Context Exploitation in Data Fusion

by

Lubos Vaci

Dissertation submitted to the
Polytechnic Department of Engineering and Architecture
of the University of Udine
in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in Industrial and Information Engineering

XXX Cycle

Supervisor:

Assist. Prof. Lauro Snidaro

Context Exploitation in Data Fusion

Lubos Vaci

Academic Year 2016/2017

To those who contributed to this work,

Abstract

Complex and dynamic environments constitute a challenge for existing tracking algorithms. For this reason, modern solutions are trying to utilize any available information which could help to constrain, improve or explain the measurements. So called Context Information (CI) is understood as information that surrounds an element of interest, whose knowledge may help understanding the (estimated) situation and also in reacting to that situation. However, context discovery and exploitation are still largely unexplored research topics.

Until now, the context has been extensively exploited as a parameter in system and measurement models which led to the development of numerous approaches for the linear or non-linear constrained estimation and target tracking. More specifically, the spatial or static context is the most common source of the ambient information, i.e. features, utilized for recursive enhancement of the state variables either in the prediction or the measurement update of the filters. In the case of multiple model estimators, context can not only be related to the state but also to a certain mode of the filter. Common practice for multiple model scenarios is to represent states and context as a joint distribution of Gaussian mixtures. These approaches are commonly referred as the join tracking and classification. Alternatively, the usefulness of context was also demonstrated in aiding the measurement data association. Process of formulating a hypothesis, which assigns a particular measurement to the track, is traditionally governed by the empirical knowledge of the noise characteristics of sensors and operating environment, i.e. probability of detection, false alarm, clutter noise, which can be further enhanced by conditioning on context.

We believe that interactions between the environment and the object could be classified into actions, activities and intents, and formed into structured graphs with contextual links translated into arcs. By learning the environment model we will be able to make prediction on the targets future actions based on its past observation. Probability of target future action could be utilized in the fusion process to adjust tracker confidence on measurements. By incorporating contextual knowledge of the environment, in the form of a likelihood function, in the filter measurement update step, we have been able to reduce uncertainties of the tracking solution and improve the consistency of the track. The promising results demonstrate that the fusion of CI brings a significant performance improvement in comparison to the regular tracking approaches.

Acknowledgments

I would like to express my sincere gratitude to my thesis supervisor Assist. Prof. Lauro Snidaro for his support, patience, encouragement and guidance throughout the study. I greatly appreciate his share in every step taken in the development of the thesis.

I would also like to thank Prof. Gian Luca Foresti for creating a good atmosphere in the group. The coordinators of my thesis committee, Prof. Paolo Gardonio and Prof. Luca Selmi, deserve my gratitude for their support and suggestions. Our secretary Diana Zannier was always arranging practical matters in a smooth way and I would like to thank her as well for her assistance and kindness.

A special thanks belongs to Dr. Nikki Martinel for his encouragement, endless discussions and in overall for being such a good friend!

I am glad for having an opportunity to work, share my time and ideas with other members of the AViReS laboratory, especially with Prof. Christian Micheloni, Assist. Prof. Claudio Piciarelli, Dr. Manuela Farinosi, Dr. Marco Vernier, Dr. Zahid Akhtar, Dr. Sanoj Saini, Nadja Yakusheva, Matteo Chini, Mattia Zanier, Marco Tortolo and Shadi Abpeykar. Thank you all for making my stay in the group such an inspiring and memorable experience!

Part of this work was conducted in collaboration with Magneti Marelli s.p.a., and therefore I would like to thank Dr. Axel Furlan and Giuseppe Giorgio for their assistance and fruitful discussions during the course of this project.

I would like to thank all other Ph.D. fellows, with whom I shared this journey, for accompanying me during the late times in the office and for the free time we spent together. Thank you Francesco, Alessandro, Dario, Filip and Marco!

I will also never forget the unending support and love my family has provided me all my life!

Contents

List of Figures

List of Tables

Preface

This thesis is the result of the author's 3-year Ph.D. study under the supervision of Assist. Prof. Lauro Snidaro. First three chapters are independent research topics on the general framework of context-aware target tacking. The last chapter is a result of a cooperation conducted in collaboration with Magneti Marelli s.p.a. on the subject of fusion of sensory and contextual data for environment perception and mapping on autonomous vehicles. While writing about the conducted research, I tried my best to integrate the material into a single compact study on the context exploitation in data fusion. Part of this research has been already submitted to academic conferences for publication, while the rest is expected to be published in the near future. The following is a list of these publications:

- L. Snidaro, L. Vaci, J. Garcia, E. D. Marti, A.-L. Jousselme, K. Bryan, D. D. Bloisi, and D. Nardi. A framework for dynamic context exploitation. *In Proceedings of 18th International Conference on Information Fusion*, volume 1, pages 1-18, Washington, DC, 2015. IEEE. [135]
- L. Vaci, L. Snidaro, and G. L. Foresti. Encoding context likelihood functions as classifiers in particle filters for target tracking. *In Proceedings on Multisensor Fusion and Integration*, pages 310-315, Baden-Baden, Germany, 2016. IEEE. [153]

The main theme of this work evolves around the idea of modeling contextual information as individual entities' actions and intents and exploitation of this inferred knowledge in an information fusion process. Design aspects and issues to be considered when developing such a context-aware information fusion system, capable of dynamical context exploitation across all levels of the fusion, are discussed in Sec. 3.

There are two main challenges associated with the context exploitation in the domain of information fusion. First, how could be contextual information learned from the environment and subsequently inferred in the process of reasoning. Second, how could be CI utilized in the process of data fusion. These tasks can not be completely separated from each other as the capability to accurately infer on context siphons into enhanced data fusion performance and vice versa. The applicability of the later into the Bayesian recursion becomes the essence of this work as problems of CI learning, classification and reasoning are already well established in the literature.

The initial attempt for building the context-adaptive framework is based on the seminal work of W. Koch & M. Ulmke [152] on constrains exploitation in the Bayesian filtering. A modification of this approach, that encodes different types of contextual information as likelihood functions via classifiers in particle filters, discussed in the Sec. 4, was presented as a solution for multi-level context representation and exploitation. However, this method is only concerned about the problem of detection and exploitation of the spatial context and not at all capable of utilizing CI at higher levels of abstraction. It therefore become apparent, that the data fusion architecture that respects the spatio-temporal relationships of the events and successfully utilizes this new knowledge in estimation process needs to be developed.

The idea of seeing target's actions as a probabilistic graphical model [49] with the relevant spatial and event-temporal context associated with each node of the graph became a backbone of the framework developed in Sec. 5. The belief that a target seeks an objective is used in aiding of the tracking by adjusting mode transition probabilities in the interactive multiple models estimation process. The mode transition probabilities are estimated recursively during the course of tracking [68]. The proposed technique utilizes a quasi-Bayesian estimator in order to predict the mode transition matrix of the IMM in terms of the multiple models mode probabilities and the joint likelihood of contextual information and measurements.

The exploitation of context is not exclusive to the Bayesian framework but could be also applied to theories used for reasoning under a uncertainty such as the Dempster-Shafer theory. As an example, a concept for the context-aware grid-based fusion and occupancy mapping designed for the autonomous system perception is discussed in Sec. 6. By realizing that occupancy-grid techniques do not utilize predefinitions of map features thus completely avoid the data association problem, they also weight the measurement grids equally unless specified otherwise by the user. When sensors readings are highly contrasting, the fusion of grids becomes increasingly more challenging. It therefore becomes apparent, that sensor weights needs to be adjusted recursively during the fusion process. Strategies that exploits discrepancies between the grid maps in such a manner where the fusion of contradicting information become less susceptible to sensor weighting are discussed and evaluated on the real world scenario in Sec. 6.

As a final remark, the process of forming this Ph.D. thesis was not an easy one but full of inspiration and challenges. As the quote of Henry Ford says: *"Obstacles are those frightful things you see when you take your eyes off your goal"*, the challenges here faced were many but the desire to overcome them made this work become a reality and made ourself stronger and better in the process.

1

Introduction

> *Contextual Information could be said to be that information that "surrounds" a situation of interest in the world. It is information that aids in understanding the (estimated) situation and also aids in reacting to the situation, if a reaction is required [134].*

1.1 Motivation

Modern surveillance systems almost exclusively rely on the target tracking techniques in order to recognize the moving objects of interest within an environment. The objective of tracking algorithm is to partition sensor readings into sets of observations, or tracks, originated from the same source and to provide location of that source, in a form of timely ordered sequence, for further processing. Therefore, target tracking can be seen as a generalization of the dynamic estimation theory, i.e. the process of inferring the value of a quantity of interest from indirect, inaccurate and uncertain observations, for the cases where:

- objects of interest are continuously changing their dynamics,
- received measurements does not necessary originate from the particular object,
- measurements could be faulty and/or missing,
- initial guesses of object states, such as position, velocity or attitude, might not be available.

Tracking objects in complex and noisy dynamic environments pose a challenge for any existing tracking algorithm. For this reason, modern solutions are trying to utilize any available information which could help to constrain, improve or explain the measurements. So called context (CI), is understood as information that "surrounds" a situation of interest in the world. It is information that aids in understanding the estimated situation and also aids in reacting to the situation, if a reaction is required [134].

For instance, maritime traffic surveillance applications require extended and continuous tracking of vessels far beyond territorial waters and within several sensor coverage areas [10]. Spatial coverage gaps of sensors, extended periods when measurements are missing or are highly cluttered, and intermittent nature of cooperative sources pose multiple challenges to conventional tracking algorithms. In order to improve track accuracy and continuity the navigation field concept, a mathematical model for the inclusion of contextual information, was introduced into maritime target tracking systems (Fig. 1.1). In this scenario, the context is encoded as the navigation force field which effects the superposition of forces acting on transiting vessel. Strength and direction of the navigation force field depend on the relative position of target w.r.t. source points, which can be either attractive (blue) or repulsive (red). These forces might be seen as representation of engine or rudder induced forces acting on the ship. It is reported in [10], that occurrence of track losses was reduced by over 15% in presence of the context.

The use of road maps and terrain information to enhance the performance of ground target tracking with GMTI (ground moving target indicator), in so called road matching framework, was presented in [152]. As shown in Fig. 1.2, road networks of the road sample can be decomposed into eleven edges (roads) and ten nodes distinguished as five junctions (•), three sources (○), two road signs (⊗) confining a bridge. At first, contextual information about roads is incorporated into target kinematics as a constraining factor. At second, detectability of a target is evaluated on bases of terrain visibility and the clutter notch of the sensor. At last, target kinematics is approximated by a number of so called "random walkers" generated at

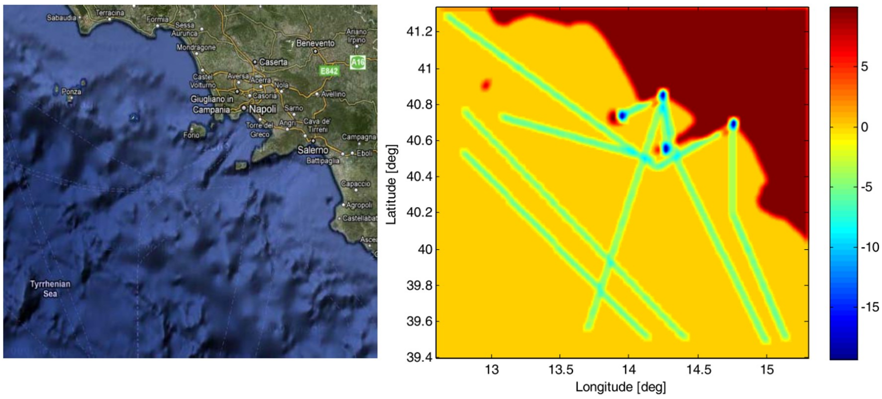
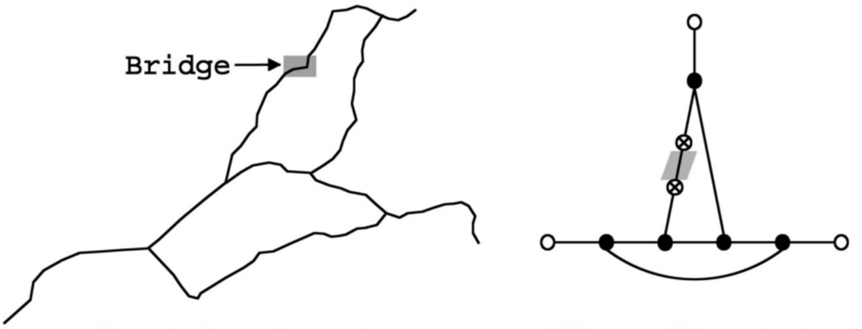


Figure 1.1: Context as an intensity map of the maritime traffic [10].



a) Sample of a road network

b) Diagram of the road network

Figure 1.2: Context as a road map diagram [152].

every intersection. Each walker propagates along the road(s) according to the target kinematics and road type. The weight of each walker is calculated after the measurement update of the tracker and one with the highest score is declared to be the true target. Simulation results performed on a realistic ground scenario show strongly reduced target location errors compared with the case where a road-map information is neglected. Furthermore, detection of stopping targets was beneficently reduced by modeling the clutter notch of the GMTI sensor.

Away from tracking, in the domain of autonomous driving systems, the context information have been recently considered for the purpose of evidential grid mapping [83]. In an autonomous driving scenario, it is of a crucial importance to distinguish static objects attached to infrastructure, e.g. road sign, from mobile road users, e.g. cars, cyclist or pedestrians. Furthermore, autonomous driving systems are tasked to distinguished the navigable spaces in the their vicinity and safety navigate through the newly built map (Fig. 1.3). In the figure, a LiDAR perception (sensor grid) scheme enhanced by geo-referenced maps (GIS or context grid) is used to capture complex dynamic environments by a multi-grid fusion framework. An adaptation of the conjunctive combination Dempster-Shafer rule is employed to resolve the conflicts occurring in the process of mapping. The method relies on temporal accumulation to make the distinction between stationary and moving objects, and applies contextual discounting in order to model information obsolescence. As a result, proposed method is able to better characterize the state of the occupied cells by differentiating moving objects, parked cars, urban infrastructure and buildings. Another benefit of this context-aware approach is the capability to separate the navigable spaces from the non-navigable ones.

These examples demonstrate the usefulness of contextual information for purposes of target tracking and data fusion in general. The exploitation of context has been a very active research topic for over a decade and the key element in design of any modern surveillance system. In this study, context information deduced from a priori observed target behavior will be further analyzed and evaluated.

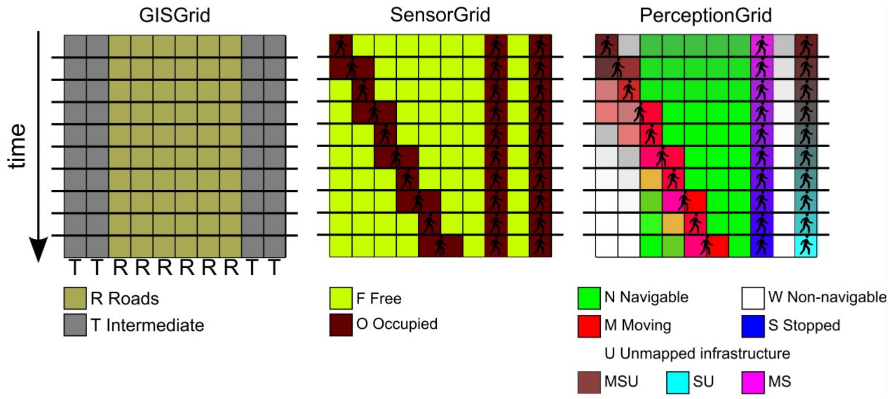


Figure 1.3: Context in support of environment map building [83].

1.2 Exploitation of Contextual Information in Information Fusion

Latest issues and challenges associated with the context based information fusion were recently surveyed and discussed in [16] and [134]. Authors concluded, that understanding and principled exploitation of context in information fusion (IF) systems is still rather limited. They claim, that domain knowledge has been traditionally acquired ad hoc from experts or static archives and applied to stove-piped solutions that could hardly scale or adapt to new conditions. Authors envision, that context, considered as locally relevant (and possibly dynamic) domain knowledge, should play a key role at any level of a modern fusion system to gain adaptability and improved performance. As can be seen on Fig. 1.4 context supports all levels of information fusion and management. Context is in a form of physical measurements and object features commonly utilized at signal refinement (sub-object assessment) and object assessment levels of fusion. So called environment, or static/physical, context can be associated within geographic maps as a feature space, which can be for instance represented as the road width or the off-road areas. The knowledge about entity's actions, plans or any information besides physical is traditionally utilized at higher levels of fusion [149], i.e. situation or impact assessment. Finally, dynamic context variables such as meteorological conditions, sea state, situation variables or inputs coming from an inference engine have also been considered. Snidaro at al. [133] observe two types of heterogeneity in a sensor fusion processes. The horizontal, in the sense that refers to same-level information produced or extracted by concurrent or cooperative sources which operate in the same environment. For instance, an object appearance can be described by features as color, shape, velocity or class, which are at the same abstraction level. The vertical, in the sense that considers that fusion can take place between different levels of information abstraction, from sensory data to features, from decisions to soft/hard data, including also high-level layers. In this case, the object of interest can be represented by multi-layered information with different degrees of refinement and detail. The process by

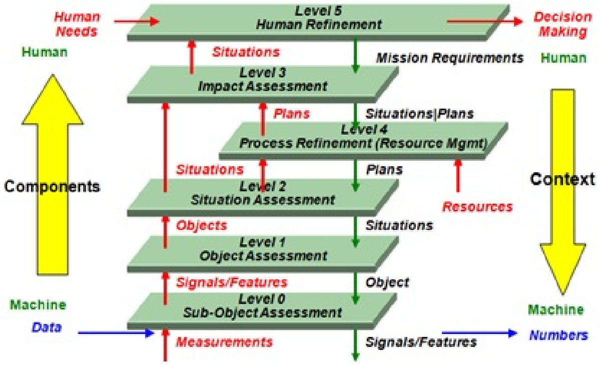


Figure 1.4: Context in support of information fusion [16].

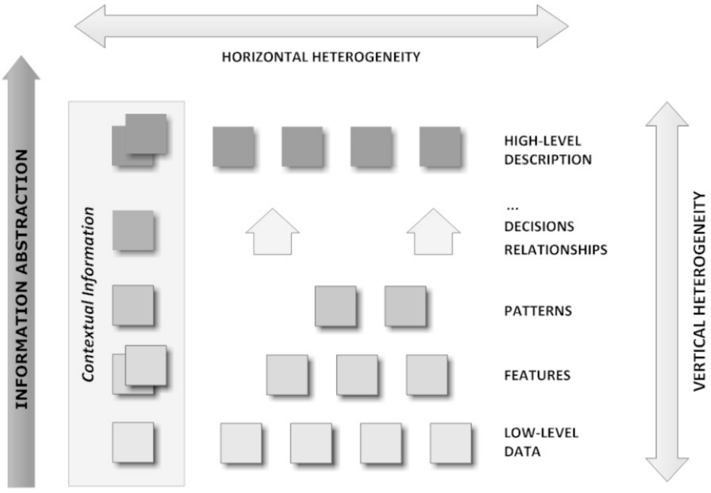


Figure 1.5: Exemplification of context heterogeneity at different abstraction levels [133].

which the entirety of the domain knowledge acquired is fractured into pieces and assigned to the proper algorithms is here referred to as context shifting and illustrated in Fig. 1.6. CI has local scope and validity and is thus pertinent to the scenario at hand. The granularity of the scope of certain information can be more fine-grained and be applicable only to sub-areas of the observed environment. Fig. 1.7 illustrates, context

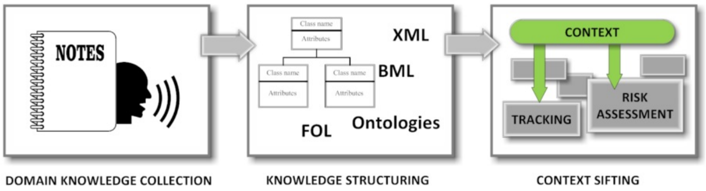


Figure 1.6: Shifting of contextual information [133].

switching, the case of two cameras observing the activities of two different parking lots. Take for example the case of two different parking areas for a public/enterprise building. The first is for visitors/customers and allows free parking while the second is dedicated to staff personnel only. Activity in the two areas is likely to be different, for example knowing that working hours should be in the 8-19 range the system could exploit this information to detect anomalous events.

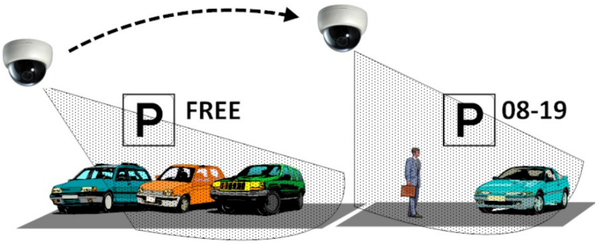


Figure 1.7: Shifting of contextual information on parking lot example [133].

In general, challenges related to the context implementation in area of information fusion are associated with:

- context formalism and representation,

- relevant context discovery and reasoning.

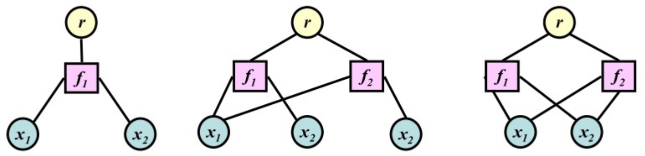
Context Formalism and Representation

By following the Stang's survey [138], the context models most applicable to information fusion are implemented:

- *Key-Value Models* are the simplest way of representing context. They provide values of context attributes as environmental information and utilize exact matching algorithms on these attributes. These models may suffice for use in Level 1 fusion, but they lack capabilities for complex structuring required by higher level fusion;
- *Ontology-based models* provide a formal and uniform way for specifying core concepts, sub-concepts, facts and their inter-relationships to enable realistic representation of contextual knowledge for reasoning, information sharing and reuse;
- *Logic-based models* define context as facts, expressions and rules. Usually contextual information is used in a logic-based system in terms of facts or information inferred from rules. McCarthy [103] defines contexts by axioms describing and interrelating concepts. Furthermore, McCarthy introduced a basic context related notation: $ist(c, p)$, meaning that a logical sentence p holds in the context c, where c is meant to capture all that is not explicit in p that is required to make p a meaningful statement;
- *Multi-context systems* [48] define context as a specific subset of the complete state of an individual entity that is used for reasoning about a given goal. Multi-context systems are seen as a (partial) theory of the world which encodes an individual's subjective perspective about it;
- *Extended Situation Theory* [3], which expands th situation theory in order to model the context with situation types. The variety of different contexts is addressed in the form of rules and beliefs related to a particular point of view;
- *The Sensed Context Model* proposed in [53] is similar to the Extended Situation Theory. It uses first-order predicate logic as a formal representation of contextual propositions and relationships.

Relevant Context Discovery and Reasoning

Relevant context is often not self-evident, but must be discovered or selected as a means to problems-solving. Therefore, context exploitation involves an integration of data fusion with planning and control functions. Discovering and selecting useful context variables is an abductive data fusion and management problem that can be characterized in a utility or uncertainty framework. In information fusion, as in other problem solving applications, contextual reasoning involves inferring desired information on the basis of other available information [142]. The concept of relations and relationships for assessment and practical exploitation of situations was proposed in [140], [141], [119]. Relations, construed to be abstractions, and relationships, which are anchored to sets of referents within a situational context. The ontology of these items could be represented by a means of the factor graphs. This is illustrated in Fig. 1.8, in which entities x_1 and x_2 and relation r form a relationships f_i. Steinberg and Rogova [142], [119] assess the context either from the inside-out or from the outside-in. Such an assessment relates to the notions of *context-of*



a. Relational instantiation **b.** Multiply instantiated relation **c.** Multiply instantiated relation with the same members

Figure 1.8: Factor graph representation of relationships [142].

(CO) and *context-for* (CF) presented in [52]. Former is defined as a situation of interest that provides constraints and expectations for constituent entities, relationships and activities, e.g. "In the context of the present economic and political situation, we would expect an increase in property crime." Alternatively, reference items can be assessed, whether individual entities or situations, in context: "The economic and political situation provides a context for understanding this crime." In the second use (CF), context is relative to one or more reference items. The reference items (RI) for which context is considered can be either an entity or an entity attribute for level 1 estimation or a set of entities and their relationships (i.e. a situation at some level of granularity). In other words, *context-for* is a notion that is relative to a problem to be solved (as in the formulations of [48] and [3]). One's assessment of RIs, whether individual entities, relationships or situations, can be conditioned on larger situations in which these occur. The specific RIs of interest can be defined as a set of state variables, so called problem variables, which an agent wishes to evaluate. A data fusion problem may be stated in terms of a set of problem variables and a utility function on the accuracy of evaluating these variables.

It is not uncommon in fusion, that the variables to be estimated are not directly measured, or are not accurate enough. In such cases, the values of problem variables may be inferred, or estimated, on the basis of other variables. Such inference, commonly performed by neural (NN) or Bayesian belief networks (BN), assumes a model of the dependencies between measurement variables and problem variables. Aforementioned models usually constitute from multiple layers of abstraction, where a set of items and relations defining a CF can be called a set of context variables, while a set of items and relations of interest (the RI) constitute the problem variables, and vice versa. Context variables are exogenous variables that the system selects as means for evaluating problem variables, thus following distinction can be made:

- latent endogenous variables η (i.e. unobserved problem variables),
- latent independent, or exogenous, variables ξ,
- γ an observed endogenous indicators of latent endogenous variables η,
- observed exogenous indicators of latent exogenous variables ξ.

As explained in [140], exogenous variables may be selected for evaluation as means to resolving observed or latent problem variables. Fig. 1.9 represents relationships among problem and contextual elements. Both context variables and problem variables may be known a priori and both can contain static and dynamic variables. For example, context variables in ground target tracking can include the static (local terrain) and dynamic variables (weather). Dependencies between problem variable and context variables can be represented by means of factor graphs that contain the problem situation as a subgraph. Fig. 1.10 represents hierarchical relationships between reference items and contexts for these items. Context variables are typically chosen on the basis of their utility in solving a given problem.

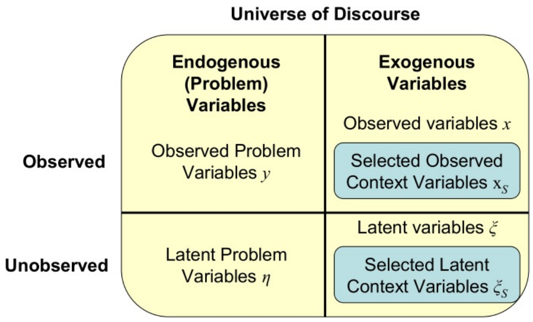

Universe of Discourse
Endogenous (Problem) Variables
Exogenous Variables
Observed
Unobserved
Observed Problem Variables y
Observed variables x
Selected Observed Context Variables x_S
Latent Problem Variables η
Latent variables ξ
Selected Latent Context Variables ξ_S

Figure 1.9: Relation of problem variables and contextual variables [140].

One of the major challenges of modeling and exploiting problem contexts (CF) is determining the selection of context variables. In general, such selection should be based on a constraining ontology of context variables and their relations with problem variables, i.e relevancy and consistency. A context variable can be called relevant to a set of problem variables, if the values of these problem variables are dependent on the context variable under consideration. Alternative measure of the context relevance can be

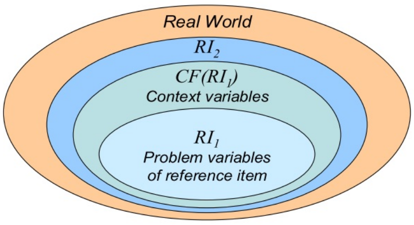


Figure 1.10: Hierarchical relation of problem variables and contextual variables [142].

constituted as the increase in information gain when a particular CI is being inferred. Therefore, relevant context variables can be inferred by means of the problem variables. This is done on the basis of three factors:

- the utility assigned to the given degree of problem variable accuracy,
- the likelihood of achieving that accuracy by some course of action (e.g. by selection, acquisition and processing of context variables),
- the cost of the course of action, which may involve costs of data collection, communications and processing, as well as lost opportunity costs.

Selection of contextual variable assumes that the ambient *"context of"* (CO) reference items are known. However, CO might be unknown or implicitly changed and thus it needs to be discovered. This scenario can happen in highly dynamic environments, in which situational items and relationships constantly change. Inconsistencies between the characteristics and behavior of the initial set and currently observed situational items could be credited to either a new CO appearance or uncertainties in the models. Analyzes of this inconstancies, from effect to cause, is an abductive process executed in the following steps:

- constructing or postulating hypotheses about possible CF,
- computing plausibility of these hypotheses,
- selecting the most plausible hypothesis.

A very important consideration here is the quality (credibility, reliability, etc) of the reference items. Abductive reasoning for context discovery in fusion applications has to take into account the lack of complete knowledge about contextual and problem variables at all required levels of granularity. The process of context discovery may also serve as a means for problem variable evaluation.

1.2.1 Context Inclusion in Low-Levels of Fusion

Data fusion processes at signal or object refinement levels [17] allows incorporating different kind of external factors, such as equality and inequality constrains, which can be modeled as linear or non-linear functions of the state vectors. In majority of cases, the context information is at JDL levels 0 and 1 implemented as a constraining factor since it bounds or limits the state evolution. The knowledge of context can be applied to any of the three low-level fusion functions, depicted as fusion nodes (Fig. 1.11), complemented with a management process [50]:

- Data alignment (also known as common referencing): normalization operations are performed, such as coordinate or units transformations and uncertainty transformations, to align data from information sources to be fused.
- Data association: multiple inputs of either estimates or measurements are examined in order to determine which (hypothetical) entity that the system believes to exist they are associated to or originate from.
- State estimation: often about entity attributes (e.g., kinematic properties, classification attributes such as color, identity, etc.) exploiting prediction models and estimation/inference processes.

- Fusion management: actions to control the output of fusion processes, such as creation, deletion, merging, etc.

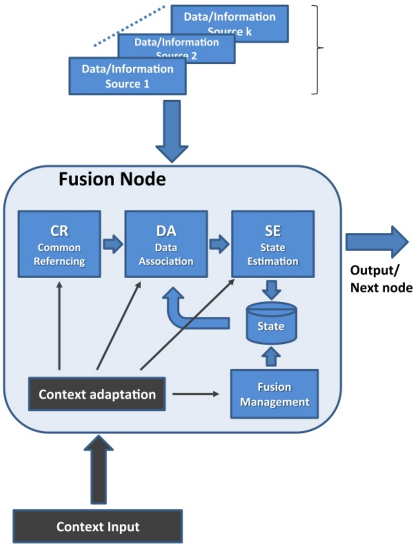


Figure 1.11: Fusion node and adaptation to context [50].

Contextual information (CI) exploitation has been recently considered in the design of modern object tracking algorithms [18]. An exhaustive overview of the context-driven fusion approaches is reported by Snidaro et. al and Simon in [134] and [127], respectively. For the purpose of this thesis, a spacial attention is given to approaches which exploits context in aiding the data association and filtering processes.

Data Association

Data association can be characterized as the process of associating uncertain measurements to existing tracks or alternatively existing tracks to measurements. This problem encompasses a number of challenges related to the track detection, maintenance, and deletion in either single or multi-senor and possibly multi-targets scenarios. Commonly used solutions to this challenging problem are based on probabilistic data association (PDA) strategies. Recent works on context exploitation for data association can be decided among:

- Adaptive association probabilities calculation [90], [90],
- Hypothesis reduction and argumentation techniques [29], [42].

