge (bottom left), they are tracked correctly. (c) The TSR is remains at a high level until the end of the sequence.

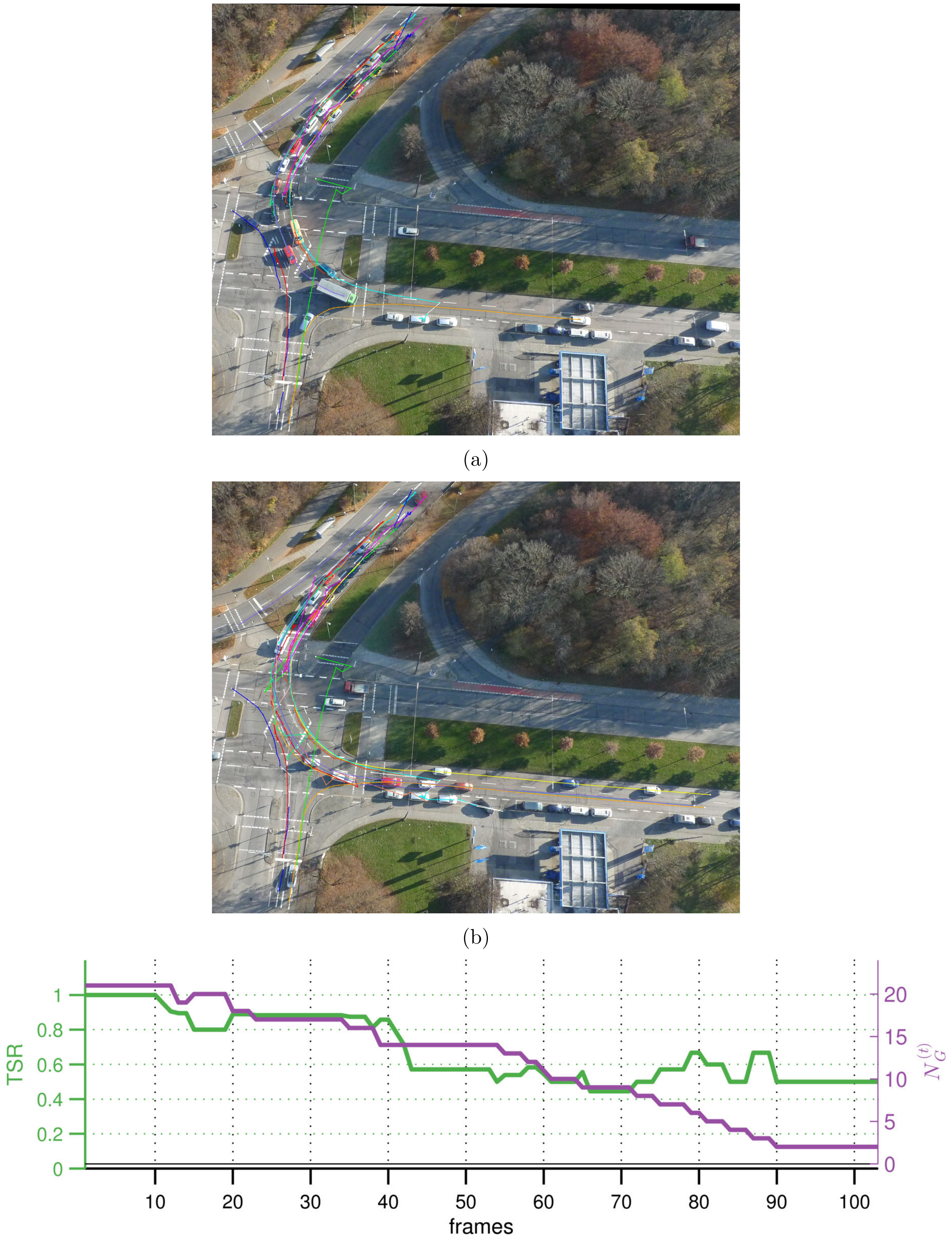


(a)

(b)

(c)

Fig. 4.24 – Trajectories of tracked vehicles in the *UAV-A* scene, (a) frame 37, (b) frame 68. Until frame 37, all vehicles are tracked successfully. After this, perspective and illumination changes together with accelerations cause the tracker to loose vehicles. (c) The TSR is remains at a high level until frame 37, it then decreases to 0.75 and drops until zero at the end of the sequence when only one vehicle from the initial set of detected vehicles is visible but lost.

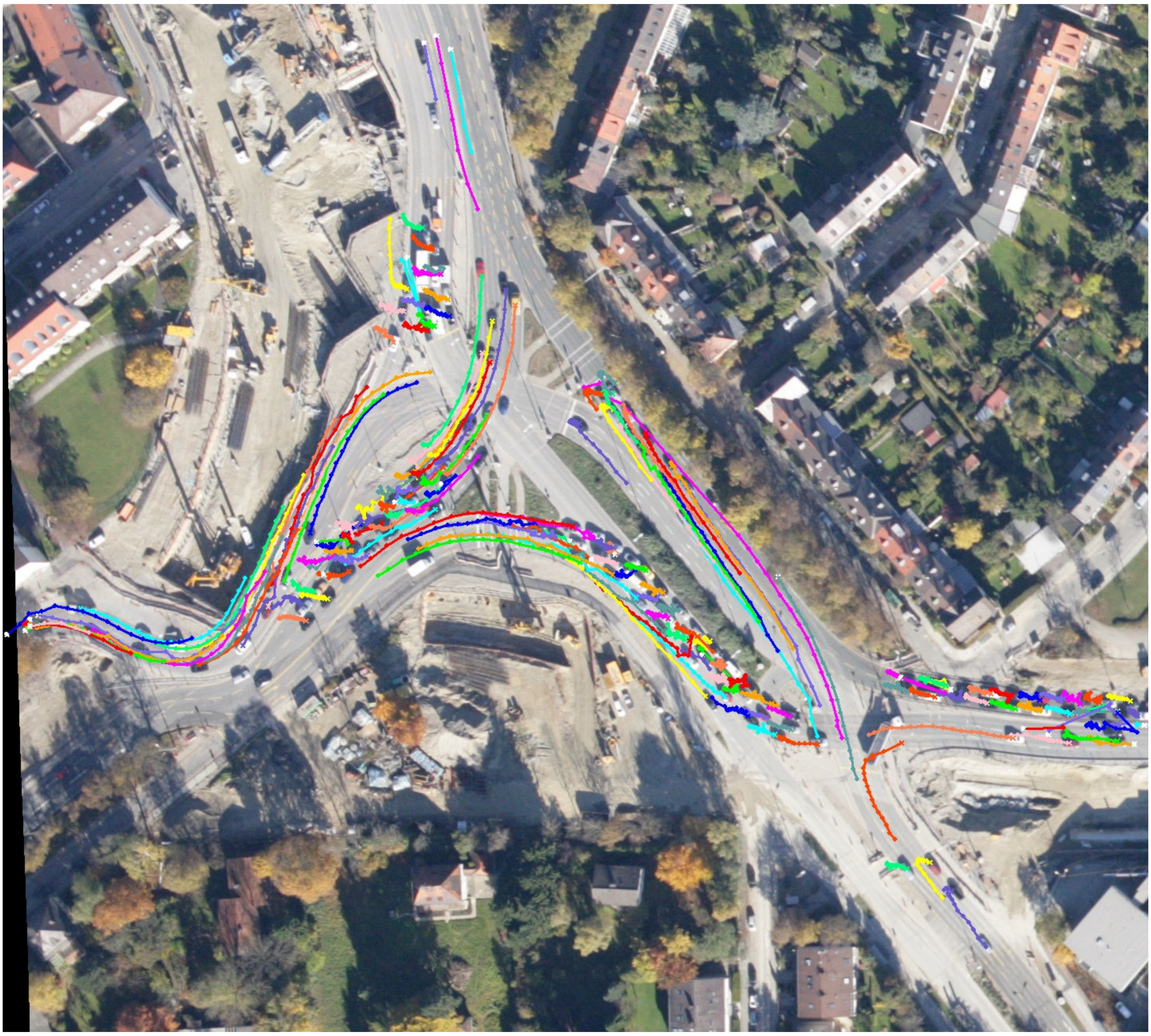


(a)

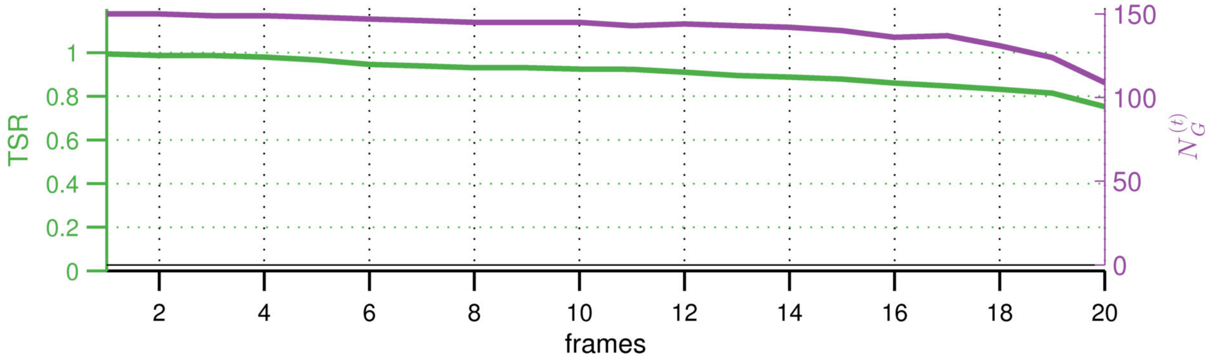
(b)

(c)

Fig. 4.25 – Trajectories of tracked vehicles in the *UAV-B* scene, (a) frame 34, (b) frame 69. Vehicles are lost during tracking mainly when they enter shadows or their perspective changes. Also, mismatches to parked vehicles occur. (c) The TSR decreases steadily from frame 10 onwards and falls down to 0.5 at the end of the sequence when only two vehicles from the initial set of detected vehicles are visible.

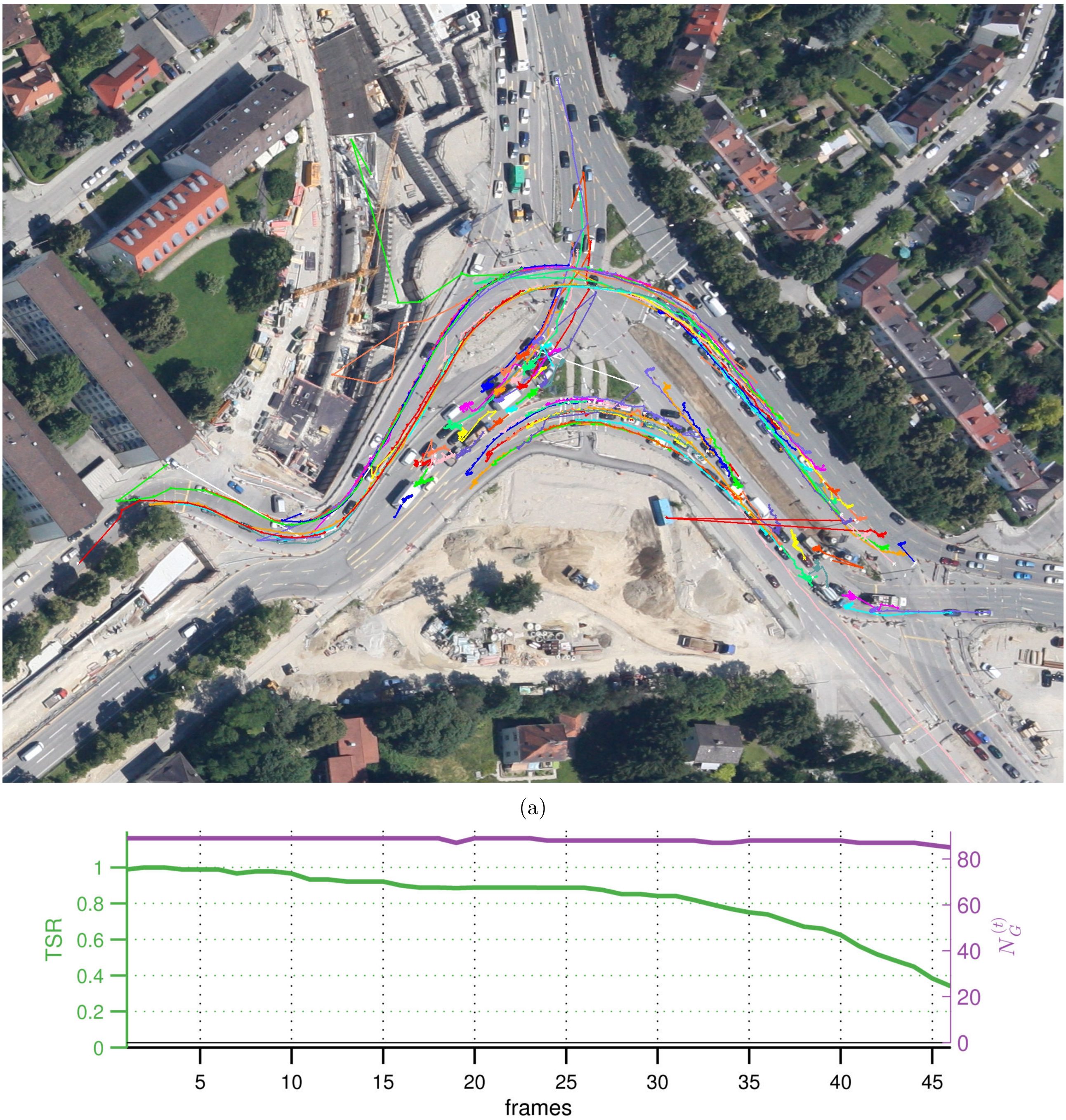


(a)



(b)

Fig. 4.26 – Trajectories of tracked vehicles in the *3K-D* scene, (a) frame 17. Vehicles are mismatched to other vehicles or to background, when they change their appearance due to perspective change. (b) The TSR decreases slowly from the beginning, but remains at a high level. At the end of the sequence, the TSR has decreased to 0.75.



(a)

(b)

Fig. 4.27 – Trajectories of tracked vehicles in the *3K+C* scene, (a) frame 46. (c) The TSR decreases slowly from the beginning, but remains at a high level until frame 26 where it has a value of 0.89. After this, it decreases and reaches 0.34 at the end of the sequence.

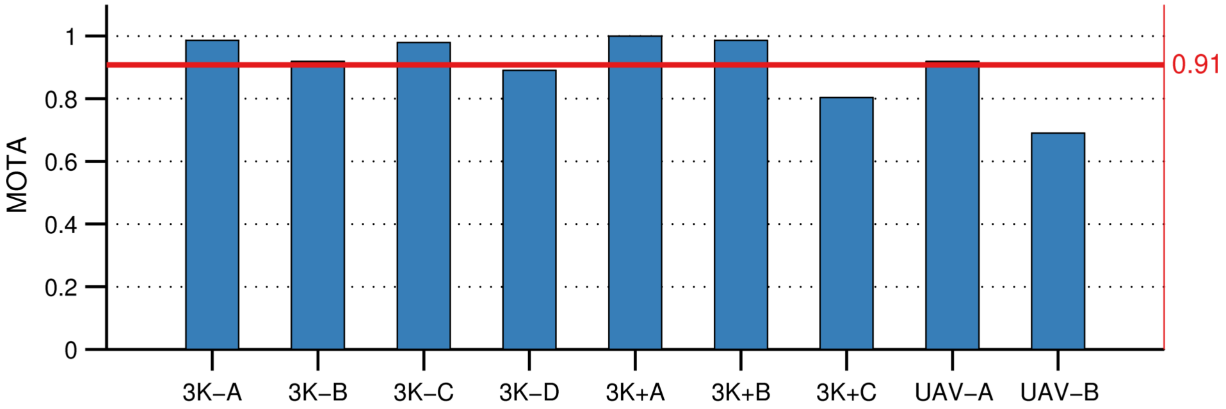


Fig. 4.28 – MOTA metric for all test scenes. The average MOTA is 0.91.

4.7.3 Limitations for occluded vehicles

The test sequences in the previous section show that occlusion is one of the main risks for loosing the tracked vehicles. Short occlusions by single trees, gantries or small bridges can often be handled successfully by the tracker, which is demonstrated in Figure 4.21 and 4.23. The effect of occlusion on target losses depends on the duration of occlusion, but also on the appearance of the occluding objects and the motion of the occluded vehicles. If the occluding objects are similar to the occluded vehicles, the tracker can produce mismatches or it can be "hijacked" by these objects. Thus, the effects of occlusion on the tracking results depend on multiple factors that cannot be described by specific parameters. Therefore, to evaluate the sensitivity of the tracker to occlusions, a classical sensitivity analysis is not feasible and the sensitivity of the tracking system to occlusion of vehicles is analyzed in characteristic situations.

Figure 4.29 shows an example with occlusion caused by two bridges. After the first bridge is passed by the group of three tracked vehicles, two vehicles are correctly tracked. This bridge has a different shape and color as the vehicles, which leads to low weights of the particle cloud. In contrast, the second bridge is very challenging, because similar vehicles are parked on it, although the duration of occlusion is comparable to the first bridge. The tracker of one vehicle successfully continues tracking after the vehicle appears again. The trackers of the other two vehicles that pass this bridge do not recover from the temporal loss, because they stick to other vehicles on the bridge.

Figure 4.30 illustrates a scene where vehicles are occluded by trees. Additionally, shadows from these trees complicate tracking after the vehicles appear again. The results show that such occlusions confuse the tracker and target losses occur. The first vehicle that passes the trees is partly occluded and is tracked correctly. However, the other smaller vehicles are almost completely occluded. The tracker of the following two vehicles resumes tracking after the first occlusion, but looses them after the second occlusion. The reason for target loss is the appearance similarity of the shadows, which distracts the tracker from the true target position.

4.7.4 Conclusion

The developed particle filter method for tracking vehicles in airborne images was evaluated on test sequences with urban dense traffic. The robustness of the tracker against variation of motion and appearance of the vehicles is improved by advanced techniques for particle sampling and weighting, target template update and online map learning. Examples of challenging scenarios demonstrated

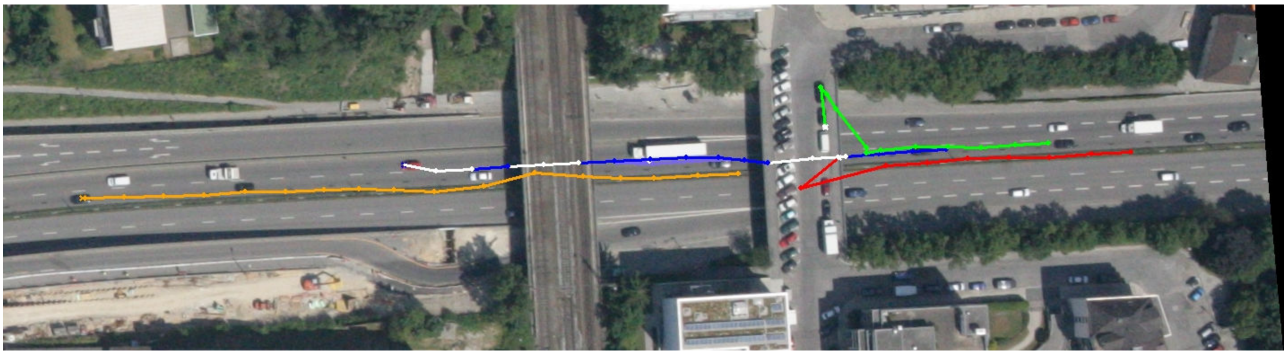


Fig. 4.29 – Trajectories of tracked vehicles at frame 16. Two bridges cause total occlusion of the vehicles. The first bridge is easily discriminated from the vehicles, which causes the tracker to resume tracking for the first two vehicles (blue, orange). The parked vehicles on the second bridge confuse the tracker, which leads to target losses of the other two vehicles (red, green).

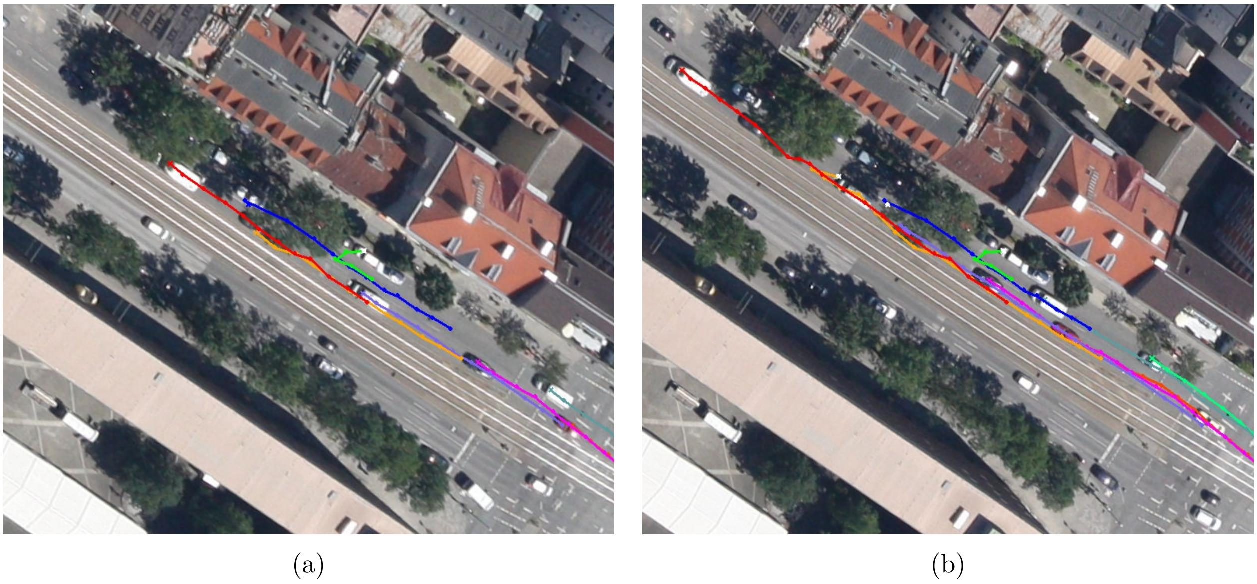


Fig. 4.30 – Trajectories of tracked vehicles, (a) frame 11, (b) frame 19. Occlusions from trees and their shadows confuse the tracker, which leads to target losses. After the first occlusion in frame 11, the first vehicle (red) and the following two (blue, orange) are tracked successfully. After the second occlusion in frame 19, only the first vehicle is captured while the other two are lost under the trees and in the shadows.

the effectiveness of these techniques to capture vehicles under discontinuous motion and appearance changes.

The evaluation of the tracking method on wide-area scenes reports an average MOTA rate of 0.91, which indicates that the performance is competitive to other airborne trackers. The particle filter tracker demonstrated superior performance in challenging situations compared to state-of-the art methods, which lack results of scenarios with discontinuous motion or appearance changes. The main reasons for mismatches or target losses are given by fast accelerations or sharp turns of the vehicles, shadows from buildings, trees or other vehicles and perspective changes caused by non-nadir view angles. Short occlusions from single trees, gantries or small bridges can be handled by the tracker, while occlusions from tree rows, road bridges poses difficulties for the tracker.

Three flight systems with different cameras and flight heights were employed that result in different coverages and different GSD. A comparison of the same test scene captured with the 3K and the 3K+ system demonstrates that the 3K+ system produces images that result in a higher MOTA when compared for the same sequence length than the 3K system. The resolution of the UAV system images is even higher than the 3K/3K+ systems, but an improved tracking quality is not observable, because UAV images suffer from a non-nadir view angle that reinforces the problem of perspective changes caused by vehicle pose changes.

The adaptive weighting strategy and the template update method mitigate the problem of appearance changes, but still situations exists where the tracker cannot capture vehicles under abrupt appearance changes especially caused by shadows. The same holds for discontinuous motion changes that are not represented by the adaptive motion model in some cases.

5 Lane-level map-matching

This chapter describes a method to solve the map-matching problem formulated as tracking the vehicle constrained to a digital map. The map-matching system is designed to meet requirements of low-cost sensors and commercial map data arising by the use of the system in series production vehicles. A particle filter method is presented that contains efficient models for particle sampling and weighting under absolute positioning and map inaccuracies. The evaluation of the proposed method focuses on ambiguous situations, where a fast and reliable decision for the correct road segment is critical.

5.1 Map-matching system

The system architecture of the map-matching system is based on the potential integration into a vehicle navigation system. In this work, a modular approach is chosen, which allows the map-matcher to be deployed in different vehicle systems. The map-matching component is designed to have minimum requirements regarding input data, which are obtained from commercially available sensors and map products. The proposed map-matcher is embedded into a navigation framework, which allows extensive onboard testing of the developed algorithms.

5.1.1 System architecture

Modern vehicle systems are characterized by a partitioned architecture, because various sensors and control units from different suppliers need to be integrated. Recent developments on hybrid navigation systems pushed on this separation for not only physically separated, but also portable systems, which can be fully integrated into the vehicle system as plug-in device [38]. This design principle suggests a modular architecture also for the navigation system, which includes the map-matcher as a separate component. Other components include the positioning module, the vision system and the human-machine interface (HMI). To account for this physical separation, the map-matcher is treated as an individual component. An overview of the decentralized architectural concept is shown in Figure 5.1.

The absolute positioning component comprises the fusion of GNSS and IMU sensor measurements and computes an estimate of the absolute vehicle position in geographical coordinates together with an estimate of its velocity and absolute heading. The onboard vision system detects lane markings of the left and right lane boundaries and lane change events. The map-matching component fuses input parameters from positioning, vision and map access interfaces and provides the map-matched position for ADAS and navigation applications. The map-matched result is visualized together with a rendering of the digital map.

The implementation of the developed map-matcher is integrated in an existing navigation platform and tested with a BMW research vehicle. The employed navigation framework has been developed at BMW Research and Technology and provides a modular framework for rapid prototyping of navigational functions. The existing components cover map access, routing, guiding, map rendering and positioning functionalities. In this way, a realistic experience of the map-matching quality is facilitated. Real-time

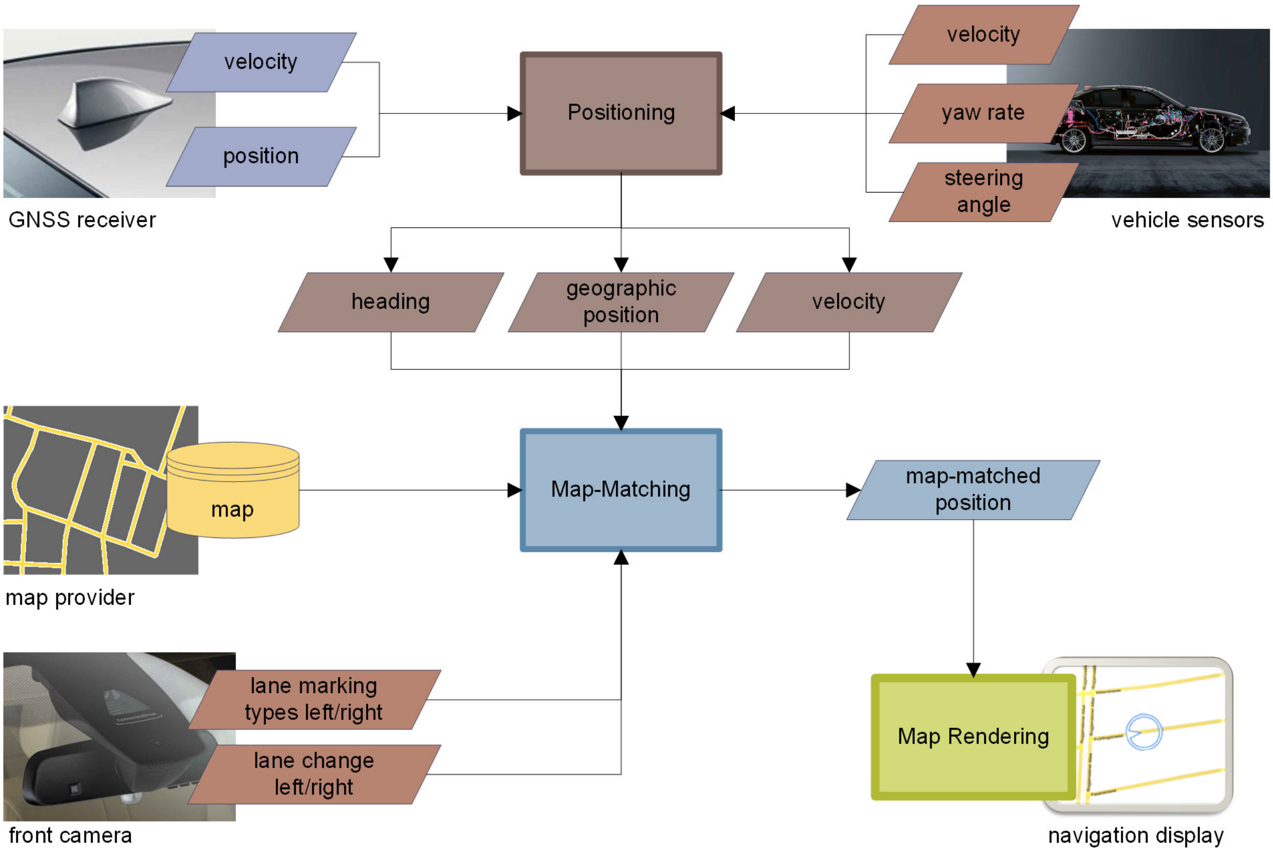


Fig. 5.1 – Overview of map-matching system components

updates of the map-matched position in the vehicle navigation display provides an intuitive evaluation of the system performance.