Whereas in many works the detection probability of PDA is set up once, Lherbier et al. [90] proposed a method that estimates the detection probability for each object dynamically using the contextual information modeled by a Bayesian Network (BN-JPDA). BN infers detectability and visibility of an object as contextual relationships to its' observation, whereas nodes between two objects are modeled as and occcultations. Proposed method shows an efficient behavior on real data compared to the usual JPDA approach. Analogously in [90], Bayesian networks are used for computing the evolution of the detection probabilities in a convoy tracking scenario within busy urban environment. Contributions of this article are threefold. First, a labeled version of the Gaussian Mixture Cardinalized Probability Hypothesis Density (GMC-PHD) that allows to differentiate the tracks was introduced. The second contribution concerns the hybridization of the GMC-PHD algorithm to the Variable Structure Interacting Multiple Model with Constraints Multiple Hypothesis Tracking (VS-IMMC-MHT) algorithm in order to improve the performances, specially for group of closely spaced objects. Finally, the third contribution concerns the convoy model by using DBN that proposes an original answer to convoy detection process. The authors in [104] apply a simple PDA where a weighted average over all feasible plot-target assignments is performed. Proposed method incorporates context information, such as road-map and terrain data, and a refined sensor model to include the clutter notch in the tracking process. Results based on simulation scenarios show a significant improvement of the tracking performance in terms of track precision and track continuity. Authors in [29] use map information to prevent unnecessary hypothesis branching during MHT tracking and improve the state estimation accuracy by considering the road network information. An analogous strategy was applied in [42] in order to boost the efficiency of aforementioned method. A framework for feedback multi-target tracking that accounts for scene context was proposed in [99]. Information about target birth and spatially persistent clutter are incrementally learned based on mixtures of Gaussians. The corresponding models are used by a PHD filter that spatially modulates strength of a particular hypothesis given the learned CI. In cases where the measurements are not well described as Gaussian random variables the standard association based on the chi-squared metric will likely fail. Stubberun et al. [144] suggest to enhance the concept of chi-squared metric by a fuzzy-logic based association, by exploiting variable scale target location region, capable of handling the non-Gaussian measurements. Article [87] describes a modification of an association algorithm for object tracking based on the evidence theory. Information such as bathymetric data is used to describe the influence on location possibilities of a submarine or a ship.

Bayesian Filtering

Contextual information can be in the form of equality and/or inequality constraints exploited for the linear or non-linear systems based on following techniques [127]:

- Linear systems with inequality constrains
  - Estimate Projection [128],
  - Gain Projection [56],
  - Probability Density Function Truncation [129],
  - Interior Point Likelihood Maximization [12],
  - Second Order Expansion of Nonlinear Constraints [162],
  - Moving Horizon Estimation (MHE) [116],
- Linear systems with equality constrains
  - Model Reduction [157],
  - Perfect Measurements [28],
  - System Projection [77],
- Non-linear Systems
  - Moving Horizon Estimation (MHE) [117],
  - Sequential Monte Carlo (SMC) [2], [152].

The Kalman Filter is an optimal estimator for the linear state space models, and thus the solution of choice for the recursive constrains exploitation. If the constraints result in non-linear functions, a linearization step is required before adopting the above mentioned techniques in the Kalman filter. If the constraints maintain their non-linearities, non-linear filtering algorithms, such as Extended Kalman filter (EKF), Unscented Kalman filters (UKF) or Particle filters (PF), need to be employed for the target estimation purposes. According to [112], an underlying aspect when comparing constrained Bayesian filters is the entry point of the constraint, that is either prediction or measurement update of the filter.

Kalman filters are commonly used to estimate the states of a dynamic system. However, in the application of Kalman filters there is often known model or signal information that is either ignored or dealt with heuristically. The idea of using state constraints to improve the tracking performance dates back to the 90s, when attempts to exploit hard linear equality constraints in Kalman filtering led to the definition of the pseudo-measurements approach [150]. A rigorous analytic method of incorporating state inequality constraints in the Kalman filter was proposed by Simon et al. [128]. At each time step the unconstrained Kalman filter solution is projected onto the state constraint surface in order to improve the prediction accuracy of the filter. A gain projection approach for KF was introduced by Gupta et. al. [56]. Another possibility to induce inequality constrains to the Kalman filter is by truncating the probability density function (PDF) [129]. In this approach, KF estimates the constraints and then computes the constrained filter estimate as the mean of the truncated PDF. Simulation results obtained from turbofan engine model demonstration shows, that incorporation of state variable constraints increases the computational effort of the filter but also improves its estimation accuracy. In [12] authors have shown, how interior point methods can be applied to maximize the Kalman-Bucy smoother likelihood subject to nonlinear inequality constraints. A key contribution of this approach is that it exploits the same decomposition that is used for unconstrained Kalman-Bucy smoothers and so the required operations scale linearly with the number of measurements. In [116] authors consider moving horizon strategies for constrained linear state estimation. Authors argue, that formulating a linear state estimation problem with inequality constraints, prevents recursive solutions such as Kalamn filtering, and, consequently, the estimation problem grows with time as more measurements become available. To bound the problem size, authors proposed moving horizon strategies for constrained linear state estimation.

Wen et al. [157] introduce an algorithm for implementing multi-sensor system in model-based environment with consideration of equality constraints. Authors suggested to improve effectiveness of sensor data fusion through consistent representation of environment geometry, effective sensor modeling and an optimal estimation algorithm. Authors utilize CAD model in order to generate geometric features and constraints of the environment. Measurement models are used to predict sensor response to certain features and to interpret raw sensory data. Constrained minimum mean squared (MMS) estimator is used to recursively predict, match and update the location of features. A constrained filtering method is proposed in [28] in order to deal with the filtering problems for nonlinear systems with constraints. Authors, convened the problem into a sequence of recursive estimation problems, in which, the system equations and constraint conditions are treated as pseudo-measurements. To resolve the singularity problem arising from the constraints, a modified maximum-likelihood method for nonlinear systems was proposed. The state estimation problem for linear systems with linear state equality constraints was thoughtfully analyzed in [77]. Authors construct the optimal estimate, which satisfies linear equality constraints, from the noisy measurements. By using the constrained Kalman filter for the projected system and comparing its filter's Riccati equation with those of the unconstrained and the projected Kalman filters authors demonstrated, that the constrained estimator outperforms the other filters for estimating the constrained system state.

State estimator design for a nonlinear discrete-time system is a challenging problem, further complicated when additional physical insight is available in the form of inequality constraints on the state variables and disturbances. One strategy for constrained state estimation is to employ on-line optimization using a moving horizon approximation [117]. In the article authors propose a general theory for constrained moving horizon estimation for linear and nonlinear state estimation. In [2] the use of Sequential Monte Carlo (SMC) methods for road constrained target tracking is considered. The proposed algorithm propagates the joint PDF of the target kinematic state and target ID in a road constrained environment. Road map assisted ground targets tracking is also considered in [152] and [143]. Authors model the target dynamics in quasi one-dimensional road coordinates which are mapped onto the ground coordinates using linear road segments. Here the author proposes the use of a Gaussian sum algorithm within a Variable Structure Multiple Models (VSMM) scheme. As long as the predicted estimate is inside the same road, a Kalman filter is used to perform the update step. When the target approaches a junction, an on-road projection is necessary, and a multiple hypotheses approach is followed. Hard inequality state constraints of airplanes flight envelope are considered in [27]. During the measurement update of the filter, only particles that satisfy the constraint are accepted and the rest are rejected from the cloud of independent particles. The proposed PF algorithm converges to the true a posteriori PDF for a sufficiently large number of particles, but might be unfeasible due to the computational load required. Mertens et al. [104] use topographic background information to enhance the tracking of ground vehicles in complex dynamic environments. Presence of CI in such a scenario significantly improved the quality and continuity of tracks, particularly during stop and go maneuvers and target masking due to Doppler blindness. Similarly, Gustafsson et al. [59] improved navigation and tracking performance of road bound vehicles by imposing road constraints on their trajectories. The so called road-assisted navigation, takes advantage of dynamic matching between the motion model of a vehicle and the road network implemented as manifold. Road map assisted ground targets tracking is also considered in [143]. The context-aided tracker or ConTracker [102] utilizes CI obtained from naval maps, such as a water depth, trade route paths, and areas/buildings with a high strategic value, in order to detect anomalies in the ship traffic. A refined GMTI sensor model with state dependent detection probability and information about the clutter notch is proposed. Both equality and inequality constraints are used to model the known road network. The latter are used to model non-zero width roads. The authors investigate the performance of both Gaussian sum and particles based approximations, in which the prediction step is performed in road coordinates, while the update step is carried out in the 2D Cartesian space.

Multiple Model Filtering

Utilizing multiple cooperating filers, each optimized for a certain mode of the target, is a common practice for dealing with targets which are either constantly maneuvering or actively avoiding the detection. In both cases, context information can not only be utilized as a parameter in kinematic equations, but it can also provide useful clues about the motion mode of the target. Most common approaches to context aware MM filtering include:

- Feature aided tracking [148], [37],
- Gaussian mixture estimators [147],
- MC joint tracking and classification [6], [7].

As of now, target recognition and feature-aided tracking are the most common examples of the mode context exploitation [148], [37]. Here, modal observations are modeled as classifiers over a set of plausible target modes. Combination of context observations and the kinematic measurements leads to the joint probability density of the likelihood function [164], [41]. According to the Bayesian recursion rule, updates of the hybrid state densities results a mixture of Gaussians with an exponentially increasing number of components. For this reason, a lot attention has been devoted for development of tractable approximate estimators with resolvable computation time [41]. So called, image-enhanced IE-IMM is capable to significantly enhance the performance of classical IMM when a readings from high quality modal sensor are available, but this advantage diminishes as the quality of the mode observations becomes poorer. Alternatively, Sworder et al. [147] decided to limit the growth of the Gaussian mixture components by a hypothesis merging techniques, which lead to the development of Gaussian wavelet estimator (GWE). The comparative simulation results indicates, that in some cases (e.g., strong context, poor kinematic data) GWE can outperform IE-IMM and IMM counterparts by a 25% and 50%, respectively. An exact hybrid filter based on change of measure that accounts for intermittent mode measurements was presented in [5], along with an approximate, GPB-type implementation and EM-based estimation of the transition probability matrix. More recent studies on feature-aided IMM tracking were presented by Blasch and Yang in [19] and [163], respectively. In their work, authors present a mutual-aided target tracking and identification scheme that exploits the couplings between the target kinematic and object features. In [101] the use of PF for littoral tracking is proposed. The authors formulate the problem as Joint Tracking and Classification (JTC), where a target class is assigned for each isolated land or water region. A similar approach is followed in [6], where the authors propose a modified version of the JTC-PF algorithm that uses class-dependent speed likelihoods. Based on recently proposed Monte Carlo techniques, a multiple model (MM) particle filter and a mixture Kalman filter (MKF) are designed for two-class identification of air targets: commercial and military aircraft. In [7] a variable structure multiple model particle filter (VSMM-PF) which uses the concept of directional process noise to model motion along particular roads. The information available through a road map is modeled using a Jump Markov system with state dependent transition probabilities. Each road segment is represented by two way-points determining direction, location, and length of each road. A binary valued probability represents the visibility constrains. The inequality constraints on target speed are applied in prediction through the generation of random variables from a truncated Gaussian.

Dempster-Shafer Theory of Evidence

A commonly used alternative to the Bayesian filters is formulated as the Dempster-Shafer Theory (DST) of fusion. DST is considered as one of major paradigm shifts for reasoning under uncertainty. Despite being capable of combining the independent pieces of evidence, the DST has been also strongly criticized because of its unexpected behavior and by providing counter-intuitive results when combining highly conflicting information. However, it is genuinely agreed that Dempsters rule provides valid results in scenarios where the initial conditions are respected and the problem is well modeled.

In [35] a real-time Generalized Data Association Multi Target Tracking systems (GDA-MTT) is proposed on bases of proportional conflict redistribution (PCR) rules. PCR allows a very efficient target type tracking and reduces drastically the latency delay for correct target type decision with respect to Demspter's rule. A Two-Layer Conflict Solving (TLCS) data fusion scheme based on Dempster-Shafer Theory and on Fuzzy-Pattern-Classification (FPC) concepts is proposed in [98]. Authors aim to provide an approach to data fusion which offers a stable conflict scenario handling. Similarly in [122], fusion of each piece of evidence is discounted in proportion to the degree that it contributes to the conflict. This way the contributors of conflict are managed on a case-by-case basis in relation to the problem they cause. Discounting is performed in a sequence of incremental steps, with conflict updated at each step, until the overall conflict is brought down exactly to a predefined acceptable level. A Demspter-Shafer multi-level data fusion composed of a multi-criterion fusion and a multi-sensor fusion was proposed in [85]. The benefits of the proposed approach are twofold: first, contrary to conventional approaches, the source independences are guaranteed. On the other hand, the consideration of the conflict as an additional source of information for the decision step, allows the speed limit assistant to stay undecided about the final speed limit. These benefits have been highlighted in the comparisons of this multi-level Speed Limit Assistant with a conventional approaches. In [65], Hasse diagrams were utilized to improve the understandability of the PCR methods. Through a comparative study, the advantage of PCR6 over PCR5 for more than three sources was presented.

While there are numerous cases where DST was implemented for the navigation purposes, the articles addressing the multi-sensor data fusion cases are nearly nonexistent. It is worth noting, that basic DST formula is in complex fusion systems further altered by a human expert knowledge in order to improve quality and reliability of the solution [160], [137]. In order to cope with a luck of expert knowledge or with an unknown and unpredictable evidence, the DST was extended to new more flexible theories such as:

- Transferable Belief Model (TBM) [33], refutes the constraint on the frame of discernment and the underlying probability model, which allows to allocate belief to the elements of the empty set,

- Dezert-Smarandache Theory (DSmT), extends the DST to allow usage of hybrid and dynamic models and solves numerical issues which originates while combining highly conflicting pieces of evidence,

- Adaptive Combination Rule (ACR), which maximizes the conjunctive and the disjunctive rules based on the distribution of the conflict according to a new choice of weighting coefficients,

- Proportional Conflict Redistribution (PCR) [130], [44], redistributes the partial conflicting masses to the elements involved in the partial conflicts only, considering the conjunctive normal form of the partial conflicts.

1.2.2 Context Inclusion in Higher-Levels of Fusion

Data fusion processes at situation assessment, impact assessment or process refinement deal almost elusively with symbolic information sources. Exploitation of context in higher levels of fusion, i.e. JDL level 2 and above, was recently documented in [134]. Situation Assessment involves inferences of the following types:

- Inferring the presence and the states of entities on the basis of relationships in which they participate,

- Inferring relationships on the basis of entity states and/or other relationships,

- Recognizing and characterizing observed situations,

- Projecting unobserved (e.g. future) situations.

In Situation Assessment, hypotheses concerning relationships and situations are built and evaluated. These relationship and situation hypotheses can also be represented as factor graphs. In situation assessment (as in all data fusion), the characteristic relationships are epistemic, i.e. between the information system (cognitive agent) and inferred world states, reflecting the beliefs of the information system. The uncertainty in these beliefs needs to be represented by an uncertainty metric, e.g. a likelihood, in the relationship function [16].

Situation Assessment

The fundamental principles of designing a situation-aware data fusion architecture has been discussed by Steinberg and Rogova [142]. Briefly, the authors suggest to model events or situations in the world as a set of relations and relationships in the form of factor graphs. In this sense, CI could seen as a background knowledge which is associated to every relation pair or in different words to each node of the graph. The uncertainty in these relations can be best described by a likelihood functions. Such a factor graphs could be then used for the context reasoning either in logical or probabilistic manner. In the information fusion (IF), contextual reasoning relates to an inference of problem variables i.e. desired information given the basis of context variables i.e available information. Generally speaking, the process of finding a relevant context is not trivial and often involves a complex integration of IF with planning, abductive logic and control functions. For instance Steinberg [140] models contextual information as situations and suggests the use of structural equation modeling (SEM) techniques for evaluating dependencies between problem and context variables. In [119] context-dependent information quality attributes, namely, credibility, reliability, and timeliness, are introduced and their incorporation into sequential decision making for pattern recognition is discussed. In short, context is represented by the time-dependent distance between an observed target and a sensor, and a situation-based time-dependent threshold on credibility.

Recent advances on artificial intelligence techniques in providing automatic and self-adaptive systems were discussed by Suarez-Tangil et al. [145]. In their work, authors proposed security information and event management (SIEM) system with self-adaptation capabilities capable of optimizing the operator intervention routines. Machine learning is applied for rule extraction to classify reported events accordingly to a context-based pattern definition of attacks. A proposal to dynamically represent context knowledge with ontologies and evaluate anomalous situations is presented by Gomez-Romero et al. in [51]. This paper proposes a double-layer fusion system for context-based situation and threat assessment with application to harbor surveillance. The first layer uses an ontological model to formally represent input data and to classify harbor objects and basic situations by deductive reasoning according to the harbor regulations. The second layer applies belief-based argumentation (BAS) in order to determine the threat level of situations which are considered to the unusual. The recent approach of Snidaro et al. [136] discusses the fusion of uncertain sensory and contextual information for maritime situational awareness. In the article, events and anomalies are key elements in the process of assessing and understanding the observed environment. Building an effective situational picture for a surveillance system involves combining high-level information with sensory data. The Markov Logic Networks framework is employed to both encode a priori and contextual knowledge and to fuse evidence from multiple sources, possibly reasoning over incomplete data. Application of agent-based systems into situation-aware information fusion has been discussed in [30], [4].

The problem of context reasoning, as an essential part of a situation and impact assessment respectively, could be conveniently scaled into the problem of a plan recognition. Plan recognition seeks to infer on a entity's plan based on the observed entity's actions and their effects. The role of CI in plan recognition is twofold. Firstly it creates additional links between events and entities. Secondly it adjust our confidence that an entity is following a certain plan. Probabilistic reasoning is the most popular approach in nowadays plan recognition and as such approaches utilizing Bayesian networks (BN) and their variants are very popular [23], [94] or [154]. Furthermore, the casualty of relations and relationships, modeled by factor graphs, could be directly translated into the BN as states/nodes at multiple hierarchical levels and at different time slices i.e. dynamic Bayesian net works (DBN). As the target progresses trough the network and forming a chain of events, then the BN could infer on target's future actions based on its past. Similarly, relevant context could be then discovered or selected based on target's future or past actions, respectably.

Situation Assessment and Target Tracking

To the best of our knowledge, BN were for the first time proposed to aid object tracking by Hautaniemi and Saarinen [61]. They intended to enhance the classic multiple target tracking (MTT) algorithm, based on interactive multiple model filter (IMM) and probabilistic data association (PDA), with the quantities other than kinematic measurements i.e. context. Authors argued that PDA was unable to deal with a contextual information (CI) and therefore BN were developed as an additional inference method to aid data association and track identification processes. In [115] DBN are used to capture the modal nature of the tracking sea species. More specifically, the state space model of a tracked ocean animal was represented as a DBN itself including both continuous (the velocities) and discrete variables (the propulsive mode of the animal, and the discrete observation variables). Such a hybrid system was then used to adapt the bandwidth of the multiple model bootstrap filter based on the most probable models given their conditional probability densities (CPDs). Schubert and Wanielik [123] argued that context in [61] does not influence the tracking directly or in case of [115] a tracking procedure is computationally very expensive. Therefore, the authors proposed their own approach for incorporating additional information into the IMM called the Meta Model filter. With structure similar to BN, the Meta Model represent causality of events in form of an adaptive transition probability matrix. Elements of such a matrix are nodes and states which represent the possible modes and transitions to other modes respectively. The usefulness of such a Meta Model filtering was demonstrated in a line change maneuver recognition algorithm for vehicles with uncertain velocities and yaw rates. In their most recent publication [124], the authors leverage their expertise into development of an Advanced Driver Assistance System (ADAS) using unified Bayesian approach for tracking and situation assessment.

Aforementioned work assumes a priori knowledge of the transition probability matrix (TPM). However, the TPM is a design parameter whose choice could significantly influence the estimation process. Therefore, algorithms which can identify the TPM during the course of tracking were extensively studied in literature [68], [67]. Furthermore, variable structure algorithms, referred to as expected-mode augmentation (EMA), for multiple-model estimation were developed [91], [92]. In EMA approach, the original model set is augmented by a variable set of models intended to match the expected value of the unknown true mode. An on-line maximum likelihood estimator for the transition probabilities associated with a jump Markov linear system (JMLS) was proposed by Orguner et al. [110]. The maximum likelihood estimator is derived using the reference probability method, which exploits an hypothetical probability measure to find recursions for complex expectations. Expectation maximization (EM) procedure is utilized for maximizing the TPM likelihood function. These models are generated adaptively in real time as probabilistically weighted sums of modal states over the model set.

1.3 Contribution and Objectives

In a short summary of Sec. 1.2, the process of context exploitation in the data fusion, especially in the field of target tracking, is application oriented and offers very little of generality or adaptability to ever changing environmental conditions. The principle frameworks capable of adaptive context exploitation across all levels of fusion are still being developed. Until now, the context has been extensively exploited as a parameter in system and measurement models which led to the development of numerous approaches for constrained estimation and target tracking. Here, the spatial or static context is the most common source of the ambient information, i.e. features, utilized for enhancement of the state variables. In the case of multiple model estimators, context can not only be related to the state but also to a certain mode of the filter. Common practice for multiple model scenarios is to represent states and context as a joint distribution of Gaussian mixtures. These approaches are commonly referred as the join tracking and classification (JTC). Alternatively, the usefulness of context was also demonstrated in aiding the measurement data association. Process of formulating a hypothesis, which assigns a particular measurement to the track, is traditionally governed by the empirical knowledge of the noise characteristics of sensors and operating environment, i.e. probability of detection, false alarm, clutter noise, which can be further enhanced by conditioning on context. CI in a form of events and entities' intents, i.e. event or intent CI, and its relation to the time, i.e. event-temporal and intent-temporal CI, has been recently considered for aiding the mode selection process. The key idea was to utilize probabilistic reasoning to infer the mode state and/or the mode transition matrix of the target. Aforementioned methodologies are only a few examples of CI utilization in the modern target tracking systems. We believe that context awareness can be further enhanced by a tighter integration of the reasoning techniques with the low-level fusion processes. In line with the idea we present the objectives of this thesis as follows:

- Investigate a feasibility of a hybrid context reasoning approaches by utilizing probabilistic or logic-based reasoning processes;
- Investigate possibility to incorporate temporal, spatial and logical context into a hybrid reasoning architecture;
- Investigate feasibility of adaptive context exploitation and reasoning;

- Evaluate proposed methodology against state of the art algorithms for information fusion, e.g. target tracking, intent recognition, anomalies detection.

Contributions of the thesis are fourfold:

- An adaptive framework that dynamically takes into consideration contextual information in order to support mission goals was presented. Furthermore, architecture concepts to be considered in the development of context-aware fusion systems were discussed and illustrated in a maritime use-case scenario (Sec. 3).
- The problem of multi-level context representation and exploitation for target tracking is presented in Sec. 4. An approach for encoding different types of contextual information as likelihood functions via classifiers in particle filters was proposed and evaluated on synthetic dataset.
- Framework discussed in the Sec. 5)views target actions as a hidden Markov process with an event context associated with each node. Relevant context is at each time step selected based on immediate and goal driven sets of actions. Inference in the HMM is conditioned on prior target measurements and the belief state conditioned on context. This posterior is then compared with a target state estimate in order to adjust switching probability in the Interactive Multiple Models (IMM) tracking process.
- Context-aware grid-based environment mapping and obstacle detection techniques are discussed in Sec. 6. A multi-sensor solution for general purpose environmental mapping, based on Bayesian occupancy filter and Dempster-Shafer theory, which combines advantages of grid-based mapping and situation assessment is presented and evaluated on autonomous driving vehicle scenario. It has been shown, that discrepancies between the grid maps can be exploited in such a manner, where fusion of contradicting information will be less susceptible to the sensor weighting and the accuracy of mapped environment can be further improved.

1.4 Outline of the Thesis

The structure of the thesis follows schematics below (Fig. 1.12), where each block represent a chapter and lanes outline the logical connections between individual chapters. Introduction Sec. 1 explains the motivation, objectives and state of the art of this work. Fundamental theory utilized for experiments throughout the work is summarized in Sec. 2. The architecture for adaptive context-aware tracking is introduced in Sec. 3. Practicality and usefulness of newly introduced context exploitation concepts is demonstrated in two tracking scenarios, extensively discussed in Sec. 4 and Sec. 5. Away from target tracking, the importance of context is also demonstrated in the field of environment map building for autonomous vehicles (Sec. 6).

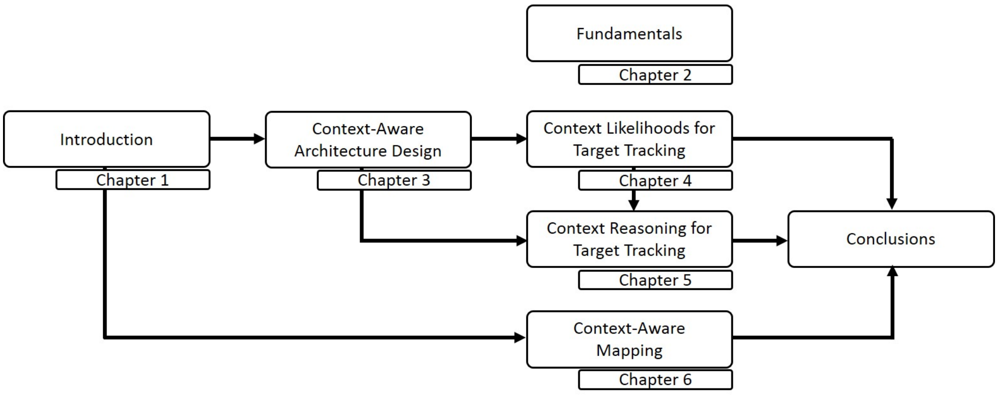


Figure 1.12: Outline of the Thesis.

2

Fundamentals

"Always pass on what you have learned."

2.1 System Theory

A physical system can be described by a general nonlinear state space model of time varying variables as

$$\begin{aligned}\dot{\mathbf{x}}_t &= \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{v}, t),\\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{e}, t).\end{aligned} \tag{2.1}$$

Above model is characterized by a system state $\mathbf{x}$, input signal $\mathbf{u}$, measurements $\mathbf{y}$, process noise $\mathbf{v}$, measurement noise $\mathbf{e}$. Functions $\mathbf{f}(.)$ and $\mathbf{h}(.)$ can be either linear or non-linear function of the system evolution and the measurement model, respectively. The linear time varying state space model can be written as

$$\begin{aligned}\dot{\mathbf{x}}_t =& \mathbf{A}_t\mathbf{x}_t + \mathbf{B}_t^u\mathbf{u}_t + \mathbf{B}_t^v\mathbf{v}_t,\\ \mathbf{y}_t =& \mathbf{C}_t\mathbf{x}_t + \mathbf{D}_t^u\mathbf{u}_t + \mathbf{w}_t,\end{aligned} \tag{2.2}$$

where matrices $\mathbf{A}_t, \mathbf{B}_t^u, \mathbf{B}_t^v, \mathbf{C}_t, \mathbf{D}_t$ represent transitions of states, direct inputs, system noise uncertainties, outputs, direct feedthrough, respectively. Linear system theory could be applied to non-linear systems by using approximations over the set of linear models. Discrete time equivalent of the system equations (2.2) can be obtained by sampling over time interval T

$$\begin{aligned}\mathbf{F}_k =& e^{\mathbf{A}_t T},\\ \mathbf{G}_k^u =& \int_0^T e^{\mathbf{A}_t \tau} d\tau \, \mathbf{B}_t^u.\end{aligned} \tag{2.3}$$

Discrete time equivalent of the nonlinear system equations (2.1) is presented as (2.11).

Nearly constant velocity (CV) (2.4) and coordinated turn (CT) (2.5) models were employed for modeling the system kinematics. 2D state space models are defined in the local level frame and read as follows

$$\begin{bmatrix} p_{k+1}^x \\ p_{k+1}^y \\ v_{k+1}^x \\ v_{k+1}^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}, \tag{2.4}$$

and

$$\begin{bmatrix} p_{k+1}^x \\ p_{k+1}^y \\ v_{k+1}^x \\ v_{k+1}^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & \sin\left(\frac{\omega_k T}{\omega_k}\right) & -\frac{1-\cos(\omega_k T)}{\omega_k} \\ 0 & 1 & \frac{1-\cos(\omega_k T)}{\omega_k} & \sin\left(\frac{\omega_k T}{\omega_k}\right) \\ 0 & 0 & \cos(\omega_k T) & -\sin(\omega_k T) \\ 0 & 0 & \sin(\omega_k T) & \cos(\omega_k T) \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}. \tag{2.5}$$

Uncertainties in state space model (2.4) and (2.5) are generated by a zero mean Gaussian distributed acceleration noise $\mathbf{a}_k = \mathcal{N}(0, \sigma_a^2)$. It is assumed, that ω_k is a priori known parameter and therefore the angular velocity is not estimated within a state vector $\mathbf{x}_k$. This simplification avoid the usage of varying structure multiple model estimators (VSMM). Simplified sensor models consisting directly from position measurement are considered in the experiments.

$$\begin{bmatrix} y_k^x \\ y_k^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} w_k^x \\ w_k^y \end{bmatrix} \tag{2.6}$$

Uncertainties in observations are modeled by a Gaussian distributed white noise $\mathbf{w}_k \approx \mathcal{N}(0, \sigma_w^2)$ both for x and y components.

2.2 Information Fusion

The general information fusion theorem of $n \in \{1, \dots, N\}$ independent sources [58] can be for a linear system equations (2.2) expressed as

$$\begin{aligned} \mathbf{I}_k =& \mathbf{I}_1 + \mathbf{I}_2 + \dots + \mathbf{I}_n, \\ \hat{\mathbf{x}}_k =& \mathbf{I}_k^{-1}(\mathbf{I}_1\hat{\mathbf{x}}_2 + \mathbf{I}_2\hat{\mathbf{x}}_1 + \dots + \hat{\mathbf{x}}_n\mathbf{I}_n), \end{aligned} \tag{2.7}$$

where $\hat{\mathbf{x}}_k$ is the estimate and $\mathbf{I}_k$ is an information matrix and product $\hat{\mathbf{x}}_k\mathbf{I}_k = \mathbf{f}_k$ is referred as an information state. The information can thus be seen as weighting of the estimates. In case of the weighted least square estimate $\hat{\mathbf{x}}^{WLS}$ leads to the well known formulas

$$\begin{aligned} \mathbf{I}_k =& \sum_{n=1}^{N} \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{H}_n, \\ \mathbf{f}_k =& \sum_{n=1}^{N} \mathbf{H}_n^T \mathbf{R}_n^{-1} \mathbf{y}_n, \\ \hat{\mathbf{x}}_k^{WLS} =& \mathbf{I}_k^{-1}\mathbf{f}_k. \end{aligned} \tag{2.8}$$

If the estimates are updated sequentially in space or time expressions (2.8) will become

$$\begin{aligned} \mathbf{I}_k =& \mathbf{I}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k, \\ \mathbf{f}_k =& \mathbf{f}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{y}_k, \\ \hat{\mathbf{x}}_k^{WLS} =& \mathbf{I}_k^{-1}\mathbf{f}_k. \end{aligned} \tag{2.9}$$

The loss function $\min \mathbf{V}_k(x)$ is for WLS estimate defined as

$$\min \mathbf{V}_k(x) = \sum_{k=1}^{K} (\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k)^T \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k). \tag{2.10}$$

2.3 Bayesian Estimation

Consider a general discrete time state space system of a form

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{g}_k(\mathbf{v}_k) \quad \text{or} \quad p(\mathbf{x}_{k+1}|\mathbf{x}_k), \tag{2.11}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{w}_k \quad \text{or} \quad p(\mathbf{y}_k|\mathbf{x}_k), \tag{2.12}$$

where $\mathbf{f}_k$ and $\mathbf{g}_k$ are nonlinear functions of the target state vector $\mathbf{x}_k$ and process noise $\mathbf{v}_k$, respectively. Variable $\mathbf{h}_k$ represents a nonlinear relationship between sensor output $\mathbf{y}_k$ and target state vector $\mathbf{x}_k$ affected by a measurement noise $\mathbf{w}_k$. The goal of estimation is to infer the state variable $\mathbf{x}_k$ with the available sensor measurements $\mathbf{y}_{1:k}$. By using the Bayesian framework, this estimation problem relates to the recursive evaluation of the probability density function (PDF) $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ in two consecutive steps, the prediction and the measurement update of the state vectors.

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Prediction}} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \tag{2.13}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Measurement}} p(\mathbf{x}_k|\mathbf{y}_{1:k}) \tag{2.14}$$

The prediction state density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ of state $\mathbf{x}_k$ is calculated from the prior PDF $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$ by using Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1}. \tag{2.15}$$

Equality (2.15) follows the 1^{st} order Markov property which assumes that $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ only depends on state $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ at time k and $k-1$ respectively. The measurement update $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is computed from the prior distribution (2.15) and measurements $\mathbf{y}_{1:k}$ by a Bayesian formula which results in

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{2.16}$$

The 1^{st} order Markov property for equation (2.16) implies that $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ only depends on measurement $\mathbf{y}_k$ at time k.