The proposed architecture of the map-matching system meets requirements of series production vehicles and makes the system prepared for deployment in a built-in navigation system.

5.1.2 Vehicle sensors and map data

The map-matching component within in the navigation framework can be integrated in any modern BMW series production vehicle with some additional hardware. A BMW 5 Grand Turismo series vehicle has been modified to test the map-matching functionality. Details of these modifications are shown in Figure 5.2.

The integration of the map-matcher result in the navigation display allows a comprehensive evaluation of the latency and accuracy of the algorithm under real driving conditions. A logging component traces all input signals of the map-matcher for data analysis outside the vehicle.

The following system components are used for experimental validation of the functionality of the lane-level map-matcher:

GNSS

To receive GNSS signals, the single frequency, low-cost GPS/GLONASS receiver Novatel FlexPak-G2 OEMStar is employed [125]. The receiver produces position, velocity and time signals at up to 10 Hz. It is equipped with an antenna attached to the vehicle roof with a magnetic mount, Figure 5.3. The

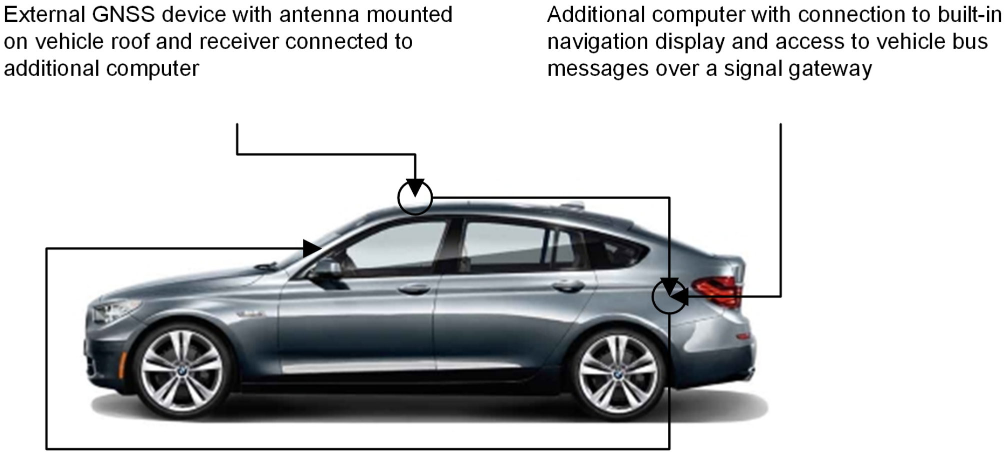

External GNSS device with antenna mounted on vehicle roof and receiver connected to additional computer
Additional computer with connection to built-in navigation display and access to vehicle bus messages over a signal gateway

Fig. 5.2 – BMW 5 series Grand Tourismo research vehicle with modifications. An external GNSS device is mounted to access all signals required for estimating the absolute vehicle position.

position estimate is obtained with disabled DGPS due to the costly subscription charge. From the manufacturer's data sheet, a single-point positioning accuracy of 1.5 m Root Mean Square (RMS) can be achieved. However, common sources of error, as described in Section 2.2.2, drastically degrade the resulting performance during practical use.

Inertial sensors

The inertial measurements are performed with accelerometers and gyrometers from the series vehicle. The measured signals also include the velocity estimated from wheel ticks, yaw rate and steering angle.

To estimate the vehicle position with dead reckoning, a model of the underlying longitudinal and lateral dynamics is required. In this work, it is assumed that the lateral vehicle dynamics follow a single-track model [147]. The longitudinal velocity is assumed to be constant. This model is characterized by two degrees of freedom, the lateral position and the yaw angle of the vehicle, with the steering angle as input parameter, Figure 5.4.

(a) (b)

Fig. 5.3 – GNSS for prototyping of positioning systems in series vehicles [125]. (a) GNSS receiver with communication ports to connect with a computer. (b) GNSS antenna with flush magnetic mount.

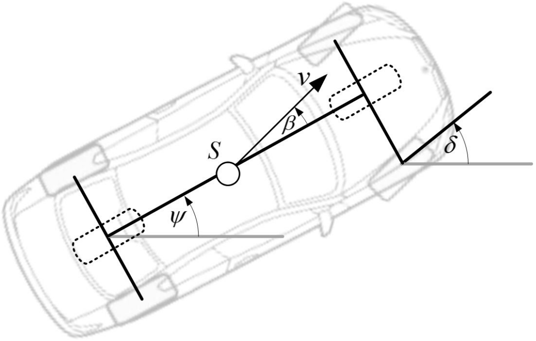


Fig. 5.4 – Single-track model of lateral vehicle dynamics, adapted from [147]. The front tires and the rear tires are combined into an imaginary contact point at the center of the front and rear axles. The vehicle's mass is located at center of gravity S, the yaw angle ψ, the side slip angle β and the steering angle δ describe the lateral motion.

Geographic vehicle localization

The absolute vehicle position is estimated by dead reckoning combined with GNSS measurements for robust absolute positioning. For this purpose, a field-tested positioning method as described in [170] is applied.

An overview of the filtering steps within this system is shown in Figure 5.5. The suggested approach first compensates errors from the velocity estimate due to tire radius variations by fusing the measured velocities from the wheel tick sensors and the GNSS receiver. The subsequent sensor fusion of the measured yaw rate and the steering angle employs an extended Kalman filter (EKF) to obtain a corrected yaw rate estimate. The corrected yaw rate and the estimated velocity serve as input parameters for the dead reckoning method, which calculates the predictions within the main EKF in the system. The signals transmitted from the GNSS receiver by the NMEA protocol including latitude and longitudes coordinates are integrated as measurement updates. Since the GNSS signals arrive at 10 Hz, the sampling period of the filter follows the same frequency. The output of the filter at time t contains the following estimates:

- absolute vehicle position in geographic coordinates $z_t^{\mathrm{x}}, z_t^{\mathrm{y}}$,
- velocity v_t
- absolute heading θ_t.

A detailed description of a similar Kalman filtering approach for GNSS/dead reckoning positioning is found in [40].

Camera system

The employed vision system consists of a standard front camera, which is used for the lane keeping assistant. This camera is integrated in the front mirror mount, close to the rain sensor, Figure 5.6.

The image processing unit detects the line markings, which mark the left and right boundaries of the ego lane. Lane markings up to a range of approximately 60 m along a straight road can be detected under optimal conditions. For curved roads, the range of foresight reduces with increasing curvatures.

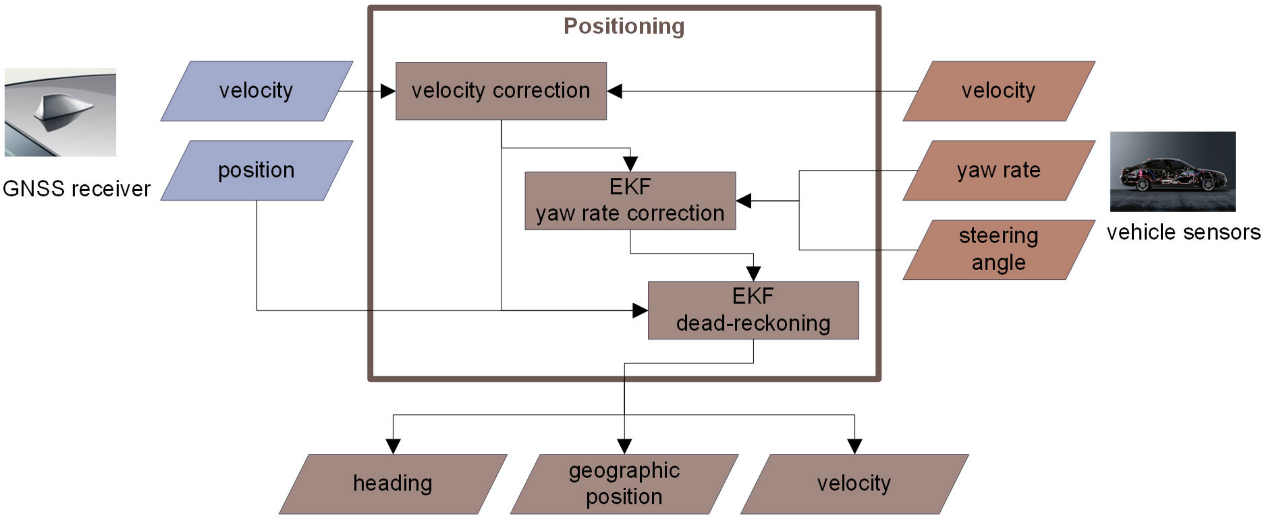


Fig. 5.5 – Overview of the absolute positioning component within the map-matching system, adapted from [170]

At sharp turns, lane detection may completely fail. The measurements of the vision system are updated at 15 fps (frames-per-second) and can contain the following parameters:

- left and right lane boundary marking type $b^{\mathrm{l}}, b^{\mathrm{r}} \in \{\text{solid}, \text{broken}, \text{unknown}\}$,
- lane change event to the right or left lane $a_t \in \{-1, 1\}$.

The detected lane markings are already processed by the system to represent the closest visible lane markings in front of the vehicle. The lane change event is triggered, when the vehicle has passed the adjacent lane boundary with a sufficiently high confidence.

The quality of the detected lane markings varies depending on the environmental conditions. Optimal conditions are given by a dry road and an cloudy sky with no direct sunlight. In this case, reflections and shadows on the road surface are minimized. Dirt or snow on the road can cause partly occluded lines, which prevent a successful detection from the vision system. Heavy rain or snow leads to a reduced visibility and a limited range of foresight, which diminishes the reliability of the measurements. Darkness is not an issue for the system, as long as the lane markings fall within the illuminated range of foresight. The detection quality also depends on the condition of the road and painted markings itself. Broken lines are likely to be missed or misinterpreted. Tar seams and other road surface structures parallel to the lane markings often cause false detections. Typical situations, where the lane detection is not reliable, occur at intersections.

Fig. 5.6 – Camera integrated in the front mirror mount of a BMW vehicle [12]

Map data

The employed commercial map follows the GDF standard with additional attributes, as described in Section 2.2.3. It is assumed that the following map attributes are available:

- link connectivity describing allowed turn maneuvers, denoted as link successors,
- number of lanes describing drivable lanes for the particular link,
- lane connectivity describing allowed turn maneuvers at the lane-level, denoted as lane successors.

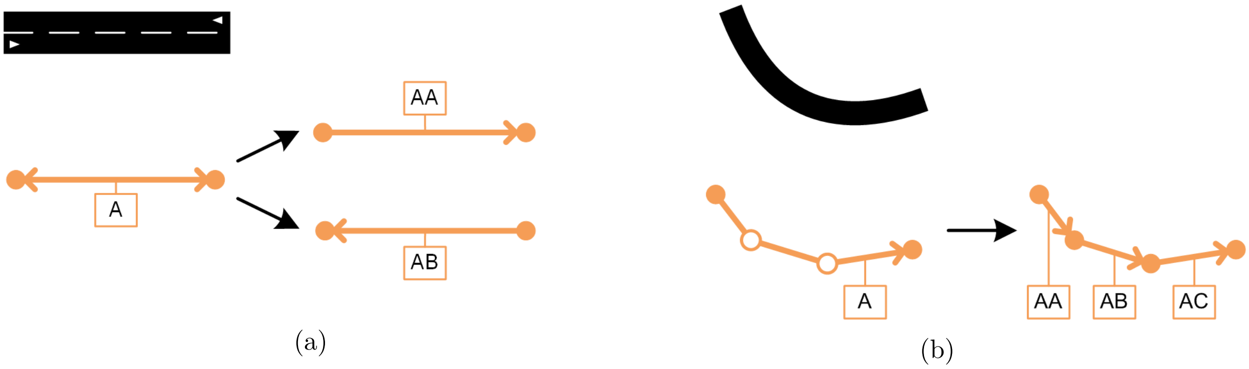
Within the map access component of the map-matching system, auxiliary functions are required to retrieve information from the road network database. The following functions are defined for a road network with a set of links K and link $k \in \mathrm{K}$:

- length of link, $\mathrm{L} : \mathrm{K} \to \mathbb{R}_+$,
- adjacent links, which can potentially be reached from k without regard of turning regulations, $\mathrm{adj} : \mathrm{K} \to \mathcal{P}(\mathrm{K})$,
- link successors, which are reachable from k according to turning regulations, $\mathrm{succ} : \mathrm{K} \to \mathcal{P}(\mathrm{K})$,
- number of lanes, $\mathcal{L} : \mathrm{K} \to \mathbb{N}$, the range of lane numbers for k is defined as $\mathrm{N}_k = [1, \ldots, \mathcal{L}(k)]$, where 1 corresponds to the rightmost lane.

For lane $l \in \mathrm{N}_k$ and link $k \in \mathrm{K}$, the lane successors are defined as $\mathrm{cnt}(k, l) : \mathrm{K} \times \mathbb{N} \to \mathcal{P}(\mathrm{K} \times \mathbb{N})$.

From the formal definition of the road topology, links can have one or two driving directions. Without loss of generality, this convention is adapted: Each driving direction is associated with an individual link, links with both driving directions are split into two links with opposite driving direction keeping their respective number of lanes, Figure 5.7. Similarly, links containing shape nodes are split into multiple links, where the shape points are transformed to the start and end points of the new links. This formalization facilitates the interpretation of the road network as a directed graph with nodes and links as edges.

Since the architecture and the sensor configuration of the map-matching system are now established, the methodological aspects of the developed algorithm can be addressed.



(a)

(b)

Fig. 5.7 – (a) Road with one lane for each driving direction. Link representation with two driving directions is transformed to modified links with separate links for each driving direction. (b) Link representation of curved road with shape points is transformed to modified links without shape points.

5.2 Bayesian system model

The map-matching problem is formalized as a Bayesian filtering problem. In the proposed approach, the map-matched position is strictly constrained to the road network. The associated state vector is defined as

$$x_t = (k_t, l_t, s_t), \tag{5.1}$$

where $k_t \in \mathrm{K}$ denotes the link, $l_t \in \mathrm{N}_k$ the lane on this link and $s_t \in [0, \ldots, \mathrm{L}] \subset \mathbb{R}_+$ the offset from the start node of the link. The map-matched position m_t in geographical coordinates is uniquely defined by the state vector as

$$m_t = (m_t^{\mathrm{x}}, m_t^{\mathrm{y}}, \psi_t), \tag{5.2}$$

where ψ_t denotes the heading of the associated link. It is computed by propagating the position of the start node along the direction of the link by the length of s_t. Note that the map-matched position is not shifted in lateral direction according to the lane number, because the error of the road centerline position in the map can be larger than the road width. Corrections of the map-matched position by the estimated lane would impose a significant risk of decreasing the accuracy.

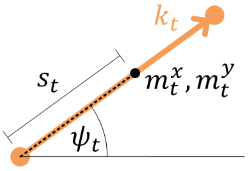


Fig. 5.8 – Definition of a map-matched position $m_t = (m_t^{\mathrm{x}}, m_t^{\mathrm{y}}, \psi_t)$ from $x_t = (k_t, l_t, s_t)$.

It is assumed that the lane is independent of the position along the link, therefore the probability for the states can be factorized as

$$p(x_t) = p(k_t, l_t, s_t) = p(k_t, l_t|s_t)p(s_t) = p(l_t|k_t, s_t)p(k_t|s_t)p(s_t) = p(l_t|k_t)p(k_t|s_t)p(s_t). \tag{5.3}$$

The observations of the system are given by the estimated absolute position z_t of the vehicle and the measurements c_t of the camera,

$$y_t = (z_t, c_t), \tag{5.4}$$

with $z_t = (z_t^{\mathrm{x}}, z_t^{\mathrm{y}}, \theta_t)$ and $c_t = (b_l, b_r)$ as defined in Section 5.1.2.

The dynamical system is defined as a Hidden Markov Model (HMM), which is described by the transition probability $p(x_t|x_{t-1})$ and the observation probability $p(y_t|x_t)$, see section 3.1.

The goal is to estimate the filtering distribution $p(x_t|y_{1:t})$, which gives the solution to the map-matching problem in the Bayesian framework,

$$p(x_t|y_{1:t}) = p(k_t, l_t, s_t|z_{1:t}, c_{1:t}). \tag{5.5}$$

A graphical representation of the dynamic Bayesian network that specifies the independence relations of the HMM is shown in Figure 5.9. Note that the encoded assumptions will be explained during the presentation of the transition and the observation probabilities in the next section.

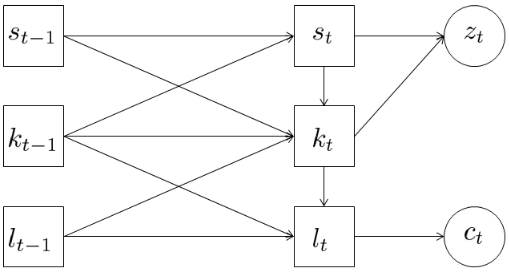


Fig. 5.9 – Hidden Markov Model for the map-matching system at time t

5.3 Particle filter for lane-level map-matching

The posterior probability of the map-matched position is expected to be multi-modal due to the road network topology and the properties of the sensor systems. This assumption is reasonable for map-matching problems at intersections and bifurcations in urban environments. Therefore, particle filtering is a suitable approach for estimating the corresponding filtering distribution.

In the general case, an analytic expression for the optimal importance function is not available and approximation of it implies substantial computational effort. To allow for an efficient implementation, the principle of bootstrap particle filtering is followed and the motion model of the system is employed as the importance function. Thus, the particles are sampled according to $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)})$. Consequently, the particle weighting process involves the evaluation of the observation model as defined in (3.16).

This section describes all steps in sequential importance resampling for the application of particle filtering to the map-matching problem. To design a robust particle filter, appropriate models for particle sampling and weighting are introduced.

5.3.1 Target dynamics

The transition probability $p(x_t|x_{t-1})$ of the HMM is defined by the motion model of the system, which describes the evolution of the state vector along the road network. The motion model of the state vector determines the one-dimensional shift of the offset along the link, the topological decision for the successor link when the map-matched position passes link ends and lane changes.

The transition probability can be factorized using (5.3) and the independence between offset s_t and lane number l_t,

$$\begin{aligned} p(x_t|x_{t-1}) &= p(l_t|l_{t-1}, k_t, k_{t-1}, s_{t-1})\, p(k_t|k_{t-1}, l_{t-1}, s_t, s_{t-1})\, p(s_t|s_{t-1}, l_{t-1}, k_{t-1}) \\ &= p(l_t|l_{t-1}, k_t, k_{t-1})\, p(k_t|k_{t-1}, l_{t-1}, s_t)\, p(s_t|s_{t-1}, k_{t-1}). \end{aligned} \tag{5.6}$$

Transition of offset along link $p(s_t|s_{t-1}, k_{t-1})$

The position along the link is propagated by the estimated distance the vehicle has traveled since the last time step. If the updated offset exceeds the link length, the corrected offset will be applied to the offset along the successor link. Thus, the transition of the offset s_t is defined as

$$s_t = s_{t-1} + \Delta s_t - \zeta_t, \tag{5.7}$$

where $\zeta_t \in \mathbb{R}$ is the correction term when the end of link is passed:

$$\zeta_t = \begin{cases} \mathrm{L}(k_{t-1}) & \text{if } s_t + \Delta s_t > \mathrm{L}(k_{t-1}), \\ \mathrm{L}(\mathrm{pred}_t - (s_t + \Delta s_t) & \text{if } s_t + \Delta s_t < 0, \\ 0 & \text{else,} \end{cases} \tag{5.8}$$

where pred_t defines the link adjacent to k_t that has been previously visited by the particles. If multiple links are traversed in this way, the correction is applied recursively until $s_t + \Delta s_t < \mathrm{L}(k_t)$. The link successor is chosen according to $p(k_t|k_{t-1}, l_{t-1}, s_t, s_{t-1})$ at each link transition, details are described below.

The error model of the offset shift Δs_t considers inaccuracies of the absolute vehicle position as well as map errors. Let u_t denote the estimate of the distance traveled as computed from the absolute positioning estimate. The error of this estimate is assumed to be Gaussian,

$$u_t = v_t \, \Delta t + \epsilon_t, \; \epsilon_t \sim \mathcal{N}(0, \sigma^2_{\mathrm{pos}}). \tag{5.9}$$

Because the transition model determines the particle spreading, robustness against map errors must be ensured. Such inaccuracies are caused by a road segment displacement or a heading adjustment during the map digitalization and generalization process. This map induced uncertainty is added to the estimate of the traveled distance as Gaussian noise. The offset shift is then given as

$$\Delta s_t = u_t + \tau_t, \; \tau_t \sim \mathcal{N}(0, \sigma^2_{\mathrm{map}}). \tag{5.10}$$

After adding the possibly negative noise terms, Δs_t can be negative, leading to an offset shift backwards to the link where the current position emerged. For correct backwards propagation, pred_t maintains the link that has been previously visited.

Note that particles that are propagated according to the presented transition model will populate the road network along links independently of link transitions. This fact avoids undesirable biases in the particle cloud caused only by the road network discretization. Such limitations were observed in prior methods for road-constrained particle filters [52].

Limiting the map-matched position to the road network guarantees that the result is always well-defined. However, in unmapped areas the result is meaningless, when the distance between the map-matched position and the real vehicle position is large. In such off-road cases an exception handling is necessary, which is not supported by the particle filter. The particle filter method needs to be suspended until candidates on the road explain the measurements reasonably well.

Transition of link $p(k_t|k_{t-1}, l_{t-1}, s_t)$

When the offset of the particle shift exceeds the link length, a successor link needs to be determined. This successor should be chosen such that it represents the true vehicle path. Link and lane connectivities from the map are used to sample particles on the successor links.

If only link connectivities are given for the current link, the decision to follow them during particle sampling has the defined probability ν. In this case, all links from the set of successor links have equal probability to be selected,

$$p(k_t = k^j | k_{t-1} = k^i, l_{t-1}) = \frac{1}{M_{\mathrm{linkSucc}}} \quad \text{if } k^j \in \mathrm{succ}(k^i), \; M_{\mathrm{linkSucc}} = \left| \mathrm{succ}(k^i) \right|. \tag{5.11}$$

With a probability of $1-\nu$ the other links that are adjacent to the current link, but not reachable with regular turn maneuvers, are added to the set of possible link successors to obtain robustness against non-conformable turns and missing connectivities in the map. In this case it holds

$$p(k_t = k^j | k_{t-1} = k^i, l_{t-1}) = \frac{1}{M_{\text{linkAdj}}} \quad \text{if } k^j \in \text{adj}(k^i),\ M_{\text{linkAdj}} = \left|\text{adj}(k^i)\right|. \tag{5.12}$$

Examples for the application of (5.11) and (5.12) at an intersection are shown in Figure 5.10.

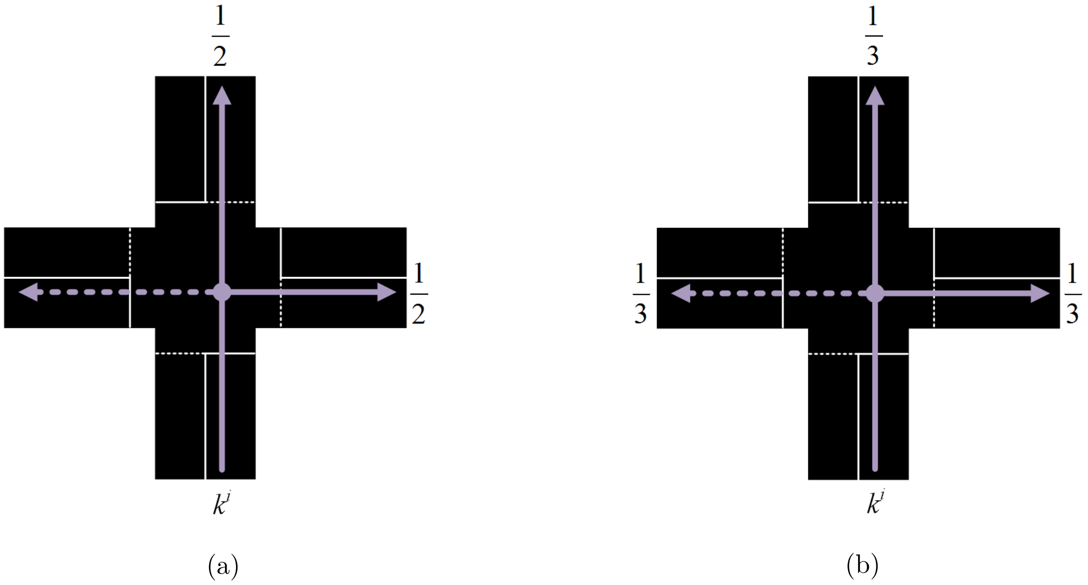


Fig. 5.10 – Examples for possible link transitions from $k_{t-1} = k^i$ and their probabilities according to connected links (purple) and an adjacent link (purple dashed). (a) The two possible successor links connected to k^i have equal probability of $\frac{1}{2}$ to be selected if the decision to follow link connectivities is chosen with probability ν. (b) If link connectivities are not applied, all adjacent links obtain the same probability of $\frac{1}{3}$.

If lane connectivities are given for the current link, the decision to follow them during particle sampling has the defined probability γ. In this case, all links from the set of successor links connected to the current lane have equal probability to be selected,

$$\begin{aligned} p(k_t = k^j | k_{t-1} = k^i, l_{t-1} = l^m) = \frac{1}{M_{\text{laneSucc}}} \quad & \text{if } k^j \in \left\{k : (k, \cdot) \in \text{cnt}(k^i, l^m)\right\}, \\ & M_{\text{laneSucc}} = \left|\left\{k : (k, \cdot) \in \text{cnt}(k^i, l^m)\right\}\right|. \end{aligned} \tag{5.13}$$

With a probability of $1-\gamma$ lane connections are not followed and only link connections are considered. This case is handled according to (5.11) and (5.12). Figure 5.11 illustrates examples at an intersection with lane connectivities.

In case the link history is traversed backwards, the predecessor of k_{t-1} is chosen,

$$p(k_t = \text{pred}_t\, | k_{t-1}) = 1, \text{ if } s_t + \Delta s_t < 0. \tag{5.14}$$

Note that the planned route is not considered as a link successor criterion. This assumption has been made to provide a fast adaption of the map-matching system when the traveled path deviates from the planned route. Otherwise the particles would stick to the planned route and bias the map-matcher towards route links under the assumption that the driver always follows the route.

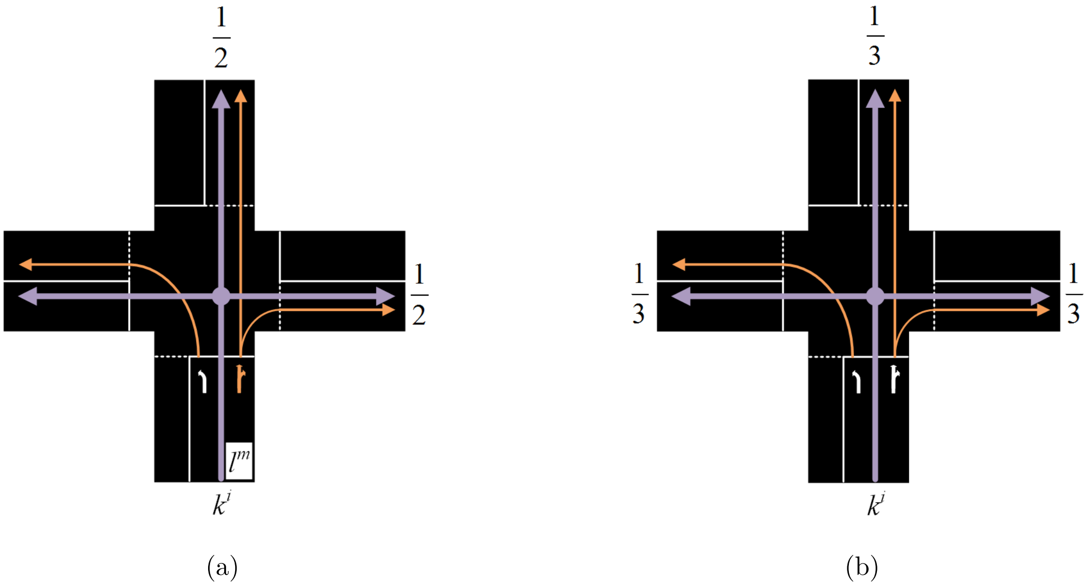


Fig. 5.11 – Examples for possible link transitions from $k_{t-1} = k^i$, $l_{t-1} = l^m$ and their probabilities according to connected links (purple) and lanes (orange). (a) The two possible successor links connected to k^i, l^m have equal probability of $\frac{1}{2}$ to be selected if the decision to follow lane connectivities is chosen with probability γ. (b) If lane connectivities are not applied, but link connectivities are followed, all successor links obtain the same probability of $\frac{1}{3}$.