2.3.1 Kalman Filter

Given that system model, represented by distributions $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ (2.11) and $p(\mathbf{y}_k|\mathbf{x}_k)$ (2.12), is linear and Gaussian distributed, the best linear unbiased estimate of the PDF $p(\mathbf{x}_k|\mathbf{y}_{1:k})$, i.e. solution to the Bayesian recursion (2.13) and (2.14), is provided by the Kalman filter (KF) [58]. The linear sate space densities for prediction (2.13) and measurement update (2.14) can be for KF reformulated explicitly as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{\Sigma}_{k|k-1}), \tag{2.17}$$

and

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}, \mathbf{\Sigma}_{k|k}), \tag{2.18}$$

respectively. The Gaussian mean $\hat{\mathbf{x}}_{k|k-1}$ and covariance $\mathbf{\Sigma}_{k|k-1}$ are for the state evolution $\mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k-1}, \mathbf{\Sigma}_{k|k-1})$ computed as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{F}_{k-1}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{G}_{k-1}\mathbf{v}_{k-1}, \\ \mathbf{\Sigma}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{\Sigma}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{G}_{k-1}\mathbf{Q}_{k-1}\mathbf{G}_{k-1}^T. \end{aligned} \tag{2.19}$$

The measurement update (2.18) of the Kalman filter reads as follows

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k|k-1}), \\ \mathbf{\Sigma}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{\Sigma}_{k|k-1}, \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T(\mathbf{H}_k\mathbf{\Sigma}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k)^{-1}. \end{aligned} \tag{2.20}$$

2.3.2 Extended Kalman Filter

Kalman filter solution to the nonlinear state estimation problem is commonly reffed as the Extended Kalman Filter (Alg.: 1). In comparison to the Kalman filter recursion, EKF linearizes the system equations (2.11, 2.12) by applying 1^{st} order Taylor series expansion around the latest state estimate, i.e.

$$\begin{aligned} \mathbf{f}_k(\mathbf{x}_k) &\approx \mathbf{f}_k(\hat{\mathbf{x}}_k) + \mathbf{F}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}), \\ \mathbf{g}_k(\mathbf{x}_k) &\approx \mathbf{f}_k(\hat{\mathbf{x}}_k) + \mathbf{G}_k\mathbf{v}_k, \\ \mathbf{h}_k(\mathbf{x}_k) &\approx \mathbf{h}_k(\hat{\mathbf{x}}_k) + \mathbf{H}_k(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k}), \end{aligned} \tag{2.21}$$

where

$$\begin{aligned}\mathbf{F}_k &= \frac{\partial \mathbf{f}_k(\mathbf{x},\mathbf{v})}{\partial \mathbf{x}} \Rightarrow \frac{\partial \mathbf{f}_k(\hat{\mathbf{x}}_{k|k},0)}{\partial \mathbf{x}},\\ \mathbf{G}_k &= \frac{\partial \mathbf{f}_k(\mathbf{x},\mathbf{v})}{\partial \mathbf{v}} \Rightarrow \frac{\partial \mathbf{f}_k(\hat{\mathbf{x}}_{k|k},0)}{\partial \mathbf{v}},\\ \mathbf{H}_k &= \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \Rightarrow \frac{\partial \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})}{\partial \mathbf{x}}.\end{aligned} \tag{2.22}$$

Single step of the Extended Kalman filter is presented as the Alg. 1 shown below.

Algorithm 1: Extended Kalman filter

```
    Step 0: Initialize
    Set $\hat{x}_{k-1|k-1} = x_0$ and $\Sigma_{k-1|k-1} = \Sigma_0$

    Step 1: Time update
        $\hat{x}_{k|k-1} = f_{k-1}(\hat{x}_{k-1}),$
        $\Sigma_{k|k-1} = F_{k-1}\Sigma_{k-1|k-1}F_{k-1}^T + G_{k-1}Q_{k-1}{G_{k-1}}^T,$

    Step 2: Measurement update
        $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - h_{k-1}(\hat{x}_{k|k-1})),$
        $\Sigma_{k|k} = \Sigma_{k|k-1} - K_k S_k {K_k}^T,$
        $S_k = H_k \Sigma_{k|k-1} {H_k}^T + R_k,$
        $K_k = \Sigma_{k|k-1} {H_k}^T {S_k}^{-1}.$
```

2.3.3 Particle Filter

Arguably, the most popular algorithm to nonlinear recursive estimation is the particle filter (PF), extensively evaluated in [57]. PF represents any arbitrary probability density function $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ by samples or particles x_k^i, i.e.

$$x_k^i \approx p(\mathbf{x}_k|\mathbf{y}_{1:k}), \quad i = 1,\dots,N. \tag{2.23}$$

The particles are used to form an approximative distribution as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \hat{p}(\mathbf{x}_k|\mathbf{y}_{1:k}) = \sum_{i=1}^{N} w_{k|k}^i \delta(\mathbf{x}_k - x_k^i), \quad \sum_{i=1}^{N} w_{k|k} = 1, \tag{2.24}$$

where $\hat{p}(\mathbf{x}_k|\mathbf{y}_{1:k})$ is an approximated distribution, $\delta(\mathbf{x}_k - x_k^i)$ is a the Dirac delta function and $w_{k|k}^i$ the weights of the particles. The time update of the Bayesian recursion (2.11) is in case of PF evaluated as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) &\approx \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) \sum_{i=1}^{N} w_{k-1|k-1}^i \delta(\mathbf{x}_{k-1} - x_{k-1}^i)\mathbf{d}\mathbf{x}_{k-1},\\ &\approx \sum_{i=1}^{N} w_{k-1|k-1}^i \int p(\mathbf{x}_k|\mathbf{x}_{k-1})\delta(\mathbf{x}_{k-1} - x_{k-1}^i)\mathbf{d}\mathbf{x}_{k-1},\\ &\approx \sum_{i=1}^{N} w_{k-1|k-1}^i p(\mathbf{x}_k|\mathbf{x}_{k-1}). \end{aligned} \tag{2.25}$$

The particles x_{k-1}^i in above equations (2.25) are sampled from proposal distribution $\pi(\mathbf{x}_k|x_{k-1}^i)$, i.e. $x_{k-1}^i \approx \pi(\mathbf{x}_k|x_{k-1}^i)$. Proposal distribution is very often defined by the state transition PDF, that is

$\pi(\mathbf{x}_k|x^i_{k-1}) = p(\mathbf{x}_k|x^i_{k-1})$. In this case, the weights updates results to

$$w^i_{k|k-1} = \frac{p(x^i_k|x^i_{k-1})}{\pi(x^i_k|x^i_{k-1}, y_k)} w^i_{k-1|k-1} = \frac{p(x^i_k|x^i_{k-1})}{p(x^i_k|x^i_{k-1})} w^i_{k-1|k-1} = w^i_{k-1|k-1}. \tag{2.26}$$

The measurement update $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ (2.12) is computed by a Bayesian formula (2.16), which can be in terms of the particles x^i_k represented as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \approx \sum_{i=1}^{N} w^i_{k|k-1} p(\mathbf{y}_k|\mathbf{x}_k)\delta(\mathbf{x}_k - x^i_k). \tag{2.27}$$

Denominator in (2.16) is only a normalizing factor independent of $\mathbf{x}_k$ thus can be safety omitted if the distribution is numerically normed as shown by (2.27). Similarly, the particle filter weights are updated as

$$w^i_{k|k} = \frac{w^i_{k|k-1} p(y_k|x^i_k)}{\sum_{j=1}^{N} w^j_{k|k-1} p(y_k|x^j_k)}. \tag{2.28}$$

The MC recursion tempt to degrade over time as all relative weights would tend to zero except for one that tends to one. Therefore, when particle depletion ratio reaches 0.5 a Sampling Importance Resampling (SIR) or Sampling Importance Sampling (SIS) techniques are applied in the recursion. A single step of the Particle filter algorithm is presented as the Alg. 2.

Algorithm 2: Particle filter

Step 0: Initialize

Initilaze particles according to $\hat{x}_0^{(i)} = p(x_0)$ *and appropiate weights* $w_{0|0}^{(i)}$ *for all* $i = 1, \dots, N$.

Step 1: Time update

Generate new particles according to the proposal distribution

$$x^i_k \approx \pi(x_k|x^i_k, x_k), \quad \forall i \in \{1, \dots, N\}$$

Update the weights according to

$$w^i_{k|k-1} = \frac{p(x^i_k|x^i_{k-1})}{\pi(x^i_k|x^i_{k-1}, y_k)} w^i_{k-1|k-1}.$$

which in case $\pi(x_k|x_{k-1}, y_k) = p(x_k|x^i_{k-1})$ *simplifies to* $w^i_{k|k-1} = w^i_{k-1|k-1}$.

Step 2: Measurement update

Claculate importance weights according to

$$w^i_{k|k} = \frac{w^i_{k|k-1} p(y_k|x^i_k)}{\sum_{j=1}^{N} w^j_{k|k-1} p(y_k|x^j_k)}, \qquad \forall i \in \{1, \dots, N\}$$

Step 3: Resampling

Apply sampling importance resampling (SIR) or sampling importance sampling (SIS) when the particle depletion ratio reaches 0.5.

2.3.4 Interactive Multiple Model Filter

Assume a discrete state space system (2.11, 2.12) with Markovian coefficients

$$\mathbf{x}_k = \mathbf{f}_k(m_k)\mathbf{x}_{k-1} + \mathbf{g}_k(m_k)\mathbf{v}_k \quad \text{or} \quad p(\mathbf{x}_k, m_k|\mathbf{x}_{k-1}) \tag{2.29}$$

$$\mathbf{y}_k = \mathbf{h}_k(m_k)\mathbf{x}_k + \mathbf{w}_k \quad \text{or} \quad p(\mathbf{y}_k|m_k, \mathbf{x}_k). \tag{2.30}$$

Parameter m_k is a mode state taking values $(1, 2, ..., N_m)$ and represents the current system model in use. The history of the mode state m_k can modeled as a homogeneous Markov chain with transition probability matrix (TPM) $\boldsymbol{\pi}^{ji}_{k-1|k-1} = p(m^i_k|m^j_{k-1})$. The goal of estimation in Markovian jump systems (MJS) framework is to infer the posterior density of the target $\mathbf{x}_k$ for a specific model history $m^i_{1:k}$ with the available sensor measurements $\mathbf{y}_{1:k}$. This estimation problem relates to the recursive evaluation of the probability density function $p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k})$ over all possible mode sequences m_k that end in the mode m^i_k in five consecutive steps (2.31 - 2.35).

$$p(m^j_{k-1}|\mathbf{y}_{1:k-1}) \xrightarrow{\text{Mixing}} p(m^i_k|\mathbf{y}_{1:k-1}) \tag{2.31}$$

$$p(\mathbf{x}_{k-1}|m^j_{k-1}, \mathbf{y}_{1:k-1}) \xrightarrow{\text{Mixing}} p(\mathbf{x}_{k-1}|m^i_k, \mathbf{y}_{1:k-1}) \tag{2.32}$$

$$p(\mathbf{x}_{k-1}|m^i_k, \mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Prediction}} p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k-1}) \tag{2.33}$$

$$p(m^i_k|\mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Mode}} p(m^i_k|\mathbf{y}_{1:k}) \tag{2.34}$$

$$p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Measurement}} p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k}) \tag{2.35}$$

Product of the posterior probability of the mode sequence $p(m^i_k|\mathbf{y}_{1:k})$ (2.34) and the posterior distribution over the system state $p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k})$ (2.35) conditioned on the mode sequence m^i_k can be seen in a view of total probability theorem as the $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ defined as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \sum_{i=1}^{N_m} p(m^i_k|\mathbf{y}_{1:k})p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k}). \tag{2.36}$$

The recursion (2.31 - 2.35), extensively discussed in Sec. 5.3.1, yields a common approximation of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ in (2.36) by a Gaussian mixture with N^2_m components i.e.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_m} \boldsymbol{\mu}^i_k \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}^i_{k|k}, \boldsymbol{\Sigma}^i_{k|k}). \tag{2.37}$$

The number of components in the Gaussian mixture (2.37) grows exponentially over time, therefore techniques such as pruning or merging are used to reduce the number of Gaussian mixture components [93]. Interactive Multiple Model (IMM) filter introduced by Blom and Bar-Shalom [20] approximates posterior mode probabilities μ^i_k in equation (2.37) with

$$\begin{aligned}\boldsymbol{\mu}^i_k \equiv& p(m^i_k|\mathbf{y}_{1:k}),\\ \boldsymbol{\mu}^i_k =& \frac{\mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}^i_{k|k-1}, \mathbf{S}^i_k)\Sigma^{N_m}_{j=1}\boldsymbol{\pi}_{ji}\boldsymbol{\mu}^j_{k-1}}{\Sigma^{N_m}_l \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}^l_{k|K-1}, \boldsymbol{S}^l_k)\Sigma^{N_m}_{j=1}\boldsymbol{\pi}_{jl}\boldsymbol{\mu}^j_{k-1}},\end{aligned} \tag{2.38}$$

Mode transition probability $\boldsymbol{\pi}^{ji}_{k-1|k-1} = p(m^i_k|m^j_{k-1})$ is genuinely assumed to be known a priori and to be constant time invariant matrix. With such an approximation the overall posterior mean $\hat{x}_{k|k}$ and covariance $\Sigma_{k|k}$ are equal to formulas (2.39) and (2.40) respectively.

$$\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{N_m} \boldsymbol{\mu}^i_k \hat{\mathbf{x}}^i_{k|k} \tag{2.39}$$

$$\boldsymbol{\Sigma}_{k|k} = \sum_{i=1}^{N_m} \boldsymbol{\mu}^i_k [\boldsymbol{\Sigma}^i_{k|k} + (\hat{\boldsymbol{x}}^i_{k|k} - \hat{\boldsymbol{x}}_{k|k})(\hat{\boldsymbol{x}}^i_{k|k} - \hat{\boldsymbol{x}}_{k|k})^T] \tag{2.40}$$

This estimate (2.39) and covariance (2.40) can be given to the user as the output. The mode conditional means $\{\hat{x}^i_{k|k}\}^{N_m}_{i=1}$, covariances $\{\Sigma^i_{k|k}\}^{N_m}_{i=1}$ and mode probabilities $\{\mu^i_k\}^{N_m}_{i=1}$ must be calculated recursively from their previous values $\{\hat{x}^i_{k-1|k-1}, \Sigma^i_{k-1|k-1}, \mu^i_{k-1}\}^{N_m}_{i=1}$. A single step of the IMM filter is presented as the Alg. 3.

Algorithm 3: Interactive Multiple Model filter

Step 1: Mixing

Calculate the mixing probabilitis $\{\mu^{ji}_{k-1|k-1}\}^{N_m}_{i,j=1}$ *as*

$$\mu^{ji}_{k-1|k-1} = \frac{\pi_{ji}\mu^j_{k-1}}{\Sigma^{N_m}_{l=1}\pi_{li}\mu^l_{k-1}}.$$

Calculate the mixed estimates $\{\hat{x}^{0i}_{k-1|k-1}\}^{N_m}_{i=1}$ *and cova- raince*s $\{\Sigma^{0i}_{k-1|k-1}\}^{N_m}_{i=1}$ *as*

$$\hat{x}^{0i}_{k-1|k-1} = \sum_{j=1}^{N_m} \mu^{ji}_{k-1|k-1}\hat{x}^j_{k-1|k-1},$$

$$\Sigma^{0i}_{k-1|k-1} = \sum_{j=1}^{N_m} \mu^{ji}_{k-1|k-1}[\Sigma^j_{k-1|k-1} + (\hat{x}^j_{k-1|k-1} - \hat{x}^{0i}_{k-1|k-1})(\hat{x}^j_{k-1|k-1} - \hat{x}^{0i}_{k-1|k-1})^T].$$

Step 2: Mode matched prediction update

For ith model, i = 1,...,N_m *caluclate the predicted estimate* $\hat{x}^i_{k|k-1}$ *and covariance* $\Sigma^i_{k|k-1}$ *from the mixed estiamte* $\hat{x}^{0i}_{k-1|k-1}$ *and covariance* $\Sigma^{0i}_{k-1|k-1}$ *as*

$$\hat{x}^i_{k|k-1} = F^i\hat{x}^{0i}_{k-1|k-1},$$

$$\Sigma^i_{k|k-1} = F^i\Sigma^{0i}_{k-1|k-1}F^{i^T} + G^iQG^{i^T}.$$

Step 3: Mode matched measurement update

For ith model, i = 1,...,N_m *caluclate the updated estimate* $\hat{x}^i_{k|k}$ *and covariance* $\Sigma^i_{k|k}$ *from the mixed estiamte* $\hat{x}^i_{k|k-1}$ *and covariance* $\Sigma^i_{k|k-1}$ *as*

$$\hat{x}^i_{k|k} = \hat{x}^i_{k|k-1} + K^i_k(y_k - \hat{y}^i_{k|k-1}), \Sigma^i_{k|k} = \Sigma^i_{k|k-1} - K^i_kS^i_kK^{i^T}_k,$$

$$\hat{y}^i_{k|k-1} = H^i\hat{x}^i_{k|k},$$

$$S^i_k = H^i\Sigma_{k|k-1}H^{i^T} + D^iRD^{i^T},$$

$$K^i_k = \Sigma^i_{k|k-1}H^{i^T}S^{i^{-1}}_k,$$

and the updated mode probability μ^i_k *as*

$$\mu^i_k = \frac{\mathcal{N}(y_k; \hat{y}^i_{k|k-1}, S^i_k)\Sigma^{N_m}_{j=1}\pi_{ji}\mu^j_{k-1}}{\Sigma^{N_m}_{l}\mathcal{N}(y_k; \hat{y}^l_{k|K-1}, S^l_k)\Sigma^{N_m}_{j=1}\pi_{jl}\mu^j_{k-1}}.$$

Step 4: Output estimate calculation

Calculate the overal state estimate $\hat{x}_{k|k}$ *and covariance* $\Sigma_{k|k}$ *as*

$$\hat{x}_{k|k} = \sum_{i=1}^{N_m}\mu^i_k\hat{x}^i_{k|k}, \Sigma_{k|k} = \sum_{i=1}^{N_m}\mu^i_k[\Sigma^i_{k|k} + (\hat{x}^i_{k|k} - \hat{x}_{k|k})(\hat{x}^i_{k|k} - \hat{x}_{k|k})^T].$$

2.4 Dempster-Shafer Theory

Suppose a following frame of discernment (FOD) (Fig. 6.6 right) for a power set $2^{\{O,F\}}$

$$2^{\mathcal{X}} = \{\emptyset, O, F, \Omega\}. \tag{2.41}$$

Symbols O, F, $\emptyset$ and Ω represent the $occupied$, $free$, $null$ and the $conflict$ set, respectively. Set $\emptyset$ will always have a mass function equal to zero $m_k(\emptyset) = 0$, since a cell must be in any of the states $m_k \in \{O, F, \Omega\}$ defined in the FOD (2.42). The latter set is especially interesting since it represents the status of a cell being neither free F nor occupied O but in conflict Ω. Even though the power set $2^{\mathcal{X}}$ is composed of four propositions, it can be seen from basic probability assignment (BPA) theorem that masses $m_k(O)$ and $m_k(F)$ are sufficient to fully describe the whole FOD (2.41).

$$\begin{aligned}\sum_{A \in 2^{\mathcal{X}}} m_k(A) = m_k(\emptyset) + m_k(O) + m_k(F) + m_k(\Omega) = 1,\\ m(\Omega) = m_k(O \cup F) = 1 - m_k(O) - m_k(F),\\ m_k(\emptyset) = 0.\end{aligned} \tag{2.42}$$

Belief X of FOD $2^{\mathcal{X}}$ is a resulting probability that accounts for all the evidence that supports the proposition X. It represents the degree to which X, in our case occupied or free $X \in \{O, F\}$, is believed to be true.

$$Bel(X) = \sum_{A \subseteq X} m(A). \tag{2.43}$$

On the other hand, plausibility of set $X \in \{O, F\}$ of FOD $2^{\mathcal{X}}$ considerate the evidence that does not provide knowledge about the preposition X. It therefore represents the degree to which X is believed not to be false.

$$Pl(X) = 1 - \sum_{A \cap X = \mathcal{X}, A \in 2^{\mathcal{X}}} m(A). \tag{2.44}$$

According to DST, two sources of information $m_{k(1)}$ and $m_{k(2)}$ are combined into a fused belief function $m_{k(12)}$ through the operator $\oplus$, which could represent either conjunction $\cap$ or disjunction $\cup$ formula, in accordance with $X \neq \emptyset$ (2.45) and $X = \emptyset$ (2.46)

$$m_{k(12)}(X) = (m_{k(1)} \oplus m_{k(2)})(X) = \frac{(m_{k(1)} \cap m_{k(2)})(X)}{1 - (m_{k(1)} \cap m_{k(2)})(\emptyset)}, \tag{2.45}$$

$$m_{k(12)}(\emptyset) = (m_{k(1)} \oplus m_{k(2)})(\emptyset) = 0. \tag{2.46}$$

The $(m_{k(1)} \cap m_{k(2)})(X)$ is the conjunctive combination rule such that

$$(m_{k(1)} \cap m_{k(2)})(X) = \sum_{A \cap B = X; A, B \in 2^{\mathcal{X}}} m_{k(1)}(A)\, m_{k(2)}(B). \tag{2.47}$$

The denominator of (2.45) is a normalization factor where the $(m_{k(1)} \cap m_{k(2)})(\emptyset)$ is a measure of a conflict between sources of information. If combining information is highly contradicting, term $(m_{k(1)} \cap m_{k(2)})(\emptyset)$ becomes close to 1 and denominator of (2.45) become close to zero. That is the fundamental weakens in DST theory and underlaying topic of criticism. Furthermore, it can be shown that the Dempster-Shafer combination rule is commutative and associative. Therefore, the fusion of n sensors $\{m_{k(1)}, m_{k(2)}, \ldots, m_{k(n)}\}$ can be performed sequentially, as follows

$$m_{k(12n)}(X) = ((m_{k(1)}(X) \oplus m_{k(2)}(X)) \oplus \ldots) \oplus m_{k(n)}(X). \tag{2.48}$$

2.5 Data Association

In general, information about target behavior is obtained by $n_o = (1, ..., N_o)$ spatially distributed sensors (observers). We denote the observer locations by $\mathbf{p}_o^{n_o}$, and allow them to vary in time. At any particular time step k the combined set of measurements from all available observers will be denoted by $\mathbf{y} = (\mathbf{y}^1, ..., \mathbf{y}^{N_o})$, where $\mathbf{y}^{n_o} = (\mathbf{y}^{n_o,1}, ..., \mathbf{y}^{n_o,N_y})$ is the vector comprising of the number of N_y measurements originated from the n_o-th observer. During the course of tracking, measurements may not only originate from the target itself but also from false alarms e.g. clutter. Additional clutter measurements could be a result of multi-path effects, spurious objects, sensor errors, etc. For this reason, number of measurements N_y originated from each sensor (observer) $\mathbf{y}^{n_o}$ is not necessary constant but varies over time. We assume that each target could generate at most one measurement per sensor at the given time k. We further assume that several or all measurements may arise due to clutter.

We refer to the assignment of either measurement to target or target to measurement as the data association process, with the later being utilized here. Both formulations carry the same information, but are used in different contexts. We define target to measurement association hypothesis as $\theta = (\theta^1, ..., \theta^{N_o})$, where $\theta^{n_o} = (\mathbf{r}^{n_o}, \mathbf{y}_c^{n_o}, \mathbf{y}_t^{n_o})$ is the target to measurement association hypothesis at the n_o-th observer, with $\mathbf{y}_c^{n_o}$ and $\mathbf{y}_t^{n_o}$ being the number of clutter and target originated measurements, respectively. Note that $\mathbf{y}^{n_o} = \mathbf{y}_t^{n_o} + \mathbf{y}_c^{n_o}$. Elements of the association vector $\mathbf{r}^{n_o} = (\mathbf{r}^{n_o,1}, ..., \mathbf{r}^{n_o,N_t})$ are given by

$$\mathbf{r}^{n_o,n_t} = \begin{cases} 0 & \text{if the target } n_t \text{ is undetected by an observer } n_o, \\ \mathbf{y}^{n_o} \in (\mathbf{y}^{n_o,1}, ..., \mathbf{y}^{n_o,N_y}) & \text{if the target } n_t \text{ generated the measurment } n_y \\ & \text{at observer } n_o. \end{cases} \tag{2.49}$$

We assume that measurements originated from a single observer are conditionally independent from each other and of those originated from other observers. Therefore, the measurement likelihood $p(\mathbf{y}|\mathbf{x})$ can be factorized based on the measurement to target association hypothesis as follows

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{n_o=1}^{N_o} \left[\prod_{n_{y_c}=1}^{N_{yc}} p_c^{n_o}(\mathbf{y}^{n_o,n_{y_c}}) \prod_{n_t=1}^{N_t} p^{n_o}(\mathbf{y}^{n_o,\mathbf{r}^{n_o,n_t}} | \mathbf{x}^{n_t}) \right]. \tag{2.50}$$

Likelihood $p_c^{n_o}$ is the clutter model defined for the observer n_o, which is genuinely assumed to be uniform over the volume of measurement space $\mathbf{V}^{n_o}$. Therefore, equation (2.50) could be simplified to

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{n_o=1}^{N_o} \left[(\mathbf{V}^{n_o})^{\mathbf{y}_c^{n_o}} \prod_{n_t=1}^{N_t} p^{n_o}(\mathbf{y}^{n_o,\mathbf{r}^{n_o,n_t}} | \mathbf{x}^{n_t}) \right]. \tag{2.51}$$

The likelihood in the second product can be rewritten as

$$p^{n_o}(\mathbf{y}^{n_o,\mathbf{r}^{n_o,n_t}} | \mathbf{x}^{n_t}) = \begin{cases} 1 & \text{if } \mathbf{r}^{n_o,n_t} = 0, \\ p_t^{n_o}(\mathbf{y}^{n_o,\mathbf{r}^{n_o,n_t}} | \mathbf{x}^{n_t}) & \text{otherwise.} \end{cases} \tag{2.52}$$

The association hypothesis θ might not be know a priori, and thus needs to be estimated or marginalized from the problem. To achieve this in a Bayesian framework it is necessary to define a prior distribution $p(\theta)$ over the association hypothesis θ. We assume the prior $p(\theta)$ to be independent of the state vector and past values of association hypothesis. Thus, the prior can be factorized over the observers n_o as follows

$$p(\theta) = \prod_{n_o=1}^{N_o} p(\theta^{n_o}). \tag{2.53}$$

Prior $p(\theta^{n_o})$ is for the each observer n_o defined as

$$p(\theta^{n_o}) = p(\mathbf{r}^{n_o} | \mathbf{y}_c^{n_o}, \mathbf{y}_t^{n_o}) p(\mathbf{y}_c^{n_o}) p(\mathbf{y}_t^{n_o}) \tag{2.54}$$

where $\mathbf{y}_c^{n_o}$ and $\mathbf{y}_t^{n_o}$ are clutter and target originated measurements at observer n_o, respectively. Measurement to target associations at observer n_o are included in vector $\mathbf{r}^{n_o}$. Furthermore, we define $p(\mathbf{r}^{n_o} | \mathbf{y}_c^{n_o}, \mathbf{y}_t^{n_o})$, $p(\mathbf{y}_c^{n_o})$ and $p(\mathbf{y}_t^{n_o})$ as follows

$$p(\mathbf{r}^{n_o} | \mathbf{y}_c^{n_o}, \mathbf{y}_t^{n_o}) = [N_{\theta^{n_o}}(\mathbf{y}_c^{n_o}, \mathbf{y}_t^{n_o})]^{-1} \tag{2.55a}$$

$$p(\mathbf{y}_c^{n_o}) = (\theta_c^{n_o})^{\mathbf{y}_c^{n_o}} \exp(-\theta_c^{n_o} / \mathbf{y}_c^{n_o}!) \tag{2.55b}$$

$$p(\mathbf{y}_t^{n_o}) = \binom{\mathbf{x}_t}{\mathbf{y}_t^{n_o}} p_d^{\mathbf{y}_t^{n_o}} (1 - p_d)^{\mathbf{x}_t - \mathbf{y}_t^{n_o}}. \tag{2.55c}$$

The number of valid hypotheses for a given number of target measurements equals to

$$N_{\theta^{n_o}}(\mathbf{y}_c^{n_o}, \mathbf{y}_t^{n_o}) = \binom{\mathbf{x}_t}{\mathbf{y}_t^{n_o}} \frac{\mathbf{y}^{n_o}!}{(\mathbf{y}^{n_o} - \mathbf{y}_t^{n_o})!} \tag{2.56}$$

Equation (2.56) enumerates the number of possible associations between the detections $\mathbf{y}_t^{n_o}$ and the targets $\mathbf{x}_t$ multiplied by the number of ways a subset $\mathbf{y}_t^{n_o}$ can be chosen from the available measurements $\mathbf{y}_t^{n_o}$. The number of clutter measurements is assumed to be Poisson distributed with rate parameter of $\theta_c^{n_o}$, which is supposed to be fixed and known. It is common practice to define the rate parameter $\theta_c^{n_o}$ in terms of the spatial density of the clutter μ^{n_o} constrained by the volume of measurement space $\mathbf{V}^{n_o}$, i.e. $\theta_c^{n_o} = \mu^{n_o}\mathbf{V}^{n_o}$. Assuming that all targets $\mathbf{x}_t$ share the same fixed and known detection probability p_d, the association prior $N_{\theta^{n_o}}$ can be factorized over the individual target associations as

$$p(\theta^{n_o}) = p(\mathbf{y}_c^{n_o}) \prod_{n_t=1}^{N_t} p(\theta^{n_o}) p(\mathbf{r}^{n_o,n_t}|\mathbf{r}^{n_o,1:n_t-1}) \tag{2.57}$$

where

$$\mathbf{r}^{n_o,1:n_t-1}) \propto \begin{cases} 1 - p_d & \text{if } \mathbf{y}^{n_o} = 0, \\ 0 & \text{if } \mathbf{y}^{n_o} > 0 \text{ and } \mathbf{y}^{n_o} \in \{\mathbf{r}^{n_o,1:n_t-1}\}, \\ p_d/\mathbf{y}^{n_o,n_t} & \text{otherwise.} \end{cases} \tag{2.58}$$

Vector of unassigned measurements $\mathbf{y}^{n_o,n_t} = \mathbf{y}^{n_o} - \mathbf{y}^{n_o,l}$, where $l : \mathbf{r}^{n_o,n_l} \neq 0 \; l \in (1, ..., n_t - 1)$, takes into account the assignment of previous associations $(n_t - 1)$. Note that this sequential factorization can be performed over any permutation of the target ordering. Note further, that the prior for the number of target detections is implicitly captured by the factorization of the association vector, and hence disappears from the expression for the prior. This factorization will aid in the design of efficient sampling strategies to combat the curse of dimensionality with an increase in the number of targets.

2.6 Bayesian Target Tracking

A high level implementation of Bayesian tracking logic (Fig. 2.1), for system defined by (2.11) and (2.12), will be discussed here.

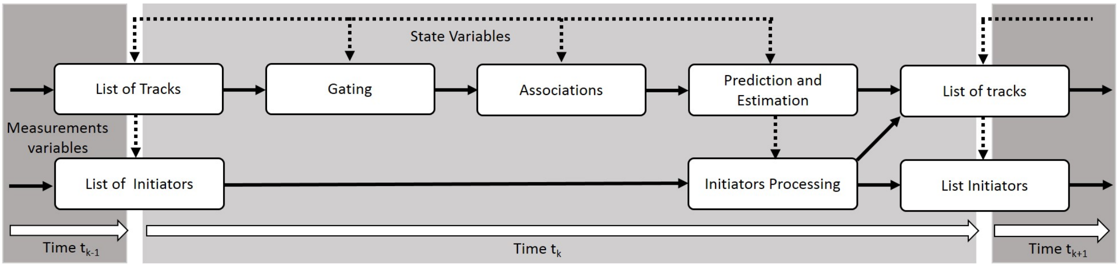


Figure 2.1: Track handling in a target tracking.

- **Initialization**, at time stamp $k = 0$ initialize tentative tracks, construct *initiator*, from all measurements $\mathbf{y}_{k=1}$.

$$\textit{initiator} = initialize(\textit{state estimate}, \textit{state covariance}, \textit{meas. estimate}, \textit{inovation covariance}, \ldots$$
$$\textit{age}, \textit{last update time}, \textit{M/N logic state}, \textit{score}, \ldots)$$

- **Prediction update**, compute prediction update of a state $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ and measurement vector $p(\hat{\mathbf{y}}_{k+1}|\mathbf{x}_k, \mathbf{y}_{1:k-1})$ according to (2.15) for each track *initiator*.

$$\textit{initiator} = predictionUpdate(\textit{initiator})$$

- **Gating**, impose a hard decisions on measurements which are not considered feasible for a given *initiator*. Gating region, in case of elliptical gates, for a *initiator* state vector $\mathbf{x}_k^j$ is centered around predicted measurement location $\hat{\mathbf{y}}_{k+1|k}^j$ with a volume proportional to

$$(\mathbf{y}_{k+1}^i - \tilde{\mathbf{y}}_{k+1|k}^j)^T \mathbf{S}_{k+1|k}^{-1} (\mathbf{y}_{k+1}^i - \tilde{\mathbf{y}}_{k+1|k}^j) > \gamma_G. \tag{2.59}$$

Here the $\mathbf{S}_{k+1|k}^{-1}$ is an innovation covariance and γ_G is χ^2 distributed threshold given the probability of gating p_G and number of measurements $\mathbf{y}_{k+1}^I$.

$$\textit{gate decesions} = gating(\textit{initiator}, \textit{measurments})$$

- **Measurement association** (Sec. 2.5), in the cases when measurements fall into the gate $\{y_k^i\}_{i=1}^{n_y}$ evaluate them against hypotheses θ^0 and θ^i for a given *initiator* $\mathbf{x}_k^j$.

$$\begin{aligned}\theta^0 &= \text{All measurments } \{y_k^i\}_{i=1}^{n_y} \text{ are false alarms (FA)}\\ \theta^i &= \text{Measurment } \{y_k^i\}_{i=1}^{n_y} \text{ belongs to the target, rest are false alarms (FA)}\end{aligned}$$

Probabilities $p(\theta^i|\{y_{0:k}^i\}_{i=1}^{n_y})$ associated with hypothesis θ^i are computed as a cases where

$$p(\theta^i|\{y_k^i\}_{i=1}^{n_y}) \propto \begin{cases}(1 - p_D p_G)\beta_{FA}^{n_y} & \text{if} \quad i = 0,\\ p(y_k^i|\{y_{0:k-1}^i\}_{i=1}^{n_y}) p_D \beta_{FA}^{n_y} & \text{otherwise.}\end{cases} \tag{2.60}$$

Within this work the probability $p(y_k^i|\{y_{0:k-1}^i\}_{i=1}^{n_y})$ is defined as $\mathcal{N}(\mathbf{y}_k; \mathbf{y}_{k|k-1}, \mathbf{S}_{k|k-1})$.

$$\textit{association decesions} = associate(\textit{initiator}, \textit{measurments}, \textit{gate decesions})$$

- **Measurement update**, process the current *initiators* with their associated measurements. Compute Bayesian update $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ (2.16) on basis of association PDF, which can be in terms of totally probability theorem expressed as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \sum_{i=1}^{n_y} p(\mathbf{x}_k|\theta_k^i, \mathbf{y}_{1:k})\, p(\theta^i|\{y_k^i\}_{i=1}^{n_y}), \tag{2.61}$$

where n_y is number of measurements in the gate.

$$\textit{initiator} = measurmentUpdate(\textit{initiator}, \textit{measurments}, \textit{association decesions})$$

- **Track management** [13], is a decision logic purpose of which is to start a new *initiators* from unassigned measurements, update track logic states and scores, confirm or delete those *initiators* which does not meet the criteria.

$$\begin{aligned}\textit{falg} =& checkCreate(\textit{initiator}),\\ \textit{falg} =& checkConfirm(\textit{initiator}),\\ \textit{falg} =& checkDelete(\textit{initiator}).\end{aligned}$$

2.7 Simultaneous Localization and Mapping

Nowadays approaches to the environment mapping are based either on feature recognition and tracking techniques, i.e. feature-based models ([36], [45], [121], [156], [80]), or operate on the occupancy based map building principle ([62], [76], [79], [108], [109]). The two approaches are by no means mutually exclusive, but they in fact supplement each others weaknesses.

2.7.1 Feature Based SLAM

The Simultaneous Localization and Mapping (SLAM) refers to the problem of determining ones pose $\mathbf{x}$ (position and orientation) using observation to landmarks $\mathbf{m}$ of unknown position. Landmarks (features, beacons) can represent any well-defined point in the world as corners, lines or markers. The autonomous system is tasked to incrementally build a map of its environment and at the same time use this map to compute its own location. Conversely, in the autonomous vehicle application the location of the vehicle is supposed to be known and the objective is to map the surrounding environment for possible obstacles and threats. Process of map building requires recursively propagate the state distribution $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and landmarks positions uncertainties $p(\mathbf{m}_{k+1}|\mathbf{m}_k)$ recursively in tree steps: a dynamic motion model (2.62), a map (2.63) and a measurement model (2.64).

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k) \quad \text{or} \quad p(\mathbf{x}_{k+1}|\mathbf{x}_k) \tag{2.62}$$

$$\mathbf{m}_{k+1} = \mathbf{m}_k \quad \text{or} \quad p(\mathbf{m}_{k+1}|\mathbf{m}_k) \tag{2.63}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{m}_k, \mathbf{u}_k) + \mathbf{w}_k \quad \text{or} \quad p(\mathbf{y}_k|\mathbf{x}_k) \tag{2.64}$$

In above equations, $\mathbf{f}$ represents a nonlinear function of the target state vector $\mathbf{x}_k$, the vector of input signals $\mathbf{u}_k$ and process noise $\mathbf{v}_k$. The list of landmarks or map features are represented by multi-dimensional matrix $\mathbf{m}_k$. Variable $\mathbf{h}$ represents a nonlinear relationship between sensor output $\mathbf{y}_k$, target state vector $\mathbf{x}_k$ and system inputs $\mathbf{u}_k$ affected by a measurement noise $\mathbf{w}_k$. The goal of nonlinear estimation is to infer the state variable $\mathbf{x}_k$ conditioned on landmarks $\mathbf{m}_k$ with the available sensor measurements $\mathbf{y}_{1:k}$. By using the Bayesian framework, this estimation problem relates to the recursive evaluation of the probability density function (PDF) $p(\mathbf{x}_k, \mathbf{m}_k|\mathbf{y}_{1:k})$ in two consecutive steps, the prediction and the measurement update of the state vectors.

$$p(\mathbf{x}_k, \mathbf{m}_k|\mathbf{y}_{1:k}) \xrightarrow[\text{Update}]{\text{Predition}} p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k}) \tag{2.65}$$

$$p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k}) \xrightarrow[\text{Update}]{\text{Measurment}} p(\mathbf{x}_{k+1}\mathbf{m}_{k+1}|\mathbf{y}_{1:k+1}) \tag{2.66}$$

The prediction state density $p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k})$ of state $\mathbf{x}_k$ and map $\mathbf{m}_k$ is calculated from the prior PDF $p(\mathbf{x}_k, \mathbf{m}_k|\mathbf{y}_{1:k})$ by using Chapman-Kolmogorov equation

$$\begin{gathered} p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k}) = \\ \iint p(\mathbf{m}_{k+1}|\mathbf{x}_{k+1}, \mathbf{m}_k) p(\mathbf{x}_{k+1}|\mathbf{m}_k) p(\mathbf{x}_k, \mathbf{m}_k|\mathbf{y}_{1:k}) d\mathbf{x}_k d\mathbf{m}_k. \end{gathered} \tag{2.67}$$

Equality (2.67) follows the 1^{st} order Markov property which assumes that $p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k})$ only depends on states $\mathbf{x}_{k+1}$, $\mathbf{m}_{k+1}$ and $\mathbf{x}_k$, $\mathbf{m}_k$ at time $k+1$ and k respectively. The measurement update $p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k+1})$ is computed from the prior distribution (2.67) measurements $\mathbf{y}_k$ by a Bayesian formula which results in

$$p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k+1}) = \frac{p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}, \mathbf{m}_{k+1}) p(\mathbf{m}_{k+1}|\mathbf{x}_{k+1}) p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k})}{p(\mathbf{y}_{k+1}|\mathbf{y}_{1:k})} \tag{2.68}$$

The 1^{st} order Markov property for equation (5.10) implies that $p(\mathbf{x}_{k+1}, \mathbf{m}_{k+1}|\mathbf{y}_{1:k+1})$ only depends on measurement $\mathbf{y}_{k+1}$ at time $k+1$.

Typical sensor observations $\mathbf{y}_k$ are taken from the scanning laser which returns horizontal distance and bearing to a landmark. Landmarks could be also obtained from the visual sensors in forms of distinct corners or patches. The measurements are inferred into the recursion as a likelihood functions $p(\mathbf{y}_k|\mathbf{x}_k)$ (2.64). Without the loss of generality, the measurement equation could be expressed at time k as follows

$$\mathbf{y}_k = \mathbf{H}_k^x \hat{\mathbf{x}}_k + \mathbf{H}_k^m \mathbf{C}_k^{1:I_k} \hat{\mathbf{m}}_k + \mathbf{e}_k. \tag{2.69}$$

Equation (2.69) reads, that a particular measurement $i \in \{1, \ldots, I\}$ obtained by a sensor in a scan k, e.g. $\mathbf{y}_k^i$, needs to be associated with an already existing landmark $\mathbf{m}_k^j$ via association matrix $\mathbf{C}_k^{1:I_k}$. The data association process (Sec. 2.5) determines weather the detection $\mathbf{y}_k^i$ in fact originates from the landmark, or it is a clutter, or it should be declared as a new landmark/obstacle through the process commonly referred

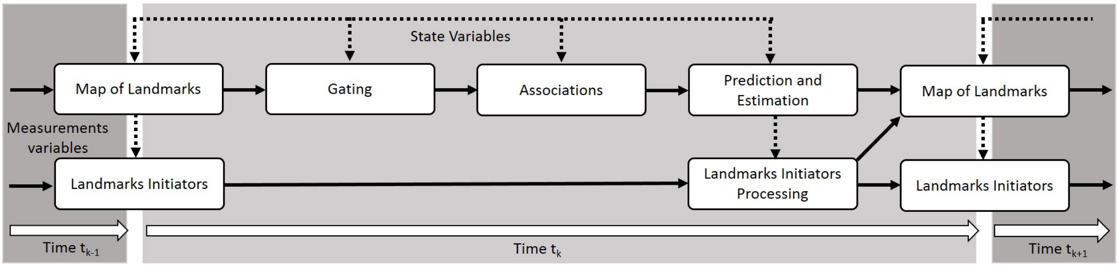


Figure 2.2: Detections handling in a SLAM framework.

in a target tracking to as a track life. The modification of the track life routine for the SLAM, e.g. landmark handling, is visualized on the Fig. 2.2.

Assume that certain number of measurements $\mathbf{y}_k^{1:I}$ were obtained at scan k and there already exist landmarks $\mathbf{m}_k$ in the map. Landmarks should be seen as an objects which keep some level of their own history. Typical landmark carries knowledge of its age, state estimate $\hat{\mathbf{x}}_{k+1|k}$, state covariance $\hat{\mathbf{P}}_{k+1|k}$, last update time, score, M/N logic state, etc. The primary goal of the landmark logic is to:

- initialize (detect) new landmarks by forming a tentative landmarks and confirm only the persistent ones,
- maintain the current landmarks location with respect to the vehicle position by feeding the confirmed landmarks with only relevant data,
- delete the landmarks with the persistent absence of data or low quality,

under the notion of false alarms (FA) and clutter. At the time k, the poses of all landmarks $\mathbf{m}_k$ are with respect to the vehicle state $\mathbf{x}_k$ propagated into the time $k+1$. The prediction step of a filter (2.65) alongside with state variables flow is visualized by dashed lines. Following the Bayesian recursion rule (2.67), observations at time k $\mathbf{y}_k^{1:I}$ needs to inferred into the predicted landmarks $\hat{\mathbf{m}}_{k+1|k}$ and vehicle states $\hat{\mathbf{x}}_{k+1|k}$. This step requires to resolve the landmark to measurement association problem i.e. to find $\mathbf{C}_k^{1:I_k}$ matrix in (2.69). Genuinely, it is not recommended to evaluate all detection to landmark combinations when the number of involved quantities is ≥ 10. Therefore, we impose a hard decisions i.e. gating on measurements which are not considered feasible for a given landmark. Gating region for a landmark $\mathbf{m}_k^j$ is centered around predicted measurement location $\hat{\mathbf{y}}_{k+1|k}^j$ with a volume proportional to

$$(\mathbf{y}_{k+1}^i - \tilde{\mathbf{y}}_{k+1|k}^j)^T \mathbf{S}_{k+1|k}^{-1} (\mathbf{y}_{k+1}^i - \tilde{\mathbf{y}}_{k+1|k}^j) > \gamma_G. \tag{2.70}$$

Here the $\mathbf{S}_{k+1|k}^{-1}$ is an innovation covariance and γ_G is χ^2 distributed threshold given the probability of gating p_G and number of measurements $\mathbf{y}_{k+1}^I$. In the cases when measurements fall into the gate y_{k+1}^i there are probabilistically evaluated against hypotheses θ_i governed by a multiple-sensor multiple-landmarks association process (JPDA) (Sec. 2.5).

2.7.2 Grid Based SLAM

Consider a system model defined by set of equations (2.62, 2.63, 2.64). The map of the environment, i.e. the occupancy grid map, is represented by multi-dimensional matrix $\mathbf{m}_k$ (2.71) conditioned on the vehicle state $\mathbf{x}_k$ and sensor observations $\mathbf{y}_k$.

$$\mathbf{m}_k = \begin{bmatrix} m_k^{11} & m_k^{12} & \cdots & m_k^{1j} \\ m_k^{21} & m_k^{22} & \cdots & m_k^{2j} \\ \vdots & \vdots & \ddots & \vdots \\ m_k^{i1} & m_k^{i2} & \cdots & m_k^{ij} \end{bmatrix} \tag{2.71}$$

Each element of the grid matrix m_k^{ij} holds the information about the probability that corresponding location on the map is occupied with a probability $p(m_k^{ij})$ ranging between $\{0,1\}$. The cell of the grid map is denoted by m_k^{ij} where $i = \{1, \ldots, N_i\}$ represents the number of grid rows N_i and $j = \{1, \ldots, N_j\}$ number of grid columns N_j. The map building step (2.64) can be formulated as an optimization problem that maximize the probability that particular cell is in fact occupied (or free) $\hat{\mathbf{m}}_{1:k}$ given the prior knowledge of the environment $\mathbf{m}_{1:k-1}$, vehicle poses $\mathbf{x}_{1:k}$ and sensor measurements $\mathbf{y}_{1:k}$. In other words, $\hat{\mathbf{m}}_{1:k}$ can be computed as the maximum a posteriori prediction of a function

$$\hat{\mathbf{m}}_{1:k} = \arg\max_{\mathbf{m}_{1:k}} p(\mathbf{m}_k|\mathbf{m}_{1:k-1}, \mathbf{x}_{1:k}, \mathbf{y}_{1:k}). \tag{2.72}$$

The probability that a single cell $p(m_{1:k}^{ij})$ is filled given the measurements $\mathbf{y}_{1:k}$ and poses $\mathbf{x}_{1:k}$ is computed by the Bayesian formula

$$p(m_{1:k}^{ij}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|m_{1:k}^{ij}, \mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})\; p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1})}. \tag{2.73}$$

The terms in above equation (2.73) are denoted as the map estimate $p(m_{1:k}^{ij}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k})$, the measurement likelihood $p(\mathbf{y}_k|m_{1:k}^{ij}, \mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$, the map prior $p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$ and the evidence or normalize factor $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{x}_{1:k})$. By assuming that current readings are independent of all previous states, e.g. by applying the 1^{st} order Markov assumption and by knowing the map prior, probability of map occupancy (2.73) can leads to

$$p(m_{1:k}^{ij}) = \frac{p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{y}_k)}{1 - p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{y}_k)} \, \frac{1 - p(m_0^{ij})}{p(m_0^{ij})} \, \frac{p(m_{1:k}^{ij}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}{1 - p(m_{1:k}^{ij}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}. \tag{2.74}$$

By applying logarithm to the occupancy grid computation (6.10) numerical errors from multiplying minuscule floating point numbers can be significantly reduced.

$$\log p(m_{1:k}^{ij}) = \log \frac{p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{y}_k)}{1 - p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{y}_k)} + \log \frac{1 - p(m_0^{ij})}{p(m_0^{ij})} + \log p(m_{1:k-1}^{ij}). \tag{2.75}$$

For this update rule, one needs only to specify $p(m_{1:k}^{ij}|\mathbf{x}_k, \mathbf{y}_k)$ e.g. the inverse sensor model, the initial map $p(m_0^{ij})$, and the prior occupancy probability of a given cell $p(m_{1:k-1}^{ij})$. The iterative Bayesian filter update for computing the new cell estimate $p(m_{1:k}^{ij})$ can be recovered from the log odds representation (2.75) by the following equation

$$p(m_k^{ij}) = 1 - \frac{1}{1 + \exp^{\log p(m_{1:k-1}^{ij})}}. \tag{2.76}$$

Please refer to the Sec. 6.3.1 for more information about the occupancy grid mapping. Alternative solution to the mapping is presented as the Dempster-Shafer Theory of Evidence discussed in detail in Sec. 2.4 and 6.3.3.

2.8 Hidden Markov Models

Inference in hidden Markov models (HMM) is in many ways similar to the Bayesian recursion, however there are fundamental differences which deserve to be highlighted [58]. Consider a simple HMM model expressed in discrete time k

$$\mathbf{x}_{k+1} = \mathbf{F}\,\mathbf{x}_k. \tag{2.77}$$

The basic HMM formulation assumes only one unknown discrete state $\boldsymbol{\xi} \in \{1, 2, \ldots, N_x\}$. The probability of a state $x_k \in \mathbf{x}_k$ being a Markov variable $\xi_k^j \in \boldsymbol{\xi}_k$ at time instance k is expressed as

$$x_k^j = p(\xi_k = j), \quad j = 1, 2, \ldots, N_x. \tag{2.78}$$

The posterior filter distribution is denoted as

$$\pi_{k|k}^j = p(\xi_k = j|\mathbf{y}_{1:k}), \quad j = 1, 2, \ldots, N_x. \tag{2.79}$$

The hidden Markov variable is at each time instant k observed as

$$\begin{aligned} p(y_k = i|\xi_k = j) =& H^{ij}, \quad i = 1, 2, \dots, N_x, \\ p(y_k = i) =& \sum_{j=1}^{N_x} H^{ij} p(\xi_k = j), \quad i = 1, 2, \dots, N_x, \\ =& \sum_{j=1}^{N_x} H^{ij} x_k^j, \quad i = 1, 2, \dots, N_x. \end{aligned} \tag{2.80}$$

Applying a time update (2.15) into formalism (2.80) results in

$$\begin{aligned} \pi^i_{k|k-1} =& \sum_j^{N_x} p(\xi_k = i|\xi_{k-1} = j)\; p(\xi_{k-1} = j|\mathbf{y}_{k-1}), \\ =& \sum_j^{N_x} \pi_{k-1|k-1}(j)\; F^{ij}. \end{aligned} \tag{2.81}$$

Similarly, applying (2.16) to formalism (2.80) gives

$$\begin{aligned} \pi^i_{k|k} =& \frac{p(\xi_k = i|\mathbf{y}_{k-1})\; p(\mathbf{y}_k|\xi_k = i)}{p(\mathbf{y}_k|\mathbf{y}_{k-1})}, \\ =& \frac{\pi^i_{k|k-1}\; H^{\mathbf{y}_k i}}{\sum_j^{N_x} \pi^j_{k|k-1}\; H^{\mathbf{y}_k j}}. \end{aligned} \tag{2.82}$$

The maximum a posteriori estimate (MAP) of $\boldsymbol{\xi}_k$ is defined as $\hat{\boldsymbol{\xi}}_k = \arg\max_i \pi^i_{k|k}$ and commonly solved by the Baum-Welch Alg. 4.

Algorithm 4: Baum-Welch Algorithm

Step 0: Initialization

Initialize $\boldsymbol{\theta}_{k|k-1} = [a^{ij}_{k|k-1}, b^{ij}_{k|k-1}, s^{i}_{k|0}]$ *with random initial conditions or with priors* $a^{ij}_{k|0}, b^{ij}_{k|0}, s^{i}_{k|0}$.

Step 1: Expectation Calculation (E-Step)

Let $\alpha^i_k = p(\mathbf{c}_k, s^i_k|\boldsymbol{\theta}_{k|k-1})$ *denotes the probability of seeing observations* $\mathbf{c}_k = \{c^j_1, c^j_2, \dots, c^j_K\}$ *while being in state* i *at time* k. *Then* α^i_k *can be found recursively by a forward procedure as shown below*

$$\alpha^i_1 = s^i_0 \; b^i_{k|0}(\mathbf{c}_1) = s^i_0 \; p(\mathbf{c}_1|s^i_0),$$

$$\alpha^i_k = b^i_{k|k-1}(\mathbf{c}_k) \textstyle\sum_{j=1}^{N_m} \alpha^j_{k-1} a^{ij}_{k|k-1} = p(\mathbf{c}_k|s^i_k) \textstyle\sum_{j=1}^{N_m} \alpha^j_{k-1} a^{ij}_{k|k-1}.$$

Let $\beta^j_{k-1} = p(\mathbf{c}_k|s^j_{k-1}, \boldsymbol{\theta}_{k|k-1})$ *denotes the probability of ending particular seqnece of observations* $\mathbf{c}_k = \{c^j_K, \dots, c^j_{k-1}\}$ *in state* s^j_{k-1} *while starting in state* i *at time* k. *Then* β^i_k *can be calculated recursively by a backward procedure as follows*

$$\beta^j_K = 1,$$

$$\beta^j_{k-1} = \textstyle\sum_{j=1}^{N_m} \beta^i_k a^{ji}_{k|k-1} b^i_{k|k-1}(\mathbf{c}_k) = \textstyle\sum_{j=1}^{N_m} \beta^i_k a^{ji}_{k|k-1} p(\mathbf{c}_k|s^i_k).$$

According to Bayes' theorem the probability of being in state i *at time* k, *i.e.* γ^i_k, *given the observation seqnece* $\mathbf{c}_k$ *and the parameters* $\boldsymbol{\theta}_{k|k-1}$ *reads as follows*

$$\gamma^i_k = p(s^i_k|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1}) = \frac{p(s^i_k, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}{p(\mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})} = \frac{\alpha^i_k \beta^i_k}{\sum_{i=1}^{N_m} \alpha^i_k \beta^i_k}$$

The probability of being in state i *and* j *at time* k *and* $k-1$ *respectively given the observed sequence* $\mathbf{c}_k$ *and the parameters* $\boldsymbol{\theta}_{k|k-1}$, *i.e.* $p(s^i_k, s^j_{k-1}|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1})$, *reads as follows*

$$\xi^{ji}_k = p(s^i_k, s^j_{k-1}|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1}) = \frac{p(s^i_k, s^j_{k-1}, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}{p(\mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})} = \frac{\alpha^j_{k-1} a^{ji}_{k|k-1} \beta^i_k b^i_{k|k+1}(\mathbf{c}_k)}{\sum_{i=1}^{N_m} \sum_{j=1}^{N_m} \alpha^j_{k-1} a^{ji}_{k|k-1} \beta^i_k b^i_{k|k+1}(\mathbf{c}_k)}$$

The denominators of γ^i_k *and* ξ^{ji}_k *are the same. Furthermore, they represent the probability of making the observations* $\mathbf{c}_k$ *given the parameters* $\boldsymbol{\theta}_{k|k-1}$.

Step 2: Maximization Calculation (M-Step)

State transition matrix a^{ij}_k *is computed as expected number of transitions from state* s^j_{k-1} *to state* s^i_k *compared to the expected total number of transitions away from state* s^j_{k-1}.

$$a^{ji}_k = \frac{\sum_{k=k-1}^{K} \xi^{ij}_k}{\sum_{k=k-1}^{K} \gamma^i_k}$$

Liklihood function β^i_k *represents the number of times observation sequence* $\mathbf{c}_k$ *have been equal to* $\mathbf{v}_k$ *while in state* s^i_k *over the expected total number of times being in state* s^i_k,

$$b^i_k(v_k) = \frac{\sum_{k=1}^{K} 1_{c_k = v_k} \gamma^i_k}{\sum_{k=1}^{K} \gamma^i_k}$$

where indicator function $1_{c_k = v_k}$ *is defined as*

$$1_{c_k = v_k} = \begin{cases} 1 & if \; c^i_k = v^i_k \\ 0 & otherwise. \end{cases}$$

3

A Framework for Dynamic Context Exploitation

While the benefits of exploiting Contextual Information (CI) are starting being recognized by the Information Fusion (IF) community, most current approaches for CI inclusion lead to stove-piped solutions that hardly scale or adapt to new input or situations. This paper makes a step in the direction of better CI exploitation by presenting some results of an international collaboration investigating the role of CI in IF and proposing an adaptive framework that dynamically takes into consideration CI to better support mission goals. In particular, we discuss some architecture concepts to be considered in the development of fusion systems including CI and we present how context can be dynamically exploited at different levels of a fusion engine. The concepts are illustrated in a maritime use-case.

3.1 Introduction

By surveying recent proceedings on contex in IF [134, 18], three important conclusions could be made about context information (CI) exploitation. First, domain knowledge is in a vast number of cases tailored for application driven solutions of limited scalability and adaptability. Second, CI is not given the same level of importance throughout the levels of fusion, which reduces system performance. Third, frameworks which will be able to capture the nature of the context regardless of the target application are merely not existent. Furthermore, it seems that nowadays context aware systems (CAS) do not consider the fact that context is typically of dynamic nature. That is, a context variable may be latent, but it could be discovered through an inference process or it could be dependent on the user's and target's goals [134].

Llinas in [96] surveyed available frameworks for IF over the last decade. Based on his findings one should consider: a) graphs/network methods for creating contextual relations between events and entities; b) Common Referencing (CR), Data Association (DA), and State Estimation (SE) as basic functionalities of all fusion nodes; c) a Resource Management module (RM) to be coupled to the fusion engine in order to promote adaptation. Frameworks are expected to accommodate hard and soft information as well. From these premises, arguably the greatest weakness of current frameworks lies in their inability to provide adaptive feedback and to dynamically control the fusion process.

Steinberg and Bowman envisioned adaptability issues in [141], by introducing the concept of adaptive context discovery and exploitation. Their proposal is to seek, discover, select and fuse CI, modeled as context variables, as a part of goal-driven decision process e.g. through problem variables. Engineering implications posed by adaptive context discovery and exploitation were addressed subsequently in [60], and [21] and led to the development of the Data Fusion and Resource Management (DF & RM) Dual Node (DNN) architecture [141]. DF & RM DNN allows any decision process to be completely characterized in terms of IF and RM processes. The architecture has proven to be particularly useful in the design and evaluation of large, complex decision systems. It is therefore particularly apparent the importance of adaptability in presence of CI that can be very transient depending on the current situation and target's and mission goals [134]. The weaknesses of current approaches are therefore stimulating the efforts for finding a truly adaptive CAS architecture in order to improve the performance of fusion processes.

In his work [96], Llinas goes beyond the survey and sets a stepping stone for further CAS development. The architecture he propose further develops ideas originated from Bowman and Steinberg [139], and from his own work [97], along with the already mentioned suggestions originated from the survey [96]. The design aspects of this architecture will be explained and expanded within the body of this paper. One way to extend the sub-structure of current concept could be, for example, achieved by Blackboard architectural concepts [146]. Blackboard systems are knowledge-based problem solvers that work through the collaboration of independent reasoning modules.

This paper presents some results of an international collaboration investigating the role of CI in IF and proposing an adaptive framework that dynamically takes into consideration CI to better support mission goals. Before introducing the architecture here proposed (Sec. 3.3), some terminology and fundamental concepts need to be recalled in the following section. A maritime use-case (Sec. 3.4) has been used to illustrate some functionalities of the designed architecture.

3.2 Fundamentals

It is a well known fact that functions which enable human-like reasoning are considered beneficial for the fusion, has a direct implication to context consideration. In this sense, Context could be said to "surround" an element of interest in the world [134] or, according to Day [34], context is "any information that can be used to characterize the situation of an entity". Finding relevant CI is not self-evident and often involves a complex integration of IF with planning, abductive logic and control functions. Contextual reasoning is therefore seen as an inference process, where desired information i.e. problem variables can be in some sense enhanced (e.g., reducing uncertainty, augmenting accuracy) by CI. As of now, no unified framework for designing such context aware system exist, but one might consider concepts for *a priori* and *a posteriori* CI exploration respectively as a good reference [96]. We provide in the rest of this section some definitions of the key concepts used in this paper.

3.2.1 Definitions

Context: Context is understood as information that surrounds an element of interest, whose knowledge may help understanding the (estimated) situation and also in reacting to that situation [134]. As pointed out by Steinberg and Rogova [142], context can be used in IF to:

- Refine ambiguous estimates,
- Explain observations,
- Constrain processing, whether in cueing or tipping-off or in managing fusion or management processes.

Architecture: Structure useful for creating solutions to a problem, which describes the parts composing a solution and how they are organized and related. Architectures can focus on different organizational aspects including physical/processes distribution and topology.
Framework: "A conceptual structure intended to serve as a support or guide for the building of something that expands the structure into something useful" [96]. A framework tailored for a specific domain (e.g., Information Fusion) may include specific components fitting a broad range of applications in that domain.
Middleware: Software layer placed on top of another component. It provides higher level, domain-specific functionalities that improve the usability of the base component by services, applications and libraries.
Fusion Node: Abstraction of a generic fusion process that can be thought as composed by four consecutive steps (Uncertainty Characterization, Common Referencing, Data Association and State Estimation). It defines an interface for exchanging information (input and output) and managing its internal state and configuration.
Problem Space Characterization: The description of a generic problem (e.g., tracking) as an observable set of variables that need to be known, and how they are related. With this information, an intelligent algorithm manager can select from a repository the best algorithms that solve a fusion problem.

It can be noticed that these definitions are very generic and may resemble human judgment to integrate context knowledge in evaluating situations. For this reason, the aim of this proposal is highlighting the separation between context inputs and information sources from an architectural approach, avoiding particular solutions where context representation and exploitation is dependent on the application.

3.2.2 Context in Information Fusion

Visentini et al. [155] observe two types of heterogeneity in a sensor fusion processes. The horizontal heterogeneity, in the sense that CI refers to same-level information produced or extracted by concurrent or cooperative sources which operate in the same environment. For instance, an object appearance can be described by features as color, shape, velocity or class, which are at the same abstraction level. The vertical heterogeneity, in the sense that CI considers that fusion can take place between different levels of information abstraction, from sensory data to features, from decisions to soft/hard data, including also high-level layers. In this case, the object of interest can be represented by multi-layered information with different degrees of refinement and detail. In this regard, Snidaro et al. [133] also introduce the concepts of context shifting and switching. The process by which the entirety of the domain knowledge acquired is fractured into pieces and assigned to the proper algorithms. CI could also have local scope and validity and is thus pertinent to the scenario at hand. The granularity of the scope of certain information can be more fine-grained and be applicable only to sub-areas of the observed environment.

Steinberg and Rogova [139], [142], [140], [141] favor to use concept of relations and relationships for assessment and practical exploitation of situations. Context in these rations could be assessed either from the inside-out i.e. “context-of” or from the outside-in i.e. “context-for”. “Context-of” is used when a certain expectations could be made based on situations on the other hand the term “context-for” is used when a relation to one or more reference or problem items or variables need to be solved. A data fusion problem may be stated in terms of a set of problem variables and a utility function on the accuracy of evaluating these variables. Such inference assumes a model of the dependencies between measurement variables and problem variables which could be conveniently modeled by a factor graphs. Authors suggested, that selection of relevant context should be based on a constraining ontology of context variables and their relations with problem variables. Selection of contextual variable assumes that context of is know. This might not be the case for all applications and therefore this context needs to be discovered trough reasoning. Processes of relevance and reasoning are throughout the article refereed to dynamic context exploitation and are considered to be the main logic of presented middleware.

3.2.3 Context Adaptive Architectures

Ideas introduced by Llinas in [96] and Gomez-Romero et al. [49] established the basis for context-aware architectural designs. In their work, CI can be fully static or dynamic, possibly changing along the same timeline as the situation. Furthermore, authors argue that full characterization and specification of CI may not be able to be known at system/algorithm design time. Therefore, an “a priori” framework, that attempts to account for the effects on situational estimation of that CI that is known at design time, was introduced. Llinas et. al. also consider that CI may, like observational data, have errors and inconsistencies itself. Accommodation of these errors in data fusion processes leads to development of hybrid algorithms for “a posteriori” context exploitation. “A posteriori” in comparison to “a priori” includes checks of the consistency for a current situational hypothesis with the newly discovered CI. Both architectures assume the existence of a “middleware” layer which will be not only able to sample CI data and shape it into a suitable form for fusion processes, but also discover new CI. Our vision on how to realize such a middleware is presented in the next sections.