Transition of lane $p(l_t|l_{t-1}, k_t, k_{t-1})$

Lane transitions in the motion model enable the particles to follow performed lane changes. If the particle remains on the same link after the link transition is applied, lane transition events are triggered by camera measurements. When a lane change event is detected by the camera at the current time step, the lane state will be shifted according to the direction of lane change with probability α. Taking into account false positive detections, the lane transition does not follow the lane change event with probability $1 - \alpha$,

$$\left.\begin{aligned} p(l_t = l^m + a_t | l_{t-1} = l^m, k_t = k^i, k_{t-1} = k^i) &= \alpha \\ p(l_t = l^m | l_{t-1} = l^m, k_t = k^i, k_{t-1} = k^i) &= 1 - \alpha \end{aligned}\right. \qquad \text{if } a_t \in \{-1, 1\}\,, \tag{5.15}$$

where $l^m \in [1, \ldots, \mathcal{L}(k^i)]$.

Lane changes can be missed by the camera when lane markings are not correctly detected or lane markings on the road are missing. To handle such false negative cases, a lane change event $\tilde{a}_t$ is triggered with probability ρ if no lane change event is detected by the camera. Left and right lane change events are assumed to have equal probability, $p(\tilde{a}_t = -1) = p(\tilde{a}_t = 1) = 0.5$. In this case, the lane transition as defined in (5.15) is applied with probability $\alpha = 1$.

The final lane number is bounded to prevent it from exceeding the defined range of lane numbers for the respective link,

$$l_t = \min\left(\max(l_{t-1} + a_t, 1), \mathcal{L}(k_t)\right). \tag{5.16}$$

At link transitions, lane changes follow the given lane connectivities between the previous link k_{t-1} and the updated link k_t after link transition. If multiple lanes are connected, each lane is assigned equal probability,

$$p(l_t = l^n | l_{t-1} = l^m, k_t = k^j, k_{t-1} = k^i) = \frac{1}{N_{\text{succ}}}, \quad \text{for all } k^i = \text{pred}_t \text{ and } (k^j, l^n) \in \text{cnt}(k^i, l^m), \tag{5.17}$$

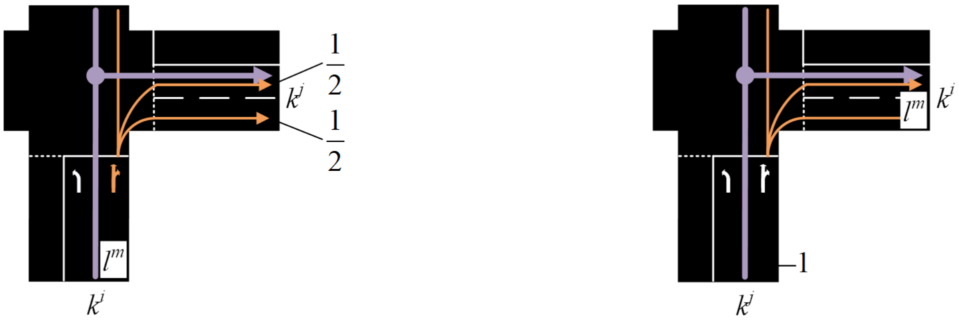
where N_{succ} is the number of successor lanes on link k^j for given lane l^m,
$N_{\text{succ}} = \left|\{l^n : (k^j, l^n) \in \text{cnt}(k^i, l^m)\}\right|$.

In case of a backward link transition $k_t = \text{pred}_t$, each predecessor lane, as given in the lane connectivities, is assigned equal probability,

$$p(l_t = l^n | l_{t-1} = l^m, k_t = k^j, k_{t-1} = k^i) = \frac{1}{N_{\text{pred}}}, \quad \text{for all } k^j = \text{pred}_t \text{ and } (k^i, l^m) \in \text{cnt}(k^j, l^n), \tag{5.18}$$

where N_{pred} is the number of predecessor lanes on link k^j for given lane l^m,
$N_{\text{pred}} = \left|\{l^n : (k^i, l^m) \in \text{cnt}(k^j, l^n)\}\right|$.

The lane transition probabilities in case of a (backwards) link transition are visualized in Figure 5.12.



(a) (b)

Fig. 5.12 – Examples for possible lane transitions from $l_{t-1} = l^m$ on $k_{t-1} = k^i$ according to the updated link $k_t = k^j$ and connected links (purple) and lanes (orange). (a) The two connected lanes from l^m have equal probability of $\frac{1}{2}$ to be selected. (b) Lane transition after backwards link transition. Only one lane on $k_t = k^j$ is connected to lane $l_{t-1} = l^m$ from $k_t = k^i$.

5.3.2 Observation model

Since the transition probability defines the particle sampling, it follows from (3.16) that the particle weights include the evaluation of the likelihood,

$$w_t^{(i)} \propto w_{t-1}^{(i)} \, p(y_t | x_t^{(i)}). \tag{5.19}$$

The likelihood is determined by the observation model, which includes two sources of measurements, the absolute vehicle positioning and the camera system,

$$p(y_t | x_t) = p(z_t, c_t | l_t, k_t, s_t). \tag{5.20}$$

It can be assumed that these two parts of the observation model can be treated separately, because of two reasons: First, the error of low-cost absolute positioning systems can be in the magnitude of the road width, which prevents using this measurement for estimating the lateral position of the vehicle on the road. Second, the lane markings detection from the camera is not affected by the offset along the link and therefore also independent from road map related parameters. These assumptions lead to the factorization of the likelihood,

$$\begin{aligned} p(y_t|x_t) &= p(z_t, c_t | l_t, k_t, s_t) = p(z_t | l_t, k_t, s_t) \, p(c_t | l_t, k_t, s_t) \\ &= p(z_t | k_t, s_t) p(c_t | l_t, k_t). \end{aligned} \tag{5.21}$$

Observations from absolute positioning system $p(z_t|k_t, s_t)$

The estimated absolute position contains the geographical position $(z_t^{\mathrm{x}}, z_t^{\mathrm{y}})$ and the absolute heading θ_t. These measurements are used for comparison with the map-matched position m_t from the particle states vector. Two criteria are applied: First, proximity of position and second, heading similarity.

For the proximity term of the observation model, the distance between the map-matched position and the estimated absolute position is defined as

$$d_z = d_z(z_t, m_t),\ d_z : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+, \tag{5.22}$$

where the metric $d_z(\cdot,\cdot)$ is the great circle distance between two geographical coordinates, Figure 5.13. The weight associated with the proximity between the map-matched position and the absolute position is defined as

$$w_z = \exp(\lambda\ d_z). \tag{5.23}$$

with $\lambda < 0$. The exponential function guarantees that larger deviations between the map-matched position and the absolute position are weighted over-proportionally, Figure 5.14a.

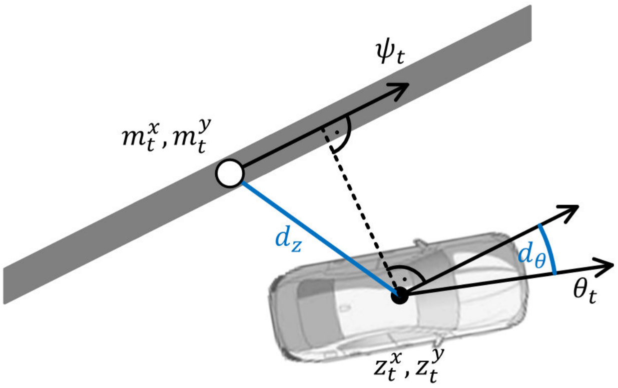


Fig. 5.13 – Distance metrics for proximity and heading difference

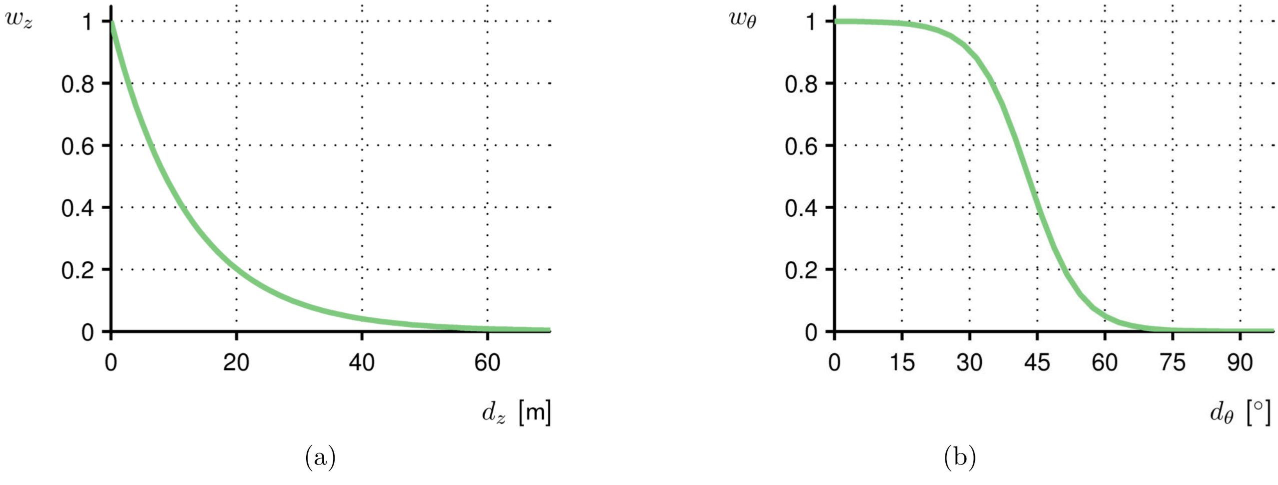


Fig. 5.14 – Examples of weighting functions. (a) Proximity weights from (5.23) with $\lambda = -0.08$. (b) Heading weights from (5.25) with $\beta = 10$ and $\mu = 7.5$.

The heading term is similarly based on a distance measure between the headings of the link of the map-matched position and the estimated absolute heading,

$$d_\theta = d_\theta(\theta_t, \psi_t),\ d_\theta : [-\pi; \pi] \times [-\pi; \pi] \to [0; \pi]\,, \tag{5.24}$$

where the metric $d_\theta(\cdot, \cdot)$ is defined as the "normalized difference" [77] between two angles, Figure 5.13. The weight associated with the similarity of headings between the map-matched position and the absolute position is defined as

$$w_\theta = \frac{1}{1 + \exp(\beta d_\theta - \mu)}, \tag{5.25}$$

with $\beta, \mu > 0$. The sigmoidal shape of this weight function models the property of small angle deviations as not being meaningful for the discrimination of map-matched candidates, Figure 5.14b. The generalized link geometry at intersections and bifurcations often prevents finding map-matched candidates with a high heading similarity. Thus, the associated heading criterion contributes to the particle weights variation only when a certain threshold of the heading difference is reached. For small deviations, particles weights are almost equal to 1. If the heading difference reaches a certain threshold, the weights are close to zero to increase the probability of the particles to be removed during resampling.

Both the proximity and the heading criteria should be met for a particle to obtain a high weight. Thus, the likelihood $p(z_t|k_t, s_t)$ is defined by the multiplication of the proximity and the heading weight,

$$p(z_t|k_t, s_t) \propto w_z w_\theta. \tag{5.26}$$

Observations from camera system $p(c_t|l_t, k_t)$

The likelihood of the lane markings detection for a given lane is based on a model that associates lanes with expected types of left and right lane markings. The proposed model is based on German-style lane markings, where the outer markings of outer lanes are mostly solid and the inner markings of inner lanes are mostly broken, Figure 5.15. For other countries, this model needs to be revised. This model is suited to distinguish lanes on multi-lane roads with less than four lanes. For four lanes and more, there is more than one inner lane that explains observations of broken lines for the left and right lane markings equally well. Since the focus of this work lies in urban scenarios, this model is sufficient.

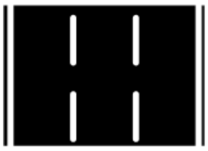


Fig. 5.15 – Model for expected lane markings on a three-lanes road

To determine the likelihood for each lane, the detected lane marking types $c_t = (b_t^\mathrm{l}, b_t^\mathrm{r})$ are compared with the lane l_t according to the lane markings model. To quantify this evaluation, different levels of likelihood are introduced, υ is rather uncertain, χ is neutral, ξ rather certain and κ quite sure about the likelihood of the measurement for the given lane. It holds by definition $0 < \upsilon < \chi < \xi < \kappa < 1$. The computation of the lane weights w_l for all combinations of lanes l_t and detected lane marking types $(b_t^\mathrm{l}, b_t^\mathrm{r})$ summarized in Table 5.1. It is defined that lane markings contribute only to the lane weight if the current link k_t has more than one lane, otherwise the neutral value χ is applied. The only exception is given at a single-lane link, where solid lane markings on both sides are observed. In this case the value ξ is assigned to that lane, in all other cases υ is assigned to a single-lane link.

left lane marking type		$l_t \in \mathrm{N}_{k_t}$			right lane marking type
b_t^{l}		$\mathcal{L}(k_t)$	$2, \ldots, \mathcal{L}(k_t) - 1$	1	b_t^{r}
			w_l		
solid		χ	χ	χ	solid
		κ	υ	υ	broken
		ξ	χ	χ	unknown
broken		υ	υ	κ	solid
	$\mathcal{L}(k_t) = 2$	χ	-	χ	broken
	$\mathcal{L}(k_t) > 2$	υ	κ	υ	
		χ	χ	ξ	unknown
unknown		χ	χ	ξ	solid
		ξ	ξ	χ	broken
		χ	χ	χ	unknown

Tab. 5.1 – Lane weights w_l according to different levels of likelihood denoted by υ, χ, ξ, and κ for links with more than one lane. If k_t corresponds to a two-lanes road, the middle column $l_t \in \{2, \ldots, \mathcal{L}(k_t) - 1\}$ is left out, unless otherwise stated.

Some examples from the definitions of the lane weight w_l illustrate the motivation of this observation model:

1. If the left lane marking is detected as solid and the right as broken, then the leftmost lane obtains a high weight, $p(c_t = (\text{solid}, \text{broken}) | l_t = \mathcal{L}(k_t), k_t) \propto w_l = \kappa$.
2. If the left and right lane marking are detected as broken, then this observation is meaningless for either lane on a two-lanes road and therefore both obtain a neutral weight, $p(c_t = (\text{broken}, \text{broken}) | l_t \in \{1, 2\}, k_t) \propto w_l = \chi$, if $\mathcal{L}(k_t) = 2$.
3. If the left lane marking is detected as unknown and the right as broken, then it is rather certain that this observation corresponds to either the left or the middle lane on a three-lanes road, $p(c_t = (\text{unknown}, \text{broken}) | l_t \in \{2, 3\}, k_t) \propto w_l = \xi$, if $\mathcal{L}(k_t) = 3$.
4. If the left lane marking is detected as solid and the right as broken, a single-lane road obtains a neutral weight, $p(c_t = (\text{solid}, \text{broken}) | l_t, k_t) \propto w_l = \chi$, if $\mathcal{L}(k_t) = 2$, whereas the leftmost lane on a three-lanes road obtains a high weight, $p(c_t = (\text{solid}, \text{broken}) | l_t = 3, k_t) \propto w_l = \kappa$, if $\mathcal{L}(k_t) = 3$.

The composition of the link weight related to the absolute position observation and the lane weight related to the lane markings observations determines the total weight of the particle. These two models are combined by a logical AND-conjunction and therefore their weights are multiplied to obtain the final particle weight,

$$w_t^{(i)} \propto w_{t-1}^{(i)} \, p(z_t | k_t^{(i)}, s_t^{(i)}) \, p(c_t | l_t^{(i)}, k_t^{(i)}) = w_{t-1}^{(i)} w_z^{(i)} \, w_\theta^{(i)} \, w_l^{(i)}. \tag{5.27}$$

5.3.3 Resampling

The same resampling strategy as described for airborne vehicle tracking in Section 4.3.3 is applied to mitigate particle degeneracy. After resampling, all particles have uniform weights and the particle weighting process in the subsequent time step is reduced to the evaluation of the likelihood,

$$w_{t+1}^{(i)} \propto p(y_{t+1} | x_{t+1}^{(i)}). \tag{5.28}$$

5.4 Parameter modeling and sensitivity analysis

The motion model and the observation model contain several parameters to control the variance of the particle spreading and weighting, respectively. The optimal choice of these parameters is related to models that balance the trade-off between robustness and precision of the estimate. The layout of the chosen parameters is explained and a parameter sensitivity analysis is presented for the complex parts of the system models.

5.4.1 Parameters of the motion model

Setting the probability parameters of the motion model depends on the reliability of map information, position estimates and camera detection of lane change events. The motion model needs to be restrictive enough to spread the particles efficiently, but also fairly robust to cover links and lanes sufficiently with particles.

The spreading of each state vector variable is controlled by an individual parameter. The offset s_t along the link is propagated by the offset shift Δs_t, which is normally distributed with the mean located at the estimated distance traveled and the variance defined by $\sigma_{\text{pos}}^2 + \sigma_{\text{map}}^2$ according to (5.9) and (5.10). The absolute position estimate noise is assumed to be constant as a property of the absolute positioning system. It is set to a fixed value in this work to evaluate the particle filter method independently of the positioning quality, which is affected by the individual sensor configuration and positioning method. The standard deviation of this noise is set to $\sigma_{\text{pos}} = 2$ meters, which represents the quality of a low-cost system.

The standard deviation of the map error controls the extend to which links are explored by the particle cloud. On the one hand, this noise parameter should allow the particles to capture portions of the road network that contain possible hypotheses for the map-matched position. On the other hand, the spreading of the particles should be small to minimize the number of particles that are required to cover links with a sufficient density. Scenarios with different complexity are selected to analyze parameter variations of σ_{map} systematically, Figure 5.16. For this experiment, all successor links are assigned equal probability and single-lane links are assumed. The particles at the previous time step are set to the same position and the mean of the distance traveled as estimated by the absolute positioning system is set to one meter, $u_t \sim \mathcal{N}(1, \sigma_{\text{pos}}^2)$. Each scenario is computed with three different values of the noise standard deviation, $\sigma_{\text{map}} \in \{1, 5, 10\}$ meters.

When the vehicle drives along a straight road, map errors in the road segment heading lead to a deviation between the true length of the road segment and the mapped link. Thus, the projection of the distance traveled on the mapped link needs to be adapted. If the true road and the mapped link run in parallel, the positioning noise alone would determine the particle spreading sufficiently. Since this is rarely the case in real world scenarios, the amount of spreading controls the robustness against heading errors in the map. Such headings errors are especially apparent at bifurcations, Figure 5.16b. Also, the uncertainty of the localization of the road split point in the map needs to be considered. When the split point is mapped behind the true location in driving direction, the complete particle cloud could already reside on the wrong link once the vehicle arrives at the split point. To increase the chance that particles arrive at the split point at the same time as the vehicle reaches the true split point, the diffusion level of the particles must correspond to the expected map generalization error. At T-junctions, a similar strategy applies: The particles need to populate a reasonable portion of the links before and after the intersection point to avoid a premature exclusion of the alternative link, Figure 5.16c. However, when the spreading increases, the density of the particles on the individual links decreases for the same number of particles. The noise level of the spreading should be sufficiently low to avoid sparse sampling at complex intersections with multiple successor links, Figure 5.16d. Otherwise, the precision of the particle filter estimate is negatively affected.

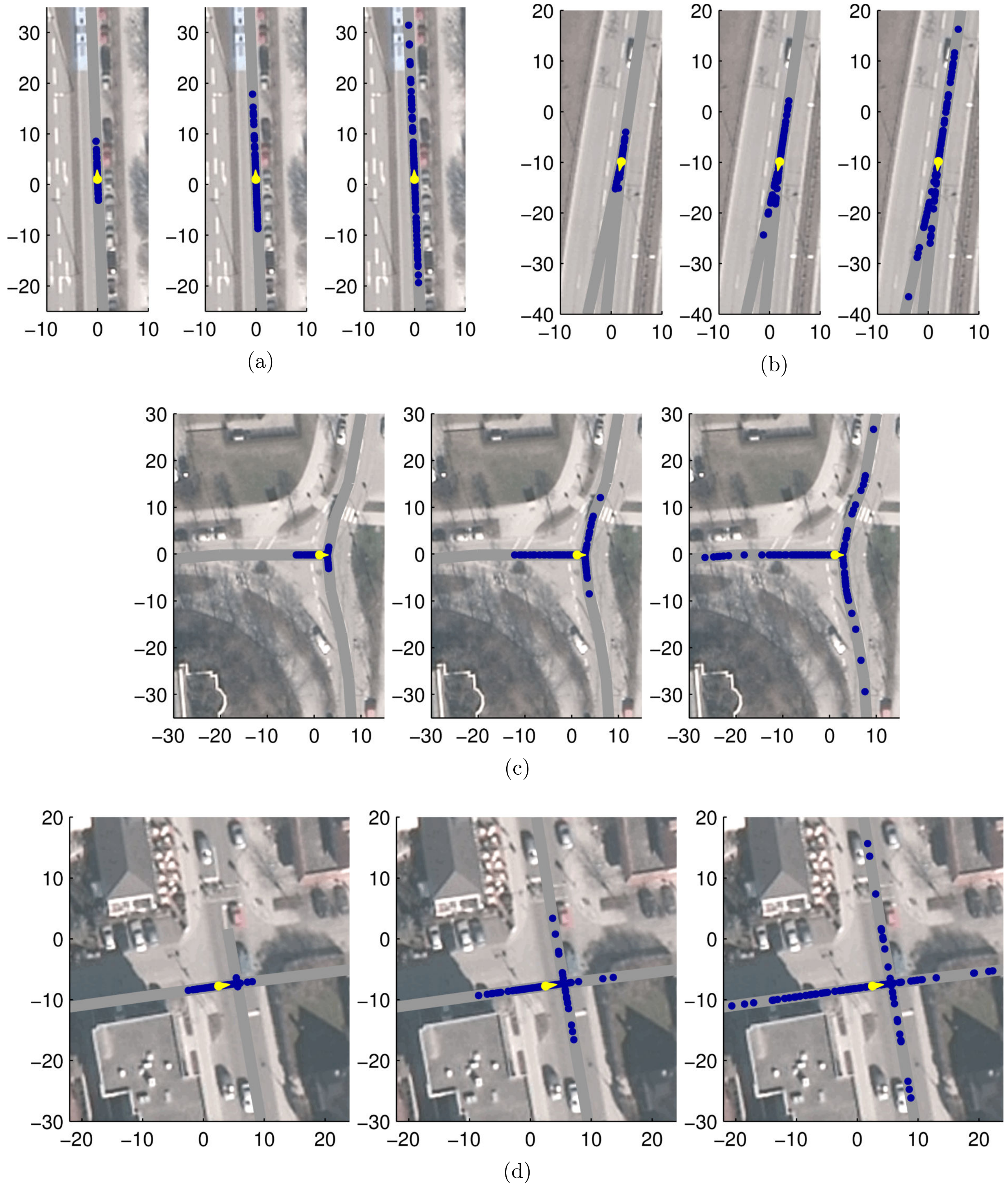


Fig. 5.16 – Particle spreading at different scenarios, (a) straight road, (b) bifurcation, (c) T-junction and (d) intersection. Blue dots show particle positions for 100 particles. The yellow marker indicates the mean position of the shifted particle cloud and the direction of travel. The particle propagation is evaluated with $\sigma_{\text{map}} = 1$ (left), 5 (middle) and 10 (right) meters.

Now the "optimal" value of σ_{map} needs to meet robustness and precision criteria in these different scenarios. From the comparison in Figure 5.16, it shows that a small value of $\sigma_{\text{map}} = 1$ meter restricts the particle cloud to possible hypotheses, which are only a few meters around the intersection or split point of multiple branching links. Especially in the ambiguous scenario at a road bifurcation, the spreading is too small to robustly handle split point displacements in the map, Figure 5.16b.

Setting the noise parameter to 10 meters increases the spreading such that the particle cloud explores the links to an extent that tolerates map errors in complex or ambiguous scenarios, Figure 5.16c. Although such noise level would be a preferable choice, the computational effort for updating the required amount of particles is infeasible for an onboard processing unit. As the comparison shows, a number of 100 particles is hardly sufficient to cover the links densely enough for reliable estimates, Figure 5.16d. Therefore, a value of 5 meters for σ_{map} is suggested to balance efficiency and robustness of the estimate.

At link transitions, the link and lane successor probabilities define the uncertainty about map information and driving maneuvers. It is assumed that the map information defines the correct connection between transition links because common routing algorithms are based on this information. Thus, the related parameter ν is set to 0.9. If lane connectivities are given, the suggested map information is trusted to a high level. However, since no commercial use of these data is known, the data is assumed to be not entirely reliable and the probability parameter γ is set to 0.7. It is further believed that turn maneuvers from or to lanes that are not according to the traffic regulations occur more frequently than entering forbidden links.

As a consequence of detected lane changes by the camera, the particles follow this lane change with a probability of α. Again, since the camera system is employed for a lane keeping assistant functionality, this event detection is trusted with $\alpha = 0.9$. The detection of a lane change event requires the stable detection of lane markings. Since situations frequently occur where lane markings are not existent or the detection fails, random lane changes are applied reasonable often to compensate missed lane changes. With a probability of $\rho = 0.2$, a lane change event is triggered.

The proposed models are tested using the values summarized in Table 5.2.

transition model	probability parameter	value
(5.9)	σ_{pos}	2 [m]
(5.10)	σ_{map}	5 [m]
(5.11), (5.12)	ν	0.7
(5.13)	γ	0.9
(5.15)	α	0.9
	ρ	0.2

Tab. 5.2 – Parameter values of the motion model

5.4.2 Parameters of the observation model

The weighting parameters within the observation models are crucial for the overall performance, because they determine the characteristics of the particle weighting. A preferable weight distribution allows to discriminate particles that represent correct hypotheses for the map-matched position from mismatched candidates while compensating measurement and map inaccuracies.

The hypotheses weights include the proximity criterion (5.23), which assigns a high value to particles close to the estimated absolute position. The factor λ takes into account uncertainty about absolute positioning and map displacement. It provides a relaxation of the assumption that the particle closest to the absolute position represents the most probable hypothesis for the map-matched position. The higher λ, the more robust the observation model is against positioning or map displacement errors. This comes at the cost of precision, because close hypotheses obtain nearly similar weights. This impact is visualized in Figure 5.17. For small values of λ, the curve rapidly drops and assigns weights close to zero to particles that are more than 20 m away from the estimated absolute position. For

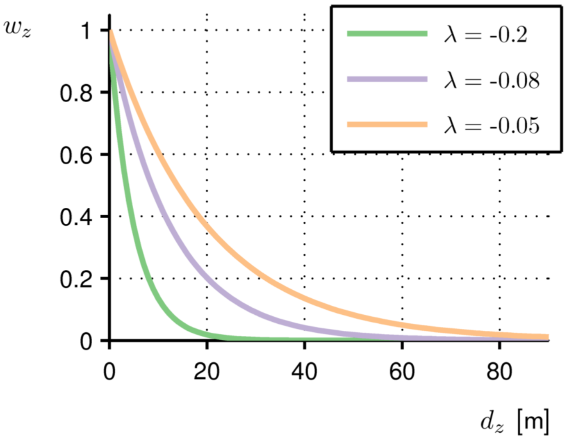


Fig. 5.17 – Proximity weighting function for variation of λ from (5.23)

larger λ, more weight is given to distant particles. At the same deviation of 20 m, a weight of 0.4 is assigned for $\lambda = -0.05$.

The effect of parameter variation is demonstrated on real road examples. A uniform, densely sampled particle cloud is simulated to evaluate continuous variation of the weighting function, Figure 5.18. On a single road segment, the most likely position on the road is uniquely determined by the closest point on that road, Figure 5.18a. Therefore, the variation of weights is only applied in longitudinal direction to the respective road. Similarly to the motion model noise, it models the uncertainty of absolute positioning and link position. To avoid that sampled particles fall in regions of low likelihood, the noise of the proximity weighting should cover a similar range. At bifurcations, map errors are apparent, Figure 5.18b. A strict weighting function with low λ can let particles on the alternative road segment die out too early. In the opposite case, a high value for λ would lead to inefficient reproduction of particles once the hypothesis for the road branch can be decided unambiguously after the vehicle has passed the bifurcation. At T-junctions and intersections, Figures 5.18c, 5.18d, the link position in the map deviates from the absolute vehicle position by a few meters. Here, the weighting function needs to balance robustness against the lateral position offset from the road and preciseness of the map-matched estimate. From these examples it can be deduced that $\lambda = -0.08$ is a good factor to meet requirements of both robustness and precision. This value will tolerate deviations of up to 60 m before setting particle weights close to zero.