3.3 Architecture Design

This section describes an architecture to integrate context sources in Information Fusion (IF) processes in a general way, so that any fusion system in which contextual knowledge is available can be developed following this architecture. The approach does not make assumptions or puts restrictions about specific fusion processes or information and context sources, but it will be defined at an abstract level, so that specific algorithms and applications can be developed based on the architecture. In the first place, the types of context sources are commented, and a general mechanism to access context from fusion processes is discussed, following a middleware paradigm. Then, the adaptive IF framework is explained. The key idea is the exploitation of context knowledge to adapt the IF processes in order to optimize their performance.

3.3.1 Context Sources

A fusion system may access a number of different sources of contextual knowledge depending on the specific domain. In many applications, it is available in static repositories such as maps, GIS databases, representations of roads, channels, bridges, etc.; in other cases, context comes through dynamic data, such as meteorological conditions. In this case, we talk about context variables, implying the need of context access and update processes running in parallel with the core fusion processes. Finally, sometimes CI cannot be observed directly, and only indirectly deduced from other sources (inferred context). In addition, we can distinguish physical and logical context. In the first case, we will have physical descriptions (like GIS files) or variables (like meteorological phenomena) which are measurable objectively. In the case of logical knowledge (such as entities engaged in a coordinated trajectory, traffic regulations, mission goals, etc.), context can come from knowledge, human reports, learned from data or result from indirect inference processes based on other pieces of information. This characterization of context sources is illustrated in Fig. 3.1. Therefore, contextual sources can classified in terms of the nature and way of accessing available information:

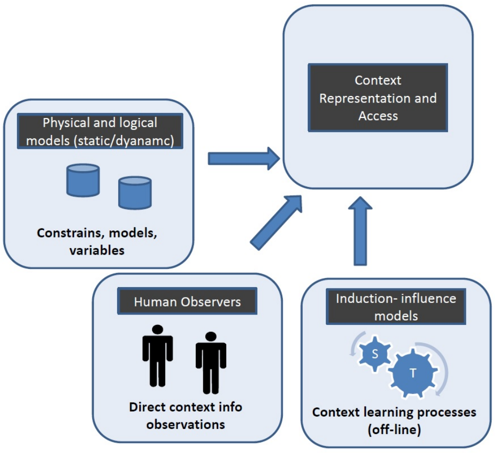

Context Representation and Access
Physical and logical models (static/dyanamc)
Constrains, models, variables
Human Observers
Direct context info observations
Induction- influence models
S
T
Context learning processes (off-line)

Figure 3.1: Context source types [135].

- Physical and logical structures:
  - Static datasets with information: roads, GIS databases, terrain characterization (navigation), urban environment, procedural information, normative, etc. In the maritime case, navigation routes or stationary areas are examples of context data sets, and some times they can be learned from historical data, as in the case of patterns of life reflecting the real behavior of entities of interest.
  - Contextual variables such as physical fields: weather, wind, sea state, clouds, etc. These variables are distribution of magnitudes, changing in space and time.
- Observed relations. Dynamic reports, human messages, and other documents represent the explicit input to the fusion process about situation (normal, labor day, anomaly, emergency, etc.), time of the day or week (working, meeting, etc.). These variables usually take discrete values indicating different contexts, coming from direct observation. The instantiated relationships are input to the system as context in some way, such as a human observation directly input to system. In the maritime case the geopolitical situation can be an example of dynamic observed relation.
- Inferred relations: Context can be deducted as dynamic relationships. A possibility is employing an automatic inference process, which may lead to the idea of a parallel representation of context process with its own processes and sources available.

3.3.2 Middleware

A way to systematically address advanced and generic context-based IF design deals with a context access and management system, in charge of providing useful context information about the entities as a transversal independent module. As mentioned, context services supporting fusion processes can be very heterogeneous, including, for example, access to reference databases, meteorological information, image repositories, GIS systems, texts, Internet, etc. Accessing such heterogeneous information represents a challenge. The middleware approach can alleviate this problem by placing a component between context data and its consumers. This solution is a popular choice in context-aware computing applications, as analyzed in the survey [75]. IF processes access contextual resources through the interface exposed by the middleware. So, the context middleware acts as a transversal independent module in charge of deciding which context information is relevant, as illustrated in the Fig. 3.2 below. The basic mechanism follows a query-response process: the middleware returns the selected relevant context information from the available sources, according to the values inferred and hypotheses raised by fusion processes. Two basic elements can be identified in both sides:

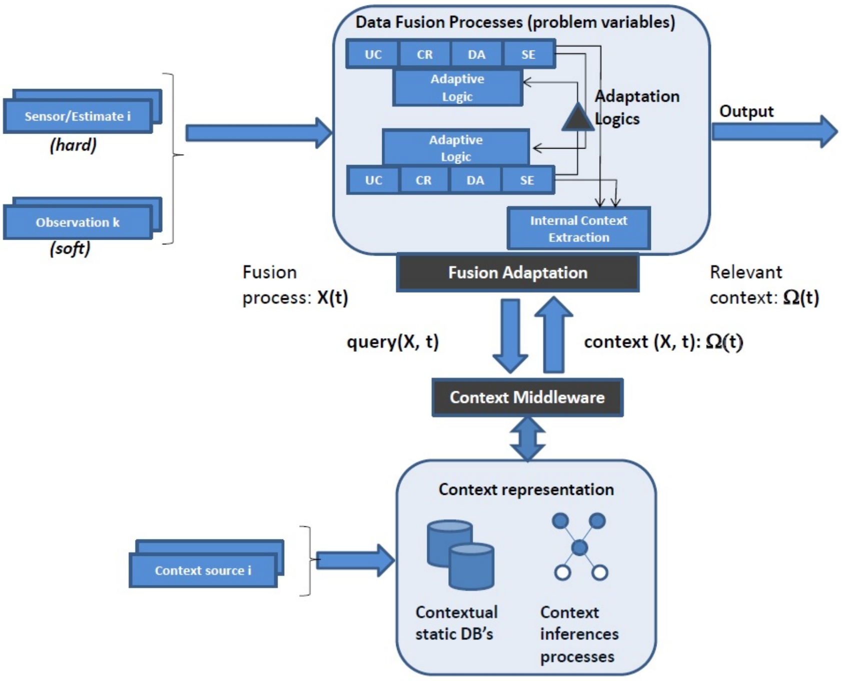


Figure 3.2: Context middleware mechanism [135].

- At the context side, the middleware is responsible for collecting, updating and making context knowledge usable by fusion processes.
- At the fusion side, the fusion adaptation logic uses the contextual inputs, so all processes and modules need to be described in terms of context input and interconnections to apply the adaptation.

In Fig. 3.2, and also in the architecture presented in the next subsection, all IF processes are abstracted as nodes consisting in four main basic functions applied to the data. This abstraction is taken from [96], but including Uncertainty Characterization as part of the fusion process. In general, any fusion node accepts either sensor data from some input source or an estimate (fused or otherwise formed) from some prior processing node. In this characterization, processing operations involve four basic functions:

- Uncertainty Characterization (UC): uncertainty associated to the information provided by the source, exploiting available models and related information.
- Common Referencing (CR): normalization operations, such as coordinate or units transformations, to align data from information sources to be fused.

- Data Association (DA): the multiple inputs (estimates or measurements) are examined in order to determine which (hypothetical) entity that the system believes to exist they are associated to or come from.
- State Estimation (SE): computation of attributes (e.g., kinematic properties, classification attributes such as color, identity, inferred relationships, etc), exploiting the associated data together with prediction models in estimation/inference processes.

Context middleware is responsible for providing usable context:

- Relevance: search for relevant pieces of context;
- It must provide up-to-date context. This means that it must integrate on-line information appropriate and potentially useful for the fusion processes;
- Granularity: it implies adaptation to the needs of fusion algorithm. For instance, in the access to wind representation, it can be 2D but needed 3D. Some aggregation or interpolation may be required to adapt the scales at both sides;
- Characterize the uncertainty in the contextual information provided, considering both the intrinsic uncertainty in contextual information and that propagated due to uncertain in query (for instance uncertainty in the location to index spatial context).

The operations to be done by the context middleware services are indicated below:

- Regarding search of applicable context to the fusion query:
  - Search of context relevant to the situation: physical (roads, bridges, channels, etc.), operational rules, etc.
  - Compatibility: validate the collected information as appropriate for query and check its compatibility (map, number of objects, etc.). In some cases, con- text maybe is not applicable (off-road, operational rules not met, etc.)
- Regarding transformation and normalization in the context response:
  - Context correlation and alignment with fusion process. This is especially relevant for use of real- time dynamic contextual sources, i.e. meteorological services;
  - Spatial alignment: fundamental for efficiency: search with appropriate representation and algorithms (maps, GIS, roads, etc.);
  - Time alignment (prediction functions): necessary when context is dynamic: simple temporal indexing, extrapolation models, etc.
  - Time alignment (prediction functions): necessary when context is dynamic: simple temporal indexing, extrapolation models, etc.

With respect to context relevance, as commented in [140] and [141], a big challenge is determining the selection of context variables. In general, such selection should be based on previous knowledge of relations among context variables and problem variables. A possibility could be the development of an ontology based on relevancy of contextual variables to problem variables and their consistency. A context variable can be called relevant to a set of problem variables defining the reference items and relations between them, if the values of these problem variables change with the value of the context variable under consideration. Another criterion for determining a particular context as relevant may be the increase in information as the result of utilizing that context variable for estimation and/or inference. Finally, the problem of selecting context variables is more complex since relevance is often time- variable. Situations of interest are often dynamic, such that the availability of any sought data may also be time-variable. Even the mission-driven information needs and fusion processes can be also dynamic, making the utility of information given by context pieces also time-variable. Therefore, the middleware is presented as an approach to generalize the context access and exploitation by fusion processes, organized as a set of operations done over the information available in different sources. The context middleware manager is responsible for searching and providing the relevant and updated information in the expected format and scale, considering the needs and requirements of the fusion node, so that fusion operations can take into account the context, independently of the specific strategy adopted. The service-oriented architecture is the key to develop a general perspective in the design and avoid particular solutions depending on the specific types and nature of the contextual sources available.

3.3.3 Architecture

The adaptive fusion architecture presented in this section is depicted in Fig. 3.3, as an extension of [96]. Raw input data, covering both hard (electronic, physics-based) sensors and soft (human observers) sources, undergo detection, semantic labeling, and flow control composite functions. Once the best qualified detections have been achieved, there is the question of assigning them to the various Fusion Nodes to be processed and generate the desired outputs. The key to keep interaction with the contextual sources, through the middleware interface presented in previous subsection, is a function module called Problem Space Characterization below the detection operations. To adaptively manage a system with a library of alternative algorithms that address a generically-common problem space (e.g., object tracking problems), knowledge of the performance bounds of any library algorithm in terms of an observable set of parameters needs to be known. With such knowledge, an intelligent algorithm manager (part of the InterNodal Adaptive Logic) can terminate and invoke the best algorithm for the current problem-space condition. An important point here is that the problems-space complexity parameters need to be observable by the system sensor set. Besides, we may distinguish a static configuration logic, describing all problem-space variable and inter-relations, and the possibility of dynamic adaptation. A typical example is the set of categories in a classification problem, which may change dynamically accordingly to the operative conditions or available context. This knowledge may also be contextually-dependent, so we have CI also feeding this knowledge base and control logic. The context middleware presented in previous subsection is in charge of providing the appropriate context pieces accordingly to the fusion variables state. This context is delivered by the adaptation manager to the different adaptive processes defined along the architecture, including the specific processes at the sources, the functions composing each individual fusion process (IN Adaptive logic boxes) and the inter-level processes, depending on the type of solution developed.

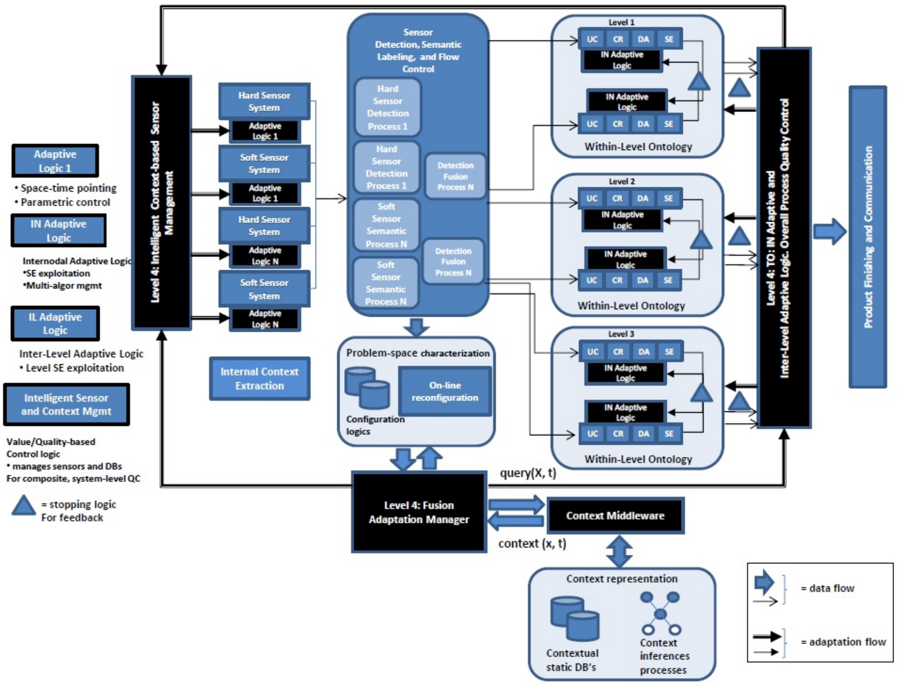

Adaptive Logic 1
• Space-time pointing
• Parametric control
IN Adaptive Logic
Internodal Adaptive Logic
•SE exploitation
•Multi-algor mgmt
IL Adaptive Logic
Inter-Level Adaptive Logic
• Level SE exploitation
Intelligent Sensor and Context Mgmt
Value/Quality-based Control logic
• manages sensors and DBs
For composite, system-level QC
= stopping logic For feedback
Level 4: Intelligent Context-based Sensor Management
Hard Sensor System
Adaptive Logic 1
Soft Sensor System
Adaptive Logic 1
Hard Sensor System
Adaptive Logic N
Soft Sensor System
Adaptive Logic N
Internal Context Extraction
Sensor Detection, Semantic Labeling, and Flow Control
Hard Sensor Detection Process 1
Hard Sensor Detection Process 1
Detection Fusion Process N
Soft Sensor Semantic Process N
Detection Fusion Process N
Soft Sensor Semantic Process N
Problem-space characterization
Configuration logics
On-line reconfiguration
Level 4: Fusion Adaptation Manager
query(X, t)
context (x, t)
Context Middleware
Context representation
Contextual static DB's
Context inferences processes
Level 1
Level 2
Level 3
UC
CR
DA
SE
IN Adaptive Logic
Within-Level Ontology
Level 4: TO: IN Adaptive and Inter-Level Adaptive Logic. Overall Process Quality Control
Product Finishing and Communication
= data flow
= adaptation flow

Figure 3.3: An adaptive information fusion framework [135].

By definition, all the adaptation processes (highlighted in black in Fig. 3.3) are part of JDL Level 4, which is one of the basic goals of the architecture: exploiting the context in order to refine and adapt the different fusion processes (including data sources). Feedback as adaptation is a fundamental aspect: the framework should show adaptive behavior such as inter- nodal feedback to allow (or perhaps require) that the Nodes share and exploit information if possible. One can see this in traditional Level 1 fusion for tracking and identification usually done in two separate Fusion Nodes; kinematics are of course helpful for identification, and identification is helpful for example to know an objects feasible dynamic motion. In turn, an adaptive Inter-Level feedback process is also shown, allowing situational estimates to feedback their estimates to other levels; an example of this would be a situational estimate that would suggest that maneuvering behaviors could be expected, informing Level 1 object tracking logic to open the tracking gates and capture the diverging measurements occurring upon the maneuver, i.e., as a maneuver-anticipation strategy instead of the (generally too-late) post-maneuver detection strategies often employed in tracking systems. As already mentioned in [141], all control loops need to define stopping criteria that terminate the otherwise-endless looping; that requirement is shown by triangles in the Fig. 3.3.

3.4 Case Study

In this section, an example of instantiation of the architecture, discussed in the previous section, is provided within a maritime use case. This use case is part of a selection of other use cases developed at CMRE to emphasize maritime security challenges and facilitate the collaboration and integration of different communities [69]. We identify the elements of context possibly considered, driven by the user's needs to take the decision.

3.4.1 Contextual Information

Significant portions of the world population live in coastal areas, and many large cities directly border the water. The maritime environment is complex, directly connecting the world via its waterways, with relatively limited regulation and a mixture of traffic ranging from large container vessels to smaller fishing boats and pleasure craft. Coastal areas are vulnerable to threats arriving from the maritime environment, as was seen in the Mumbai hotel bombings in 2008. Civil authorities are responsible for monitoring harbor areas and protecting ports and critical infrastructure from threats arriving via maritime routes. Generally, some form of surveillance will be in place for major port areas, with any suspicious activity monitored, according to current threat levels and typical types of activity in the port. In heightened levels of threat, all unauthorized vessels approaching the port would be detected and monitored, with an assessment made of its behavior and intent assessed in order to allow early intervention if required. Intelligent systems making of use data and information fusion technologies are certainly an asset for harbor protection (e.g., [126], [46], [51]) and as an example, the fusion architecture (Fig. 3.3) is instantiated within the following use case.

The scenario takes place in a port loosely based on the port of La Spezia, Italy, due to the variety and complexity of its activities. Some physical contextual information directly related to harbor zones characteristics is available such as water depths, channels, restricted areas, fishing areas, borders, harbors (fishing, recreational, etc), shipping lanes, ferry lane, military and LNG (Liquid Natural Gas) anchorage areas. A fair degree of Pattern of Life (PoL) is known about the area from experience and automated traffic pattern extraction routines [111]. There is significant fishing in the area and fishing vessels behavior and fishing areas is generally understood. There are also several regular smaller passenger vessels for local tourism and private yachts and small boats. Other large vessels' including cargo vessels, tankers, and cruise ships operate normally in the area. Large passenger ships are required to report their Estimated Time Of Arrival and AIS information to the port authorities but smaller vessels do not have formal reporting requirements.

In this scenario, it is peacetime, there is no specific terrorist threat, but we are still in a post 9/11 security environment with a risk of potential malicious acts, from a variety of motivations. There is also an increased resentment after a recent wave of illegal immigration caused by political and economical instability of neighborhood regions. Thus, the geopolitical context is relatively quiet and the Harbour Protection Level (HPL) is set to ONE over a scale of three levels. For the environmental context, the meteorological conditions are clam (the weather is clear, sunny, there is no fog, the sea state is at the lowest level) within the port.

3.4.2 Response Event

The use case presented here is a civil harbor protection response where the national authorities have just alerted the local authorities of a possible recent or imminent Improvised Explosive Device (IED) drop within the port [69]. After the notification, the local security coordinator executes the pre-planned response to confirm or disconfirm the credibility of the threat, including actions such as: (1) Elevate HPL to level TWO, (2) Increase local security measures (e.g., divert traffic and classify all real-time small vessel traffic), (3) Notify the investigation team, (4) Request for the deployment of Autonomous Underwater Vehicles (UUVs), to check the sea bed within the port and clear the area. The investigation team will conduct historical analysis of the electronic media and data (radar, SAR imagery, video from a Pan-Tilt-Zoom (PTZ) camera, phone traffic, AIS messages exchange, twitter, etc) of the last hours, interview local witnesses (e.g., harbor pilots, local fishermen, etc), looking for any suspicious or abnormal event missed during routine surveillance. An event of interest may have been missed because of the surveillance team was unaware of the threat at that time. The UUVs will adapt their search path based on any finding of the investigation team (e.g., localization of a suspicious activity).

The user context is defined by the user's needs to take his decision. Based on the information provided by the investigation team together with the UUVs team, the local security coordinator will decide whether the threat is real or not and then, to step up the level of security or to return to normal security posture respectively [136]. The local security coordinator evaluates the risk regarding the probability of the threat (was it a hoax or not), the vulnerabilities of the port (e.g., the LNG terminal, ferries, container terminal, etc) and the consequences of the event (e.g., loss of life, economical). Based on some prior intelligence information, the evaluation of the threat by the investigation team first focuses on small vessels (fishing boat, pleasure craft, etc). Immediately, real time small vessel traffic is to be classified by type. Further, among other aspects of the investigation, the captured data from the previous 24 hours will be reviewed and revisited in the light of the new threat declaration to possibly detect any suspect behavior from small vessels.

3.4.3 Instantiated Fusion Architecture

Table (Tab.: 3.1) provides exemplar tasks to support the local security coordinator across the different levels of the JDL model [17]. In rows, are listed the JDL levels 1 to 3 (level 0 is not considered here) while the four main fusion functions of Uncertainty Characterization, Common Referencing, Association and State Estimation and Prediction (Sec. 3.3) are listed as columns. To emphasize that the refinement process (level 4) applies at each level, it is added as a last column. Problem variables (observational, decisional and contextual) are also mentioned for each level. Let us denote by x a vector of measurements (or observations) about different attributes (e.g., position, speed, heading, length, type) provided by several sources such as the coastal radar, the PTZ camera, the SAR imagery or AIS if available. Let us also denote by $\mathcal{X}_A^s$ the domain of the variable associated with attribute A for a given source, distinguishing between possible different domains across the different sources.

	Variables			Fusion Node Functions				
	Observation	Decision	Context	UC	CR	DA	SA	Level 4 Process Refine.
Level 1 Object Assessment	*vessel_length* *vessel_width*	*vessel_behavior*	$\mathcal{X}_T^{User1}$	track split and merge	$\mathcal{X}_T^{SAR} \leftrightarrow \mathcal{X}_T^{Rad}$	$x \mapsto Track$	$x \mapsto \hat{T} \in \mathcal{X}_T^{User1}$	$\mathcal{X}_T^{User1} \mapsto \mathcal{X}_T^{User2}$
Level 2 Situation Assessment	*vessel_speed* *vessel type* *vessel length*	*vessel_type*	*route_set* *desig._areas* *sea_state*	route extraction	Grid align. SAR, AIS, RADAR	$x \mapsto V$	$V \mapsto \{normal\}$ $\{abnormal\}$	Adjustment of camera parameters
Level 3 Impact Assessment	*vessel_behav.* *vessel_flag*	*vessel_identity*	HPL	threat statistics and costs	$x \mapsto$ standard categories	$x \mapsto V$	$V \mapsto ID \in$ standard categories	Detailed intervention plan

Table 3.1: Example of fusion node functions across the JDL levels for the use case [135].

Level 1: *The State Estimation* $\hat{T}$ of the type of the vessel corresponding to a suspicious track (i.e., a small vessel) is performed, based on the vector of measurements x. As the type is a perennial property, no prediction is required. The Association assigns any new declaration or measurement from the sources to the suspicious track. The different sources report over different attributes (e.g., the *vessel_width* and/or *vessel_length* for the SAR analyst or for the camera analyst, the *vessel_length* for the radar) and over different domains $\mathcal{X}_T^s$: Fishing vessels vs cargo vessels vs tankers vs service ships for the SAR imagery analyst, specific types of fishing vessels for the camera analyst). The *Common Referencing* at this level aligns for instance (but not only) the different type scales to a common one, as being suggested by the user context driven by his mission goal. The *Uncertainty Characterization* identifies some uncertainty origins such as the source's reliability, or the measurements' likelihoods and transforms the uncertainty into a suitable mathematical model of a dedicated mathematical framework [69].

Level 2: The behavior analysis of each detected small vessel aims at detecting any behavior such as "Speed too high for the type of vessel", "Fishing pattern while not in a fishing area", "Loitering in the port area", "Rendezvous". The anomaly detection task can rely on several State Estimations for a further global State Estimation (e.g., Normal vs Abnormal). Anomaly detection essentially compares the estimated attributes at level 1 (*vessel_speed*, *vessel_heading*, *vessel_type*) to expected ones as represented by predefined patterns of life of routes or dedicated areas [111]. The *Common Referencing* aligns the spatial scales of the different sources (AIS, RADAR, SAR), regarding the *vessel_position*. In addition to the *Uncertainty Characterization* of contextual information (routes) in routes' representation (contextual knowledge), the UC at this level is essentially similar to UC at level 1 and some likelihood functions may be elicited from past AIS records. However, other dimensions such as the uncertainty derivation (objective vs subjective), may be characterized as well for a better interpretation of uncertainty representation by the user. The Association identifies any piece of information contributing to the task and being possibly related to the vessel's behavior. For instance, an phone or radio call associated to the vessel may be used.

Level 3: In case of the detection of an abnormal behavior, the impact is assessed involving some risk analysis elements such as the cost of (relevant vs non-relevant) intervention need to be considered. The *State Estimation and Prediction* is the classification of the vessel as the threat (i.e., the one dropping the IED) which considers both its behavior and static information. The *Association* ensures that all ID statements from concern indeed the suspect vessel. The *Uncertainty Characterization* includes some aspects of threat assessment (probability of abnormal behavior) from Level 2 but also the assessment of the vulnerability and cost of critical assets in the area, for a further risk assessment at the *Prediction* task. The *Common Referencing* aligns the identification statements of the different sources to the standard categories applied by the local Harbor Protection team.

Level 4: The refinement step influences each of the three above JDL levels, to adapt to some contextual change:

- Level 1: The classification is refined based on new user's needs: At a first instance, the local security coordinator was interested in distinguishing between small and large vessels as represented by $\mathcal{X}_T^{User1}$. A finer assessment was then required to discriminate between different types of small fishing vessels and pleasure craft, as represented by $\mathcal{X}_T^{User2}$.

- Level 2 the anomaly detectors' performance is directly impacted by the speed estimation. An updated meteorological information requires to adjust sensors' parameters for an updated assessment of *vessel_speed* and an improved anomaly detection.

- Level 3 the path planning of the UUVs may be adapted and modified on the fly based on the past location of a suspect vessel.

3.4.4 Dynamic Parameter Adaptation

The system can exploit contextual information for adapting the sensor parameters. A possible way of performing the dynamic parameter adaptation is to establish a relationship between the context variables and the parameters of the sensors. Given the context variables in (Tab.: 3.1), they can be represented as quadruple $< T, r, a, l >$, where T is the vessel type, r the expected route, a the designed area, and l the HPL. A set of different context instances can be obtained by combining their values: $< T, r, a, l > \rightarrow \{C_1, \ldots, C_n\}$. For example, in case of $<$ ferryboat, toSlickville, ferrylane, TWO $>$, the associated context C_i can be labeled as "ferry boat of 5 PM". Given a particular context, a set of parameters for the sensors can be established though a relationship $C_i \rightarrow \{p_1, \ldots, p_n\}$, where pi is a single parameter of a sensor in the system. In the case of the ferry boat, the position of the PTZ camera can be set to point on the ferry lane, with a zoom level adequate to the estimated distance of the boat from the camera site. As another example, in the case of a possible threat coming from a small boat, radar parameters can be changed to be more sensitive for the detection and tracking of small vessels. The SAR imagery parameters can be adapted by estimating the speed of the suspect vessel. In the same way, the parameters of the fusion nodes can be updated. For example, if the context variable tuple contains fishing area as designed area, the parameters of the vessel route analysis process can be set to account for non-linear trajectories, since fishing vessels are expected to perform circular trajectories. This example illustrates the potential use of context information to adapt fusion processes in a maritime scenario. However, in order to implement the framework some steps are needed to obtain a full functionality. First, the context middleware should access the available sources, represented in a convenient way in order to provide the relevant and updated context. Second, it is necessary to develop appropriate interfaces to access the real fusion processes and adapt their parameters based on available context inputs, and manage the adaptation flows from the context middleware to each data processing node.

3.5 Conclusions

This work addresses several concepts and issues to be taken into account in developing a context-aware fusion system. We have discussed an architecture for dynamically exploiting context at different levels in a fusion engine. The solution adopts a middleware approach which provides a convenient way of designing an interface level between data/information sources and the fusion functions, brokering all relevant contextual data sources to the correct data sinks. The concept has been applied to a port protection use-case and will be further developed as part as an international collaboration investigating the role of CI in fusion systems.

4

Encoding Context Likelihood Functions as Classifiers in Particle Filters for Target Tracking

While several attempts have already been made in exploiting contextual information (CI) for target tracking, the vast majority of the approaches focuses only on one specific type of CI (e.g. terrain information). In this chapter, we address the problem of multi-level context representation and exploitation for target tracking. Specifically, we present an approach for encoding different types of contextual information (CI) as likelihood functions via classifiers in particle filters. The proposed solution is sufficiently versatile as to be able to couch different types of CI. Promising results have been obtained from our simulations on synthetic data.

4.1 Introduction

Context-aided object tracking has received significant interest recently [18]. In information fusion (IF), context is understood as information (context variables) that surrounds an object or situation of interest, whose knowledge allows refined understanding of the problem variables. It is information that aids in understanding the (estimated) situation and also aids in reacting to the situation, if a reaction is required [134]. As pointed out by Steinberg and Rogova [142], context can be used in IF to refine ambiguous estimates, explain observations or constrain processing. However, most of the existing approaches on context-aware tracking propose ad-hoc solutions for encoding some specific type of contextual information (CI) (e.g. terrain conditions that can affect movement in ground target tracking). Since the movement and behavior of tracked entities can be subject to several external influences, in this paper we propose a principled solution to encode different types of contextual knowledge in the form of likelihood masks [155] within the formalism of particle filters. The idea is to represent different forms of contextual information, possibly ranging from sensor-related to high-level domain expert knowledge, as conditioning factors in the

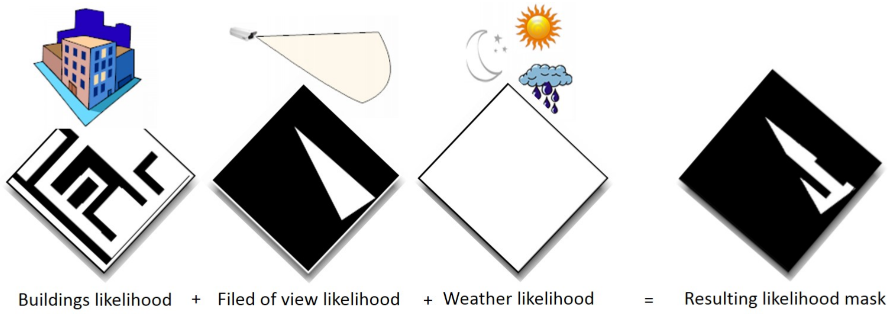


Figure 4.1: Example of likelihood masks for different context [155].

process of target tracking. Fig. 4.1 provides an intuitive depiction of the underlying concept. Starting from the left, the picture shows the mask related to the presence of buildings and structures that influence the observation capability of the sensor and constrain a target's location on the roads. The next mask represents the field of view of the sensor, excluding the possibility of a detection outside its range. The weather mask is completely white meaning that weather is not likely to be playing any influence for the time being. The fusion of the three masks yields the one shown on the right and its generation will be discussed in Sec. 4.3.3. The masks here illustrated are merely exemplary and others could be developed depending on the application. Moreover, all contextual masks are in principle dynamic and could be updated as soon as new information is available. As we show in Sec. 4.3.1, context integration can be conveniently couched in the formalism of constrained Bayesian filtering, while likelihood masks can be straightforwardly applied in the measurement update of the filter (Sec. 4.3.2). We evaluate the proposed concept through simulations performed on a synthetic dataset (Sec. 4.4.2) and discuss the results detailing where and how context could aid the tracking.

4.1.1 Overview of the Framework

Here, we assume context to be implemented as a constraining factor in the form of a likelihood function or likelihood mask. An overview of the algorithms taking into account constraints in the estimation process can be found in [127]. Based on the survey, a distinction can be made between available techniques for linear and non-linear dynamic systems utilizing either linear or non-linear constraints. Knowledge of soft or hard constraints could be applied in Bayesian recursion either in the prior density function or in the likelihood [112]. While both approaches are equivalent from a Bayesian perspective [104], in our case the inclusion of sensor-related CI is more straightforward in the update step. Our choice to employ a particle filter (PF) to drive the tracking process is motivated by the nature of likelihood masks. As shown in Fig. 4.1, likelihood masks are layers of contextual knowledge regarding the tracking area. One could consider any information, related to the sensor measurements, as a plausible likelihood mask. The main challenge here is to find a representation for these masks that will encode the different contextual information and successfully utilize this knowledge in the tracking process.