The second weighting term is defined by the heading deviation between the estimated absolute position and the particle. In (5.25), the variation is defined by the parameters μ and β, which have different effects on the weighting function. While μ defines the shift along the x-axis, β determines the gradient of weights decrease. Within the given parameter ranges for μ and β as shown in Figure 5.19, only angle differences between 0 and 90° obtain weights well above zero. Large values of μ lead to an immediate discrepancy of particle weights, while small values of μ result in similar weights for small heading differences. An intermediate values of $\mu = 7.5$ assigns similar weights to heading differences below 15°, before the weighting function starts to differentiate between particle heading angles. For the parameter β, larger values lead to a rapid drop of particle weights, while small values of β induce a moderate decrease. For $\mu = 7.5$ and $\beta = 10$, the strongest weights discrepancy happens approximately between 30° and 60° heading difference.

To examine the effect of parameter variation in practice, the heading difference weighting function is evaluated at bifurcation and intersection scenarios. At the bifurcation scenario, the road splits into two branching segments, which are characterized by slowly diverging headings, Figure 5.20. The map representation models this scenario as one road segment, which matches the real course of road well and the second segment with a significantly different heading at the splitting point. If the vehicle takes the right turn in driving direction, its real heading will not match with the correct road segment in the

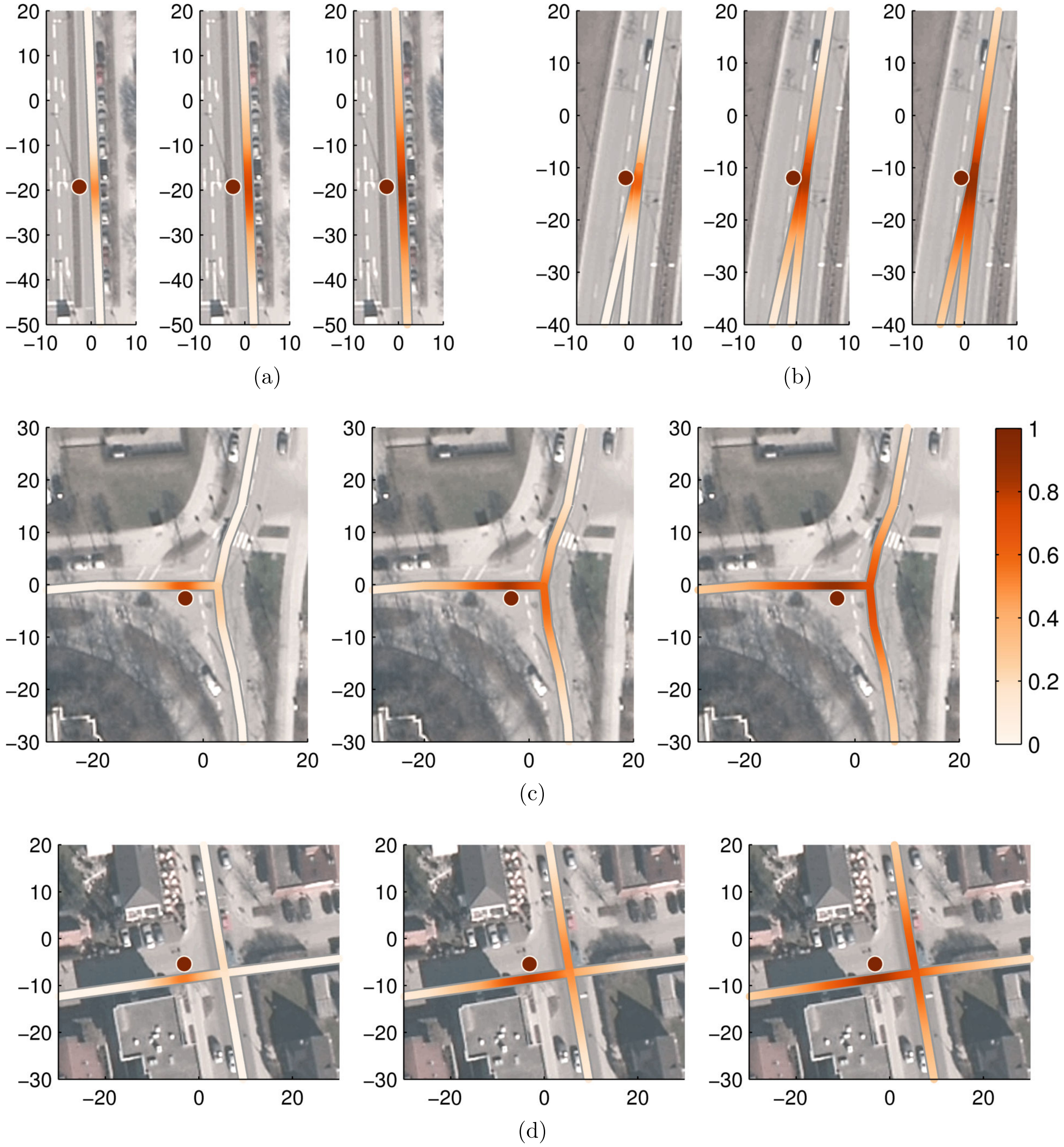


Fig. 5.18 – Link proximity weighting at different scenarios, (a) straight road, (b) bifurcation, (c) T-junction and (d) intersection. Proximity weighting is evaluated with $\lambda = -0.2$ (left), -0.08 (middle) and -0.05 (right).

map. This scenario is a typical representative of diverging roads, where the heading resolution of the map is heavily generalized. Such situations are very critical for correct map-matching, because a high risk exists that the ambiguity is biased towards the link that initially matches better with the absolute vehicle heading. Therefore, the observation model needs to be relaxed to avoid the interpretation of the heading difference as the deciding measure for particle weighting.

First, the parameter μ is fixed to 7.5 and β is varied, Figure 5.20a. For $\beta = 18$, a significant weight decrease for small heading deviations penalizes hypotheses on the exit branch of the bifurcation. Since the link heading from the map does not exactly represent the diverging characteristic of the

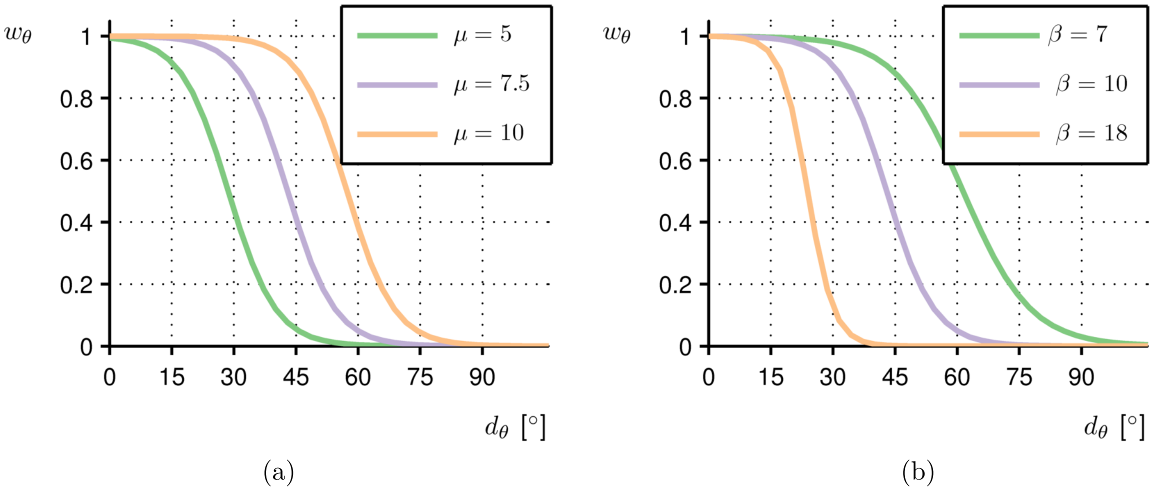


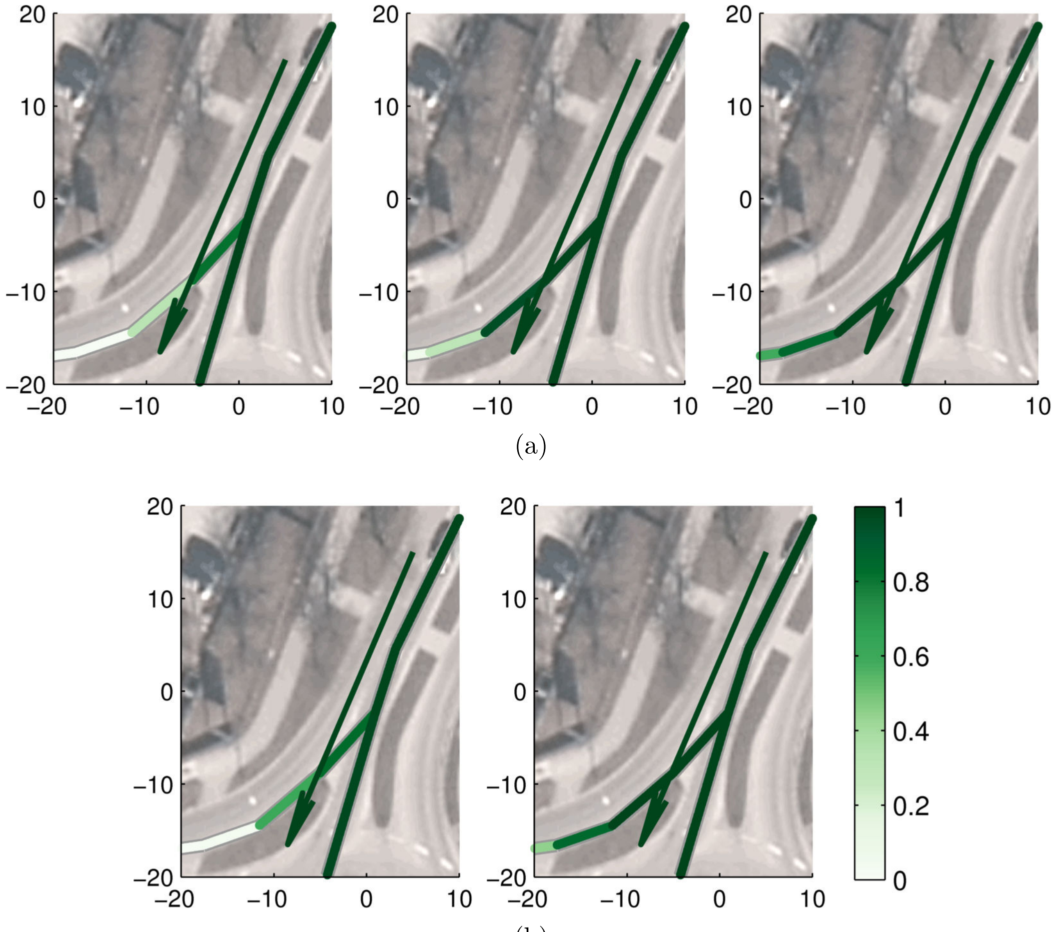
Fig. 5.19 – Heading weighting function for variation of μ and β from (5.25). (a) Parameter $\beta = 10$ is fixed, (b) parameter $\mu = 7.5$ is fixed.

real course of the road, an immediate decline of hypotheses weights with small heading difference would pose a high risk of losing particles on the alternative branch too early . For $\beta = 10$, this ambiguity is handled more appropriately. This weighting function does not distinguish hypotheses at the early stage of branching roads by their heading difference. However, once the link heading clearly deviates, the according hypotheses slowly decrease in weight. For $\beta = 7$, larger heading differences are required before the weight significantly decreases. At a road bifurcation, this parameter value would inefficiently maintain particles, which have low probability of belonging to the correct map-matched road segment.

Next, the effect of variation of μ is studied, Figure 5.20b. When β is fixed to 10, a value of $\mu = 5$ prematurely discriminates hypotheses on the exit branch. For $\mu = 10$, the branching roads are equally rated, but neither branch dominates until they clearly diverge. Thus, the parameter combination of $\beta = 10, \mu = 7.5$ is the preferable choice, because this configuration tolerates initial map heading errors at bifurcations without keeping particles unnecessarily alive when their heading is clearly out of a feasible range.

Similar characteristics of parameter variation effects can be observed at an intersection scenario, Figure 5.21. At a T-junction, two turning road segments are generalized into links in the map, which have a discontinuous heading transition at the intersection point. Thus, within in the intersection area, neither link matches with the real absolute vehicle position. Starting with a fixed value of $\mu = 7.5$ and $\beta = 18$ obviously no distinction between successor links is possible, Figure 5.21a. For $\beta = 10$, both the predecessor link and the most likely successor link are assigned comparable weights. This weighting property matches the expected heading weights, because at this stage, the situation is undecidable considering only heading difference information. For $\beta = 7$, also the opposite successor link obtains significant weights, although it is very unlikely that it contains the correct map-matched position. In case of setting $\beta = 10$ and varying μ, the observation model becomes too strict for $\mu = 5$, Figure 5.21b. For $\mu = 10$, the weights distribution is acceptable, but the opposite successor link obtains a higher weight than for $\mu = 7.5$. Therefore, just as in the bifurcation scenario, $\beta = 10, \mu = 7.5$ seems to be the optimal parameter configuration in terms of efficiency and robustness against map generalization errors.

The lane hypotheses are weighted by comparing the observed lane marking types from the camera with the expected lane markings from the lanes model (5.1). Different levels of likelihood can be assigned according to the matching result. These levels are defined by the parameters υ, χ, ξ and κ. The lowest value υ is set to 0.2. It is applied, when both the left and the right lane markings have



(a)

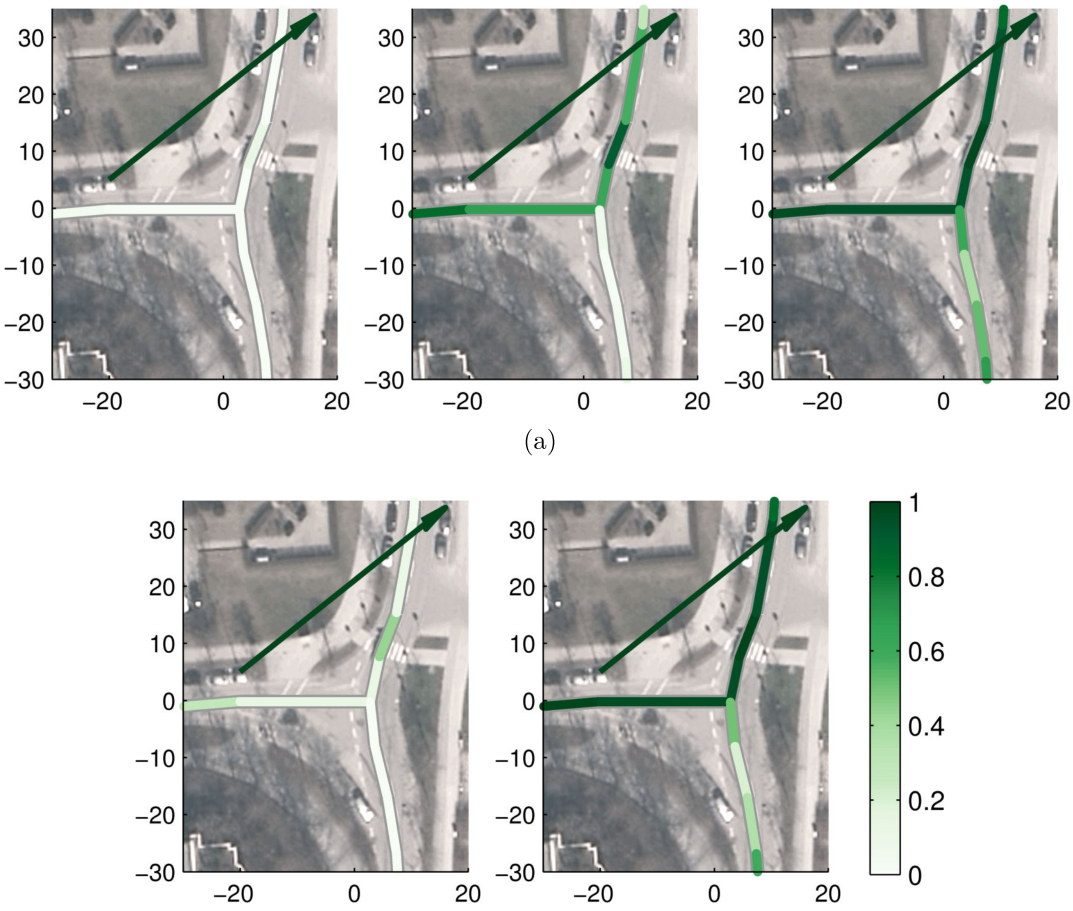
(b)

Fig. 5.20 – Particle heading weighting at a bifurcation for (a) $\beta \in \{18, 10, 7\}$, $\mu = 7.5$ (from left to right) and (b) $\beta = 10$, $\mu = 5$ (left), $\mu = 10$ (right)

been classified as broken or solid and the expected lane markings from the hypothesized lane do not match with observed ones. If they match, then the highest likelihood value of $\kappa = 1$ is assigned to the lane hypothesis. If only the lane marking on one side is visible or successfully classified, then both the probability values for the matching case is lowered, $\xi = 0.8$. In the non-matching case if the expected lane marking on the classified side does not match with the observation, a neutral value of $\chi = 0.5$ is assigned. This value is also applied if solid lane markings are observed on both sides or lane markings on both sides cannot be classified, regardless of the lane hypothesis.

Since the resulting lane weight is multiplied by the link weight to obtain the particle weight, the lane weight implicitly controls the particle replication after resampling. Thus, the lane weight probability parameters are all above zero to avoid particle exclusion caused by the result of lane matching.

Typical results of lane weighting and the influence on link hypothesis evaluation are analyzed at example scenarios, Figure 5.22. When the vehicle approaches a road bifurcation, the observed lane marking types indicate the correct lane matching. For left and right observed broken lane markings, the middle lane(s) are more likely than the outer lanes, Figure 5.22a. If the lane markings were detected correctly by the camera, a single-lane link is a very unlikely hypothesis for the map-matched position. However, for robustness against false detections it obtains a neutral weight. Observations of a left broken line and a right solid line indicate a strong hypothesis for the rightmost lane, Figure 5.22b. Again, the single-lane link obtains a neutral weight for robustness reasons. If lane markings on both



(a)

(b)

Fig. 5.21 – Particle heading weighting at a T-junction for (a) $\beta \in \{18, 10, 7\}$, $\mu = 7.5$ (from left to right) and (b) $\beta = 10$, $\mu = 5$ (left), $\mu = 10$ (right)

sides are detected as solid, the lane hypothesis on the single-lane link is higher than for the multiple-lane links, Figure 5.22c.

When the vehicle drives on a road that merges with another one, lane weights support the decision for the correct lane and link hypothesis. If the left and right lane markings are detected as broken, both lane hypotheses on the two-lanes link obtain a lower weight than the middle lane on the three-lanes link. This weighting strategy biases the hypothesis towards the successor link, which increases the probability of correct map-matching if the camera measurements are correct, Figure 5.22d. Otherwise, the neutral lane weights on the two-lanes link facilitates robustness against false classifications. If the lane markings model represents the true road environment and the observed lane marking types are classified correctly, the correct lane can be identified with high confidence. This is the case as simulated in Figure 5.22e, where a solid lane marking on the left and a broken lane marking on the right indicate that the rightmost lane is traveled. However, the lane weights interpretation is critical, if the joining roads were separated by a solid line, which should prevent drivers from immediate lane changes. Now it is assumed that the vehicle enters the successor link on the middle lane, which is separated to the left by a solid line. In this case, the lane model does not match with reality and a left solid line and a right broken line can corrupt the map-matched result, because the middle lane obtains a low weight, Figure 5.22f. Therefore, a small probability larger than zero must be maintained on the alternative, but unlikely lane hypotheses.

To conclude, the power of the observation models and from these the derived particle weighting functions lies in their ability to robustly handle heading inconsistencies between map, observations

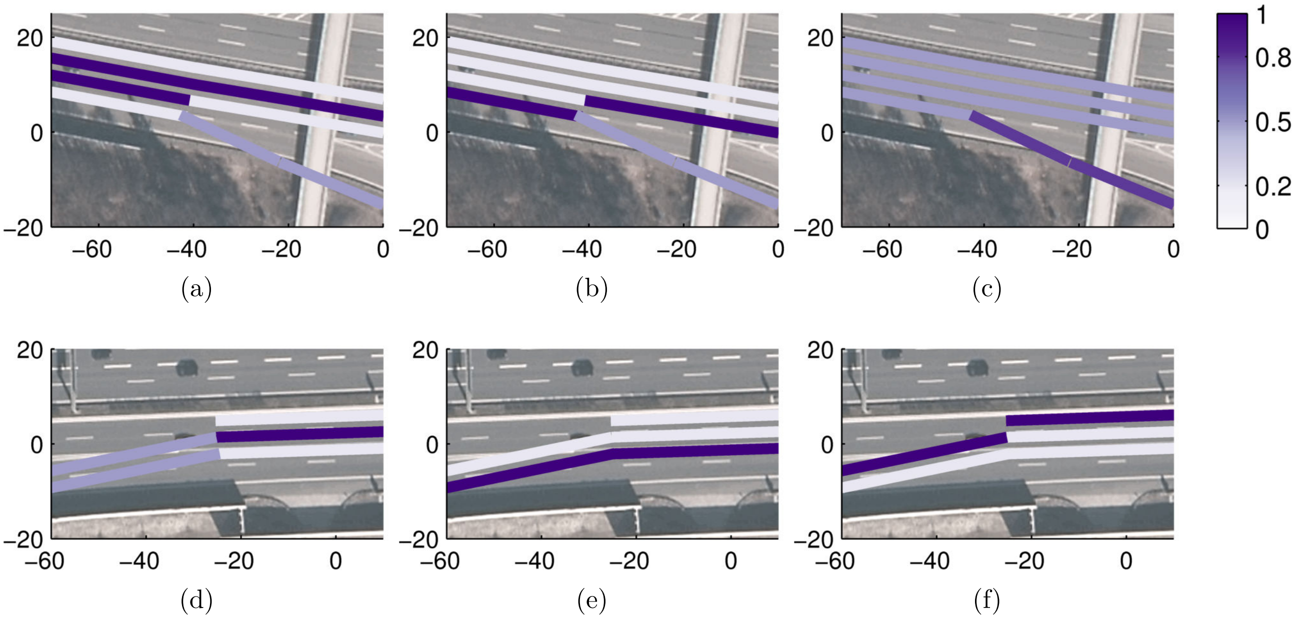


Fig. 5.22 – Lane weights for different simulated camera observations of the lane marking types. Road bifurcation with (a) left broken, right broken, (b) left broken, right solid, (c) left solid, right solid observed lane markings. Lane weights on joining roads with (d) left broken, right broken, (e) left broken, right solid, (f) left solid, right broken observed lane markings.

and reality. In this work, the proposed observation models are tested using the values summarized in Table 5.3.

5.5 Mode seeking

The output of the particle filter method for map-matching is a set of weighted particles, which represent candidates for the map-matched position. To obtain a meaningful result for navigation or map display applications, this output is transformed into a condensed representation of the most likely hypotheses about the map-matched position. Since the particle cloud represents a multi-modal distribution, e.g. at road bifurcations, simple statistical measures are not suitable for robust state estimation. The goal is to find a state estimation method that computes the unique map-matched hypothesis when the situation is unambiguous and multiple hypotheses together with their probabilities when the situation is not decidable. Clustering is applied to find the dominant hypotheses about the link-based map-matched position together with the probability of each hypothesis. In this section, a clustering

observation model	probability parameter	value
(5.23)	λ	-0.08
(5.25)	β	10
	μ	7.5
Table 5.1	υ	0.2
	χ	0.5
	ξ	0.8
	κ	1

Tab. 5.3 – Parameter values of the observation model

technique for map-matching is introduced to find the dominant modes of this multi-modal distribution of the map-matched position constrained to the road network.

5.5.1 Link hypotheses

In a Euclidean space, (weighted) mean-shift clustering provides an effective method to divide the state space into clusters and find the representative state for each cluster [30]. However, this method cannot be directly applied in the map-space, because the mean in this non-Euclidean space is not directly defined. As an alternative approach, medoid-shift provides a generalization of mean-shift to any metric space [151]. Medoid-shift replaces the local mean by the medoid, which is defined as the sample with the smallest distance to the mean. This algorithm proceeds iteratively similar to mean-shift until each sample has converged to its cluster center. The resulting mode-seeking method iterates over the medoids, which requires explicit medoid calculation only for the first iteration. The subsequent iterations associate the current sample with its medoid, which has already been computed. In this way, effective cycling within the sample set is performed. Another clustering approach operates in the same manner as medoid-shift, but replaces the mean with the median [150]. Similar advantages as for medoid-shift clustering hold, because the median is also an element of the sample set.

To apply median-shift clustering on samples in the road network, an appropriate definition of the median in this space is required. Since the clustering algorithm operates on the weighted set of particles, a definition of the weighted median needs to be found. Examples of median definitions in multi-dimensional spaces are given by the geometric median, which minimizes the L_1 norm, or the Tukey median [150].

Finding the median of map-referenced positions relates to the problem of finding the 1-median in a graph. In general, the solution of the p-median problem in the graph $\mathcal{G} = (V, E)$ with edge weights $w : V \to \mathbb{R}$ is given by $V_p \subset V$ containing p vertices such that the sum of weighted distances from V_p to V is minimized, [85]. Thus, the 1-median is defined as the vertex v that minimizes

$$\sum_{x \in V} w(x) d(v, x), \tag{5.29}$$

where $d(v, x)$ is the length of the shortest path in $\mathcal{G}$ between v and x.

In the context of particle clustering in the road network, a subgraph is constructed around the local neighborhood of the respective particle $p^i, i = 1, \ldots, N$. This subgraph contains all particles that can be reached from p^i with a path shorter than some threshold D. For each particle within this range, a vertex v^i is created and the particle link weight obtained from (5.26) is assigned to that vertex. Edges are constructed between neighboring particles and the edge weights are determined by the distance between these particles along the road network. It can be assumed that this local subgraph of the road network is a tree, which is a graph with no cycles. In this case, the 1-median can be found by Goldman's algorithm in polynomial time [64].

After median-shift has been applied for each particle until convergence, one or multiple clusters together with the cluster centers are returned. The dominant hypotheses are represented by the cluster centers and the hypothesis weight is computed as the sum of link weights of the particles that belong to that cluster. The best hypothesis is defined as the hypothesis with the highest weight.

The distance threshold parameter D [m] controls the distance between the final cluster centers. A small value produces more, closely spaced hypotheses, whereas a large value decreases the number of more widely spaced hypotheses. On the one hand, this parameter value should be small enough to distinguish individual links at intersections or branching roads. On the other hand, it should

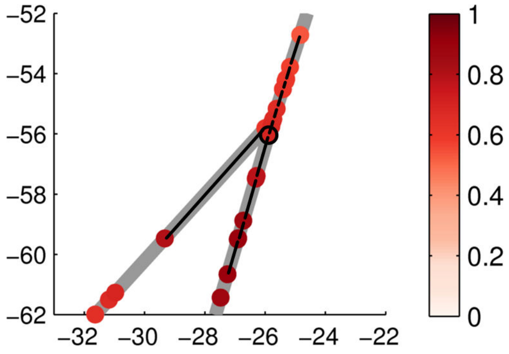


Fig. 5.23 – Local subgraph around the current particle (black circle) during one median-shift iteration. A tree is constructed with all particles that can be reached from the current particle. The link-based weight defines the particle color intensity.

be large enough to produce clusters that represent only the dominant hypotheses about the map-matched position. When D is larger than the spreading of the particle cloud, only one hypothesis is returned.

The evolution of hypotheses when the vehicle passes a bifurcation is presented in Figure 5.24. Different values for D are applied to demonstrate the effect on the number and distribution of hypotheses. The preferred choice is $D = 20$, because a single hypothesis is produced until the particle cloud arrives at the road split point and the hypotheses separate when the particle cloud covers both road branches. For $D = 10$, multiple hypotheses are produced when the particle cloud is still some meters away from the bifurcation, although the situation is unambiguously decidable at this stage. For $D = 30$, the hypotheses split further behind the bifurcation than for smaller values of D and therefore the ambiguity in this situation would be detected later.