The process of likelihood mask generation starts with the clustering of contextual knowledge. Any existing classifier could be used in this step, however most algorithms are designed with certain assumptions and favor some type of biases [161]. Each cluster has assigned a label which carries the contextual knowledge as a normalized weighting factor projected over the tracking area. We will demonstrate the process of likelihood mask generation in a road network example as shown in the Fig. 4.2 below. The idea

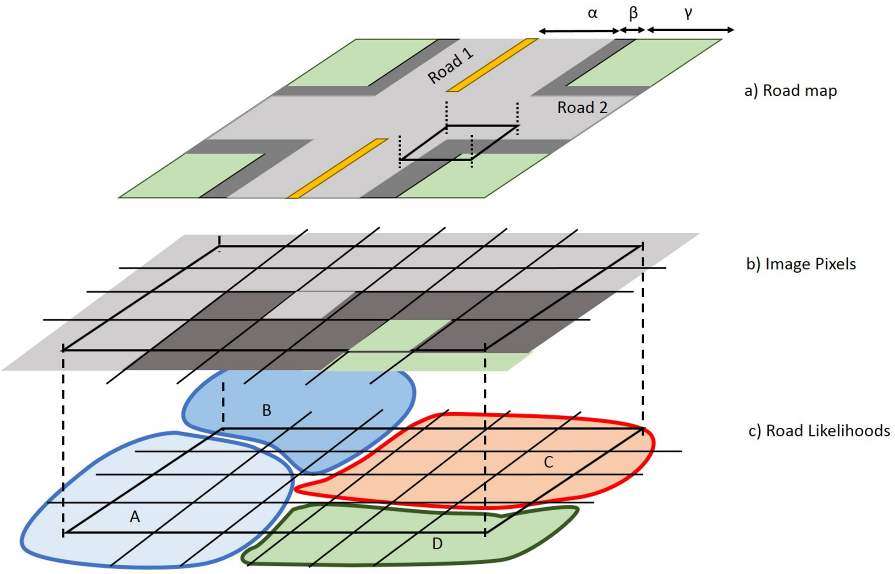


Figure 4.2: Illustration of likelihood mask generation for road constraints.

is to utilize the knowledge of traffic routines (position of detected targets) and aerial images in order to extract a topologically correct road network. In a recent study [106], authors argue that topological completeness of the road network is more important than pixel-accurate segmentation. Therefore, authors pose road extraction as a pixel-wise labeling task with two classes "road" and "background", while exploiting a local context and long-range structure dependencies between clusters. Local context is learned directly from data by training a classifier on rich appearance features. The pixel-wise road scores are formulated as candidate likelihood functions of the position and road width. The hierarchical graphs refine the section of candidate likelihoods such that they best explain the image evidence i.e. they optimally cover the road network. Without loss of generality we assume that road images (Fig. 4.2a) consist of three distinct areas: roadway α, sidewalk β and off-road γ. In reality, road extraction especially in urban environments is challenged by varying road appearance, occlusions as well as heterogeneous background. Aforementioned effects are, for the purpose of this study, treated as a time varying bias and random noise in map pixels (Fig. 4.2b). We further assume that classifier results in a road likelihood mask which is optimal for a given road topology (Fig. 4.2c). That the candidate likelihoods A and B are in fact a consecutive segments of a $Road1$ with appropriate width α. The likelihoods C and D are segments of a $Road2$ and $off-road$, respectively. Resulting road likelihood mask is a sum of all road likelihood segments.

In summary, we can generate likelihood masks for any sensor in two steps. At first, the source of CI should be aligned w.r.t. the global frame so that context features could be extracted. Here, the functional relation between measurement and contextual space needs to be defined. Secondly, the contextual space needs to be clustered and labeled from the dataset. Labels, projected over the tracking area, carry the contextual knowledge in form of weights or constraints. Likelihood masks are functions of the sensor state vector conditioned on the label space. Combining likelihood masks in Fig. 4.1 translates to the process of merging of probability density functions (PDF), i.e. likelihood mask mixtures, over the target space [155].

The resulting multivariate mixture model could be highly nonlinear and therefore we have chosen to approximate the posterior state distribution with the Monte Carlo approaches, i.e. particle filter. Since we assumed, that the likelihood mask is aligned with the target reference frame, state vector particles could be conveniently used for sampling both measurement and contextual likelihood within a single filter step (Sec. 4.3.2). By adding the contextual likelihood in the measurement update step we constrain the spread and weight of the particles. Thus, allowing the filter to account for and react upon the situations that the object could encounter during tracking. As shown in Fig. 4.3, the shape of the likelihood could change significantly and as such dramatically influence the performance of tracking. In Fig. 4.3a particles are spread alongside the road (indicated by the constrained likelihood ellipsoid), thus encouraging the filter to expect new measurements of the target from the longitudinal direction. On the other hand, in case of an intersection (Fig. 4.3b), it is desired that spread of particles will be larger w.r.t constrain and unconstrained scenario to account for possible change of a direction. Here, likelihood mask increases the ambiguity of measurements. This could result in decreased accuracy of the estimate, but on the contrary the filter

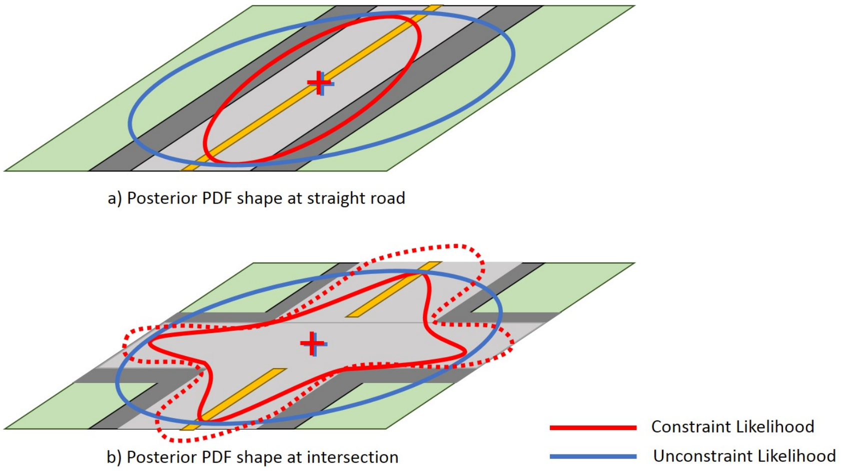

a) Posterior PDF shape at straight road
b) Posterior PDF shape at intersection
Constraint Likelihood
Unconstraint Likelihood

Figure 4.3: Illustration of prior PDF shape affected by a road constrains.

becomes more robust towards sudden trajectory changes and could prevent track losses. In summary, we present an approach for encoding different types of contextual information (CI) as likelihood functions via classifiers in particle filters.

4.1.2 Main Contributions

The contribution of this work are threefold:

- Particular object tracking scenario under inclusion of context was evaluated.
- Methodology for generation of contextual likelihood masks was introduced.
- A framework for object tracking utilizing likelihood masks was proposed.

4.1.3 Relations to the Context Exploitation Framework

Likelihood masks (Fig. 4.1) are functions of contextual observations conditioned on the state vector expressed as the mixtures of Gaussian distributions (4.16). As outlined in Sec. 4.1.1, likelihood masks are generated in the process of clustering preceded by the context space alignment (Fig. 4.2). By assuming that the FN (Fig. 3.2) is tasked to track the objects according to the block schematic (Fig. 2.1), context information can be applied in the low level fusion (Fig. 1.11) either in the prediction step or in the measurement update of filters (SE processes), or alternatively influence the hypothesis generation procedure during the data association (DA) process (Sec. 4.3.1). In this work, so called, spatial context enters the fusion node (Fig. 3.2) through a context middleware (CM) whose main function is to constrain the posterior densities during a state estimation process discussed in Sec. 4.3.1 and Sec. 4.3.2. The context is considered to be static and only applied at the object assessment level. For this reason the adaptation management functions are not required in the proposed methodology.

4.2 Related Work

Contextual information (CI) exploitation has been recently considered in the design of modern object tracking algorithms [18]. For example, [104] use topographic background information to enhance the tracking of ground vehicles in complex dynamic environments. Presence of CI in such a scenario significantly improved the quality and continuity of tracks, particularly during stop and go maneuvers and target masking due to Doppler blindness. Similarly, Gustafsson et al. [59] improved navigation and tracking performance of road bound vehicles by imposing road constraints on their trajectories. The so called road-assisted navigation, takes advantage of dynamic matching between the motion model of a vehicle and the road network implemented as manifold. Utilizing road constrains for target tracking was also considered in [2]. The proposed algorithm propagates the joint PDF of the target kinematic state and target ID in a road-constrained environment. Inequality constraints on target speed and off-road distance are treated using a min-max saturation approach, which requires a low computational load while leading to suboptimal constraints satisfaction. Road map assisted ground targets tracking is also considered in [143]. Here the author proposes the usage of a Gaussian sum algorithm within a Variable Structure Multiple Models (VSMM) scheme. As long as the predicted estimate is inside the same road, a Kalman filter is used to perform the update step. When the target approaches a junction, an on-road projection is necessary, and a multiple hypotheses approach is followed. Good results are presented, but the approach cannot be extended to nonlinear inequality constraints. The context-aided tracker or ConTracker [102] utilizes CI obtained from naval maps, such as a water depth, trade route paths, and areas/buildings with a high strategic value, in order to detect anomalies in the ship traffic. Ground targets tracking with airborne GMTI sensor measurements is considered in [152]. A refined GMTI sensor model with state dependent detection probability and information about the clutter notch is proposed. Both equality and inequality constraints are used to model the known road network. The latter are used to model non-zero width roads. The authors investigate the performance of both Gaussian sum and particles based approximations, in which the prediction step is performed in road coordinates, while the update step is carried out in the 2D Cartesian space. Hard inequality state constraints are considered in [27]. The paper is focused on tracking of airplanes with a known flight envelope (i.e. minimum and maximum velocities). The authors propose to obtain samples from a truncated distribution using a Rejection-Sampling approach, where particles are re-proposed if they do not meet the constraints. The proposed PF algorithm converges to the true a posteriori PDF for a sufficiently large number of particles, but might be unfeasible due to the computational load required. In [7] a VSMM-PF is used for tracking of ground targets with GMTI sensors. The information available through a road map is modeled using a Jump Markov system with state dependent transition probabilities. Each road segment is represented by two way-points determining direction, location, and length of each road. The visibility is defined as a binary valued probability, and an entry/exit condition is given by a Boolean variable for a subset of the roads. The inequality constraints on target speed are applied in prediction through the generation of random variables from a truncated Gaussian.

4.3 Model Formulation

4.3.1 Contextual Bayesian Estimation

In this section we will briefly recall constrained Bayesian filtering based on pivotal work published by Papi at al. [112]. The formalism will be tailored towards exploitation of CI in the measurement update in the form of contextual constraints. Consider a discrete-time state-space system of a form

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{v}_k \quad \text{or} \quad p(\mathbf{x}_{k+1}|\mathbf{x}_k), \tag{4.1}$$

$$\mathbf{y}_k = \mathbf{h}_{y\,k}(\mathbf{x}_k) + \mathbf{w}_k \quad \text{or} \quad p(\mathbf{y}_k|\mathbf{x}_k), \tag{4.2}$$

$$\mathbf{c}_k = \mathbf{h}_{c\,k}(\mathbf{x}_k) \quad \text{or} \quad p(\mathbf{c}_k|\mathbf{x}_k), \tag{4.3}$$

where $\mathbf{f}_k$ is a nonlinear function of the target state vector $\mathbf{x}_k$ and process noise $\mathbf{v}_k$. Variable $\mathbf{h}_{y\,k}$ represents a nonlinear relationship between sensor output $\mathbf{y}_k$ and target state vector $\mathbf{x}_k$ affected by a measurement noise $\mathbf{w}_k$. The contextual information $\mathbf{c}_k$ is represented as nonlinear dependency $\mathbf{h}_{c\,k}$ on state vector $\mathbf{x}_k$. The goal of nonlinear estimation is to infer the state variable $\mathbf{x}_k$ with the available sensor measurements $\mathbf{y}_{1:k}$ conditioned on context $\mathbf{c}_{1:k}$. By using the Bayesian framework, this estimation problem relates to the recursive evaluation of the probability density function (PDF) $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{c}_{1:k})$ in two consecutive steps, the prediction and the measurement update of the state vectors.

$$p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Prediction}} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1}) \tag{4.4}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Measurement}} p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{c}_{1:k}) \tag{4.5}$$

The prediction state density $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1})$ of state $\mathbf{x}_k$ is calculated from the prior PDF $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1})$ by using Chapman-Kolmogorov equation

$$\begin{gathered} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1}) = \\ \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1}) d\mathbf{x}_{k-1}. \end{gathered} \tag{4.6}$$

Equality (4.6) follows the 1^{st} order Markov property which assumes that $p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1})$ only depends on state $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ at time k and $k-1$ respectively. The measurement update $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{c}_{1:k})$ is computed from the prior distribution (4.6), measurements $\mathbf{y}_k$ and context $\mathbf{c}_{1:k}$ by a Bayesian formula which results in

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{c}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{c}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{c}_k) p(\mathbf{c}_k|\mathbf{c}_{1:k-1})} \tag{4.7}$$

The 1^{st} order Markov property for equation (4.7) implies that $p(\mathbf{x}_k|\mathbf{y}_{1:k}, \mathbf{c}_{1:k})$ only depends on measurement $\mathbf{y}_k$ and $\mathbf{c}_k$ at time k.

4.3.2 Particle Filter with Constrained Measurement Update

In comparison to the standard particle filter implementation, extensively studied by Gustafsson et al. [59], the constrained particle filter utilizes both measurement $p(\mathbf{y}_k|\mathbf{x}_k)$ and context likelihood $p(\mathbf{c}_k|\mathbf{x}_k)$ during the measurement update step. This slight change to the standard particle filter equations will be presented in the following. As outlined in [112], adding an additional likelihood function in the weight evaluation process is very straightforward.

$$w_k^i = \frac{1}{c_k} w_{k-1}^i \frac{p(\mathbf{y}_k|x_k^i)p(\mathbf{c}_k|\mathbf{x}_k)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, \mathbf{y}_k)}. \tag{4.8}$$

Here, the state vector $\mathbf{x}_k$ should not be confused with particles x_k^i and their associated weights w_k^i indexed by i, where $i = 1 : N$. The normalization factor c is defined as

$$c_k = \sum_{i=1}^{N} w_{k-1}^i \frac{p(\mathbf{y}_k|x_k^i)p(\mathbf{c}_k|\mathbf{x}_k)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, \mathbf{y}_k)}. \tag{4.9}$$

By far the most common implementation of the particle filter suggests the conditional prior of the state vector to be used as proposal distribution i.e.

$$q(x_k^i|x_{k-1}^i, \mathbf{y}_k) = p(x_k^i|x_{k-1}^i) \tag{4.10}$$

where $p(x_k^i|x_{k-1}^i)$ is referred to as the prior of x_k for each trajectory. This yields to simplified version of (4.8)

$$w_k^i = \frac{w_{k-1}^i p(\mathbf{y}_k|x_k^i)p(\mathbf{c}_k|\mathbf{x}_k)}{\sum_{i=1}^{N} w_{k-1}^i p(\mathbf{y}_k|x_k^i)p(\mathbf{c}_k|\mathbf{x}_k)}. \tag{4.11}$$

The context likelihood function $p(\mathbf{c}_k|\mathbf{x}_k)$ could be expressed as constraint factor of two cases

$$p(\mathbf{c}_k|x_k^i) = \begin{cases} 1, & \text{if} \quad \mathbf{a}_k \leq \mathbf{h}_{ck}(x_k^i) \leq \mathbf{b}_k \\ 0, & \text{otherwise.} \end{cases} \tag{4.12}$$

Here the $\mathbf{a}_k$ and $\mathbf{b}_k$ are hard constraints. The generation of the likelihood function $p(\mathbf{c}_k|x_k^i)$ is discussed in section 4.3.3.

4.3.3 Likelihood Masks

We outlined the process of likelihood masks $p(\mathbf{c}_k|x_k)$ generation by using a road map likelihood example (Fig. 4.2). Here, the source of context are image pixels (features) obtained from the areal images. However, without loss of generality, the process of likelihood mask generation remains unchanged regardless of context origin. Obtaining contextual knowledge from the data is a governed by feature detectors, a problem beyond the scope of this study. For the purpose of this work, we assume that context features are correctly acquired by the system, and therefore, we focus on the process of context modeling and classification.

We propose to model context as a parametric probability density function $p(\mathbf{c}_k)$ represented as a weighted sum of probability densities.

$$f(\mathbf{c}_k; \alpha_1, .., \alpha_k) = \sum_{k=1}^{K} w_k p(c_k, \alpha_k). \tag{4.13}$$

If we assume that $p(\mathbf{c}_k)$ is a Gaussian, then (4.13) could be expressed as a Gaussian mixture model (GMM).

$$p(\mathbf{c}_k|w_k, \mu_k, \sigma_k) = \sum_{k=1}^{K} w_k \mathcal{N}(\mu_k, \sigma_k). \tag{4.14}$$

Here, the $k - th$ component is characterized by normal distributions with weights w_k, means μ_k and covariance matrices Σ_k. Gaussian mixtures are flexible and offer relatively high model scalability achieved by changing the number of components K in the mixture (4.14). Furthermore, GMM is able to capture the complex spatial topology and utilize the Bayesian framework to iteratively adapt for changes in the environment. The covariance matrices Σ_k can be full rank or constrained to be diagonal. Additionally, parameters (w_k, μ_k, Σ_k) can be shared, or tied, among the Gaussian components, such as having a common covariance matrix for all components. The choice of model (4.14) configuration is determined by the amount of data available and how well the GMM fits to a particular contextual information. The parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or maximum a posteriori (MAP) estimation from a well-trained prior model. That is, to find the model parameters (w_k, μ_k, Σ_k) which maximize the likelihood of the GMM given the training data. For a context sequence $\mathbf{c}$ of k training vectors $\mathbf{c} = \{c_1, c_2, ..., c_K\}$, the GMM likelihood, assuming independence between the vectors, can be written as,

$$p(\mathbf{c}|w_k, \mu_k, \sigma_k) = \prod_{k=1}^{K} p(\mathbf{c}_k|w_k, \mu_k, \sigma_k). \tag{4.15}$$

In order to incorporate (4.14) into a Bayesian estimation, it is multiplied with the known distribution $p(\mathbf{x}_k|\mathbf{c}_k)$ of the state vector $\mathbf{x}_k$ conditioned on the context $\mathbf{c}_k$. The contextual likelihood $p(\mathbf{c}_k|\mathbf{x}_k)$ is also a Gaussian mixture model of the form

$$p(\mathbf{c}_k|\mathbf{x}_k, w_k, \mu_k, \sigma_k) = \sum_{k=1}^{K} w_k \mathcal{N}(\mu_k, \sigma_k). \tag{4.16}$$

As shown in (4.12), this likelihood is used to express the contextual constraints. As already mentioned, any classifier could be used to learn the context likelihood function. Gaussian mixtures where here employed as they allowed both unsupervised data clustering and classification. The result of GM classification is dependent on amount of dataset points, number of Gaussian mixtures K employed, as well as the amount of parameters w_k, μ_k, Σ_k shared among the Gaussian components. Too many clusters K complicates the classification result, therefore, makes it hard to interpret and analyze. A division with too few clusters K causes the loss of information and misleads the final interpretation of data. Good practices on how to estimate the number of Gaussian mixtures K in (4.16) are summarized in [161]. However, we argue that K should be no less than a number of intersections given a particular road network.

4.4 Experimental Results

4.4.1 Simulation Design

Nearly constant velocity model was employed in the synthetic dataset generation and estimation process. 2D state space model is defined in the local level frame and reads as follows

$$\begin{bmatrix} p^x_{k+1} \\ p^y_{k+1} \\ v^x_{k+1} \\ v^y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p^x_k \\ p^y_k \\ v^x_k \\ v^y_k \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} a^x_k \\ a^y_k \end{bmatrix}. \tag{4.17}$$

Acceleration $\mathbf{a}_k$ is driving the system noise generation defined as $\mathcal{N}(0, \sigma_a = 2ms^{-2})$. Initial conditions for the system are set to $\mathbf{x}_0 = [5km\ 5km\ 25ms^{-1}\ 25ms^{-1}]^T$. We are using simplified sensor model consisting directly from position measurements.

$$\begin{bmatrix} y^x_k \\ y^y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p^x_k \\ p^y_k \\ v^x_k \\ v^y_k \end{bmatrix} + \begin{bmatrix} w^x_k \\ w^y_k \end{bmatrix} \tag{4.18}$$

Uncertainties in observations are modeled by a white noise $\mathbf{w}_k \approx \mathcal{N}(0, \sigma_w = 20m)$ both for x and y components. This model is used for generating the target originated measurements with the detection probability of $P_D = 0,9$. Generated measurements are added to 200 sets of clutter by selecting a random target start time t_s distributed uniformly over the interval $t_s \in (0, 50)$ seconds. Clutter measurements are Poisson distributed with the rate $\beta_{FA} = 1.10^{-7}$. The spatial distribution of the false alarms is uniform in the region $x, y \in (0, 10km)$.

The implemented tracking algorithm follows the logic and advices published in [13]. Gating of the measurements $\mathbf{y}_k$ with the current initiators $\mathbf{y}_{k|k-1}$ is evaluated against gating threshold γ_G.

$$(\mathbf{y}_k - \tilde{\mathbf{y}}_{k|k-1})^T \mathbf{S}_{k|k-1}^{-1} (\mathbf{y}_k - \tilde{\mathbf{y}}_{k|k-1}) \tag{4.19}$$

Here the $\mathbf{S}_{k|k-1}^{-1}$ is an innovation covariance and γ_G is χ^2 distributed given the probability of gating $P_G = 0,9$ and number of measurements $\mathbf{y}_k$. In the cases when measurements fall into the gate $\{y_k^i\}_{i=1}^{n_y}$ we evaluate them against hypotheses θ_0 and θ_i.

$$\begin{aligned}\theta_0 &= \text{All measurments } \{y_k^i\}_{i=1}^{n_y} \text{ are FA}\\ \theta_i &= \text{Measurment } \{y_k^i\}_{i=1}^{n_y} \text{ belongs to the target, rest are FA}\end{aligned}$$

Probabilities $p(\theta_i|\{y_{0:k}^i\}_{i=1}^{n_y})$ associated with hypothesis θ_i are computed as a cases where

$$p(\theta_i|\{y_k^i\}_{i=1}^{n_y}) \propto \begin{cases}(1-P_D P_G)\beta_{FA}^{n_y} & \text{if} \quad i=0,\\ p(y_k^i|\{y_{0:k-1}^i\}_{i=1}^{n_y}) P_D \beta_{FA}^{n_y} & \text{otherwise.}\end{cases} \tag{4.20}$$

Here the probability $p(y_k^i|\{y_{0:k-1}^i\}_{i=1}^{n_y})$ is defined as $\mathcal{N}(\mathbf{y}_k; \mathbf{y}_{k|k-1}, \ \mathbf{S}_{k|k-1})$.

The particle filter uses conditional prior of the state vector as a proposal distribution, i.e. $q(x_k^i|x_{k-1}^i, \mathbf{y}_k) = p(x_k^i|x_{k-1}^i)$, sampled with 500 particles. When the particle depletion reaches a ratio of 0.5, the filter commences re-sampling. During the weights update, measurement $p(\mathbf{y}_k|x_k^i)$ and context $p(\mathbf{c}_k|x_k^i)$ likelihoods are sampled with the same set of particles x_k^i. For computation of context weight factor cw_k^i, we first evaluate the Mahalanobis distance of a particle x_k^i from a multi-variable Gaussian mixture distribution. This results in a vector of distances $\mathbf{d} = [d_1, d_2, \ldots, d_l]$ in a length l equivalent to the number of components i.e. Gaussian PDFs, the mixture comprises from. That is

$$\mathbf{d}_{x_k^i} = \sqrt{(x_k^i - \mu_{1:l})^T \Sigma^{-1} (x_k^i - \mu_{1:l})} \tag{4.21}$$

with a Σ being slices of covariances of the Gaussian mixture. Towards the goal of evaluating context weight factor cw_k^i, the minimum distance of $\mathbf{d} = [d_1, d_2, \ldots, d_l]$ is chosen. Assuming the road map scenario, the hard constraint affects the context weights as

$$cw_k^i = \begin{cases}0, & \text{if} \quad \mathbf{d} \geq 3\sigma\\ 1, & \text{if} \quad \mathbf{d} < 3\sigma.\end{cases} \tag{4.22}$$

4.4.2 Results and Discussion

The reference scenario used for data evaluation is shown in Fig. 4.4. The scenario consists of three roads, visualized by dot-dashed lines, intersecting at two points. The process of trajectory and measurements generation follows the motion model discussed in Sec. 4.4.1. The resulting trajectory is a sum of pseudo-randomly generated segments which fits into the road map. The target appears at random time in the bottom section of lateral road (labeled as $Road3$) and continues trough first intersection ($Road2$ and $Road3$ crossing) upwards till it reaches second intersection ($Road1$ and $Road2$ crossing). Here the vehicle could continue straight or turn. The vehicle trajectory is $90s$ long from the time it first appears. At the time of writing the paper, the trajectory generator was undergoing development, and as a consequence, we are not able to demonstrate stop-and-go maneuvers. Acceleration is treated as a random input variable which simulates vehicle speed changes over the course of its trajectory. Furthermore, the trajectory changes at the intersections are very rapid leaving the tracker susceptible to track losses.

Fig. 4.5 reports the tracking result estimated by a particle filter without the presence of context. In the figure, gating ellipsoids are associated with the tracks initiators. Relative error of the $90s$ long simulation run is indicated by a green line on the Fig. 4.8. The average error in position achieved during the simulation run was $\bar{x}_x = 31.61m$ in x direction and $\bar{x}_y = 32.29m$ in y direction.

Having the nominal solution been created, the task is to generate the road likelihood and then incorporate this contextual information into the tracking process. From the road map, scaled and aligned w.r.t global coordinate frame, road features (pixels) were extracted. By choosing a number of Gaussian mixtures

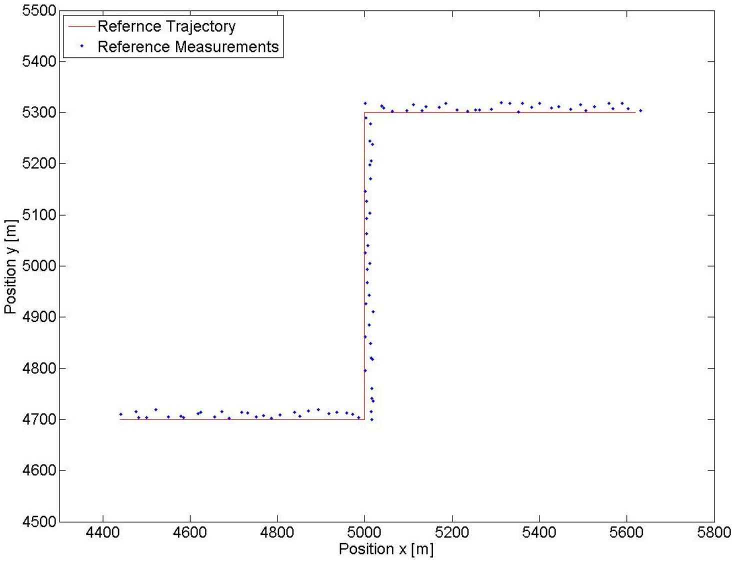


Figure 4.4: Reference trajectory (red) accompanied with the reference measurements (blue) in a road scenario (black).

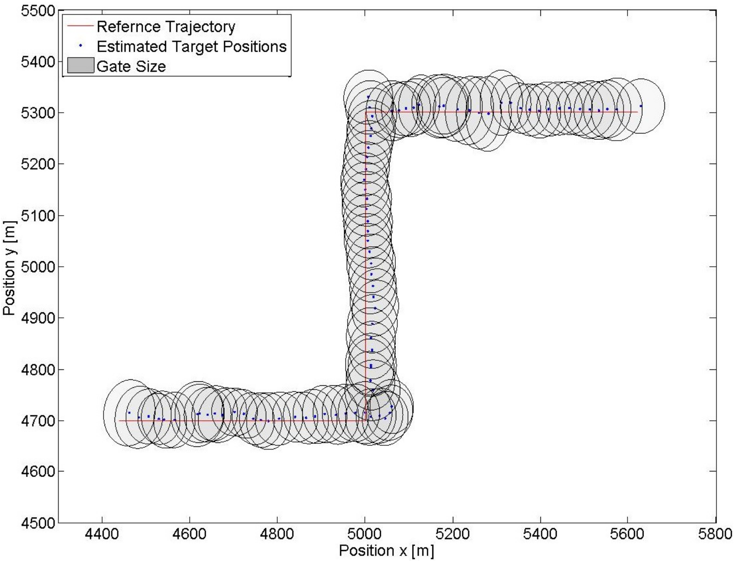


Figure 4.5: Reference trajectory compared against PF tracking.

$k = 3$, covariance matrices Σ_k constrained to be diagonal and parameters (w_k, μ_k, Σ_k) to be tied among the Gaussian components, the following GM model (Fig. 4.6) was obtained as a result of classification. The road likelihood mask was obtained as a result of classification process where a number of Gaussian mixtures $k = 3$, covariance matrices Σ_k constrained to be diagonal and parameters (w_k, μ_k, Σ_k) to be tied among the Gaussian components. $Road3$, $Road2$ and $Road1$ GMs are carrying the knowledge about road width equal to $21m$, $7m$ and $14m$ std. in position, respectively. It is worth mentioning, that sensor std. is set to be $20m$. Therefore, we do not expect context knowledge to have much influence on performance in the blue section of the trajectory.

Fig. 4.7 shows the performance of the particle filter, this time, augmented with the contextual knowledge of the road network. In comparison to Fig. 4.5, the spread of particles during contextual tracking was reduced in accordance to the road constraints, which lead to improved state estimation. This fact is also

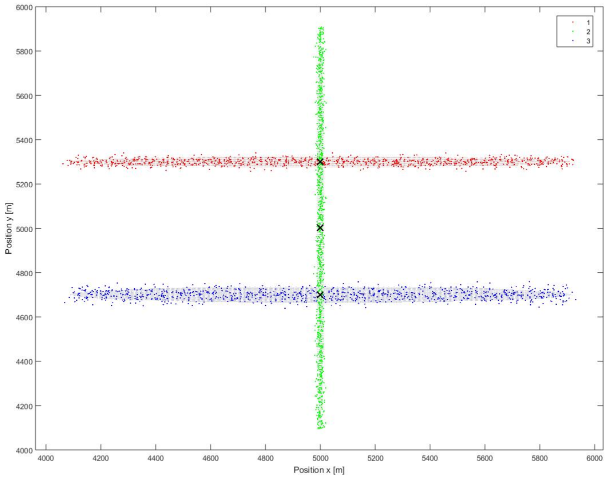


Figure 4.6: Road likelihood mask represented as a Gaussian mixture model.

reflected by mean error values, which were reduced from $\bar{x}_x = 31.61m$ to $\bar{x}_x = 4.17m$ in x direction and from $\bar{x}_y = 25.15m$ to $\bar{x}_y = 32.29m$ in y direction.

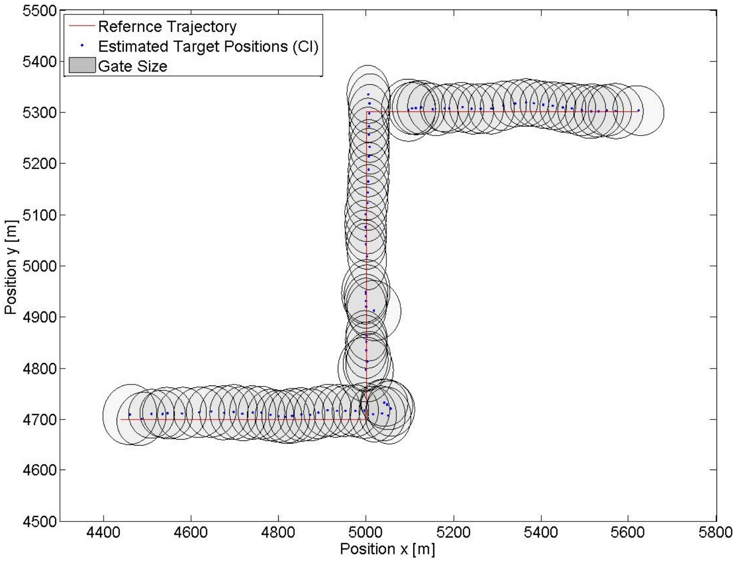


Figure 4.7: Reference trajectory compared against contextual PF tracking.

Plots of relative errors for PF with and without contextual information knowledge are shown on the figure (Fig. 4.8).