5.5.2 Lane hypotheses

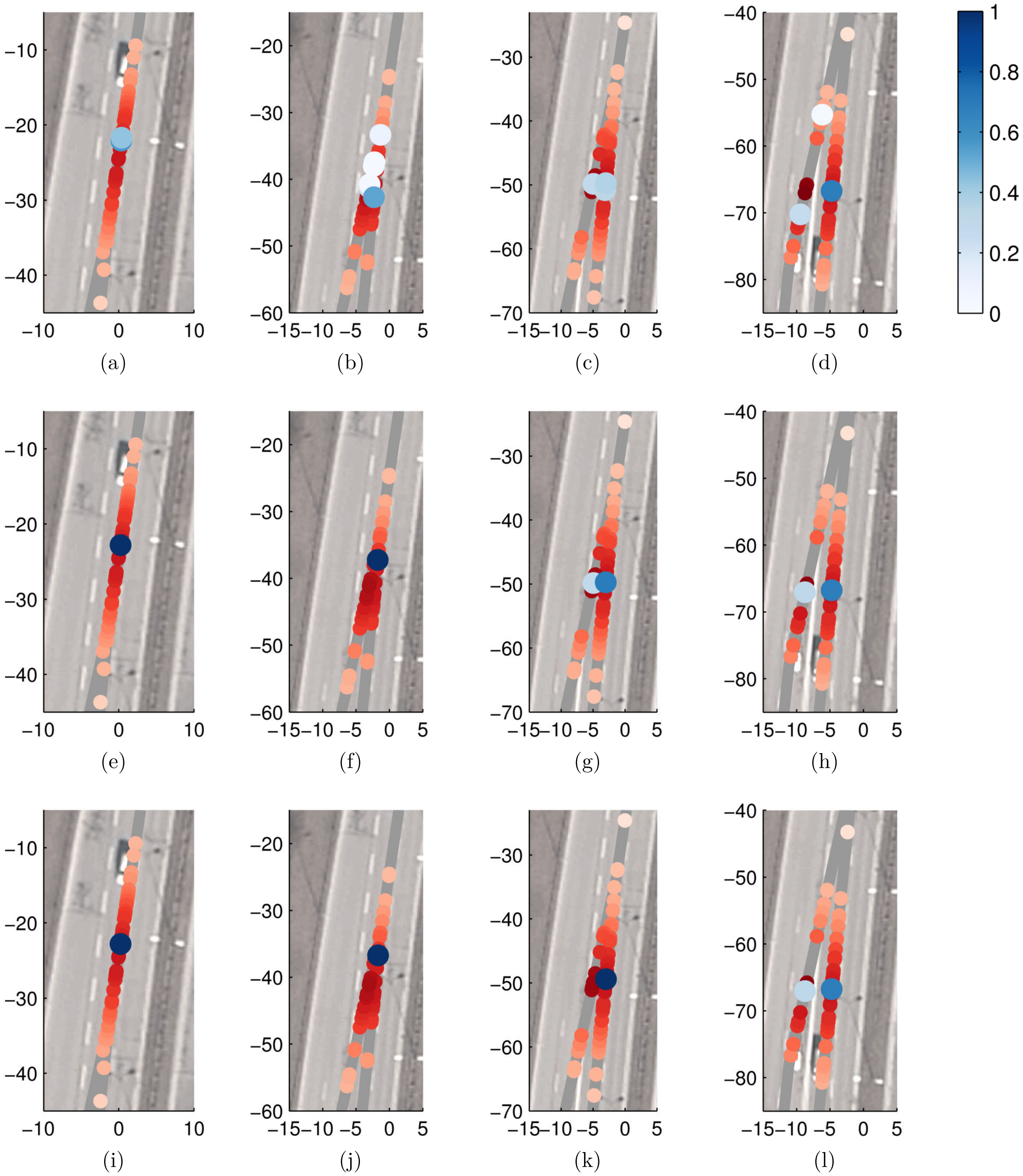
After the best hypothesis for the link-based map-matched position is determined, the best lane hypothesis is computed. For this, all particles on the same link as the best link-based hypothesis are selected. For each lane on this link, the lane hypothesis weight is computed as the sum of all lane weights (Table 5.1) of particles on the respective lane. The lane with the highest lane hypothesis weight is returned as the final lane estimate.

5.6 Results

The map-matched traces are evaluated to demonstrate the quality of the map-matching system. It is suggested to measure the quality of the map-matched result in five categories: Accuracy, stability, confidence, robustness and computational performance.

The solution needs to be the accurate such that the correct assignment between the current absolute position and the link of the road network together with the optimal position along the link is found. For navigation and ADAS applications, accuracy requirements of the map-matched results are clearly lower than for automated driving. An accuracy of better than 10 meters guarantees that driving assistance and guiding systems can precisely react to turn maneuvers or defined locations for example of sharp turns or speed limits ahead.

The stability of the map-matcher determines the deviation of the result in comparison to the input data. Small changes in the measurements of the absolute position and the camera system



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 5.24 – Evolution of particles (red) with link weights obtained from (5.26) and hypotheses (blue) after median-shift clustering in the road network with (a)-(d) $D = 10$, (e)-(h) $D = 20$, (i)-(l) $D = 30$. Hypotheses weights are normalized to sum up to one.

should lead to small changes in the map-matched solution. Especially in ambiguous situations, a stable assignment to the correct link without jumps across links even in ambiguous situations is crucial.

If the solution is ambiguous, a measure for the confidence of the result is essential for driving functions that use the map-matched position as input for further computations. In navigation systems, the confidence of the vehicle position is referred to as integrity [137]. A confidence value provides the degree of certainty of the map-matched result and can be used to predict failures of the map-matching system.

Robustness of the map-matcher guarantees that the correct map-matched solution is found even under map inaccuracies and environmental conditions that cause measurement errors. This criteria also includes a fast recovering from mismatches, which is required to detect deviations from the planned route and to adapt the guiding information immediately. A robust method ensures that the vehicle is not lost under adverse conditions.

Since map-matching is operating online, real-time computations at the frequency of the incoming measurements are necessary to enable the deployment on an embedded system.

The implementation of the map-matcher has been tested within the navigation framework in a BMW research vehicle as described in Section 5.1. The positioning component computes an estimate of the geographic vehicle position based on the fusion of GNSS, velocity, yaw rate and steering angle as described in Section 2.2.2. Traces of estimated absolute position and camera measurements were recorded for offline replay and reproduction of results. The number of particles was set to 50.

The quality of the map-matcher in the above criteria is best assessed at challenging situations for map-matching. Therefore, instead of performing field tests and computing statistical measures of quality for complete traces, this work focuses on the analysis of trace segments at characteristic scenarios. Such scenarios are defined by the road geometry and topology and include standard situations with straight roads or simple intersections, but also complex intersections or road bifurcations. The scenarios for investigation of the map-matching quality are focused on urban areas, where challenging situations are frequently observed. Difficult scenarios also occur when the environmental conditions become adverse and measurement errors lead to map-matching ambiguities. In general, deviations between the road geometry in the map and the absolute position trace complicate map-matching.

5.6.1 Accuracy

The correctness of the matched links and lanes and the accuracy of the map-matched position along the link are evaluated. The results allow to estimate the expected error of the map-matcher.

Comparing the results against the ground-truth solution would be the preferable way to evaluate the accuracy of the map-matcher. However, since the map-matched solution is constrained to an inaccurate link geometry, the truth solution for the map-matched position along the link is not computable. An illustration of this property is shown in Figure 5.25.

Therefore, traces of the estimated absolute position and map-matched result are compared to each other. This visualization scheme has proven to be useful for visual inspection of the map-matched result [133]. For a rough estimate of the positioning and the map error, an aerial image is added to the background of the visualizations, Figure 5.26a. All coordinates and measures are given in meters. Some of the presented scenarios have been rotated for better comparison and optimized page layout.

Under perfect conditions, the map and the estimated absolute position would match exactly. However in practice a deviation between the map and the absolute position is observed. The larger this deviation, the more difficult the situation is for map-matching. The optimal map-matched position is defined as the orthogonal projection of the absolute position to the nearest link along the actual traveled route in the map. The determination of the actual traveled route follows the map topology. This distance between the absolute position and the optimal map-matched position is considered as a disturbing variable q_t in the map-matching system, which corresponds to an upper bound of the map-matching

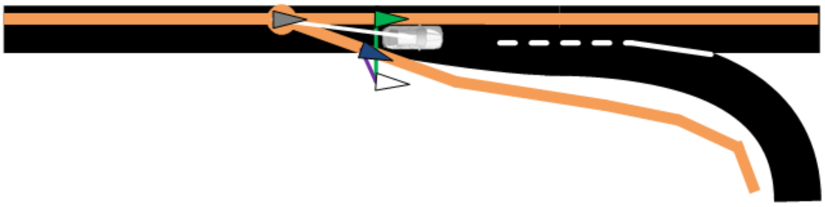


Fig. 5.25 – Situation where the ground truth solution cannot be determined without knowledge of the true absolute position, the true road map and the traveled route. The true vehicle position is at the splitting point of the real road, where the driver decides for one of the branching roads. This point is displaced in the given map (orange). The ground truth position should be assigned to the position of the splitting point in the map (gray), because both successor links can be traveled at that time. The map-matching system obtains an estimate of the absolute position (white) and computes a map-matched solution (blue). Knowledge of the actual traveled road is required to determine the optimal map-matched position as the nearest position (green) along the traveled link.

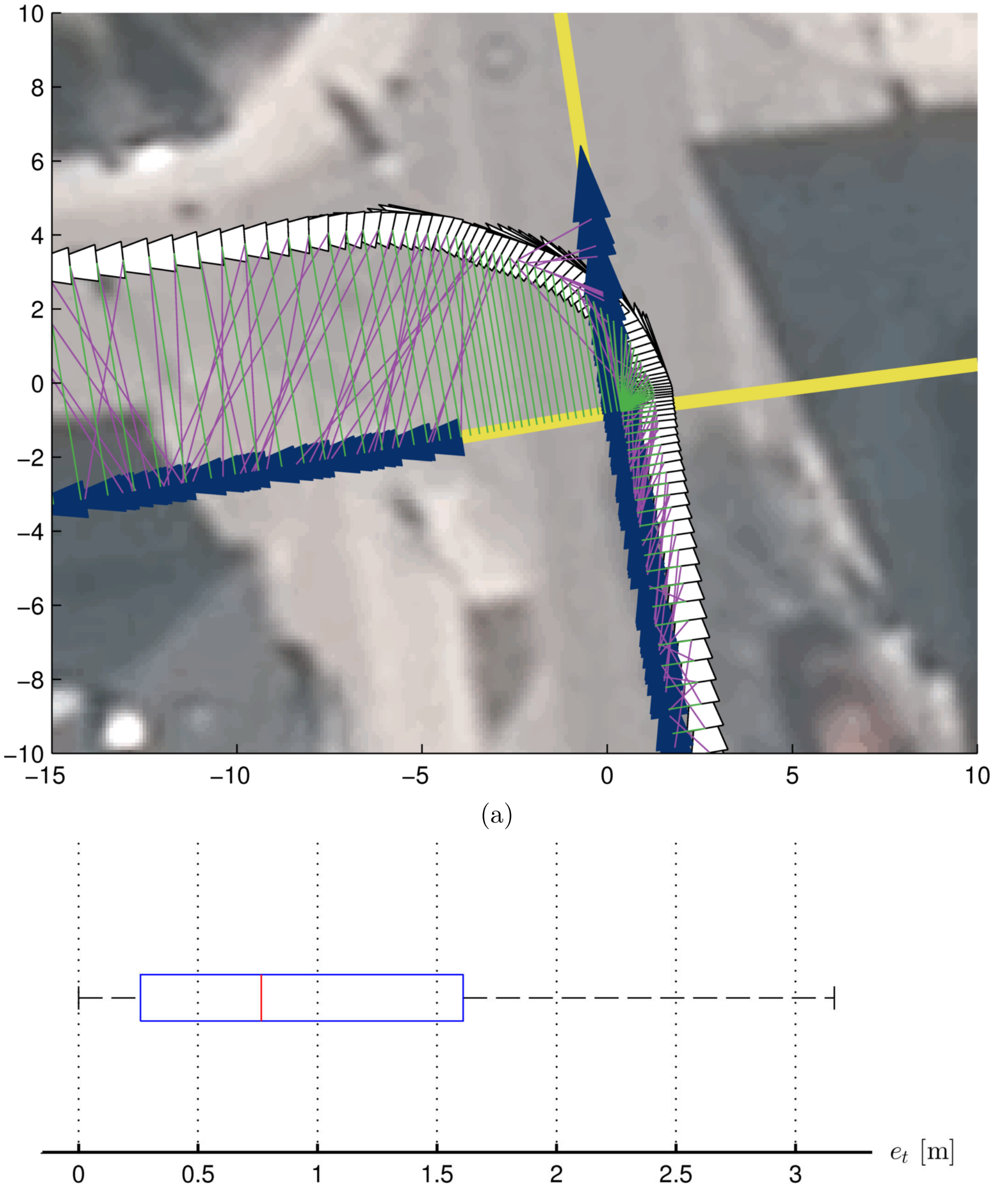
accuracy. Now the map-matching error e_t at each time step is defined as the additional deviation from the optimal map-matched position given the disturbance value q_t,

$$e_t = r_t - q_t, \quad \text{if } r_t > q_t, \tag{5.30}$$

where r_t denotes the distance between the absolute position and the computed map-matched position. This error definition allows to evaluate situations where the map-matched result is worse than the theoretical optimum. A small map-matching error corresponds to a high accuracy. If the absolute position is mismatched to a link closer than q_t, the map-matching error is not defined and omitted in the statistical evaluations. The statistical evaluation is shown in a boxplot, Figure 5.26b. In this example, the map-matcher computes matches on the link that continues straight across the intersection, before it switches to the correct link. However, this behavior does not increase the map-matched error, because the disturbance value is larger than the distance to the mismatched link. If the turning link in the map would not show a displacement error to the real road, then the map-matched result would be perfectly acceptable. One could even argue that the computed solution does better reflect the optimal map-matched position under this link displacement, if map topology is not considered. Such cases are studied in detail during the robustness analysis of the map-matcher in Section 5.6.4.

The achievable along-track accuracy of link-based map-matching is best observed at roads without turns, when both the absolute vehicle position and the road geometry from the map are accurate. Results of map-matched traces at typical scenarios with straight roads show that the medians of the map-matching error e_t range below 0.5 meters, Figure 5.27. The maximum error in these scenarios is observed at 2.3 meters and more than 75% of the selected trace sections have an error below 1 meter.

Simple turn maneuvers at two intersecting roads belong to standard scenarios for map-matching. The road geometry from commercial automotive map data is typically abstracted and does not represent the curved shape of the true road course within the intersection area. Such situations involve map-matching inaccuracies along-track, but also across road segments. Comparing absolute position traces and the map-matched results shows that the correct link assignment is quickly resolved, Figure 5.28. In some cases, short mismatches to the wrong link or to a non-optimal position are observed, Figures 5.28b, 5.28c and 5.28e. These mismatches contribute to the map-matched error only if the distance between the matched position and the absolute position is greater than the disturbance value. The boxplot for Figure 5.28c shows that the matching error increases in this case.



(a)

(b)

Fig. 5.26 – (a) Example of the estimated absolute positions (white) and the map-matched (blue) trace. Triangles point to the driving direction and the road direction respectively. The position is given by the middle of the triangle baseline. The connection between the absolute vehicle position and map-matched position indicates the same time-step. Green line: connection to the optimal map-matched position. Purple line: connection to the position computed by the map-matcher. Thick yellow line: road center line from map data. (b) Boxplot of the map-matching error in the trace section shown in (a) indicating the median (red), the minimum and maximum (black) and the 25% and 75% quartiles (blue box).

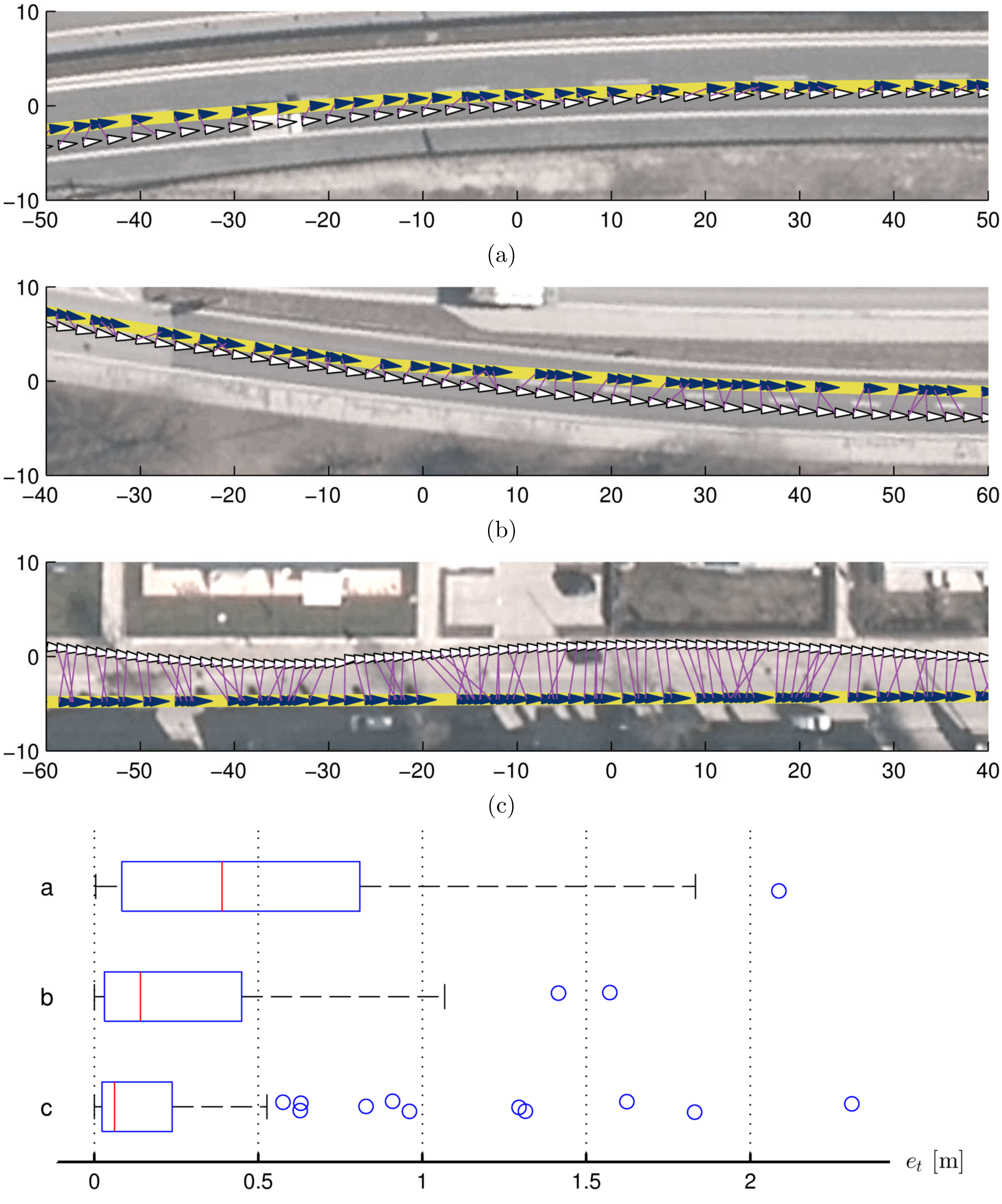
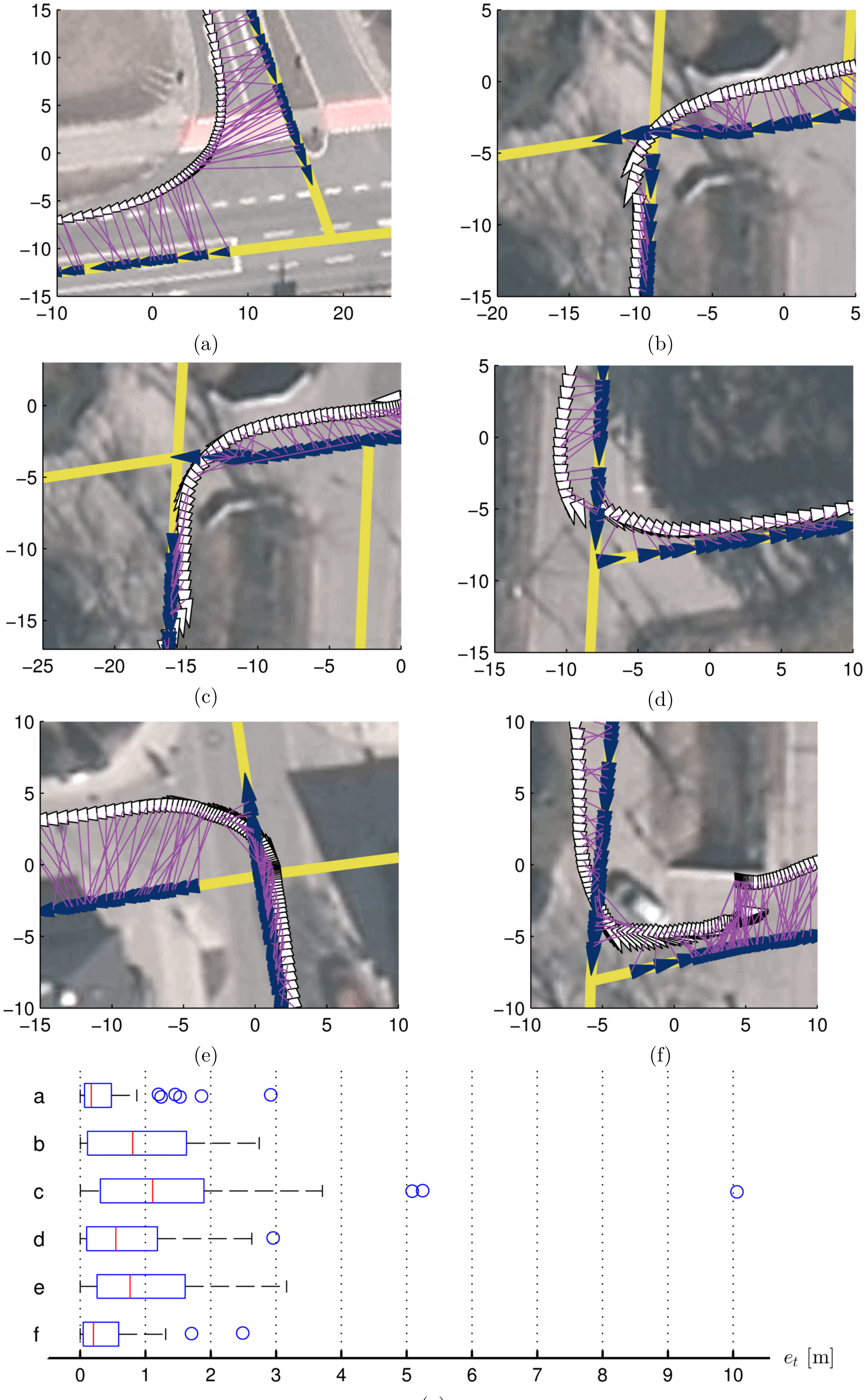


Fig. 5.27 – (a)-(c) Scenarios with map-matched results along straight roads. (d) Boxplot of map-matching error for the shown scenarios.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

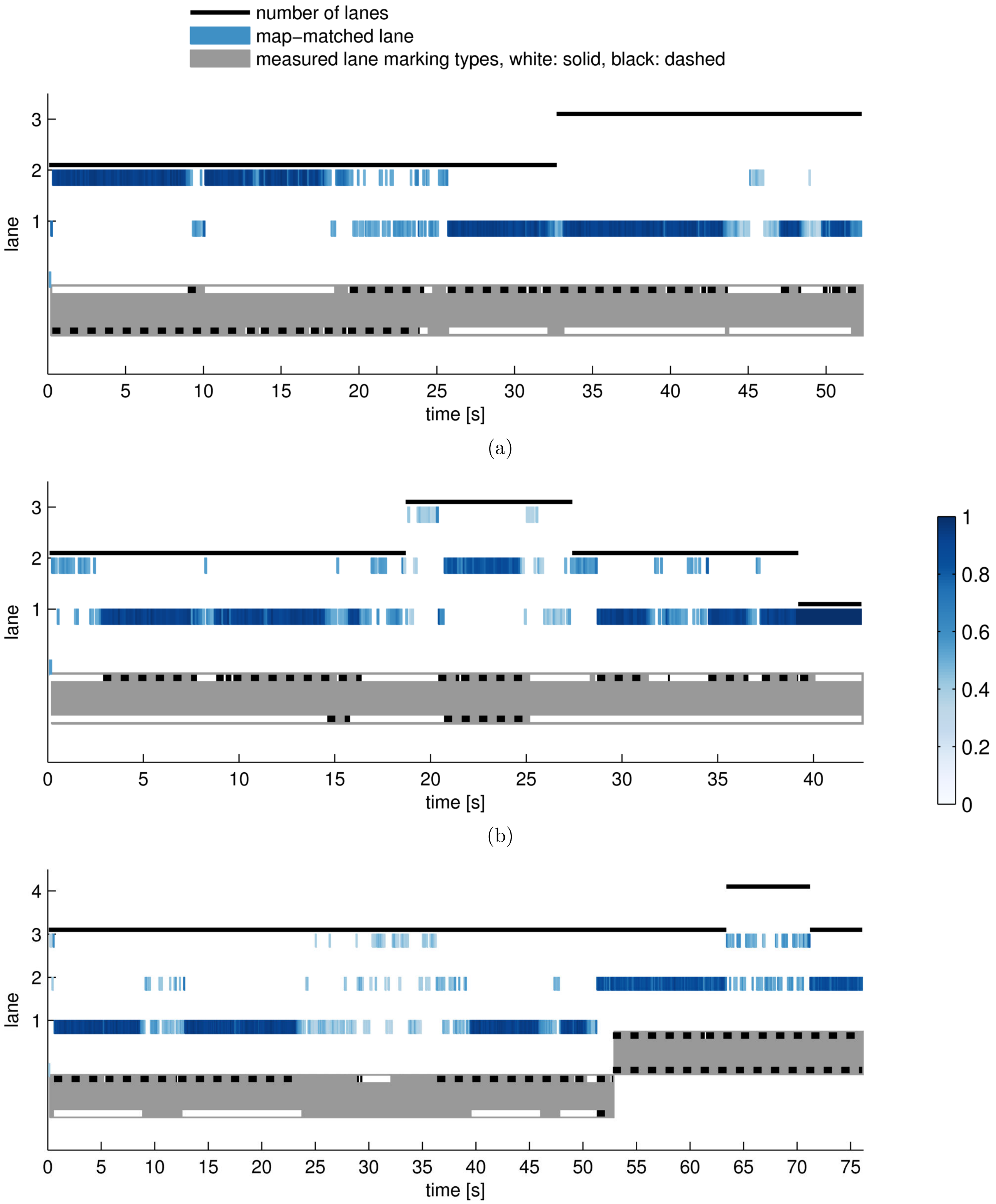
Fig. 5.28 – (a)-(f) Scenarios with map-matched results at simple intersections. (g) Boxplot of map-matching error for the shown scenarios.

More complex scenarios typically involve multiple lanes. A correct estimate of the traveled lane before an intersection or bifurcation increases the chance to identify the correct link. The accuracy of the lane estimate depends mainly on the correctness of the camera measurements, which detect the lane marking types and lane changes. The map-matcher is expected to return the lane that matches best with the camera measurements and the model developed in Section 5.3.2.

Results of matched lanes and the related lane marking detections are shown in Figure 5.29. At the beginning of the trace in Figure 5.29a, the correct leftmost lane out of two lanes is returned as long as the lane marking types fit to the model. Two given lanes and left and right broken lanes contradict with the model, leading to an ambiguous lane estimate between second 20 and 25. After this, the lane marking types indicate the rightmost lane, which is correctly computed by the map-matcher. In Figure 5.29b, the map-matcher correctly starts to identify the rightmost lane. When solid lines are measured on both sides and a two-lanes road is given in the map, the situation becomes undecidable for the map-matcher. After a third lane opens, broken lane markings on both sides are measured, which leads to the correct estimate of the middle lane. The third example in Figure 5.29c shows that all three lanes have almost equal chance to be selected by the map-matcher when no camera measurements are available between second 12 and 24. The lane change and measured broken lanes on both sides at second 40 cause the map-matcher to follow the lane change to the left. At second 52, a fourth lane opens leading to an ambiguous lane estimate between the middle lanes. All three examples show that the map-matcher immediately reacts to changes of the camera measurements without noticeable latency. Outliers for the correct lane estimate occur only when implausible or unstable camera measurements are observed. When the lane is unambiguously decidable from the camera measurements and the lane markings model, the map-matcher returns the correct lane.

Large intersections with multiple lanes receive a larger search space of the map-matcher to find the correct solution. The scenario in Figure 5.30 presents an intersection with four lanes, two leading straight and two turning left. The four absolute position traces in this scenario differ in location, the right middle lane is closest to the link the map, Figure 5.30c. In all four cases, the map-matcher finds the correct links. The evaluation of the map-matching error shows that the median for the two middle lanes is higher than for the two outer lanes, Figure 5.30e. This corresponds to the fact that the ambiguity of the matched link between the middle lanes is higher than for the outer lanes, which decreases the accuracy of the result. The map-matching error is below one meter and outliers do not exceed 8 meters, which is comparable to the results of the simple turns in Figure 5.28. When the map-matching errors of the straight traces in Figure 5.30c and Figure 5.30d are compared with the straight scenarios in Figure 5.27, it appears that they are higher in the multi-lanes scenario. The increased error is produced by the alternative link in the intersection, which causes the particle cloud to split across the link leading to a reduced population of the particles along each link. Thus, the search space around the optimal solution is sampled more sparsely by the particles, which decreases the accuracy. This interpretation is evident when the map-matched experiment is repeated with twice the number of particles, 5.30f. It can be seen that the medians of the map-matched errors are smaller or equal than in Figure 5.30e. Especially for 5.30c the median and the outliers of the map-matching error are smaller. However, because the increase of accuracy with 100 particles is small it is not worth twice the computation time compared to 50 particles.