By looking at the relative error characteristics, neither filter was able to prevent the track loss at the intersections i.e. at times $t_1 = 30s$ and $t_2 = 60s$ to the simulation. However, the context tracking solution was able to recover from the track loss slightly faster. The shape and orientation of the Gaussian mixture has significant effect on filter performance (Fig. 4.9c, 4.9d). It can be observed, that the contextual constraints of $Road2$ (Fig. 4.9c) apply only in longitudinal direction, thus constraining particles in x and not at all in y direction. On the contrary, $Road1$ (Fig. 4.9d) constrains particles in the y direction and not at all in x direction.

We have performed 30 Monte-Carlo runs of a scenario, where a vehicle appears at random time some-

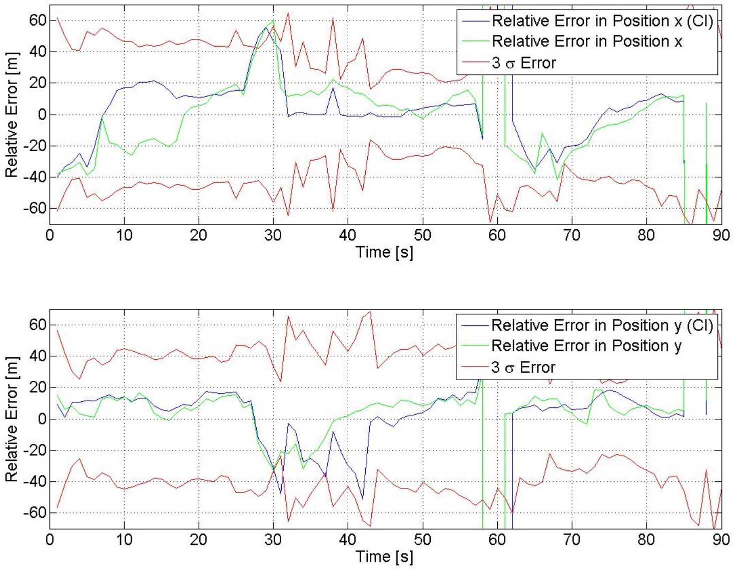


Figure 4.8: Reference trajectory compared against PF tracking.

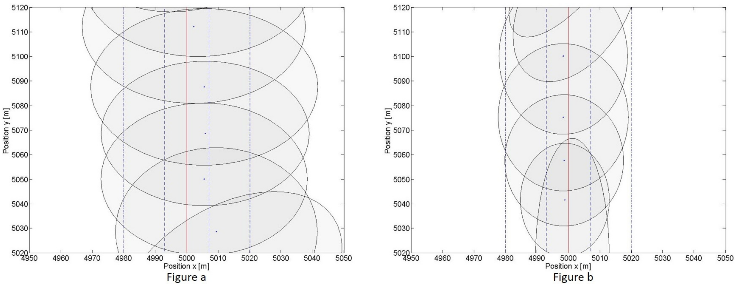


Figure 4.9: Likelihood mask effect on shape of gating ellipsoids. Figure a correspond to conventional tracking while Figure b to context enhanced tracking

where in $Road1$. The vehicle randomly selected segments of the trajectory which involve turning towards intersection of $Road2$ and $Road3$. At the 2^{nd} intersection the vehicle chooses to either turn or continue straight. Collected statistic of relative error and variance in position for target tracking with and without context is summarized in the table (Tab.: 4.1). Given the scenario, we can observe approximately 27% improvement in tracking performance by using the road context.

Target Tracking	Relative Error Mean [m]	Relative Error Variance [m]
Context	15,62	3.25
No-Context	21,59	13.76

Table 4.1: Comparison of target tracking with and without context

4.5 Conclusion

We proposed a solution to the problem of multi-level context representation and exploitation for target tracking. The presented approach encodes different types of CI in the form of likelihood functions. Likelihood functions are applied as constrains in the particle filter measurement update. Given the scenario, we have been able to improve accuracy of the target estimate approximately by 27%. Despite achieving promising results, there are multiple factors and limitations which needs to be considered during the algorithm design.

As one might expect, informativeness of context is the primary contributor to improved tracking performance. Therefore, generation of accurate and information rich likelihood masks is of crucial importance. We addressed the issue of accuracy by employing Bayes classifiers on training datasets, and adopted the context model with the best fit for the data. Regarding information richness, we assumed context features to be known as a priori. The proposed solution of employing classifiers to cluster and partition contextual data proved to be very versatile by providing: the ability to couch different types of knowledge, a principled functional representation exploitable as likelihood in the tracking process, the ability to dynamically update contextual likelihoods by re-training.

Constraining particles in the measurement update could lead to large amount of particles with small or vanishing weights. By reducing the size of the likelihood Particle filter become less efficient and eventually lead to track losses. For this reason the constrains has to be applied with the care.

It has become apparent, that presented solution experienced difficulties, while tracking the target at the intersections. An immediate solution, would suggest to the apply constrains into the prediction update of the filter and constrain the prior distribution. However, constraining both prediction and measurement update of the filter at the same time will make particle filter ineffective for tracking from Bayesian point of view.

5

Context-Based Goal-Driven Reasoning for Improved Target Tracking

Tracking objects in complex dynamic environments can be less challenging once their intents are recognized. Inferring on a targets' future actions based on their past can be addressed via probabilistic reasoning. Context information plays a crucial role in the reasoning process as it provides additional clues about targets' intents. However, architectures combining context reasoning with target tracking are merely not existing. The framework here discussed views target's actions as a Hidden Markov Model (HMM) with relevant context associated with each node. Context is at each time step selected based on immediate and goal driven sets of actions. Inference in the HMM is conditioned on prior target's measurements and the belief state conditioned on context. This posterior is then compared with the target's state estimate in order to adjust the switching probability in the Interactive Multiple Models (IMM) tracking process.

5.1 Introduction

The significance of Contextual Information (CI) for object tracking has been demonstrated in an increasing number of works [18], [16]. In information fusion (IF), any information that surrounds an object or situation of interest could be considered as contextual. However, only relevant context (contextual variables) provides useful clues in understanding the estimated situation and events [134]. However, the process of finding a relevant context is not trivial and often involves a complex integration of IF with planning, abductive logic and control functions. As of now, no unified framework for design of such context-aware event-driven system exist, but one might consider concepts for "a priori" and "a posteriori" CI exploration respectively as a good reference [49]. Situations and events in the world rarely happen independently [140], [141], [119], the spatial layout of events and their sequential patterns could provide useful clues not only for situations understanding, but also for learning of relevant context and ultimately improve accuracy and robustness of a tracking process.

Consider the scenario depicted in (Fig. 5.1), where a vehicle is being tracked by an airborne sensors in an urban environment of a priori known topology. Low visibility, complex dynamics of the targets, high clutter and target density make tracking in the real world equivalents of such scenarios extremely difficult [104]. In these cases, the exploitation of a priori information, i.e. road topology, become essential in order to maintain track accuracy and continuity [59], [102], [153]. However, the understanding of context is not only limited to spatial relationships (location in space), but could also include higher levels of abstraction. In our case a spatio-temporal relationships. As the vehicle progresses throughout the road network, it creates a sequence of events denoted by event state s at any given time k. Every time a vehicle reaches an intersection, it could decide (based on the topology) to drive *"right"*, *"left"* or going *"straight"* and by doing so it creates an event which could be related to a specific mode of a tracking algorithm. Events can be seen as the nodes of the graph tied together by a series of relationships. Relationships which binds the events together in a space and a time will be referred to a spatio-temporal context $\mathbf{c}$ [166]. Spatio-temporal

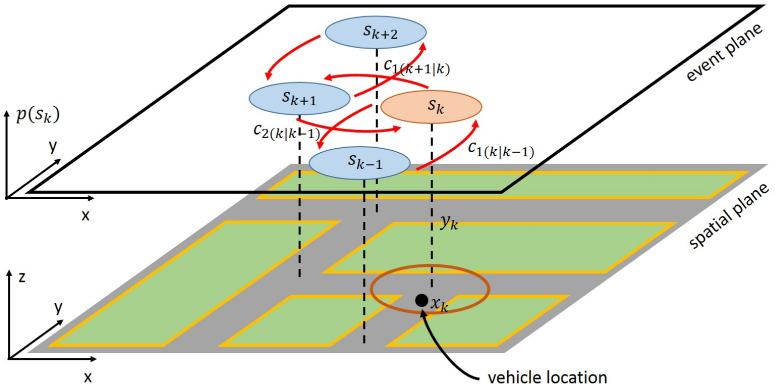


Figure 5.1: Relationship between event and spatial context.

context is a latent variable which needs to be learned from causalities naturally occurring between the objects in events. Having learned a priori knowledge of spatio-temporal context, with respect to certain observable variables characterizing the targets in relation to specific behaviors, the future actions of the target could be anticipated by abducing that the target is pursuing a certain goal given the context. This example shows the importance of spatial and temporal context relationships for target tracking as well as emphasizes the synergies between information fusion, i.e. object and situation features, at different layers of abstraction.

5.1.1 Overview of the Framework

Many existing works on target tracking exploit CI at the object assessment level [93], [17] also known as JDL fusion level 1 [14]. At this level, CI is generally considered as a constraining factor which affects evolution of the kinematic parameters of the system [127]. Knowledge of soft or hard constrains could be applied in Bayesian recursion either in the prior density function or in the likelihood [112]. We will be referring to all these solutions as object/target tracking (TT) with CI exploitation at the level 1 (Fig. 5.2) or shortly CE1 tracking.

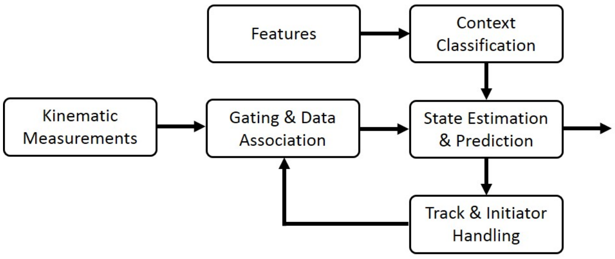

Features
Context Classification
Kinematic Measurements
Gating & Data Association
State Estimation & Prediction
Track & Initiator Handling

Figure 5.2: Target tracking with CI exploration at JDL level 1

The CE1 architecture allows the utilization of spatial context, such as topographic background information, road constraints, naval maps as successfully demonstrated in [104], [59], [102], [112] and our own work [153], respectively. However, we are not only concerned about the problem of detection and exploitation of spatial context but all available context. The main challenge is to develop a representation of the tracking scene that respects the spatio-temporal relationships of the events and successfully utilizes this new knowledge in the fusion process. To achieve this goal, we build upon existing spatio-temporal context representations, situation assessment strategies and interactive multiple model techniques that, when combined together, provide a powerful framework for context exploitation.

Having the situation depicted in (Fig. 5.1) in mind, we describe causalities occurring between situations and a single event as a Hidden Markov Model (HMM) (Fig. 5.3). An event occurs at/between each

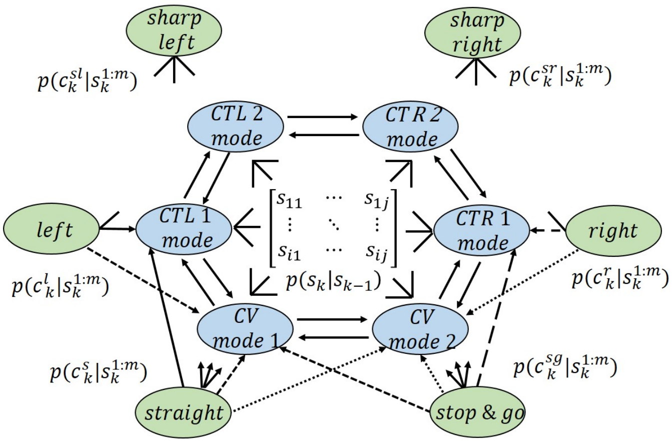


Figure 5.3: Event model structure represented as HMM

intersection and encaptures the notion of the vehicle driving *"straight"* or *"turn"* with either *"high"* or *"low"* kinematics. The tracking algorithm perceives the event at time k as a probability of modal state s_k^i associated with the nearly constant velocity models *"CV mode 1, 2"* and the coordinated turn models *"CT modes 1, . . . , 4"*. In our demonstration scenario, we observe only three types of maneuvers c_k^j, e.g. turning *"right"*, turning *"left"* and continuing *"straight"*, which could be either of *"high"* or *"low"* kinematics. However, one could think of adding *"U-turn"*, *"reverse"* or *"lane change"* maneuvers to increase the complexity of a simulation. The unknowns in the presented HMM model are the emission probabilities $p(c_k^j|s_k^i)$, which define the observation to event relationships, and the mode transition probabilities $p(s_k^i|s_{k-1}^j)$. As it will be described later, we consider the transition mode probability $p(\mathbf{s}_k|\mathbf{s}_{k-1})$ as a problem variable, which we intent to infer from the observations, context and the goals pursued by targets. The emission probability $p(\mathbf{c}_k|\mathbf{s}_k)$ represents spatio-temporal contextual variables $\mathbf{c}_k$ condition on event $\mathbf{s}_k$ we learn from the environment. Individual events, modal s_k^i or contextual c_k^j, are combined together in space and time in order to form sequences $\{\mathbf{c}\}_1^{N_s}$ or $\{\mathbf{s}\}_1^{N_s}$ of length N_s, respectively. Each sequence represents an unique goal that target seeks to reach within given topology. As the vehicle progresses through the network, we infer on target future actions based current evidence and a priori knowledge. The belief that a vehicle seeks an objective is used to aid the tracking process. Events regions (Fig. 5.1) are not necessarily constant and they vary in size and shape based on the event they represent. As the vehicle approaches an intersection the event becomes more relevant the shorter the distance to centroid of a context event is. Assuming that the relevance is an exponential function we decided to represent it as a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ of an event centroid μ and area of effect Σ.

Estimation of a target kinematics in scenario (Fig. 5.1) is usually referred to as a maneuvering target tracking problem under motion-mode and measurement-origin uncertainties, best described by a semi-Markov processes (Fig. 5.4). The evolution of mode state m_{k-1}^j to the next mode m_k^i is in the semi-Markov

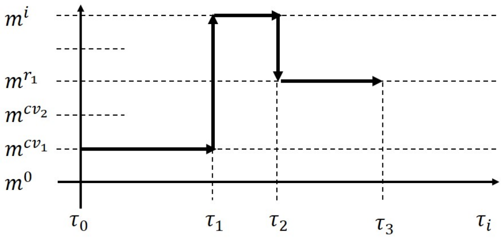


Figure 5.4: Semi-Markov Process

process is chosen at random according to the transition probability $\pi_{k-1|k-1}^{ji}$. The time between mode m_k^i

and m^j_{k-1}, i.e. the sojourn time τ^i in mode m^i_k, is selected based on the sojourn time PDF $f(\tau^{ji})$. By denoting the mode state of an estimator by m^i_k instead of s^i_k, we distinguish between Gaussian Model (GM) and Hidden Markov Model (HMM). Despite these two models are both belonging to the same family of Bayesian estimators, they are fundamentally different and their estimated mode probabilities $p(m^i_k|m^j_{k-1})$ and $p(s^i_k|s^j_{k-1})$ are not necessarily equal. By realizing that mode of a system in semi-Markov setting is not time invariant, i.e. $\mathbf{m}_k \neq \mathbf{m}$ for $\forall k$, a class of sojourn-time dependent Markov (STDM) cooperating multiple model (CMM) algorithms are arguably the most relevant solutions for described maneuvering target tracking problem [93]. However, the most popular CMM realization, the Interactive Multiple Models (IMM) [20], ignores the sojourn-time dependencies as the mode dependent state estimate $\hat{\mathbf{x}}^{ji}_{k|k}$ is equal to the expectation $E[\mathbf{x}^i_k|m^j_{k-1}, m^i_k, \mathbf{y}_{k-1}] = E[\mathbf{x}^i_k|m^j_{k-1}, \mathbf{y}_{k-1}]$. In other words, the time behavior of the mode state m^i_k is modeled as a Markov chain with a fixed time-invariant transition probability matrix (TPD) i.e. $\mathbf{\Pi}_{k-1|k-1} = \boldsymbol{\pi}^{ji}_{k-1|k-1} = p(m^i_k|m^j_{k-1})$. Therefore, an extension of the IMM configuration to the case of STDM process was proposed in [26]. Usefulness of the STDM IMM in real life applications is rather limited as the knowledge of sojourn-time dependent transition probability matrices is rarely available and hard to obtain. For this reason, we decided to utilize and slightly alter the classical IMM by adding a TPM estimation step into the recursion (Alg. 3). By doing so, the mode dependent conditional means $E[\mathbf{x}^i_k|m^j_{k-1}, \mathbf{x}^j_{k-1}, \mathbf{y}_{k-1}]$, covariances $\Sigma[\mathbf{x}^i_k|m^j_{k-1}, \mathbf{x}^j_{k-1}, \mathbf{y}_{k-1}]$, mode and transition probabilities $p(m^i_k|m^j_{k-1})$ will be computed recursively from prior values. An approximate Bayesian recursions for the TPM posterior, i.e. $p(\mathbf{\Pi}_k|m^i_k, \mathbf{y}_k, \mathbf{y}_{k-1})$, were in terms of the multiple models probabilities and likelihoods proposed by Jilkov and Li in [67] and [68], respectively. We build upon Jilkov's and Li's work, and by extending the TPM posterior for the prospect of events, i.e. $p(\mathbf{\Pi}_k|m^i_k, \mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})$, we seek to compensate for the lack of sojourn time knowledge and for ambiguities occurring between mode transitions by inclusion of a context $\mathbf{c}$. The unknown sojourn time PDF $f(\tau^{ji})$ of the mode m^i_k can be expressed by an event-temporal relationship also known as a context relevance. The ambiguities occurring between different modes can be mitigated by an event context and by a goal driven reasoning.

Utilizing event models is not uncommon in the information fusion, and proposed architecture (Fig. 5.5: Evaluation phase) leverages from concepts developed for situation assessment purposes [124]. Context features are classified either as a spatial context and directly utilized in CE1 tracking process (Fig. 5.2), or as a CI where meaning is discovered during reasoning in the event-relational model. By inferring on observed features, the probability of an event or situation outcome could be evaluated. Assessing the situations and events in process of object tracking is the backbone of proposed TT architecture with CI exploration at JDL level 2 (Fig. 5.5) or shortly CE2 tracking.

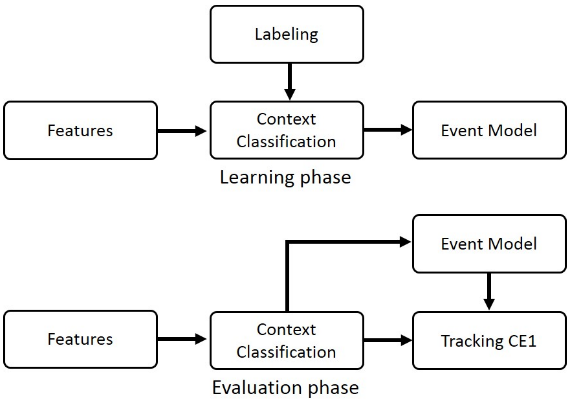


Figure 5.5: Target tracking with CI exploitation at JDL level 2

5.1.2 Main Contributions

The contribution of this work are threefold:

- A methodology for situational context exploitation via goal-driven reasoning (Fig. 5.3) is introduced and evaluated on a ground vehicle tracking scenario (Fig. 5.1).
- A context adaptive framework for unified situation assessment and target tracking is introduced and evaluated. In comparison to existing approaches. The proposed framework allows for context exploitation at JDL level 1 and 2 simultaneously (Fig. 5.5).
- The tracking filter adapts to target's modes recursively based on target's originated measurements and actions. Existing adaptive solutions utilize only an a priori defined set of modes which are engaged under certain conditions. Existing solutions rely heavily on domain expert knowledge, which is seen as a major drawback, that our solution avoids.

5.1.3 Relations to the Context Exploitation Framework

In this section we establish connections between the CE2 tracking (Fig. 5.5) and the adaptive context exploitation framework (Fig. 3.3) presented in Sec. 3. We assume that measurements originate from sensors insensitive to the context, thus contextual and measurement sources are conditionally independent from each other. Measurements enters the fusion node (Fig. 3.2) which is performing object tracking of maneuvering targets, i.e. the object assessment process (Fig. 2.1). Within the FN (Fig. 3.3), unfeasible observation are excluded from the processing during the ellipsoidal gating, i.e. the uncertainty characterization (UC) process. The state estimation (SE) process is governed by the IMM filter which recursively estimates not only the state vector of the target but also its' mode (Sec. 5.3.1). The data association (DA) problem is resolved by a JPDA methodology (Sec. 2.5). The problem of common referencing (CR) is ignored as only simplified sensor models are employed during the experimentation.

The context information represents the habits of a specific target navigating throughout the road network. The meaning of the contextual information, i.e. the relation to a specific mode of the target at given time, is inferred during the reasoning process (Sec. 5.3.4), described as the HMM (Fig. 5.3), and feed into the adaptation management module. The design of such a context-adaptive logic is presented in this chapter as a recursive mode transition estimation algorithm discussed in Sec. 5.3.2 and Sec. 5.3.3. As opposed to our previous work (Sec. 4), CI is dynamic in nature and evolves over time. As mode evidence about the target objective is being accumulated the stronger our belief about the target's objective becomes (Fig. 5.6). This belief is further adjusted by the relevance function, which is exponentially scaled by the distance to each intersection (Fig. 5.7).

5.2 Related Work

Contextual information (CI) exploitation has been recently considered in the design of modern object tracking algorithms [16]. For example, Mertens and Ulmke [104] used topographic background information to enhance tracking of ground vehicles in complex dynamic environments. Presence of CI in such a scenario significantly improved quality and continuity of tracks, particularly during stop and go maneuvers and target masking due to the Doppler blindness. Similarly, Gustafsson et al. [59] improved navigation and tracking performance of road bound vehicles by imposing road constraints on their trajectories. So called road-assisted navigation, takes advantage of dynamic matching between the motion model of a vehicle and a road network implemented as manifold. The context-aided tracker or ConTracker [102] utilizes CI obtained from naval maps, such as a water depth, trade route paths, and areas/buildings with a high strategy value, in order to detect anomalies in the ship traffic. Target features observed by visual sensors proved to be invaluable and by far the largest source of CI. Image features are more directly related to the target motion mode rather than the kinematic measurements. As such, incorporation of image context in tracking is closely related to target recognition and feature-aided tracking. For instance, one of the early study [148] utilizes the observations of aircraft shape in order to adjust the time constants in the range-bearing path tracker and subsequently discern changes in the target motion patterns. Another method for classification aided target tracking presented in [37] is concerned about processing of features and attributes in the data association process and successful utilization of these features in multiple sensors tracking scenarios. A model for image-based observation for discrete target mode estimation was presented in [164]. Loosely speaking, modal sensors are in aforementioned studies modeled as classifiers. Mode observations alongside with the sensors measurements are related to the states and target modes in a Bayesian sense via joint posterior distribution, respectively. Evans at. al. [41] argue, that optimal filter capable of couping with such a hybrid state estimation is computationally prohibitive. Therefore, authors propose the image-enhanced (IE) IMM filter as a practical solution for effective reduction of Gaussian mixtures components. However, in light of [17] exploitation of CI in the field of target tracking [16] is still rather bound to the low level information fusion (LLIF) i.e. object assessment. By neglecting higher levels of the information fusion (HLIF) the capability of the tracking algorithm to reason or to find relevant context is very limited.

In order to justify the above statement one might want to pay a closer attention to the recent works of Steinberg and Rogova et al. [142], [140], [141], [119]. Briefly, the authors suggest to model events or situations in the world as a set of relations and relationships in the form of factor graphs. In this sense, CI could seen as a background knowledge which is associated to every relation pair or in different words to each node of the graph. The uncertainty in these relations can be best described by a likelihood function. Such a factor graphs could be then used for the context reasoning either in logical or probabilistic manner. In the information fusion (IF), contextual reasoning relates to an inference of problem variables i.e. desired information given the basis of context variables i.e available information. Generally speaking, the process of finding a relevant context is not trivial and often involves a complex integration of IF with planning, abductive logic and control functions. As of now, no unified framework for design of such context aware system exist, but one might consider concepts for "a priori" and "a posteriori" CI exploration respectively as a good reference [49]. In conclusion, inferring on target's intents and goals is on mutual benefit to target tracking and supports the trend in the IF community to combine deferent levels of a fusion into a single framework [15], [14].

The problem of context reasoning, as an essential part of a situation and impact assessment respectively, could be conveniently scaled into the problem of a plan recognition. Plan recognition seeks to infer on a entity's plan based on the observed entity's actions and their effects. The role of CI in plan recognition is twofold. Firstly it creates additional links between events and entities. Secondly it adjust our confidence that an entity is following a certain plan. Probabilistic reasoning is the most popular approach in nowadays plan recognition and as such approaches utilizing Bayesian networks (BN) and their variants are very popular [23], [94] or [154]. Furthermore, the causality of relationships, modeled by factor graphs, could be expressed by multiple hierarchical levels of Bayesian networks at different time slices i.e. dynamic Bayesian net works (DBN). As the target progresses through the network and forming a chain of events, then the BN could infer on target's future actions based on its past. Similarly, relevant context could be then discovered or selected based on target's future or past actions, respectably.

To the best of our knowledge, BN were for the first time proposed to aid object tracking by Hautaniemi and Saarinen [61]. They intended to enhance the classic multiple target tracking (MTT) algorithm, based on interactive multiple model filter (IMM) and probabilistic data association (PDA), with quantities other than kinematic measurements i.e. context. Authors argued that PDA was unable to deal with a contextual information (CI) and therefore BN were developed as an additional inference method to aid data association and track identification processes. In [115], DBN are used to capture the modal nature of the tracking sea species. More specifically, the state space model of a tracked ocean animal was represented as a DBN itself including both continuous (the velocities) and discrete variables (the propulsive mode of the animal, and the discrete observation variables). Such a hybrid system was then used to adapt the bandwidth of the multiple model bootstrap filter based on the most probable models given their conditional probability densities (CPDs). Schubert and Wanielik [123] argued that context in [61] does not influence the tracking directly or in case of [115] a tracking procedure is computationally very expensive. Therefore, the authors proposed their own approach for incorporating additional information into the IMM called the Meta Model filter. With a structure similar to a BN, the Meta Model represents causality of events in form of an adaptive transition probability matrix. Elements of such a matrix are nodes and states which represent the possible modes and transitions to other modes respectively. The usefulness of such a Meta Model filtering was demonstrated in a line change maneuver recognition algorithm for vehicles with uncertain velocities and yaw rates. In their most recent publication [124], the authors leverage their expertise into development of an Advanced Driver Assistance System (ADAS) using unified Bayesian approach for tracking and situation assessment. In [114], an alternative method for learning dynamic models from training datasets of observed state space trajectories was presented. So called the Switching Linear Dynamic Systems (SLDS) describe a complex nonlinear dynamics by the successive linear models indexed by a switching variable. In order overcome the exponential complexity of the exact inference of the DBN, the authors proposed approximation techniques based on the Viterbi algorithm and the structural variational inference. Different from the previous appearances, a methodology that represents the environment by the attractive forces was recently proposed in [25]. Such a framework assumes the existence of attractive spots in the environment which could be used for altering the motion of agents. A switching model is used to describe near and far velocity fields, which are in turn used to learn attractive characteristics of environments.

The aforementioned work assumes a priori knowledge of the transition probability matrix (TPM). However, the TPM is a design parameter whose choice could significantly influence the estimation process. Therefore, algorithms which can identify the TPM during the course of tracking have been extensively studied in literature [68], [67]. Furthermore, variable structure algorithms, referred to as Expected-Mode Augmentation (EMA), for multiple-model estimation were developed [91], [92]. The EMA approach, the original model set is augmented by a variable set of models intended to match the expected value of the unknown true mode. An on-line maximum likelihood estimator for the transition probabilities associated with a Jump Markov Linear System (JMLS) was proposed by Orguner et al. [110]. The maximum likelihood estimator is derived using the reference probability method, which exploits an hypothetical probability measure to find recursions for complex expectations. The Expectation Maximization (EM) algorithm is utilized for maximizing the TPM likelihood function. These models are generated adaptively in real time as probabilistically weighted sums of modal states over the model set.

In conclusion, spatial context represents only a small portion of the background information suitable for aiding the target tracking processes. In this work, we propose a realization of the multi-level data fusion framework (Fig. 5.5) capable of exploiting spatial and situational context. The presented methodology models the relations between contextual observations alongside with sensors measurements and the state variables as conditional dependencies, and thus completely avoids the use of JPDs in the Bayes recursion [37], [147]. Approaches that model the conditional dependencies between events and situations in the form of DBNs, utilize pre-set TPM matrices in order to model the behavior of targets [124], [61] and [115]. On the contrary, our approach models the transition probabilities as parameters which are recursively estimated during the IMM tracking process. Differently from [68] and [110], we expanded the TPM estimator process for the inclusion of context.

5.3 Model Formulation

5.3.1 Maneuvering Target Tracking

The state dynamics of the maneuvering target could be conveniently modeled as a discrete system with Markovian coefficients

$$\mathbf{x}_k = \mathbf{f}_k(m_k)\mathbf{x}_{k-1} + \mathbf{g}_k(m_k)\mathbf{v}_k \quad \text{or} \quad p(\mathbf{x}_k, m_k|\mathbf{x}_{k-1}) \tag{5.1}$$

with observations

$$\mathbf{y}_k = \mathbf{h}_k(m_k)\mathbf{x}_k + \mathbf{w}_k \quad \text{or} \quad p(\mathbf{y}_k|m_k, \mathbf{x}_k). \tag{5.2}$$

Where $\mathbf{x}_k$ is the system state vector estimated from the observations $\mathbf{y}_k$. Parameter m_k is a mode state taking values $(1, 2, ..., N_m)$ and represents the current system model in use. Functions $\mathbf{f}(m_k)$, $\mathbf{g}(m_k)$ and $\mathbf{h}(m_k)$ are mode dependent state transitions. Vectors $\mathbf{v}_k \sim \mathcal{N}(\mathbf{w}_k; 0, \mathbf{Q})$ and $\mathbf{w}_k \sim \mathcal{N}(\mathbf{w}_k; 0, \mathbf{R})$ are mutually uncorrelated Gaussian distributed white noises with covariances $\mathbf{Q}$ and $\mathbf{R}$ respectively.

Suppose a target which is moving according to the kinematic model defined by (5.1) and (5.2). During the time period $(1 : k)$ a total number of N_m different model histories $m_{1:k}$ might have occurred. The history of the mode state $m_k \in (1, 2, .., N_m)$ can modeled as a homogeneous Markov chain with transition probability matrix (TPM) $\boldsymbol{\pi}^{ji}_{k-1|k-1} = p(m^i_k|m^j_{k-1})$. The goal of estimation in Markovian Jump Systems (MJS) framework is to infer the posterior density of the target $\mathbf{x}_k$ for a specific model history $m^i_{1:k}$ with the available sensor measurements $\mathbf{y}_{1:k}$. By using Bayesian framework this estimation problem relates to the recursive evaluation of the probability density function (PDF) $p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k})$ over all possible mode sequences $m_k \in (1, 2, .., N_m)$ that end in the mode m^i_k in five consecutive steps (5.3-5.7).

$$p(m^j_{k-1}|\mathbf{y}_{1:k-1}) \xrightarrow{\text{Mixing}} p(m^i_k|\mathbf{y}_{1:k-1}) \tag{5.3}$$

$$p(\mathbf{x}_{k-1}|m^j_{k-1}, \mathbf{y}_{1:k-1}) \xrightarrow{\text{Mixing}} p(\mathbf{x}_{k-1}|m^i_k, \mathbf{y}_{1:k-1}) \tag{5.4}$$

$$p(\mathbf{x}_{k-1}|m^i_k, \mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Prediction}} p(\mathbf{x}_k|m^i_k, \mathbf{y}_{1:k-1}) \tag{5.5}$$

$$p(m_k^i|\mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Mode}} p(m_k^i|\mathbf{y}_{1:k}) \tag{5.6}$$

$$p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k-1}) \xrightarrow[\text{Update}]{\text{Measurement}} p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k}) \tag{5.7}$$

The product of the posterior probability of the mode sequence $p(m_k^i|\mathbf{y}_{1:k})$ (5.6) and the posterior distribution over the system state $p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k})$ (5.7) conditioned on the mode sequence m_k^i can be seen in a view of the total probability theorem as $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ defined by

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \sum_{i=1}^{N_m} p(m_k^i|\mathbf{y}_{1:k}) p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k}). \tag{5.8}$$

The mode (5.6) and measurement update (5.7) are computed from their prior distributions $p(m_k^i|\mathbf{y}_{1:k-1})$ and $p(m_k^i|\mathbf{y}_{1:k-1})$ and measurements $\mathbf{y}_{1:k}$ by a Bayesian formula which result in (5.9) and (5.10), respectively.

$$p(m_k^i|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|m_k^i)p(m_k^i|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{5.9}$$

$$p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|m_k^i, \mathbf{x}_k)p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \tag{5.10}$$

The 1^{st} order Markov property for equations (5.9) and (5.10) implies that $p(m_k^i|\mathbf{y}_{1:k})$ and $p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k})$ only depends on measurement $\mathbf{y}_k$ at time k. The prediction state density $p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k-1})$ of state $\mathbf{x}_k$ in (5.5) is calculated from the prior PDF $p(\mathbf{x}_{k-1}|m_k^i, \mathbf{y}_{1:k-1})$ by using Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|m_k^i, \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|m_k^i, \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}. \tag{5.11}$$

By the law of the total probability expression (5.4) becomes

$$\begin{aligned} p(\mathbf{x}_{k-1}|m_k^i, \mathbf{y}_{1:k-1}) = \sum_{j=1}^{N_m} & p(m_k^i|m_{k-1}^j, \mathbf{y}_{1:k-1}) \times \\ & p(\mathbf{x}_{k-1}|m_{k-1}^j, \mathbf{y}_{1:k-1}). \end{aligned} \tag{5.12}$$

Density $p(m_k^i|m_{k-1}^j, \mathbf{y}_{1:k-1})$ relates to mode mixing (5.3) and represents the Chapman-Kolmogorov equation for a Markov chain (5.13).

$$p(m_k^i|\mathbf{y}_{1:k-1}) = \sum_{j=1}^{N_m} p(m_k^i|m_{k-1}^j) p(m_{k-1}^j|\mathbf{y}_{1:k-1}) \tag{5.13}$$

Probability $p(\mathbf{x}_{k-1}|m_{k-1}^j, \mathbf{y}_{1:k-1})$ is assumed to be know from the previous step and to be Gaussian distributed i.e.

$$p(\mathbf{x}_{k-1}|m_{k-1}^j, \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{k-1|k-1}^i, \mathbf{\Sigma}_{k-1|k-1}^i). \tag{5.14}$$

Substituting Gaussian distribution (5.14) into (5.12) and evaluating the whole recursion (5.3-5.7) yields a common approximation of $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ in (5.8) by a Gaussian mixture with N_m^2 components i.e.