The most challenging scenarios for map-matching are described by road bifurcations with slowly diverging roads. In such situations, the estimated vehicle position rarely matches exactly with the map geometry causing ambiguity of the correct map-matched link. Figure 5.31 demonstrates that the correct link is matched when the deviation between the absolute position and the map is moderate. The hypotheses weights decrease around the road split point, because the ambiguity is handled by hypotheses on the alternative link. The accuracy along the link decreases to a small extent compared to the simple intersection scenarios in Figure 5.28 and the less ambiguous situations in the multi-lane intersection in Figure 5.30, as the boxplot in Figure 5.31d shows. In comparison with the ambiguous middle lanes in Figures 5.30b, 5.30c, the accuracy is similar. These results suggest that

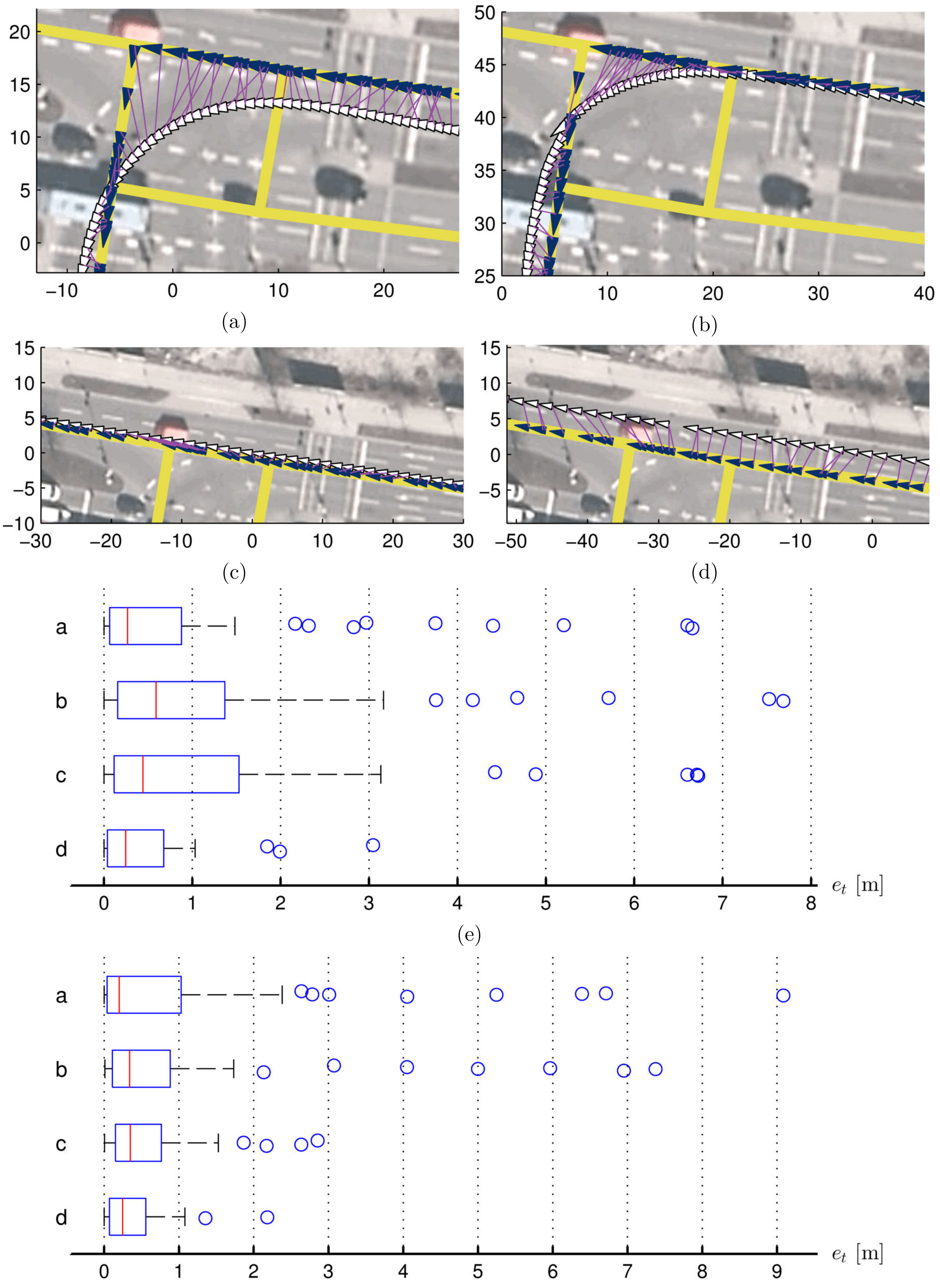


(a)

(b)

(c)

Fig. 5.29 – Trace segments with lane matching and camera measurements. The gray ribbon shows the measured lane marking types for the left (upper) and the right (lower) line over time. The lanes are counted from the right, lane 1 corresponds to the rightmost lane. The color intensity of the matched lane corresponds to the weight of the strongest lane hypothesis. For better visibility of overlapping lines, the number of lanes are drawn with a small vertical offset. The vertical displacement of the lane ribbon indicates measured lane changes to the left or right.



(a)

(b)

(c)

(d)

(e)

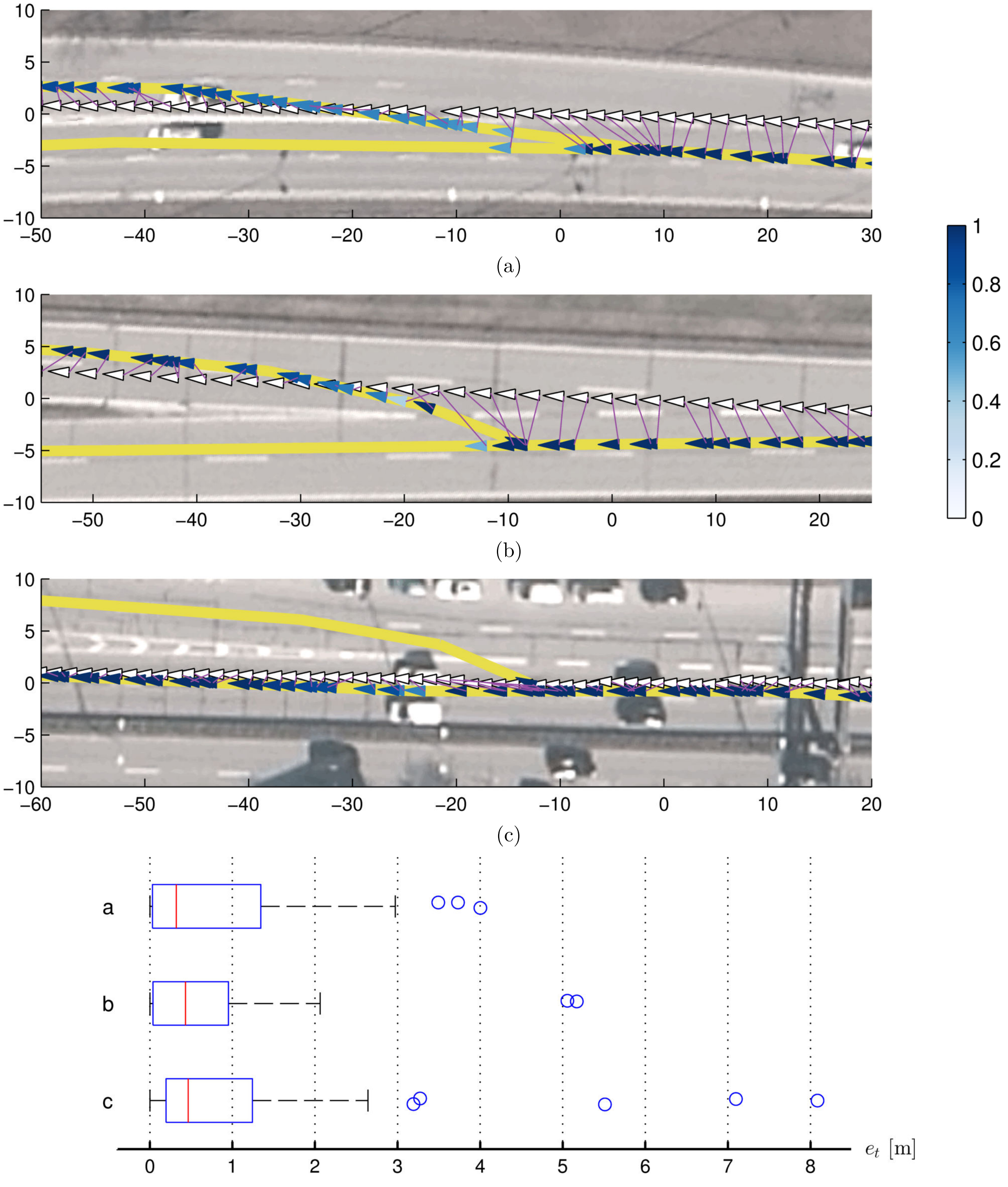
(f)

Fig. 5.30 – Scenarios with map-matched results at the same multi-lane intersection for traces starting from different lanes, (a) left lane, (b) middle left lane, (c) middle right lane, (d) right lane. (e) Boxplot of map-matching error for the shown scenarios with the default number of 50 particles. (f) Boxplot of map-matching error for the shown scenarios with 100 particles.

ambiguity decreases the accuracy, but the impact remains constant and is independent of the degree of ambiguity.

In cases where the absolute position lies between two branching roads, mismatches are more likely to occur. The accuracy across links is more critical than the accuracy along the link, which has been demonstrated in the previous examples to decrease to an acceptable level in ambiguous situations. Figure 5.32 presents characteristic situations where the map-matcher temporarily switches to the alternative link. After the absolute position approaches the correct link, the situation becomes decidable and the map-matcher recovers quickly from the mismatch.

Bifurcations with multiple lanes often contain lane connectivity information in the map. A correct lane estimate before the bifurcation increases the chance for a correct link matching after the bifurcation, because the particles follow the lane connections. To evaluate the effect of lane-matching, the map-matcher is run on the same scenarios where the camera measurements are ignored and the lane weights are set equal to one. The example in Figures 5.33a, 5.33b shows that short mismatches on the alternative link occur and the hypothesis weights of the correct link are lower when lane information are ignored. Lane matching supports link matching as long as the camera measurements correspond to the correct lane. A lane mismatch can lead to a wrong link matching and actually corrupt the correct result. In Figure 5.33c, the lane marking model does not fit to the camera observations. The camera detects a broken line on the left and a solid line on the right, because the beginning of the road split is marked with a solid line between the two middle lanes. This causes the particles on the rightmost lane to obtain a high weight. As a consequence, more particles are routed to the alternative link, which produces strong hypotheses on the wrong link.



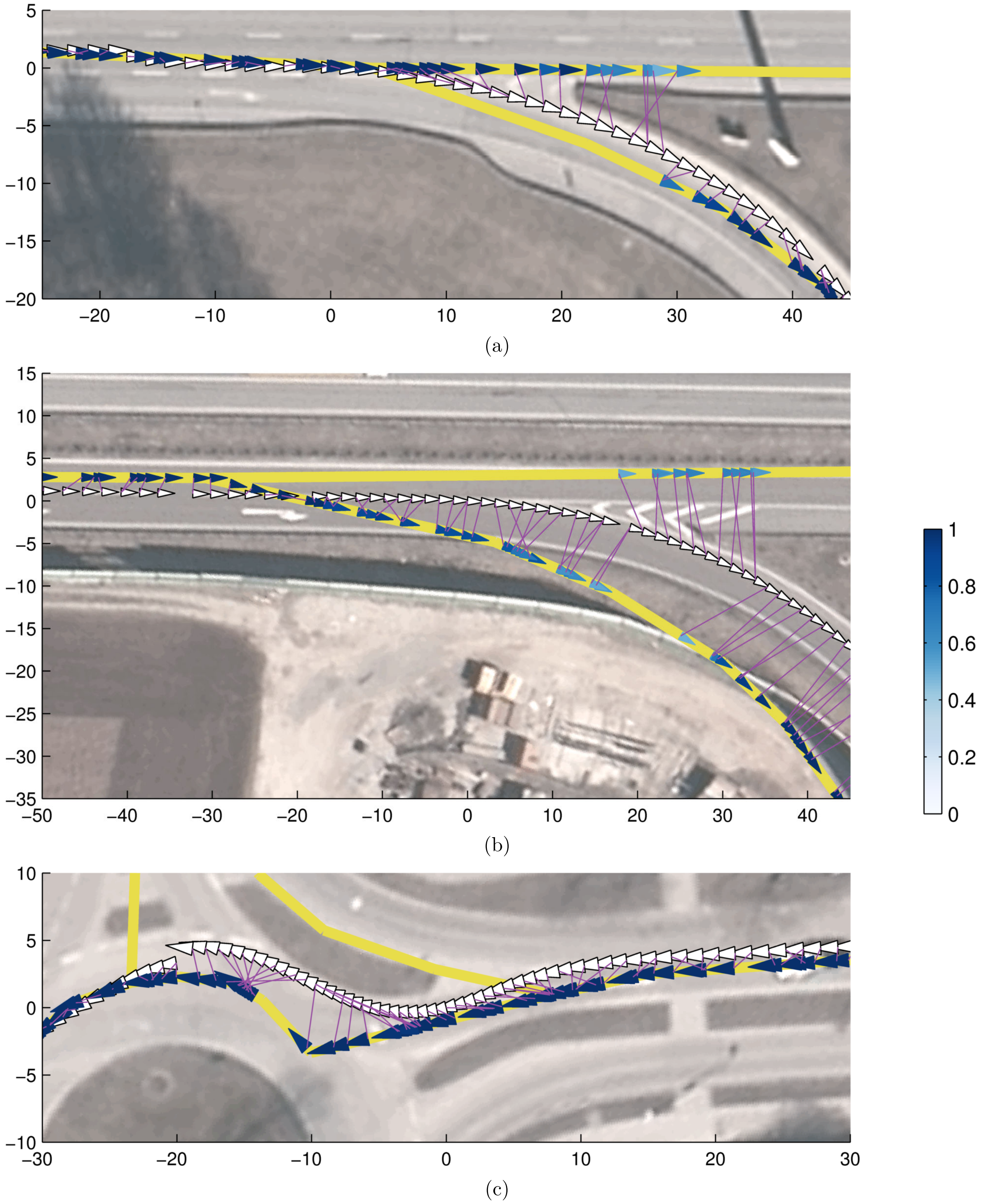
(a)

(b)

(c)

(d)

Fig. 5.31 – (a)- (c) Scenarios with map-matched results at bifurcations with moderate deviation between the absolute position and the map. The color intensity of the map-matched position corresponds to the weight of the strongest link hypothesis. Decreasing hypotheses weights indicate ambiguity. (d) Boxplot of the map-matched error for the shown scenarios.



(a)

(b)

(c)

Fig. 5.32 – (a)-(c) Scenarios with map-matched results at bifurcations with significant deviation between the absolute position and the map. Temporal mismatches occur. Decreasing hypotheses weights anticipate switches to the alternative link.

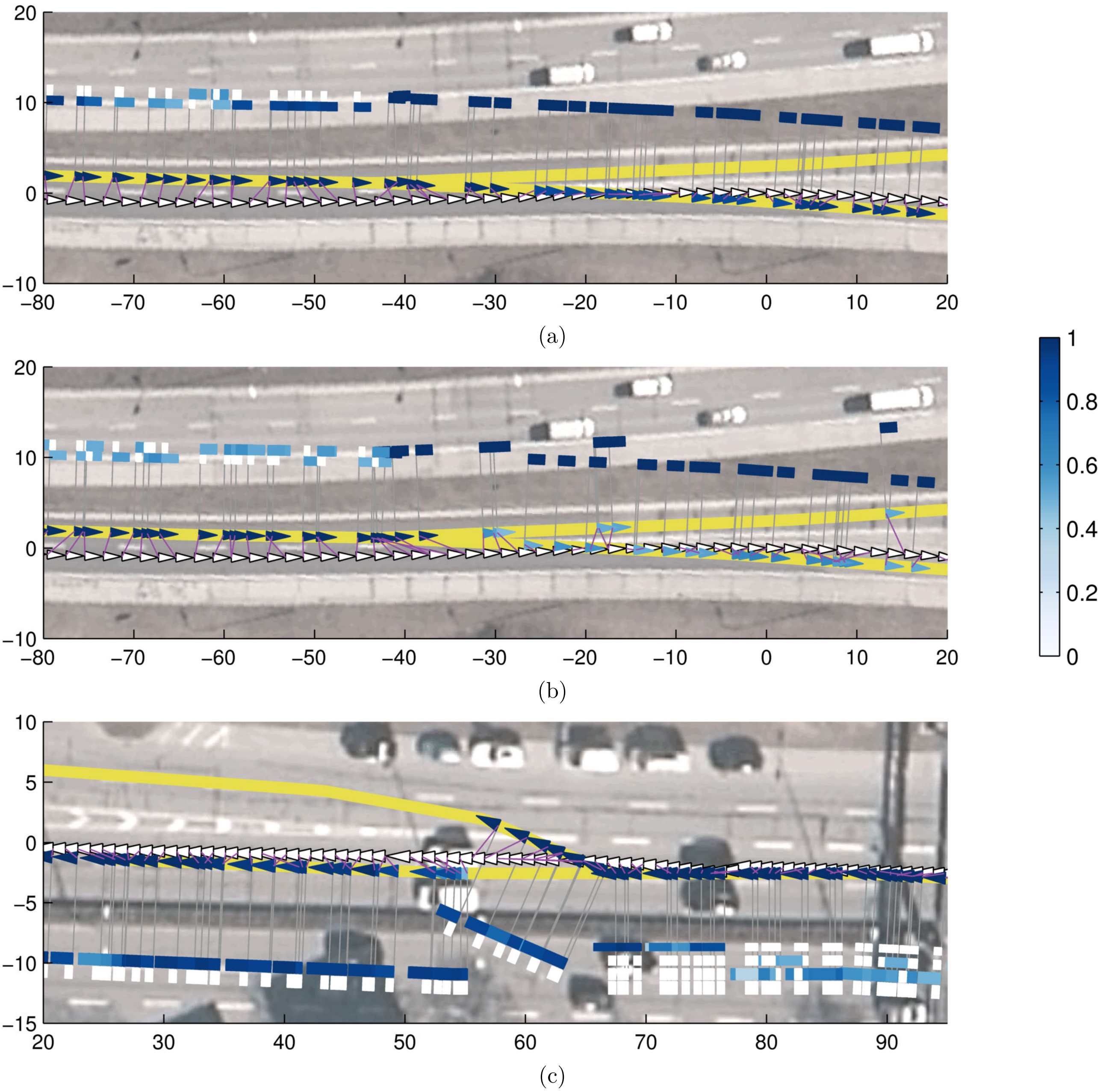


Fig. 5.33 – Scenarios with results of the link matching together with the lane matching. Flags on each map-matched position indicate the number of lanes and the matched lane (blue). The color intensity of the flag corresponds to the lane hypothesis weight of the strongest lane hypothesis. (a) Lane weights lead to more particles on the correct link and support the particle propagation to the correct link. (b) Same scenario ignoring camera measurements and assigning uniform lane weights for each lane. The ambiguity of the strongest link hypothesis increases. (c) Different scenario with camera measurements considered. A mismatched lane before the bifurcation causes a temporal mismatch of the link.

5.6.2 Stability

The accuracy results show that the map-matched position jitters around the optimal position along the link. When the map-matched result is used for map display, the driver would expect a continuous movement of the map-matched position based on the vehicle speed. The magnitude of instability is evaluated by measuring the distance between two consecutive map-matched positions. This measure is put in relationship to the desired shift, which is determined by the distance between two consecutive optimal map-matched positions as defined in the accuracy section. The derivative of the optimal map-matched position shift corresponds to the optimal speed of the map-matched result along the map. Assuming a constant time step length, the derivatives of the map-matched position shift and the optimal map-matched position shift are approximated by their difference between consecutive time steps. These measures are denoted by $ds_{\text{map-matched}}$ and ds_{optimal}, respectively. The comparison of these two measures provides insight into the magnitude of jittering of the map-matched position. The evaluation on a bifurcation scenario shows that the difference of the optimal map-matched position shift remains constant except at the road split point, but the difference of the computed map-matched position shift produces overshooting values, Figure 5.34.

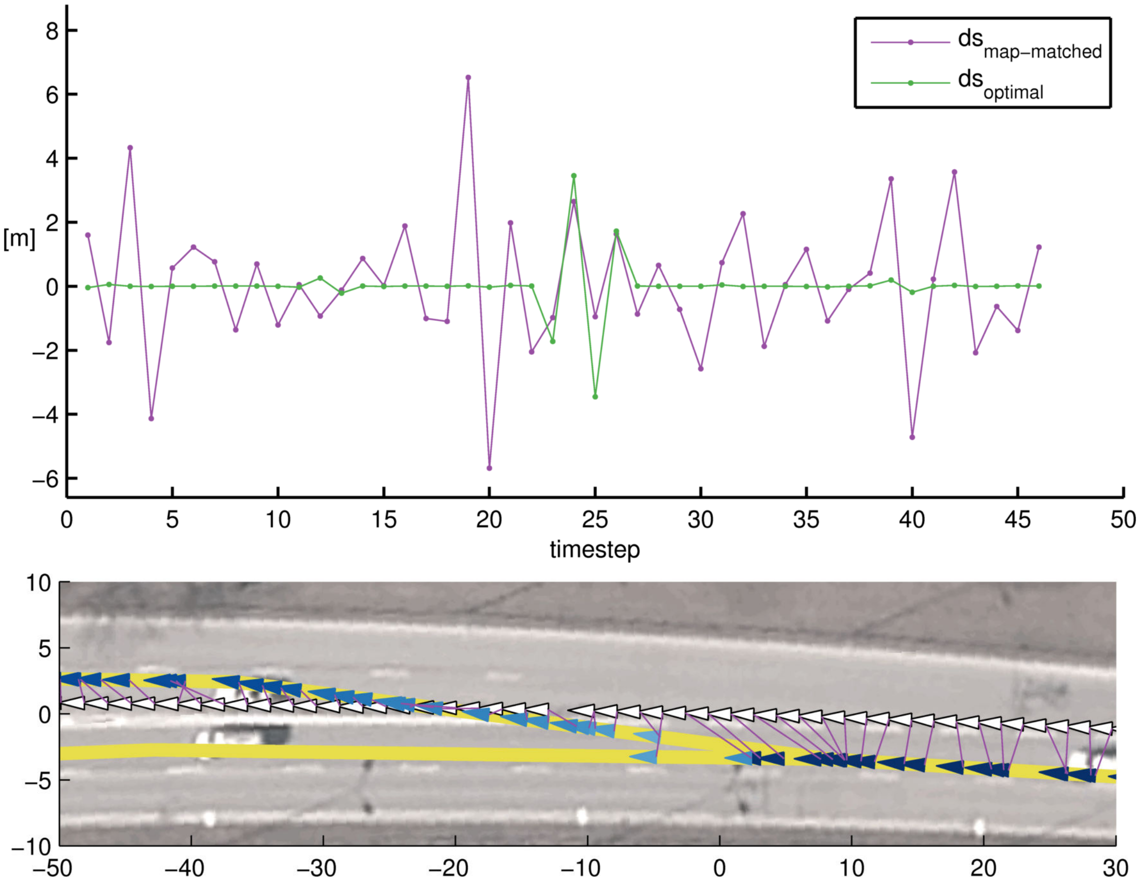
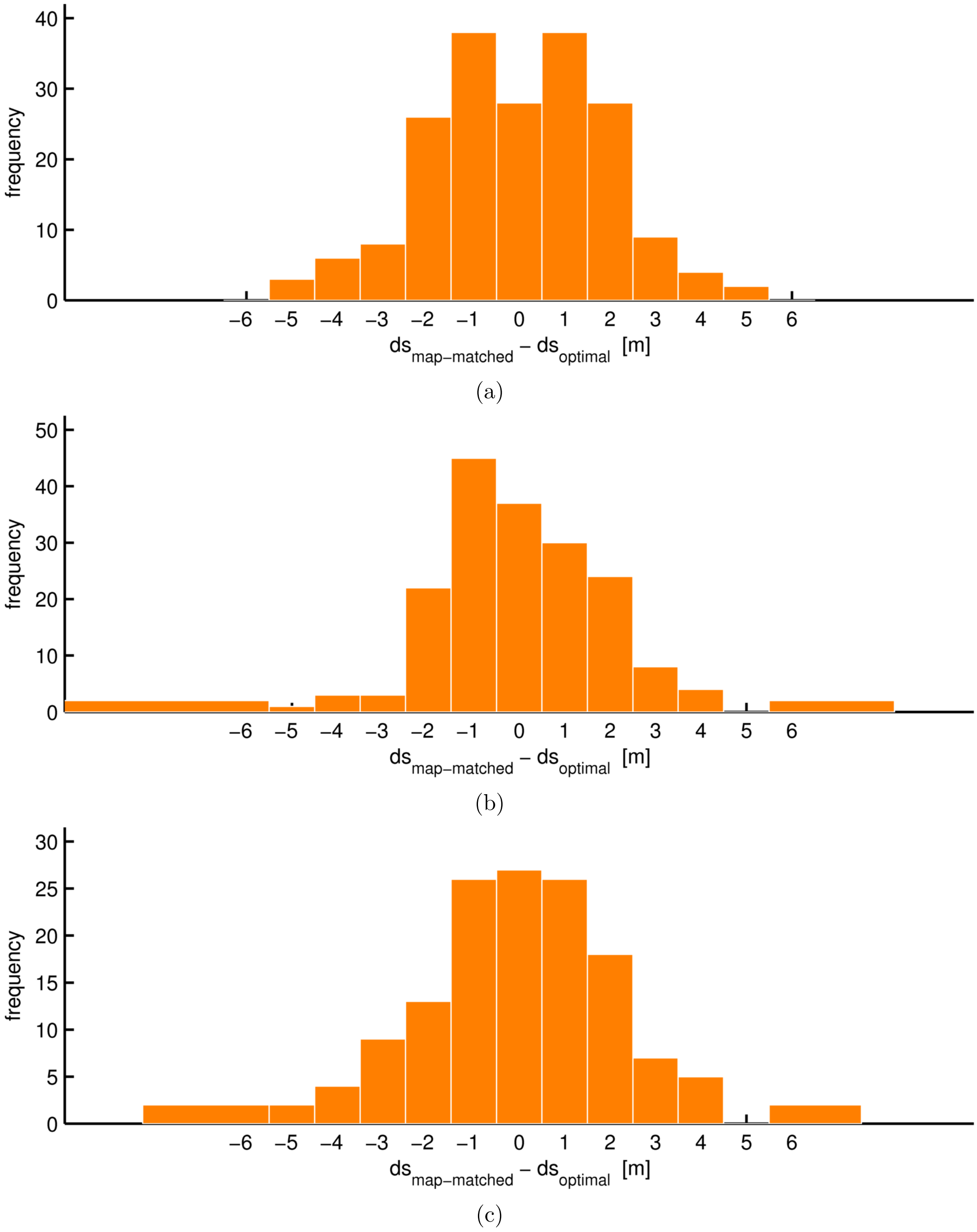


Fig. 5.34 – Difference between consecutive shifts of the map-matched and the optimal map-matched position evaluated at the scenario from Figure 5.31a. Jittering of the map-matched position occurs at time steps with large difference between the two curves.

The residual between $ds_{\text{map-matched}}$ and ds_{optimal} at each time step quantifies the instability of the map-matched result. This instability measure is computed for the scenarios from the accuracy evaluation and jointly visualized for each scenario class, Figure 5.35. According to the histograms, the majority of map-matched points fall into the instability range of between -1.5 and 1.5 meters. Outliers with more than 5.5 meters and maximum value of 10 meters are observed. The instabilities appear to



(a)

(b)

(c)

Fig. 5.35 – Histograms of the instability measure for three scenario classes, (a) straight roads from Figure 5.27, (b) simple turns from Figure 5.28, (c) simple bifurcations from Figure 5.31

have a similar magnitude and distribution in the three scenario classes regardless of the degree of ambiguity.

The other aspect of stability concerns switches across links. The examples in the accuracy section show that the degree of ambiguity increases the risk for instabilities of this kind. For straight roads (Figure 5.27), simple turns (Figure 5.28) and bifurcations with little ambiguity (Figure 5.31), only short mismatches at the road split points occur. At ambiguous bifurcations, switches to the alternative link may occur immediately after the road split point or later as an interruption of the correctly matched link, Figure 5.32. When lane measurements are considered, the ambiguity is reduced and link switches are avoided as the comparison of Figures 5.33a and 5.33b shows. These results demonstrate that instabilities across links are difficult to predict, because they do not follow a pattern and they depend on the individual configuration of the map and the absolute position.

The stability of the matched lane largely depends on the availability of stable camera measurements. The inspection of Figure 5.29 indicates that a stable lane match is observable when the lane is decidable from the lane markings model and the camera measurements. The matched lane remains stable as long as the camera measurements do. When the lane is not decidable, which applies to the middle lanes of the four lanes road at around second 55 in Figure 5.29c, multiple switches between the matched lane occur. Thus, the map-matcher produces instabilities in the lane matching, but they can be directly predicted from the stability of the camera measurements and the lane markings model. For more than three lanes, a precise estimate of the matched lane is not supported by this model.

5.6.3 Confidence

The link and the lane hypotheses weights provide an intrinsic quality parameter of the map-matched result. The weights of the matched link hypotheses at the different scenario classes indicate the ambiguity of the situation. Along straight roads and at simple and multi-lane turns, only one link hypothesis exists, which can be seen from the dark blue color of the map-matched positions in Figures 5.27, 5.28 and 5.30. When the vehicle approaches a road bifurcation, the weight of the strongest link hypothesis decreases after the road split point, Figure 5.31. This observation is in good agreement with the property of the map-matcher to maintain multiple hypotheses.

The relationship between confidence and ambiguity is evident from Figures 5.33a and 5.33b. In the simulated absence of camera measurements, the situation at the slowly diverging road becomes highly ambiguous. It can be observed that the confidence is drastically reduced in this case compared to the situation where the detected lane marking types indicate which lane has been taken before the road branches.

Monitoring the hypotheses weights allows to anticipate a forthcoming link switch, as demonstrated in Figures 5.32a and 5.32b. A suitable threshold for deciding that the map-matched link is unreliable is suggested at 0.5, because the sum of weighted particles belonging to hypotheses on the alternative link equals this value, which corresponds to a probability of 50 % that the correct link is matched. In this way, a low degree of confidence can be used as a warning for failure of the map-matching system.

In contrast, a high hypothesis weight cannot guarantee correctness of the result. Figure 5.32a illustrates that the vehicle traveled 20 meters on the wrong link before the ambiguity is detectable from a decrease in the link hypothesis weights. A similar result is visible in Figure 5.33c.

The confidence of the matched lane is analogously determined by the weight of the strongest lane hypothesis. The results of the matched lane show that the confidence is directly related to the ambiguity, Figure 5.29. In the accuracy section, it has already been stated that the lane is correctly matched if ambiguity is not present. Comparing confidence with accuracy, it is confirmed that high confidence corresponds to correct matches of the lane. Low confidence only occurs when the situation is ambiguous and therefore not decidable for the map-matcher. Thus, the matched lane can be trusted when the confidence is high and the result should be discarded if the confidence is low. The threshold for indication of lane matching failures is proposed to be set to the inverse of the number of lanes for the currently matched link.

5.6.4 Robustness

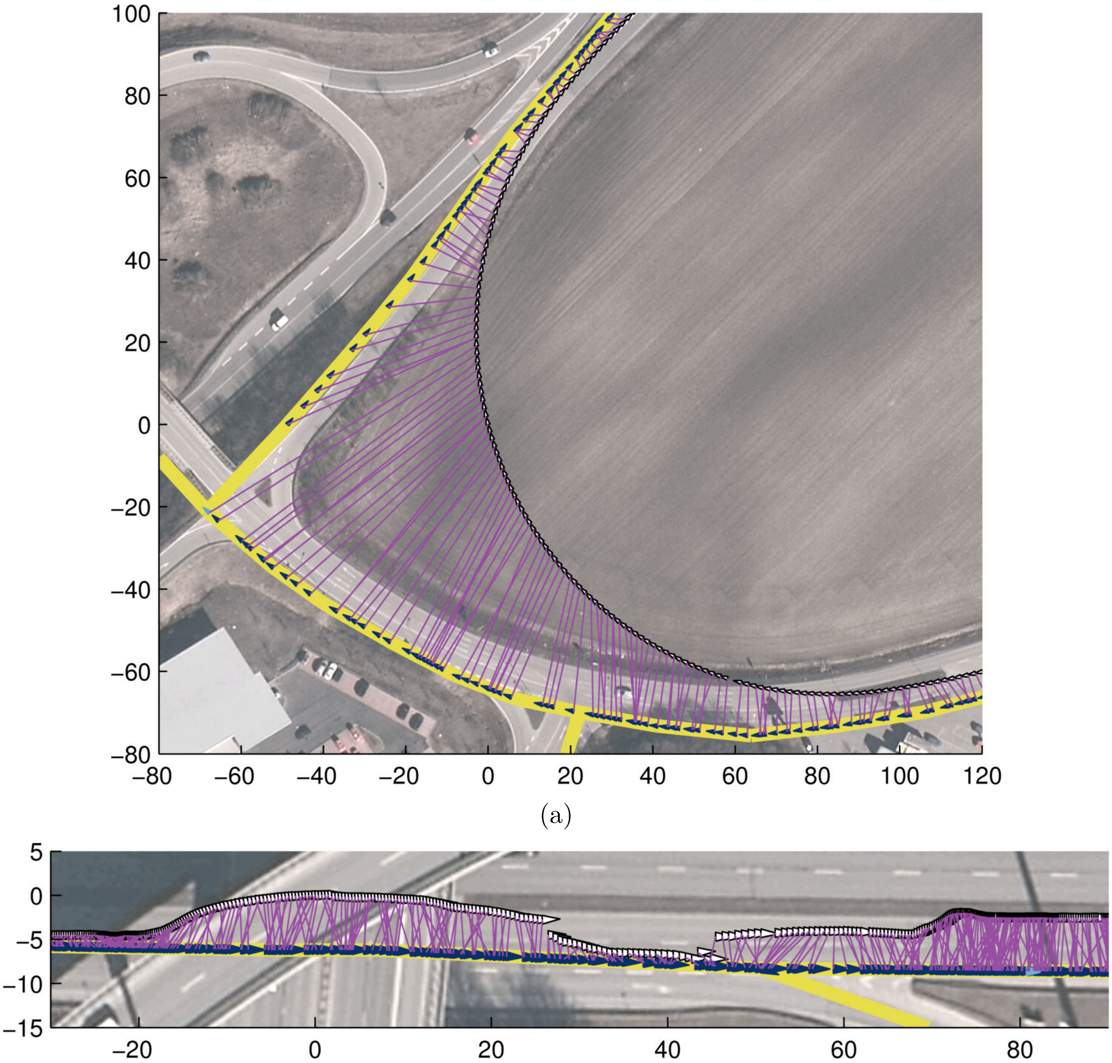
The results show that the map-matcher is capable to compute the correct solution under inaccuracies of the absolute positioning and the road geometry in the map. In general, challenging situations may occur when the discrepancy between the map and the absolute position is high. This situation is for example demonstrated in Figure 5.32c, where road geometry of the true link is less similar from the absolute position trace than the alternative link. Nevertheless, the correct link is matched, because the map-matched position is propagated after the road split point along the correct link.

The discrepancy between map and absolute positioning leads to confusion of the map-matcher only if it causes ambiguities. Figure 5.36a demonstrates that even a large error of the map can produce an acceptable result of the map-matcher, if the alternative links are plausible. In the same way, positioning errors of several meters may be less confusing for the map-matcher if only one link matches reasonably well, Figure 5.36b.

However, situations exists that expose the limits of the map-matcher under inaccuracies. Figure 5.37 presents an example where map-matching errors are inevitable. The absolute position trace is almost exactly in the middle of the two branched roads for more than 50 m. Uncertainty of the matched lane before the bifurcation adds ambiguity to this situation. Multiple switches to the wrong link occur, before the map-matcher decides for the correct link. Still, the strength of this map-matching method lies in its ability to quantify the reliability of the result by the confidence measure. The developed models for particle motion and weighting ensure that both alternatives for the matched link are tracked without loosing particles on either link too early. This strategy allows to track multiple hypotheses, which enables a quick recovery from mismatches. Hence, the information is evolved over time along the road network and reinitialization of the map-matching system is avoided.

The need for reinitialization of the particle filter method has been rarely observed. It occurs for examples in long tunnels, where the absolute position suffers from a significant drift from the true road due to the blocking of GNSS. The problem can be mitigated by feeding the map-matched position back into the positioning system in order to snap the absolute position to the road, as proposed in [170]. However, this method comes at the risk of misleading the absolute position when mismatches occur. The effects of positive feedback in such a system need to be further investigated.

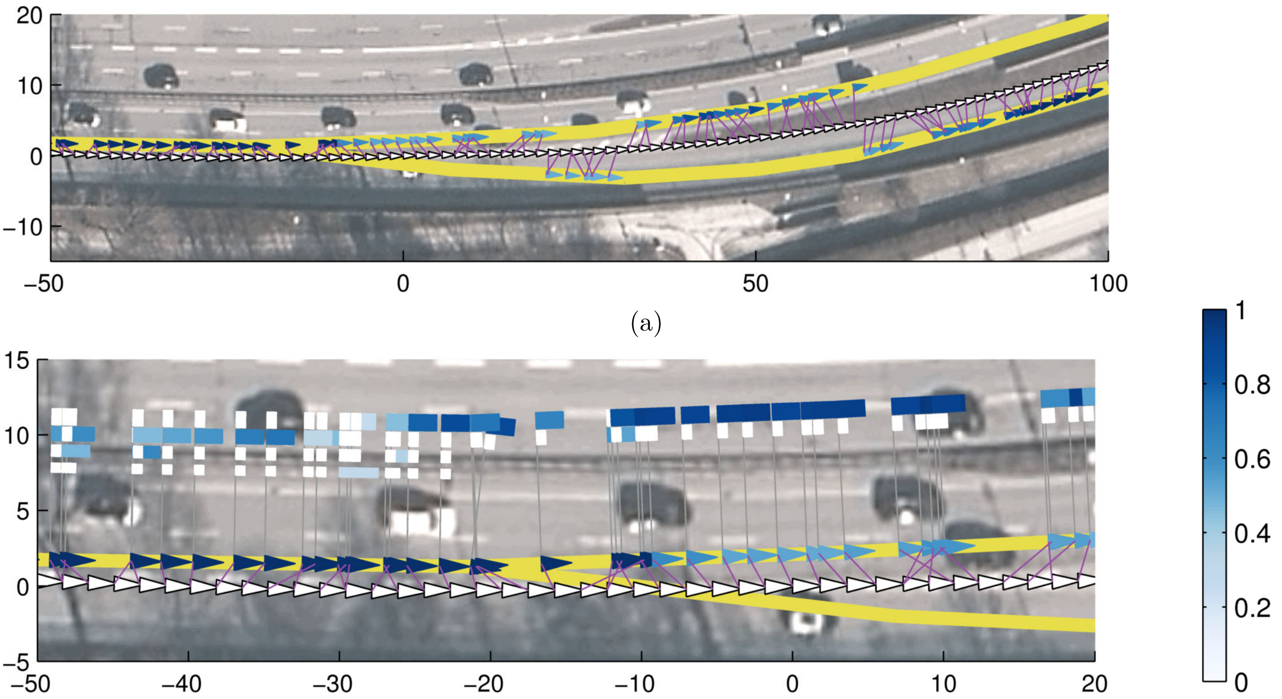
Another situation has been observed where the map-matcher produces a mismatch on the wrong link and is not able to recover from it without reinitialization, Figure 5.38a. The heading and the proximity of the matched link fit very good to the absolute position trace, which causes the particles on the alternative link to die out early. It can be assumed that the absolute position trace is affected by a displacement, because another trace followed a very similar path, but is correctly matched to the other link after the wrong hypothesis was tracked for some distance, Figure 5.38b. Therefore, this situation is crucial, because two similar traces need to be matched to different links.



(a)

(b)

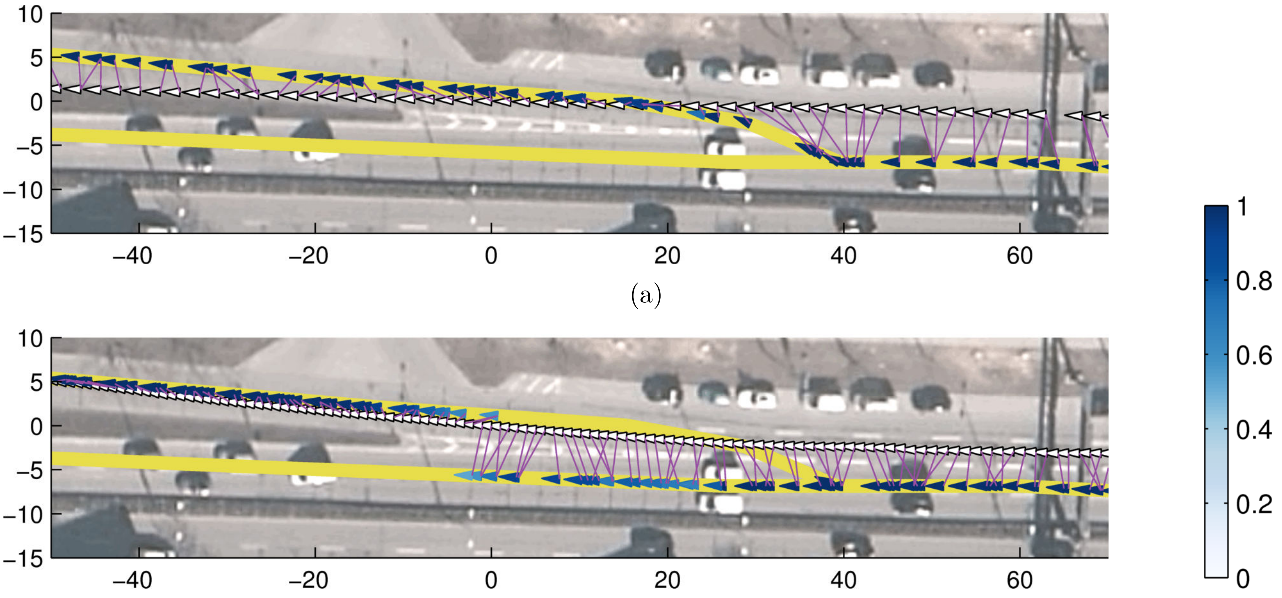
Fig. 5.36 – Scenarios with large discrepancies between the map and the estimated absolute position. (a) The absolute position follows a newly constructed road, which is not contained in the map. (b) Absolute positioning errors occur without degrading the map-matching quality.



(a)

(b)

Fig. 5.37 – Scenarios with discrepancy between the map and the absolute position, which cause ambiguities. (a) The correct map-matched link is the exit link, but the map-matcher produces multiple switches to the alternative link. (b) Zoomed view of the same situation with the matched lane visualized. The matched lane is undecidable before the road split point, because four lanes exist, which increases the ambiguity of the correct road.



(a)

(b)

Fig. 5.38 – Scenarios with results of the link matching showing the limits of the method. (a) Mismatch of the link occurs, because the correct link follows the main road. The map-matcher cannot recover from the mismatch, because no particles survived on the correct link. (b) Same location with a very similar absolute position trace, but in this case the true solution is given by the exit link.

5.6.5 Computational performance

The map-matcher is run on a 2.40 GHz Dual Core Pentium machine onboard the vehicle with output to the integrated navigation display. The map-matched result is generated in real-time at a sampling rate of 10 Hz when up to 100 particles are used. This is verified by observing the output on the navigation display, which produces a smooth output of the result. Since the map-matcher requires constant computation time at each time step for a fixed number of particles, the observed computation time is sufficient to generalize the real-time performance on all traveled routes.

5.6.6 Discussion in the context of related work

The evaluation of the map-matching quality showed that the developed method is well suited for industrial applications. Because of the practical relevance of the map-matching problem, different state-of-the art approaches developed in the past also provide solutions for it. In this section, the results of the proposed map-matching method are reviewed and compared with existing approaches. The proposed quality criteria from the previous sections are applied for comparison to prove the superiority of the developed method in real-world scenarios with industrially applicable sensor systems.

However, a direct comparison between these methods can only be done to a certain extent, because published results were obtained under different conditions. For example as it was shown before, map errors have a large impact on the quality of the map-matched results. In most work, the map error is not quantified, which impedes a direct transfer of the results to different map data. Similarly, the accuracy of the absolute positioning system and the characteristics of the employed sensor system affect the map-matching quality. The implemented map-matcher is tested with absolute positions from fusing GPS and odometry measurements. Other systems incorporate DGPS, or DGPS with odometry or GPS only, which differ in the expected absolute positioning quality. Therefore, the properties of state-of-the art map-matching methods found in literature are compared with the results of the developed method in this work in the categories accuracy, stability, confidence, robustness and computational performance. Table 5.4 provides an overview of the examined alternative methods and the addressed quality criteria.

		accuracy	stability	confidence	robustness	computational performance
Mandel and Laue [116]	Particle filter	*				
Fouque and Bonnifait [52]	Particle filter	*		*		
Davidson et al. [39]	Particle filter	*	*			
Bonnifait et al. [17]	Particle filter			*		*
Thiébaux and Lamb [163]	Kalman filter and Markov localization	*	*			
Mazhelis [118]	Recursive Bayesian estimation	*				
Smaili et al. [154]	Switching Kalman Filter	*				
Hummel [76]	Iterative Bayesian inference in Hidden Markov Model	*			*	*

Tab. 5.4 – Related work on map-matching. An asterisk marks the category that has been addressed in the related work.

The accuracy of map-matching methods from related work is analyzed in similar situations to the ones that were tested in this work. Although it is generally accepted that road bifurcations and parallel roads pose great challenges to any map-matcher, the performance evaluation of other methods does not provide exhaustive examination of the accuracy in such situations, e.g. the work of Mandel and Laue, Fouque and Bonnifait and Thiébaux and Lamb. For the qualitative comparison with these methods it must be assumed that they suffer from map-matching errors in ambiguous situations. Other approaches include tests in ambiguous situations, where they experience temporary mismatches, as the work of Davidson et al. and Mazhelis and Hummel shows. One reason for the degraded performance is due to the fact that road connectivities are not considered for road candidate identification. In this work, outliers and mismatches are avoided, because the link connectivities from the map are included into the road selection during particle sampling. This strategy to take advantage of the map topology is also found in the work of Smaili et al., where good results in ambiguous situations are demonstrated. However, when multiple roads are connected, Hummel demonstrates that ambiguities cannot be eliminated and cause temporary mismatches. The presented work extends the approach of road topology integration to the lane-level, which has proven to increase the accuracy of the road estimate when multiple links are connected (Figure 5.33).

A stability analysis of the results has not gained much attention in related work so far. Benchmarking the developed method with alternative approaches in terms of stability across and along links is therefore not possible. An implicit evaluation of stability can be performed by analyzing examples of trajectories that show the assignment of the absolute position and the map-matched position, which is however missing in most literature. The results of Davidson et al. indicate that jittering along the road occurs, when particle filtering is applied for map-matching. A similar behavior is observed by Thiébaux and Lamb for Markov localization methods. Instabilities across links are naturally present, when temporary mismatches occur. Thus, this work is the first to addresses the problem of instabilities in the context of map-matching and provides a qualitative analysis of the amount of instability.

Map-matching methods that aim for industrial applications should typically provide some kind of confidence measure for the result. The proposed method in this work developed a clustering approach to deliver hypotheses about the map-matched positions and their confidences. Related work does not always explicitly contain a confidence measure. Bonnifait et al. introduced the concept of integrity monitoring of multi-hypothesis map-matching, which provides a confidence region of the map-matched hypotheses along links and allows to classify the situation in terms of ambiguity. However, this approach does not give a decision for the best hypothesis, when the situation has been classified as ambiguous. An extension of this work by Fouque and Bonnifait estimates multiple hypotheses about the map-matched position using Gaussian mixture models, one per estimated road segment. The disadvantage of this approach is that it depends on the discretization of the road network in the underlying map. Mandel and Laue perform a clustering method to estimate the output of the map-matcher, but without computation of a confidence measure. In contrast to related approaches, the proposed clustering method of the present work is suited to determine multiple hypotheses in ambiguous situations together with an interpretable confidence measure.

Industrial map-matching systems require an assessment of the limits of the employed method. This includes the analysis of its robustness considering limit cases of map errors and absolute positioning inaccuracies. Unfortunately, literature on map-matching systems and methods rarely contains hints about the limitations of the methods under adverse conditions. Good examples for limitations of the map-matching method in complex situations are found in the work of Hummel, which presents cases of temporary and long-term mismatches of the road segment. The evaluation in this work is oriented towards real-world applications and includes an accuracy analysis of cases where the map is outdated and large deviations from the true absolute position occur (Figure 5.36). The limits of the proposed method under moderate map and positioning errors are demonstrated in a case where ambiguity could not be eliminated and leads to long-term mismatches of the estimated road (Figure 5.37) or ambiguity is eliminated prematurely (Figure 5.38). These results allow to estimate the potential of the proposed methods for navigation and ADAS systems at an industrial scale. Comparing the failure situations of the map-matcher in the present work and the alternative method of Hummel shows that similar patterns exist. Both methods experience mismatches when the absolute position and heading are more similar to the incorrect link. However, the proposed method in this work has an increased chance to resolve ambiguities correctly, because link and lane connectivities are considered and multiple hypotheses are tracked.

The final criterion for successful application of a map-matching method is given by its computational performance, which needs to fulfill requirements of real-time capability of the respective system. Most relevant map-matchers are tested in real-time at 1 Hz. In this work, the particle filter method runs in real-time at a higher frequency of 10 Hz, which is the frequency of the absolute positioning signals. Alternative particle filter approaches are presented to run in real-time at even higher frequencies e.g. 100 Hz, which was tested by Bonnifait et al. Although this result indicates that particle filtering at such higher frequency is a feasible approach, the advantage of increasing the update rate of the map-matcher to this extend has not been demonstrated yet.

5.6.7 Conclusion

The presented method in this work was evaluated under quality criteria that need to be fulfilled for application of the map-matcher in industrial systems. The evaluation of the map-matched results according to the defined quality criteria has shown that the ambiguity of the situation is a critical factor for the accuracy, stability, confidence and robustness of the solution. Ambiguity of the link matching was found to be caused by the discrepancy between the road geometry of the true link in the map and the absolute position estimate. For lane matching, ambiguity arises when the camera measurements and the lane marking model are not in accordance.

Inaccuracies in the form of mismatches and instabilities caused by switches across links appear only occasionally with short duration at less ambiguous situations. Road bifurcations belong to ambiguous situations and pose challenges to the map-matcher where the risk and the duration of mismatches and instabilities across links increases. However, the intrinsic confidence given by the hypotheses weights is suitable measure for estimating the reliability of the solution.

The results of the map-matching taking into account the correctly matched lane confirm that lane-level map-matching supports the correctness, stability and confidence of the link matching. Again, the confidence of the matched lane is an indicator for the failure of the system. The quality of the lane matching itself strongly depends on the quality of the camera observations and their compatibility with the assumed lane markings model.

Inaccuracies and instabilities along links are within few meters and therefore uncritical for the application purpose of the map-matcher in the field of navigation and ADAS. An increased stability of the solution could be achieved with a postprocessing operation. A spatial filtering method after the hypotheses estimation step can mitigate this noise effect.

The investigation of the robustness of the method at highly ambiguous situations demonstrated the strength of the particle filter method to maintain multiple hypotheses. Although mismatches are likely to occur in such situations, the risk of failures can be anticipated by the confidence measure. A fast consolidation of the correct hypothesis is observed when the situation becomes decidable. The limits of the system are reached, when severe drifts of the absolute position occur or the absolute position is significantly more similar to the incorrect link.

The developed map-matcher is deployed on an onboard vehicle system and operates in real-time at 10 Hz. It is used for research purposes as a functional component within a navigation prototyping framework.

The qualitative comparison of the developed method with related work revealed that suitable benchmarking criteria are unavailable, because they depend on the sensor configuration and map characteristics. Instead, the results of the proposed map-matcher in the five developed quality criteria are compared with reported results of alternative approaches. The developed map-matcher was tested on a representative number and variety of example situations to demonstrate its competitiveness in all categories. Especially the detailed analysis of accuracy and robustness in ambiguous situations demonstrates the advantages of the presented method compared to related work found in literature.

6 Conclusion and outlook

Two particle filter methods were developed to address the individual challenges of airborne vehicle tracking and lane-level map-matching in low-cost sensor systems. The developed methods were evaluated in challenging situations, where the advantage of a multi-hypothesis method is apparent especially when ambiguities occur.

The particle filter for airborne vehicle tracking includes novel adaptive models for target motion and observations. The introduction of an iterative sampling concept allowed to efficiently incorporate current measurements into the motion model. The variance of the observation model is adaptively increased depending on the evolution of the template matching scores. Examples with abrupt target motion and appearance changes demonstrate the increased robustness of the tracker with these adaptive models.

Further strategies to handle target motion discontinuities and appearance changes are introduced. Updating the target template of the observation model has shown to reduce the risk of target losses. The context of the current vehicle is included into the motion model by learning the predicted driving direction from estimated positions of preceding vehicles. This online map learning approach has shown to stabilize the tracker when uncertainty about the target state exists.

The airborne vehicle tracker was evaluated at typical urban scenarios with dense traffic. Image sequences from three different flight systems and cameras, two mounted on an airplane and one on a UAV, are used with a GSD between 5 and 15 cm and a frequency of 2 Hz. The test sequences include wide-area scenarios with up to 151 detected vehicles and sequence lengths of up to 51 seconds. The tracking quality is measured with the MOTA rate, which ranges between 0.69 and 1 with an average of 0.91.

The results show that the tracker performs very well in standard situations with continuous motion and constant appearance. Discontinuous motion is handled in many cases, but very fast accelerations of vehicles can lead to target losses. The tracker is able to adapt to appearance changes caused by pose changes of the vehicles or illumination changes. Only severe and abrupt appearance changes, e.g. when the vehicle enters a shadow or is partly occluded produce mismatches or target losses. Total occlusions pose the highest challenge to the tracker. A good chance exists to re-capture vehicles after they appear again from occlusions caused by trees or bridges. However, rows of trees or large bridges increase the risk of target losses.

In future work, target losses could be handled by re-initialization of the tracker with a vehicle detection method. If only single vehicles are lost within a group of vehicles, the search space of the detector could be bounded by the position of the surrounding vehicles. This vehicle group tracking technique could further improve the performance of the tracker, because the spatial relation of the vehicle group can be used to increase the precision of the motion model. The toleration of appearance changes could be increased, because the spatial constraints would improve the reliability of the motion model.

Since the road map learning technique has demonstrated to increase the robustness, the map definition could be extended to further features of the vehicle motion and appearance. In this way, not only a prediction of the driving direction would be possible, but the motion and observation model could adapt beforehand to shadows, accelerations or occluded areas.

For solving map-matching problems at the lane-level another particle filter method was developed. A novel motion model constrained to the road network was implemented that considers the lane connectivities included in the map to increase the reliability of the lane prediction across junctions. The developed observation model allows to evaluate lane candidates by line marking detections from the vehicle camera system. A detailed analysis of the probabilistic model parameters indicates the preferred parameter settings. To estimate the dominant hypotheses about the map-matched position, a new weighted medoid-shift clustering method was presented that operates on particles in the road network.

The map-matching method was deployed in a series production vehicle sensor system with commercial map data. The system was evaluated in challenging urban situations where ambiguities about the correct link estimate occurs. The quality of the map-matched results was measured in five categories, accuracy, stability, confidence, robustness and performance. An accuracy measure was developed that estimates the map-matching error relative to the imperfectly digitalized map. The results demonstrate that the median of the error in accuracy is less than one meter, even at very ambiguous situations at road bifurcations. Further, the positive effect of the availability of lane estimates on the link estimate accuracy, stability and confidence is demonstrated. The stability analysis has revealed an increased instability of the map-matched position compared to the optimal solution. However, the amount of jittering is mostly below 1.5 meters, which could be controlled with additional filtering mechanisms. The confidence measure from the hypothesis clustering method provides an intrinsic measure of reliability of the computed solution. Examples demonstrate that a low confidence value is a suitable indicator for map-matching failures. Large discrepancies between the map and the estimated absolute position challenge the robustness of the map-matcher. The results demonstrate the advantage of the multi-hypothesis method, because multiple alternatives for the map-matched link and position are maintained until the situation becomes decidable. The performance evaluation of the map-matcher shows that the developed system runs in real-time at a sampling rate of 10 Hz, which is sufficient for typical ADAS applications.

To increase the robustness, further investigations in the direction of reducing the gap between map and absolute positioning should be conducted. This can be achieved by considering results from research on positioning sensor technology or aiming to obtain a map with a higher resolution and accuracy of the digitized road geometry. Another way to improve lane-level map-matching is the integration of additional features of the vehicle environment from camera or radar systems. Further work is needed to define, recognize and map suitable features at a scale that allows commercial deployment.

Methods from control theory that take the distance traveled by the vehicle into account to determine the desired value could stabilize the result. This could be an alternative methodological approach beyond the selected particle filter.

Both developed systems were evaluated on realistic datasets from low-cost sensors, which facilitates their transition to industrial applications. The focus of the performance evaluation was put on challenging, urban scenarios, where both approaches performed well considering the deficiencies of the employed sensors.

Bibliography

[1] F. Abdallah, G. Nassreddine, and T. Denoeux, “A multiple-hypothesis map-matching method suitable for weighted and box-shaped state estimation for localization,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1495–1510, 2011.

[2] B. D. Anderson and J. B. Moore, *Optimal Filtering*. New-Jersey: Prentice-Hall, 1979.

[3] M. S. Arulampalam, S. Maskell, and N. Gordon, “A tutorial on particle filters for online nonlinear/ non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002.

[4] Ascending Technologies GmbH. (2014). Asctec falcon 8, [Online]. Available: http://www.asctec.de/uav-applications/aerial-imaging/asctec-falcon-8/asctec-falcon-8/.

[5] C. L. Azevedo, J. L. Cardoso, M. Ben-Akiva, J. P. Costeira, and M. Marques, “Automatic vehicle trajectory extraction by aerial remote sensing,” *Procedia - Social and Behavioral Sciences*, vol. 111, no. 0, pp. 849 –858, 2014.

[6] D. Bernstein and A. L. Kornhauser, “An introduction to map matching for personal navigation assistants,” Princeton University, TIDE Center, New Jersey.

[7] D. Betaille and R. Toledo-Moreo, “Creating enhanced maps for lane-level vehicle navigation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 786–798, 2010.

[8] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, “A real-time computer vision system for measuring traffic parameters,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 495–501.

[9] J. Black, T. Ellis, and P. Rosin, “A novel method for video tracking performance evaluation,” in *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2003.

[10] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House Publishers, 1999.

[11] S. Blackman, “Multiple hypothesis tracking for multiple target tracking,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[12] BMW Group. (2014). BMW Connected Drive, [Online]. Available: http://www.nzrf.co.nz/techdocs/conferencepapers2013/Mark_Keene_BMW_ConnectedDrive.pdf.

[13] BMW Group. (2015). BMW ConnectedDrive: History, [Online]. Available: http://www.bmw.com.au/com/en/insights/technology/connecteddrive/2010/philosophy/history/index1.html.

[14] BMW Group, *Internal documentation*, 2015.

[15] BMW Group. (2015). Navigation system Professional, [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2013/connectivity_technologies/index.html#navigation.

[16] BMW Group. (2011). Research project Augmented Reality - contact-analogue Head-Up Display, [Online]. Available: https://www.press.bmwgroup.com/global/photoDetail.html?title=research-project-augmented-reality-contact-analogue-head-up-display-10-2011&docNo=P90083062.

[17] P. Bonnifait, J. Laneurit, C. Fouque, and G. Dherbomez, “Multi-hypothesis map-matching using particle filtering,” in *16th ITS World Congress*, 2009.

[18] A. Börner, I. Ernst, M. Ruhé, S. Zujew, and M. Hetscher, “Airborne camera experiments for traffic monitoring,” in *10th Congress International Society for Photogrammetry and Remote Sensing*, 2004.

[19] C. Boucher and J.-C. Noyer, “Hybrid filtering for map-aided vehicle navigation,” in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2006, pp. 46–50.

[20] N. Buch, S. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011.

[21] R. Bucy and K. Senne, “Digital synthesis of non-linear filters,” *Automatica*, vol. 7, no. 3, pp. 287 –298, 1971.

[22] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[23] X. Cao, C. Gao, J. Lan, Y. Yuan, and P. Yan, “Ego motion guided particle filter for vehicle tracking in airborne videos,” *Neurocomputing*, vol. 124, pp. 168–177, 2014.

[24] X. Cao, J. Lan, P. Yan, and X. Li, “KLT feature based vehicle detection and tracking in airborne videos,” in *Sixth International Conference on Image and Graphics*, 2011, pp. 673–678.

[25] X. Cao, Z. Shi, P. Yan, and X. Li, “Tracking vehicles as groups in airborne videos,” *Neurocomputing*, vol. 99, no. 0, pp. 38 –45, 2013.

[26] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. Robert, “Adaptive importance sampling in general mixture classes,” *Statistics and Computing*, vol. 18, no. 4, pp. 447–459, 2008.

[27] D. Cerutti-Maori, I. Sikaneta, and C. Gierull, “Optimum SAR/GMTI processing and its application to the radar satellite RADARSAT-2 for traffic monitoring,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3868–3881, 2012.

[28] C. Chang and R. Ansari, “Kernel particle filter: iterative sampling for efficient visual tracking,” in *International Conference on Image Processing*, vol. 3, 2003, III–977–80 vol.2.

[29] F. Chausse, J. Laneurit, and R. Chapuis, “Vehicle localization on a digital map using particles filtering,” in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 243–248.

[30] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.

[31] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[32] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[33] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000, pp. 142–149.

[34] Continental AG. (2014). Media center, [Online]. Available: http://mediacenter.conti-online.com/internet/generator/MAM/index,templateId=Folder_2FrenderDefault.jsp.html?method=show&action=/details.do&oid=5340540.

[35] Continental AG. (2014). Media center, [Online]. Available: http://mediacenter.conti-online.com/internet/generator/MAM/index,templateId=Folder_2FrenderDefault.jsp.html?method=show&action=/details.do&oid=1578720.

[36] I. J. Cox, “A review of statistical data association techniques for motion correspondence,” *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[37] J. Czyz, B. Ristic, and B. Macq, “A particle filter for joint detection and tracking of color objects,” *Image and Vision Computing*, vol. 25, no. 8, pp. 1271 –1281, 2007.

[38] Daimler AG. (2014). Becker® MAP PILOT, [Online]. Available: http://www.mercedes-benz-accessories.com/content/mba/mpc/mpc_mba_website/en/mpc_home/mba/home/accessories/models/c-class_coupe/map_pilot.0004.html.

[39] P. Davidson, J. Collin, and J. Takala, “Application of particle filters to a map-matching algorithm,” *Gyroscopy and Navigation*, vol. 2, no. 4, pp. 285–292, 2011.

[40] P. Davidson, J. Hautamaki, J. Collin, and J. Takala, “Improved vehicle positioning in urban environment through integration of GPS and low-cost inertial sensors,” in *European Navigation Conference, ENC-GNSS*, 2009.

[41] Digital Photography Review. (2015). Panasonic Lumix DMC-LX7, [Online]. Available: http://www.dpreview.com/products/panasonic/compacts/panasonic_dmclx7/specifications.

[42] DigitalGlobe. (2014). Quickbird, [Online]. Available: https://www.digitalglobe.com/sites/default/files/QuickBird-DS-QB-PROD.pdf.

[43] DigitalGlobe. (2014). WorldView-2, [Online]. Available: https://www.digitalglobe.com/sites/default/files/DG_WorldView2_DS_PROD.pdf.

[44] DigitalGlobe. (2015). WorldView-3, [Online]. Available: https://www.digitalglobe.com/sites/default/files/DG_WorldView3_DS_forWeb_0.pdf.

[45] digitalkamera.de. (2014). Sony NEX-7 Testbericht, [Online]. Available: http://www.digitalkamera.de/Testbericht/Sony_NEX-7/7667.aspx.

[46] R. Douc and O. Cappe, “Comparison of resampling schemes for particle filtering,” in *4th International Symposium on Image and Signal Processing and Analysis*, 2005, pp. 64–69.

[47] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, “An introduction to sequential Monte Carlo methods,” in *Sequential Monte Carlo Methods in Practice*, A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds., New York: Springer-Verlag, 2001.

[48] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *Statistics and Computing*, vol. 10, pp. 197–208, 3 2000.

[49] J. Du and M. Barth, “Next-generation automated vehicle location systems: positioning at the lane level,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 48–57, 2008.

[50] M. El Badaoui El Najjar and P. Bonnifait, “Road selection using multicriteria fusion for the road-matching problem,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 279–291, 2007.

[51] N. Enda, Y. Iwahori, and R. J. Woodham, “Improvement of particle filtering for intersection of targets with similar patterns,” in *IAPR Conference on Machine Vision Applications*, 2009, pp. 223–226.

[52] C. Fouque and P. Bonnifait, “Matching raw GPS measurements on a navigable map without computing a global position,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 887–898, 2012.

[53] D. Fox, “Adapting the sample size in particle filters through KLD-sampling,” *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003.

[54] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte Carlo localization: efficient position estimation for mobile robots,” in *Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, 1999, pp. 343–349.

[55] R. García-García, M. Sotelo, I. Parra, D. Fernández, J. Naranjo, and M. Gavilán, “3D visual odometry for road vehicles,” *Journal of Intelligent and Robotic Systems*, vol. 51, no. 1, pp. 113–134, 2008.

[56] A. Gelb, J. F. Kasper, R. A. Nash, C. F. Price, and A. A. Sutherland, Eds., *Applied Optimal Estimation.* Cambridge, MA: MIT Press, 1974.

[57] Geoawesomeness. (2015). HERE’s Automated Driving Project, [Online]. Available: http://geoawesomeness.com/heres-automated-driving-project-hereinterview/.

[58] J. Georgy, A Noureldin, and C. Goodall, “Vehicle navigator using a mixture particle filter for inertial sensors/odometer/map data/GPS integration,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 544–552, 2012.

[59] German Aerospace Center (DLR). (2014). 3K Camera System, [Online]. Available: http://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-5431/9230_read-17835/.

[60] German Aerospace Center (DLR). (2014). Antares DLR-H2, [Online]. Available: http://www.dlr.de/tt/desktopdefault.aspx/tabid-4935/8219_read-13587/.

[61] German Aerospace Center (DLR). (2014). Research aircraft, [Online]. Available: http://www.dlr.de/dlr/en/desktopdefault.aspx/tabid-10203/.

[62] German Aerospace Center (DLR). (2014). Vabene, [Online]. Available: vabene.dlr.de.

[63] J. Geweke, “Bayesian inference in econometric models using Monte Carlo integration,” *Econometrica*, vol. 57, no. 6, pp. 1317–39, 1989.

[64] A. J. Goldman, “Optimal center location in simple networks,” *Transportation Science*, vol. 5, no. 2, pp. 212–221, 1971.

[65] N. Gordon, D. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings F on Radar and Signal Processing*, vol. 140, no. 2, pp. 107–113, 1993.

[66] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[67] M. Gruber, M. Ponticelli, S. Bernoegger, and F. Leberl, “UltraCamX, the large format digital aerial camera system by Vexcel Imaging / Microsoft,” in *21st International Congress on Photogrammetry and Remote Sensing*, ser. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVII, ISPRS, 2008, pp. 665–670.

[68] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation, and tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, 2002.

[69] S. Hinz, D. Lenhart, and J. Leitloff, “Detection and tracking of vehicles in low framerate arial image sequences,” in *ISPRS Hannover Workshop on High-Resolution Earth Imaging for Geospatial Information*, C. Heipke, K. Jacobsen, and M. Gerke, Eds., 2007.

[70] S. Hinz, D. Weihing, S. Suchandt, and R. Bamler, “Detection and velocity estimation of moving vehicles in high-resolution spaceborne synthetic aperture radar data,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–6.

[71] S. Hinz, D. Lenhart, and J. Leitloff, “Traffic extraction and characterisation from optical remote sensing data,” *The Photogrammetric Record*, vol. 23, no. 124, pp. 424–440, 2008.

[72] S. Hinz, F. Meyer, M. Eineder, and R. Bamler, “Traffic monitoring with spaceborne SAR-theory, simulations, and experiments,” *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 231–244, 2007.

[73] J. Hol, T. Schon, and F. Gustafsson, “On resampling algorithms for particle filters,” in *IEEE Nonlinear Statistical Signal Processing Workshop*, 2006, pp. 79–82.

[74] S. Hoogendoorn and R. Hoogendoorn, “Calibration of microscopic traffic-flow models using multiple data sources,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4497–4517, 2010.

[75] C. Hue, J.-P. Le Cadre, and P. Pérez, “Sequential Monte Carlo methods for multiple target tracking and data fusion,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 309–325, 2002.

[76] B. Hummel, “Map matching for vehicle guidance,” in *Dynamic and Mobile GIS*, ser. Innovations in GIS, CRC Press, Nov. 2006.

[77] D. Huynh, “Metrics for 3D rotations: comparison and analysis,” *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[78] International Organization for Standardization, *Intelligent transport systems – Geographic Data Files (GDF) – GDF5.0*, ISO 14825, 2011.

[79] M. Isard and A. Blake, “CONDENSATION-conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[80] M. Jabbour, P. Bonnifait, and V. Cherfaoui, “Map-matching integrity using multihypothesis road-tracking,” *Journal of Intelligent Transportation Systems*, vol. 12, no. 4, pp. 189–201, 2008.

[81] T. Kailath, “The divergence and Bhattacharyya distance measures in signal selection,” *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.

[82] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[83] W.-W. Kao, “Integration of GPS and dead-reckoning navigation systems,” in *Vehicle Navigation and Information Systems Conference*, vol. 2, 1991, pp. 635–643.

[84] E. Kaplan, *Understanding GPS - Principles and applications*, 2nd edition. Artech House, 2005.

[85] O. Kariv and S. Hakimi, “An algorithmic approach to network location problems. II: the p-medians,” *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.

[86] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, “Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319–336, 2009.

[87] A. Kelly, “Linearized error propagation in odometry,” *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 179–218, 2004.

[88] Z. Khan, T. Balch, and F. Dellaert, “MCMC-based particle filtering for tracking a variable number of interacting targets,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805–1819, 2005.

[89] G. Kitagawa, “Monte Carlo filter and smoother for non-Gaussian nonlinear state space models,” *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[90] G. Kitagawa, “Non-Gaussian state-space modeling of nonstationary time series,” *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1032–1041, 1987.

[91] J. Kittler, M. Hatef, R. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[92] A. Kong, J. S. Liu, and W. H. Wong, “Sequential imputations and Bayesian missing data problems,” *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, 1994.

[93] F. Kost, “Grundlagen der Fahrphysik,” in *Fahrstabilisierungssysteme und Fahrerassistenzsysteme*, K. Reif, Ed., Vieweg+Teubner, 2010, pp. 20–33.

[94] K. Kozempel and R. Reulke, “Fast vehicle detection and tracking in aerial image bursts,” in *Object Extraction for 3D City Models, Road Databases and Traffic Monitoring - Concepts, Algorithms and Evaluation*, 2009, pp. 175–180.

[95] T. Krauß, R. Statter, R. Philipp, and S. Brauninger, “Traffic flow estimation from single satellite images,” in *SMPR Conference*, H. Arefi, M. A. Sharifi, P. Reinartz, and M. R. Delavar, Eds., vol. XL-1/W3, 2013, pp. 241–246.

[96] G. Kravaritis and B. Mulgrew, “Variable-mass particle filter for road-constrained vehicle tracking,” *EURASIP Journal on Advances in Signal Processing*, vol. 2008, no. 1, p. 321 967, 2008.

[97] C. Kreucher, K. Kastella, and A. Hero, “Multitarget tracking using the joint multitarget probability density,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1396–1414, 2005.

[98] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[99] P. Kumar, M. Brooks, and A. van den Hengel, “An adaptive Bayesian technique for tracking multiple objects,” in *Pattern Recognition and Machine Intelligence*, ser. Lecture Notes in Computer Science, A. Ghosh, R. De, and S. Pal, Eds., vol. 4815, Springer Berlin Heidelberg, 2007, pp. 657–665.

[100] H. Kuntze, C. Frey, I. Tchouchenkov, B. Staehle, E. Rome, K. Pfeiffer, A. Wenzel, and J. Wollenstein, “Seneka - sensor network with mobile robots for disaster management,” in *IEEE Conference on Technologies for Homeland Security*, 2012, pp. 406–410.

[101] F. Kurz, D. Rosenbaum, O. Meynberg, G. Mattyus, and P. Reinartz, “Performance of a real-time sensor and processing system on a helicopter,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 189–193, Nov. 2014.

[102] F. Kurz, D. Rosenbaum, J. Leitloff, O. Meynberg, and P. Reinartz, "Real time camera system for disaster and traffic monitoring," in *International Conference on Sensors and Models in Photogrammetry and Remote Sensing*, editor, Ed., Teheran, Iran, 2011, pp. 1–6.

[103] F. Kurz, S. Türmer, O. Meynberg, D. Rosenbaum, H. Runge, P. Reinartz, and J. Leitloff, "Low-cost optical camera systems for real-time mapping applications," *PFG Photogrammetrie, Fernerkundung, Geoinformation*, vol. 2012, no. 2, pp. 159–176, May 2012.

[104] T. Laue and T. Röfer, "Pose extraction from sample sets in robot self-localization - a comparison and a novel approach," in *4th European Conference on Mobile Robots*, I. Petrović and A. J. Lilienthal, Eds., 2009, pp. 283–288.

[105] J. Leitloff, S. Hinz, and U. Stilla, "Vehicle detection in very high resolution satellite images of city areas," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 7, pp. 2795–2806, 2010.

[106] D. Lenhart, S. Hinz, J. Leitloff, and U. Stilla, "Automatic traffic monitoring based on aerial image sequences," *Pattern Recognition and Image Analysis*, vol. 18, no. 3, pp. 400–405, 2008.

[107] H. Li, F. Nashashibi, and G. Toulminet, "Localization for intelligent vehicle by fusing mono-camera, low-cost gps and map data," in *International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1657–1662.

[108] Q. Li, B. Lei, Y. Yu, and R. Hou, "Real-time highway traffic information extraction based on airborne video," in *12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1–6.

[109] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking: dynamic models," *Proc. SPIE*, vol. 4048, pp. 212–235, 2000.

[110] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: a cascade particle filter with discriminative observers of different life spans," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1728–1740, 2008.

[111] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[112] J. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statistics and Computing*, vol. 6, no. 2, pp. 113–119, 1996.

[113] N. Longbotham, C. Bleiler, C. Chaapel, C. Padwick, W. Emery, and F. Pacifici, "Spatial classification of WorldView-2 multi-angle sequence," in *Joint Urban Remote Sensing Event*, 2011, pp. 105–108.

[114] E. Maggio, F. Smerladi, and A. Cavallaro, "Adaptive multifeature tracking in a particle filtering framework," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1348–1359, 2007.

[115] E. Maggio and A. Cavallaro, *Video Tracking - Theory and Practice*. Wiley, 2011.

[116] C. Mandel and T. Laue, "Particle filter-based position estimation in road networks using digital elevation models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5744–5749.

[117] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[118] O. Mazhelis, "Using recursive Bayesian estimation for matching GPS measurements to imperfect road network data," in *International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1492–1497.

[119] L. Meng and J. Kerekes, "Object tracking using high resolution satellite imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 1, pp. 146–152, 2012.

[120] Microsoft. (2014). UltraCam Eagle, [Online]. Available: http://www.microsoft.com/ultracam/en-us/UltraCamEagleSpecs.aspx.

[121] M. Mitschke and H. Wallentowitz, "Reifen," in *Dynamik der Kraftfahrzeuge*, Springer Fachmedien Wiesbaden, 2014, pp. 9–54.

[122] B. Morris and M. Trivedi, “Learning, modeling, and classification of vehicle track patterns from live video,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, pp. 425–437, 2008.

[123] MVTec Software GmbH, *HALCON Reference - find_shape_model*, 2012.

[124] F. K. Nejadasl, B. G. Gorte, and S. P. Hoogendoorn, “Optical flow based vehicle tracking strengthened by statistical decisions,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 61, no. 3-4, pp. 159 –169, 2006.

[125] NovAtel Inc. (2014). FlexPak-G2 OEMStar, [Online]. Available: http://www.novatel.com/products/gnss-receivers/enclosures/flexpak-g2-with-oemstar/.

[126] M.-S. Oh and J. O. Berger, “Adaptive importance sampling in Monte Carlo integration,” *Journal of Statistical Computation and Simulation*, vol. 41, no. 3-4, pp. 143–168, 1992.

[127] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, “A boosted particle filter: multitarget detection and tracking,” in *Computer Vision - ECCV 2004*, ser. Lecture Notes in Computer Science, T. Pajdla and J. Matas, Eds., vol. 3021, Springer Berlin Heidelberg, 2004, pp. 28–39.

[128] Oxford Technical Solutions. (2014). RT3000, [Online]. Available: http://www.oxts.com/Downloads/Products/RT3000/RT3000.pdf.

[129] P. Pan and D. Schonfeld, “Video tracking based on sequential particle filtering on graphs,” *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1641–1651, 2011.

[130] A. Peker, O. Tosun, and T. Acarman, “Particle filter vehicle localization and map-matching using map topology,” in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 248–253.

[131] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, “You’ll never walk alone: modeling social behavior for multi-target tracking,” in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 261–268.

[132] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based probabilistic tracking,” in *Computer Vision - ECCV 2002*, ser. Lecture Notes in Computer Science, A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, Eds., vol. 2350, Springer Berlin Heidelberg, 2002, pp. 661–675.

[133] O. Pink and B. Hummel, “A statistical approach to map matching using road network geometry, topology and vehicular motion constraints,” in *11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 862–867.

[134] M. K. Pitt and N. Shephard, “Filtering via simulation: auxiliary particle filters,” *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[135] U. Pless, A. Somieski, M. Ammann, and C. Fenger. (2014). U-blox’ 3D dead reckoning for vehicle applications. u-blox AG, Ed., [Online]. Available: http://www.u-blox.com/images/downloads/Product_Docs/u-blox_3D_Dead_Reckoning_for_Vehicle_Applications.pdf.

[136] J.-S. Pyo, D.-H. Shin, and T.-K. Sung, “Development of a map matching method using the multiple hypothesis technique,” in *IEEE Intelligent Transportation Systems*, 2001, pp. 23–27.

[137] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, “Integrity of map-matching algorithms,” *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 4, pp. 283 –302, 2006.

[138] M. Quddus and N. Velaga, “Enhancing vehicle positioning data through map-matching,” in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed., Springer London, 2012, pp. 343–364.

[139] R. Rad and M. Jamzad, “Real time classification and tracking of multiple vehicles in highways,” *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1597 –1607, 2005.

[140] J. Ren and J. Hao, “Mean shift tracking algorithm combined with Kalman filter,” in *5th International Congress on Image and Signal Processing*, 2012, pp. 727–730.

[141] D. B. Rubin, “Using the SIR algorithm to simulate posterior distributions,” in *Bayesian Statistics 3*, M. H. Bernardo, K. M. Degroot, D. V. Lindley, and A. F. M. Smith, Eds., Oxford University Press, 1988, pp. 395–402.

[142] H. Runge, J. Kallo, P. Rathke, T. Stephan, F. Kurz, D. Rosenbaum, and O. Meynberg, “Chicago - an airborne observation system for security applications,” in *Future Security*, ser. Commu-

nications in Computer and Information Science, N. Aschenbruck, P. Martini, M. Meier, and J. Tölle, Eds., vol. 318, Springer Berlin Heidelberg, 2012, pp. 488–492.

[143] J. Saboune and R. Laganiere, “People detection and tracking using the explorative particle filtering,” in *IEEE 12th International Conference on Computer Vision Workshops*, 2009, pp. 1298–1305.

[144] R. Sandau, “Introduction,” in *Digital Airborne Camera*, R. Sandau, Ed., Springer Netherlands, 2010, pp. 1–30.

[145] R. Sandau, “Structure of a digital airborne camera,” in *Digital Airborne Camera*, R. Sandau, Ed., Springer Netherlands, 2010, pp. 143–259.

[146] T. Schon, R. Karlsson, and F. Gustafsson, “The marginalized particle filter in practice,” in *IEEE Aerospace Conference*, 2006.

[147] D. Schramm, M. Hiller, and R. Bardini, “Single track models,” in *Vehicle Dynamics*, Springer Berlin Heidelberg, 2014, pp. 223–253.

[148] D. Schulz, W. Burgard, D. Fox, and A. Cremers, “Tracking multiple moving targets with a mobile robot using particle filters and statistical data association,” in *IEEE International Conference on Robotics and Automation*, vol. 2, 2001, pp. 1665–1670.

[149] A Selloum, D. Betaille, E. Le Carpentier, and F. Peyret, “Lane level positioning using particle filtering,” in *12th International IEEE Conference on Intelligent Transportation Systems*, 2009, pp. 1–6.

[150] L. Shapira, S. Avidan, and A Shamir, “Mode-detection via median-shift,” in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 1909–1916.

[151] Y. Sheikh, E. Khan, and T. Kanade, “Mode-seeking by medoidshifts,” in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[152] J. Shi and C. Tomasi, “Good features to track,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[153] P. Skoglar, U. Orguner, D. Tornqvist, and F. Gustafsson, “Road target tracking with an approximative Rao-Blackwellized particle filter,” in *12th International Conference on Information Fusion*, 2009, pp. 17–24.

[154] C. Smaili, M. Najjar, and F. Charpillet, “A hybrid Bayesian framework for map matching: formulation using switching Kalman filter,” *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 725–743, 2014.

[155] V. Smidl and R. Hofman, “Adaptive importance sampling in particle filtering,” in *16th International Conference on Information Fusion*, 2013, pp. 9–16.

[156] Sony. (2014). Sony NEX-7, [Online]. Available: http://www.sony.co.uk/electronics/interchangeable-lens-cameras/nex-7-body-kit/specifications.

[157] A. Soto, “Self adaptive particle filter,” in *19th International Joint Conference on Artificial Intelligence*, 2005, pp. 1398–1403.

[158] C. Steger, “Similarity measures for occlusion, clutter, and illumination invariant object recognition,” in *23rd DAGM-Symposium on Pattern Recognition*, B. Radig and S. Florczyk, Eds., ser. Lecture Notes in Computer Science, vol. 2191, Springer, 2001, pp. 148–154.

[159] C. Steger, “System and method for object recognition,” pat. EP 1193642, 2000.

[160] J. Stephen and G. Lachapelle, “Development of a GNSS-based multi-sensor vehicle navigation system,” in *2000 National Technical Meeting of The Institute of Navigation*, 2000, pp. 268–278.

[161] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, “Lane marking aided vehicle localization,” in *16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 1509–1515.

[162] Z. Tao, P. Bonnifait, V. Fremont, and J. Ibanez-Guzman, “Mapping and localization using GPS, lane markings and proprioceptive sensors,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 406–412.

[163] S. Thiébaux and P. Lamb, “Combining Kalman filtering and Markov localization in network-like environments,” in *PRICAI 2000 Topics in Artificial Intelligence*, ser. Lecture Notes in

Computer Science, R. Mizoguchi and J. Slaney, Eds., vol. 1886, Springer Berlin Heidelberg, 2000, pp. 756–766.

[164] R. Toledo-Moreo, D. Betaille, F. Peyret, and J. Laneurit, “Fusing GNSS, dead-reckoning, and enhanced maps for road vehicle lane-level navigation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 798–809, 2009.

[165] Toll Collect GmbH. (2014). Toll collect, [Online]. Available: http://www.toll-collect.de/rund-um-ihre-maut/mautkontrolle.html.

[166] u-blox AG. (2009). GPS - Essentials of Satellite Navigation, [Online]. Available: http://www.u-blox.com/images/stories/Resources/gps_compendiumgps-x-02007.pdf.

[167] N. R. Velaga, M. A. Quddus, and A. L. Bristow, “Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems,” *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 6, pp. 672 –683, 2009.

[168] J. Vermaak, S. Godsill, and P. Perez, “Monte Carlo filtering for multi target tracking and data association,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 309–332, 2005.

[169] VIB Verkehrsinformationsagentur Bayern GmbH. (2014). Webcams Stadt Nürnberg. Bayerisches Staatsministerium des Innern, für Bau und Verkehr sowie Konsortium VIB und VIB Verkehrsinformationsagentur Bayern GmbH, Ed., [Online]. Available: http://www.bayerninfo.de/webcams/webcams-stadt-nuernberg.

[170] J. Wagner, C. Isert, A Purschwitz, and A Kistner, “Improved vehicle positioning for indoor navigation in parking garages through commercially available maps,” in *International Conference on Indoor Positioning and Indoor Navigation*, 2010, pp. 1–8.

[171] C. E. White, D. Bernstein, and A. L. Kornhauser, “Some map matching algorithms for personal navigation assistants,” *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1 -6, pp. 91–108, 2000.

[172] A. Wiechert and M. Gruber, “News from the ultracam camera line-up,” in *Proceedings of American Society for Photogrammetry and Remote Sensing 2013 Annual Conference*, 2013.

[173] J. Xiao, H. Cheng, H. Sawhney, and F. Han, “Vehicle detection and tracking in wide field-of-view aerial video,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 679–684.

[174] C. Yang, R. Duraiswami, and L. Davis, “Fast multiple object tracking via a hierarchical particle filter,” in *Tenth IEEE International Conference on Computer Vision*, vol. 1, 2005, pp. 212–219.

[175] Z. Yin and R. Collins, “Object tracking and detection after occlusion via numerical hybrid local and global mode-seeking,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[176] Q. Yu and G. Medioni, “Motion pattern interpretation and detection for tracking moving vehicles in airborne video,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2671–2678.

[177] S. Zhou, R. Chellappa, and B. Moghaddam, “Visual tracking and recognition using appearance-adaptive models in particle filters,” *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[178] X. Zhou and Y. Lu, “Abrupt motion tracking via adaptive stochastic approximation Monte Carlo sampling,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1847–1854.