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N_m} \boldsymbol{\mu}_k^i \mathcal{N}(\mathbf{x}_k; \hat{\mathbf{x}}_{k|k}^i, \mathbf{\Sigma}_{k|k}^i). \tag{5.15}$$

One might notice that the number of components in the Gaussian mixture (5.15) grows exponentially over time. Therefore, numerous techniques which reduce the number of components in the Gaussian mixture either by pruning or merging were developed [93]. Among them, the Interactive Multiple Model (IMM) filter introduced by Blom and Bar-Shalom [20] becomes a solution of choice. IMM approximates posterior mode probabilities μ_k^i in equation (5.15) with

$$\boldsymbol{\mu}_k^i \equiv p(m_k^i|\mathbf{y}_{1:k}) \tag{5.16}$$

Exact expression for the mode probability $\boldsymbol{\mu}_k^i$ is shown in the IMM algorithm (Alg.: 3). Furthermore the mode transition probability $\boldsymbol{\pi}_{k-1|k-1}^{ji} = p(m_k^i|m_{k-1}^j)$ in expression (5.13) is genuinely assumed to be known a priori and to be constant time invariant matrix. With such an approximation the overall posterior mean $\hat{x}_{k|k}$ and covariance $\Sigma_{k|k}$ are equal to formulas (5.17) and (5.18) respectively.

$$\hat{\mathbf{x}}_{k|k} = \sum_{i=1}^{N_m} \boldsymbol{\mu}_k^i \hat{\mathbf{x}}_{k|k}^i \tag{5.17}$$

$$\boldsymbol{\Sigma}_{k|k} = \sum_{i=1}^{N_m} \boldsymbol{\mu}_k^i [\boldsymbol{\Sigma}_{k|k}^i + (\hat{\boldsymbol{x}}_{k|k}^i - \hat{\boldsymbol{x}}_{k|k})(\hat{\boldsymbol{x}}_{k|k}^i - \hat{\boldsymbol{x}}_{k|k})^T] \tag{5.18}$$

This estimate (5.17) and covariance (5.18) can be given to the user as the output. The mode conditional means $\{\hat{x}_{k|k}^i\}_{i=1}^{N_m}$, covariances $\{\Sigma_{k|k}^i\}_{i=1}^{N_m}$ and mode probabilities $\{\mu_k^i\}_{i=1}^{N_m}$ must be calculated recursively from their previous values $\{\hat{x}_{k-1|k-1}^i, \Sigma_{k-1|k-1}^i, \mu_{k-1}^i\}_{i=1}^{N_m}$. A single step of the IMM filter is presented as an algorithm (Alg.: 3).

5.3.2 Adaptive Transition Probability Matrix

In the introduction Sec. 5.1.1 we discussed the importance of intent context for tracking targets with MJS properties. In order to utilize the contextual information in a IMM framework (Sec. 5.3.1), we propose to compute the mode transition probability matrix $\boldsymbol{\Pi}_k$ or π_k^{ji} recursively. In light of [68] and [67], we present the Bayesian recursion (5.20) for updating the TPMs posterior density $p(\boldsymbol{\Pi}_k)$ given the prior PDF $p(\boldsymbol{\Pi}_{k-1})$ and two observations formulated as the measurement and context likelihood functions.

$$\begin{aligned}
&p(\boldsymbol{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) \\
&\quad = \frac{p(\boldsymbol{\Pi}_k, \mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})}{p(\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})} \\
&\quad = \frac{p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) p(\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})} \\
&\quad = \frac{p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) p(\mathbf{c}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})} \\
&\qquad p(\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_{k-1}) \\
&\quad = \frac{p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}, \mathbf{c}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) p(\mathbf{c}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})} \\
&\qquad \frac{p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}) p(\mathbf{y}_{k-1}, \mathbf{c}_{k-1})}{p(\mathbf{y}_{k-1}, \mathbf{c}_{k-1})}
\end{aligned} \tag{5.19}$$

By exploiting the conditional independence between contextual c_k and the measurement y_k variables and by assuming the 1^{st} order Markov assumption recursion (5.20) simplifies to

$$\begin{aligned}
&p(\boldsymbol{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) \\
&\quad = \frac{p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1})\, p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1})\, p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})}{p(\mathbf{y}_k|\mathbf{y}_{k-1})\, p(\mathbf{c}_k|\mathbf{c}_{k-1})}.
\end{aligned} \tag{5.20}$$

In the above equation, probability $p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1})$ is referred to a measurement likelihood, $p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1})$ is context likelihood, $p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})$ is the TPM prior, terms $p(\mathbf{y}_k|\mathbf{y}_{k-1})$ and $p(\mathbf{c}_k|\mathbf{c}_{k-1})$ are measurement and context normalization factors, respectively.

Following the derivation presented in [68], the measurement and context likelihood in (5.20) are, in light of total probability theorem, defined as

$$\begin{aligned} & p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & = \sum_{j=1}^{N_m} p(\mathbf{y}_k|m^j_{k-1}, \boldsymbol{\Pi}_k, \mathbf{y}_{k-1})\; p(m^j_{k-1}|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & = \sum_{j=1}^{N_m} p(\mathbf{y}_k|m^j_{k-1}, \boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & \quad \times \sum_{i=1}^{N_m} p(m^j_{k-1}|m^i_{k-1}, \boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & \quad \times p(m^i_{k-1}|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & \approx \sum_{j=1}^{N_m} \Lambda^j_k \sum_{i=1}^{N_m} \pi^{ji}_k \mu^i_{k-1} \\ & = {\boldsymbol{\Lambda}'}_k {\boldsymbol{\Pi}'}_k \boldsymbol{\mu}_{k-1} = {\boldsymbol{\mu}'}_{k-1} \boldsymbol{\Pi}_k \boldsymbol{\Lambda}_k \end{aligned} \tag{5.21}$$

and similarly

$$\begin{aligned} & p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1}) \\ & = \sum_{j=1}^{N_m} p(\mathbf{c}_k|m^j_{k-1}, \boldsymbol{\Pi}_k, \mathbf{c}_{k-1})\; p(m^j_{k-1}|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & = \sum_{j=1}^{N_m} p(\mathbf{c}_k|m^j_{k-1}, \boldsymbol{\Pi}_k, \mathbf{c}_{k-1}) \\ & \quad \times \sum_{i=1}^{N_m} p(m^j_{k-1}|m^i_{k-1}, \boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & \quad \times p(m^i_{k-1}|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \\ & \approx \sum_{j=1}^{N_m} \Gamma^j_k \sum_{i=1}^{N_m} \pi^{ji}_k \mu^i_{k-1} \\ & = {\boldsymbol{\Gamma}'}_k {\boldsymbol{\Pi}'}_k \boldsymbol{\mu}_{k-1} = {\boldsymbol{\mu}'}_{k-1} \boldsymbol{\Pi}_k \boldsymbol{\Gamma}_k \end{aligned} \tag{5.22}$$

with the following approximation being applied in derivations (5.21) and (5.22)

$$p(\mathbf{y}_k|m^j_{k-1}, \boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \approx \Lambda^j_k \tag{5.23}$$

$$p(\mathbf{c}_k|m^j_{k-1}, \boldsymbol{\Pi}_k, \mathbf{c}_{k-1}) \approx \Gamma^j_k \tag{5.24}$$

$$p(m^i_{k-1}|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1}) \approx \mu^i_{k-1}. \tag{5.25}$$

Measurement and context likelihoods (5.21) and (5.22) assume the knowledge of TPM $\boldsymbol{\Pi}_k$ at time k which is not feasible to estimate within the same recursive cycle $(k|k-1)$. Therefore, the TPM $\boldsymbol{\Pi}_k$ is approximated with the transition probability matrix estimate from the previous cycle e.g. $\overline{\boldsymbol{\Pi}}_{k-1}$. Terms Λ^j_k, Γ^j_k and μ^i_{k-1} approximate the nonlinear dependences $p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1})$, $p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1})$ and $p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1})$ on $\boldsymbol{\Pi}_k$ by a linear function ${\boldsymbol{\mu}'}_{k-1}\boldsymbol{\Pi}_k\boldsymbol{\Lambda}_k\boldsymbol{\Gamma}_k$.

The measurement $p(\mathbf{y}_k|\mathbf{y}_{k-1})$ and context $p(\mathbf{c}_k|\mathbf{c}_{k-1})$ normalization factors are under local linear approximation (5.21) and (5.22) defined by (5.26) and (5.27), respectively.

$$\begin{aligned} & p(\mathbf{y}_k|\mathbf{y}_{k-1}) \\ & = \int p(\mathbf{y}_k|\boldsymbol{\Pi}_k, \mathbf{y}_{k-1})\; p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}) d\Pi \\ & = \int {\boldsymbol{\mu}'}_{k-1} \boldsymbol{\Pi}_k \boldsymbol{\Lambda}_k\; p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}) d\Pi \\ & = {\boldsymbol{\mu}'}_{k-1} \overline{\boldsymbol{\Pi}}_{k-1} \boldsymbol{\Lambda}_k \end{aligned} \tag{5.26}$$

$$\begin{aligned} & p(\mathbf{c}_k|\mathbf{c}_{k-1}) \\ & \quad = \int p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1})\; p(\boldsymbol{\Pi}_k|\mathbf{c}_{k-1}) d\Pi \\ & \quad = \int \boldsymbol{\mu'}_{k-1}\boldsymbol{\Pi}_k\boldsymbol{\Gamma}_k\; p(\boldsymbol{\Pi}_k|\mathbf{c}_{k-1}) d\Pi \\ & \quad = \boldsymbol{\mu'}_{k-1}\overline{\boldsymbol{\Pi}}_{k-1}\boldsymbol{\Gamma}_k \end{aligned} \tag{5.27}$$

By substituting (5.21), (5.22), (5.26), (5.27) into (5.20), we obtain the simplified Bayes rule of the form

$$\begin{aligned} & p(\boldsymbol{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1}) \\ & \quad = \frac{(\boldsymbol{\mu'}_{k-1}\boldsymbol{\Pi}_k\boldsymbol{\Lambda}_k)(\boldsymbol{\mu'}_{k-1}\boldsymbol{\Pi}_k\boldsymbol{\Gamma}_k)}{(\boldsymbol{\mu'}_{k-1}\overline{\boldsymbol{\Pi}}_{k-1}\boldsymbol{\Lambda}_k)(\boldsymbol{\mu'}_{k-1}\overline{\boldsymbol{\Pi}}_{k-1}\boldsymbol{\Gamma}_k)} p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}). \end{aligned} \tag{5.28}$$

Recursion (5.28) can be further simplified with respect to the marginal PDFs $p(\boldsymbol{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})$ for each row $i \in m_k$ of the TPM $\boldsymbol{\Pi}_k^i$, i.e.

$$\begin{aligned} p(\boldsymbol{\pi}_k^i|\mathbf{y}_k, \dots, \mathbf{c}_{k-1}) &= \int_{i=1}^{N_m} p(\boldsymbol{\Pi}_k|\mathbf{y}_k, \dots, \mathbf{c}_{k-1}) d\boldsymbol{\Pi}/d\boldsymbol{\pi}^i, \\ &= \frac{A^i}{(\boldsymbol{\mu'}_{k-1}\overline{\boldsymbol{\Pi}}_{k-1}\boldsymbol{\Lambda}_k)(\boldsymbol{\mu'}_{k-1}\overline{\boldsymbol{\Pi}}_{k-1}\boldsymbol{\Gamma}_k)}, \end{aligned} \tag{5.29}$$

where $d\boldsymbol{\Pi}/d\boldsymbol{\pi}^i = d\boldsymbol{\pi}^1, \dots, d\boldsymbol{\pi}^{i-1}, d\boldsymbol{\pi}^{i+1}, \dots d\boldsymbol{\pi}^{N_m}$. Term A^i refers to a marginal integral of the i-th row of $p(\boldsymbol{\pi}_k^i|\mathbf{y}_k, \dots, \mathbf{c}_{k-1})$ and can be solved as follows

$$\begin{aligned} A^i = & \int_{i=1}^{N_m} (\boldsymbol{\mu'}_{k-1}\boldsymbol{\Pi}_k\boldsymbol{\Lambda}_k)(\boldsymbol{\mu'}_{k-1}\boldsymbol{\Pi}_k\boldsymbol{\Gamma}_k) \\ & \quad p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})\; d\boldsymbol{\Pi}/d\boldsymbol{\pi}^i \\ = & \int_{i=1}^{N_m} \sum_{l=1}^{N_m} (\mu'^l_{k-1}\boldsymbol{\pi'}^l_k\boldsymbol{\Lambda}_k)(\mu'_{k-1}\boldsymbol{\pi'}^l_k\boldsymbol{\Gamma}_k) \\ & \quad p(\boldsymbol{\Pi}_k|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})\; d\boldsymbol{\Pi}/d\boldsymbol{\pi}^i \\ = & \sum_{l=1}^{N_m} \mu'^l_{k-1} \int_{i=1}^{N_m} \boldsymbol{\pi'}^l_k p(\boldsymbol{\pi}_k^i|\mathbf{y}_{k-1}, \mathbf{c}_{k-1})\; d\boldsymbol{\Pi}/d\boldsymbol{\pi}^i\boldsymbol{\Lambda}_k\boldsymbol{\Gamma'}_k \end{aligned} \tag{5.30}$$

By assuming the marginalization of the form

$$\begin{aligned} & \int_{l=1}^{N_m} \boldsymbol{\pi}_k^l\; p[\boldsymbol{\pi}^1, \dots, \boldsymbol{\pi}^{N_m}|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}] d\boldsymbol{\pi}^1, \dots, d\boldsymbol{\pi}^{N_m} \\ & \qquad = \overline{\boldsymbol{\pi}}_{k-1}^l\; p[\boldsymbol{\pi}^i|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}] \end{aligned} \tag{5.31}$$

expression (5.30) further simplifies to

$$\begin{aligned} A^i = & \Big[\sum_{i \neq l}^{N_m} \mu_{k-1}^l \overline{\boldsymbol{\pi}}'^l_{k-1} + \mu_{k-1}^i \boldsymbol{\pi'}^i_{k-1}\Big] \\ & \quad \times \boldsymbol{\Lambda}_k\boldsymbol{\Gamma'}_k\; p[\boldsymbol{\pi}^i|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}] \\ = & \Big[\sum_{l=1}^{N_m} \mu_{k-1}^l \overline{\boldsymbol{\pi}}'^l_{k-1} + \mu_{k-1}^i \boldsymbol{\pi'}^i_{k-1} - \mu_{k-1}^i \overline{\boldsymbol{\pi}}'^i_{k-1}\Big] \\ & \quad \times \boldsymbol{\Lambda}_k\boldsymbol{\Gamma'}_k\; p[\boldsymbol{\pi}^i|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}] \\ = & \{\boldsymbol{\mu'}_{k-1}\overline{\boldsymbol{\Pi}}_{k-1}\boldsymbol{\Lambda}_k\boldsymbol{\Gamma'}_k \\ & \quad + \mu_{k-1}^i[\boldsymbol{\pi'}^i_k - \overline{\boldsymbol{\pi}}'^i_{k-1}]\boldsymbol{\Lambda}_k\boldsymbol{\Gamma'}_k\}\; p[\boldsymbol{\pi}^i|\mathbf{y}_{k-1}, \mathbf{c}_{k-1}]. \end{aligned} \tag{5.32}$$

The result of (5.29) and (5.32) represents the polynomial approximation of posterior density $p(\boldsymbol{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})$ estimation which is the subject of discussion of Sec. 5.3.3.

5.3.3 A Quasi-Bayes TPM Estimator

The goal of the estimation process is to compute the optimal conditional probability $\hat{\boldsymbol{\Pi}}_k$ that maximize the TPM likelihood function $p(\mathbf{y}_{1:k}, \mathbf{c}_{1:k}|\boldsymbol{\Pi}_k)$, approximated by equalities (5.21) and (5.22), given the prior knowledge of the transition probabilities $\boldsymbol{\Pi}_{k-1}$, sensor measurements $\mathbf{y}_{1:k}$ and context knowledge $\mathbf{c}_{1:k}$, i.e. to compute the maximum a posteriori prediction $\hat{\boldsymbol{\Pi}}_k = \arg\max_{\Pi} p(\mathbf{y}_{1:k}, \mathbf{c}_{1:k}|\boldsymbol{\Pi}_k)$. Solutions to this estimation problem were extensively studied in [68], where the approaches based on recursive estimation of finite mixture components offered the best trade-off between accuracy and computational complexity.

Following the derivations in previous Sec. 5.3.2, the marginal TPM likelihood $p(\mathbf{y}_k, \mathbf{c}_k|\boldsymbol{\pi}^i_k, \mathbf{y}_{k-1}, \mathbf{c}_{k-1})$, where $i = 1, \ldots, N_m$, can be seen as the Gaussian mixture of the form

$$p(\mathbf{y}_k, \mathbf{c}_k|\boldsymbol{\pi}^i_k, \mathbf{y}_{k-1}, \mathbf{c}_{k-1}) = \sum_{j=1}^{N_m} \pi^{ji}_k g^{ji}_k \tag{5.33}$$

where

$$g^{ji}_k \approx \tilde{g}^{ji}_k \; p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) \; p(\mathbf{c}_k|\mathbf{c}_{1:k-1}) \tag{5.34}$$

with marginal likelihood functions

$$\tilde{g}^{ji}_k = 1 + \eta^i_k[\lambda^j_k\gamma^j_k - \boldsymbol{\pi}^i_{k-1}\boldsymbol{\Lambda}_k\boldsymbol{\Gamma'}_k]. \tag{5.35}$$

Having realized that TPM likelihood can be expressed as a mixture (5.33), the MMSE estimation of marginalized transitional probabilities $\hat{\boldsymbol{\pi}}^i$ can be conviniently transformed into a classification problem referred to as Prior Probability Estimation of finite mixtures. In general, PPE is formulated for mixture models as

$$f(\mathbf{y}_k|\boldsymbol{\Pi}) = \sum_{j=1}^{N_m} \pi^j_k \; g^j_k(\mathbf{y}_k) \tag{5.36}$$

with known mixture components $g^j_k(\mathbf{y}_k)$, where the goal is to estimate weights $\hat{\pi}^j_k$, where $\sum_{j=1}^{N_m} \hat{\pi}^j_k = 1$, for a given sequence of observations $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k$ with PDF $f(\mathbf{y}_k|\boldsymbol{\Pi})$. Jilkov at al. [68] argue, that a Quasi-Bayesian TPM estimator [131] is the solution of choice since it provides a quasi-posterior mean of the PPE, and thus meets the MMSE TPM estimation criteria perfectly. Similarly to the Gaussian mixture approximations, the underlaying idea of the Quasi-Bayesian (QB) approach is to approximate the posterior weighted sum of N_m Dirichelt distributions by a single Dirischlet distribution at each time step. Within an iteration, the QB estimator computes the recursive estimate of TPM $\hat{\boldsymbol{\pi}}^j_k$ from a Dirichelt prior $p(\boldsymbol{\pi}^j_{k-1})$ and a likelihood (5.36) via (5.37) and (5.38).

$$\hat{\boldsymbol{\pi}}^j_k = \frac{\boldsymbol{\alpha}^j_k}{k + \alpha_0} \tag{5.37}$$

$$\boldsymbol{\alpha}^j_k = \boldsymbol{\alpha}^j_{k-1} + \frac{\boldsymbol{\alpha}^j_{k-1}\boldsymbol{g}^j_k}{\sum_m^{N_m} \boldsymbol{\alpha}^l_{k-1}\boldsymbol{g}^l_k} \tag{5.38}$$

5.3.4 Context Likelihood and Hidden Markov Model

A Hidden Markov Model (HMM) is a probabilistic graphical model that represents the sequence of domain variables $\mathbf{s}_k$ with $k = 1 : K$ and their conditional dependencies in the form of a directed acyclic graph (Fig. 5.3). A domain variable, in our case the mode of a tracking filter s^i_k at time k, is seen as a node of the graph and its relation to the other nodes represented as a conditional probability $p(s^i_k|s^j_{k-1})$ of that node to its parents. In graph theory, probability $p(s^i_k|s^j_{k-1})$ is commonly referred to as the state transition matrix $\mathbf{A}_{k|k-1}$ with elements $a^{ji}_{k|k-1}$. As the vehicle navigates throughout the road network (Fig. 5.1) it creates a time sequence $\mathbf{s}_k = \{s^i_1, \ldots, s^i_k, \ldots, s^i_K\}$ consisting of events s^i_k where $i \in 1 : N_m$. At any given time k, the probability of a state s^i_k is conditionally independent from its non-descendant s^j_{k-1} as stated by the $1st$ order Markov assumption, e.g. $p(s^i_k|s^j_1, \ldots, s^j_{k-1}) \approx p(s^i_k|s^j_{k-1})$ where $j \in 1 : N_m$. It is worth noting, that both mode transition probability matrices $\boldsymbol{\pi}^{ji}_{k-1|k-1} = p(m^i_k|m^j_{k-1})$ and $p(s^i_k|s^j_{k-1})$ express the evolution of modal state from time $k-1$ to k, but they are by no means the same as former is conditional dependent on observations $\mathbf{y}_{1:k}$ and context $\mathbf{c}_{1:k}$ i.e. $p(\mathbf{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})$ and later merely depends on context $\mathbf{c}_{1:k}$ i.e. $p(\mathbf{\Pi}_k|\mathbf{c}_k, \mathbf{c}_{k-1}) \approx p(\mathbf{s}_k|\mathbf{c}_k, \mathbf{c}_{k-1})$.

Having HMM setting in mind, state sequence $\mathbf{s}_k$ is considered to be a hidden variable the knowledge of which is inferred from the sequence of contextual observations $\mathbf{c}_k$ as a joint probability distribution of a form

$$p(\mathbf{c}_k, \mathbf{s}_k) = p(\mathbf{c}_k|\mathbf{s}_k) \times p(\mathbf{s}_k|\mathbf{s}_{k-1}) = \prod_{k=1}^{K} p(c_k|s_k) \times \prod_{k=1}^{K} p(s_k|s_{k-1}). \tag{5.39}$$

We assume, that the length of contextual sequence $\mathbf{c}_k$ is equal to the length of state sequence $\mathbf{s}_k$ i.e. $\mathbf{c}_{1:K} = \mathbf{s}_{1:K}$. As visualized on the Fig. 5.3, six kinds of context variables are utilized in the presented scenario, namely $c^j \in \{$*straight*, *stop&go*, *right*, *sharp right*, *left*, *sharp left*$\}$. Context variables $\mathbf{c}_k^j$ are related to the states $\mathbf{s}_k^i$ through a likelihood function $p(c_k^j|s_k^i)$ commonly referred to as an emission probability matrix $\mathbf{B}_{k|k-1} = b_{k|k-1}^{ji}$. By knowing the joint probabilities of all observations with a particular hidden state sequences $\{p(\mathbf{c}_k, \mathbf{s}_k)\}_1^{N_s}$, the total probability of contextual observations $p(\mathbf{C}_k)$ can be computed by summing over all possible hidden state sequences

$$p(\mathbf{C}_k) = \sum_{\mathbf{c}=1}^{N_s} p(\mathbf{c}_k, \mathbf{s}_k) = \sum_{\mathbf{c}=1}^{N_s} p(\mathbf{c}_k|\mathbf{s}_k)p(\mathbf{s}_k). \tag{5.40}$$

Alternatively, the total probability of the state sequence $p(\mathbf{S}_k)$ can be represented as

$$p(\mathbf{S}_k) = \sum_{\mathbf{s}=1}^{N_s} p(\mathbf{s}_k, \mathbf{c}_k) = \sum_{\mathbf{s}=1}^{N_s} p(\mathbf{s}_k|\mathbf{c}_k)p(\mathbf{c}_k). \tag{5.41}$$

The role of HMM for context reasoning is twofold. First, HMM captures the state transition probabilities $p(\mathbf{s}_k|\mathbf{s}_{k-1})$ and individual contextual likelihoods, i.e. emission probabilities $p(\mathbf{c}_k|\mathbf{s}_k)$ from the all observed sequences $\mathbf{C}_k$ for set of possible states s_k^i during the training phase. Second, HMM infers on likelihood $p(\mathbf{c}_k|\mathbf{s}_k)$ and estimates the probability of state variables $\mathbf{s}_k$ during the evaluation phase.

The training phase of the HMM is an expectation maximization problem governed by the Baum-Welch (forward-backward) algorithm (Alg.: 4). Baum-Welch iterative search method seeks to find the local maximum of parameter space $\boldsymbol{\theta}_{k|k-1} = [\mathbf{A}_{k|k-1}, \mathbf{B}_{k|k-1}, \mathbf{s}_{k|0}]$ that maximizes the probability of contextual observations $p(\mathbf{C}_k)$ i.e. maximizes the loss function of the form $\hat{\boldsymbol{\theta}}_k = \arg\max_\theta p(\mathbf{C}_k|\boldsymbol{\theta}_{k|k-1})$ where $\mathbf{S}_k \subset \boldsymbol{\theta}_{k|k-1}$. Intuitively, the maximum likelihood estimate of the probabilities $a_{k|k-1}^{ji}$ or $b_{k|k-1}^{ji}$ can be computed by counting the number of times the transition $\xi(s_{k-1}^j \to s_k^j)$ or $\gamma(c_k^j \to s_k^i)$ between states i and j was taken, and then normalizing by the total count of all times a transition from state i was made. That would for a matrix $a_{k|k-1}^{ji}$ mean that

$$a_{k|k-1}^{ji} = \frac{\xi(s_{k-1}^j \to s_k^j)}{\sum_{s \in S} \xi(s_k^i \to s_k^s)}. \tag{5.42}$$

However, this intuition (5.42) could not be directly applied for HMM, since path taken by states $\mathbf{s}_k$ is considered to be hidden from an input sequence $\mathbf{c}_k$. For this reason, the Baum-Welch Alg. 4 iteratively estimates the expected state occupancy count γ_k^i, i.e.

$$\gamma_k^i = p(s_k^i|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1}) = \frac{p(s_k^i, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}{p(\mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}, \tag{5.43}$$

and the expected state transition count ξ_k^{ji}, i.e.

$$\xi_k^{ji} = p(s_k^i, s_{k-1}^j|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1}) = \frac{p(s_k^i, s_{k-1}^j, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}{p(\mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}, \tag{5.44}$$

from the previous values of $b_{k-1|k-1}^{ji}$ and $a_{k-1|k-1}^{ji}$, respectively in so called E-step. Probability $p(s_k^i|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1})$ in (5.43) is a result of multiplication of forward $\alpha_k^i = p(\mathbf{c}_k, s_k^i|\boldsymbol{\theta}_{k|k-1})$ and backward probabilities

$\beta^j_{k-1} = p(\mathbf{c}_k|s^j_{k-1}, \boldsymbol{\theta}_{k|k-1})$ evaluated by well-known Forward-backward algorithm. According to the Bayes theorem point of view γ^i_k in (5.43) becomes

$$\gamma^i_k = p(s^i_k|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1}) = \frac{p(\mathbf{c}_k, s^i_k|\boldsymbol{\theta}_{k|k-1})\; p(\mathbf{c}_k|s^j_{k-1}, \boldsymbol{\theta}_{k|k-1})}{p(\mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}. \tag{5.45}$$

Evaluation of the expected state transition count ξ^{ji}_k requires incorporation of prior likelihood $b^{ji}_{k-1|k-1}$ and transition matrix $a^{ji}_{k-1|k-1}$ into (5.44), which can be represented in view of forward and backward probabilities as

$$\xi^{ji}_k = p(s^i_k, s^j_{k-1}|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1}) = \frac{p(\mathbf{c}_k, s^i_k|\boldsymbol{\theta}_{k|k-1})\; p(s^i_k|s^j_{k-1})\; p(\mathbf{c}_k|s^j_{k-1}, \boldsymbol{\theta}_{k|k-1})\; p(\mathbf{c}_k|s^i_k)}{p(\mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})}. \tag{5.46}$$

In the subsequent maximization or M-step, ξ^i_k and γ^{ji}_k are used to refine $a^{ji}_{k|k-1}$ and $b^{ji}_{k|k-1}$ estimates, respectively.

By observing a particular sequence of contextual measurements $\mathbf{c}_k$, the fundamental task of HMM during the evaluation phase is to determine the most probable mode sequence $\mathbf{s}_k$, which is among all possible sequences $\mathbf{S}_k$, the underlying source of context. During the so called decoding, HMM infers the likelihood of the observation sequence for each possible hidden state $p(\mathbf{c}_k|s^i_k)$ by computing the forward probabilities of mode states $\mathbf{s}_k$, i.e. solving (5.40). By utilizing the Baum-Welch algorithm, the joint probability likelihood $p(s^i_k, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})$ is a product of forward $p(\mathbf{c}_k, s^i_k|\boldsymbol{\theta}_{k|k-1})$ and backward $p(\mathbf{c}_k|s^j_{k-1}, \boldsymbol{\theta}_{k|k-1})$ processes as indicated by the numerator of expression (5.43). Therefore (5.40) can be rewritten as

$$p(\mathbf{C}_k) = \sum_{c=1}^{N_s}\sum_{i=1}^{N_m} p(s^i_k, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1}) = \sum_{c=1}^{N_s}\sum_{i=1}^{N_m}\sum_{j=1}^{N_m} p(\mathbf{c}_k, s^i_k|\boldsymbol{\theta}_{k|k-1})\; p(\mathbf{c}_k|s^j_{k-1}, \boldsymbol{\theta}_{k|k-1}). \tag{5.47}$$

The most probable hidden state sequence $\hat{\boldsymbol{s}}_k$ of all possible sequences $\mathbf{S}_k$ is the one which maximizes the contextual observation likelihood $p(\mathbf{C}_k)$ (5.47), that is the maximum of the loss function $\hat{\boldsymbol{s}}_k = \arg\max_s p(\mathbf{C}_k|\boldsymbol{\theta}_{k|k-1})$ where $\mathbf{S}_k \subset \boldsymbol{\theta}_{k|k-1}$.

5.4 Experimental Results

In this section we discuss and address the challenges associated with synthetic dataset generation, simulation design and analyze the effect of context on the tracking performance. Our vision on the context-aware target tracking (Fig. 5.5), motivated and briefly outlined in section (5.1.1), combines the situation assessment module with the target tracking algorithm into a multi-layer data fusion architecture. The multi-target maneuvering target tracking scenario, depicted in the Fig. 5.1, can be conveniently modeled as an iterative multiple model estimation problem under motion mode and measurement origin uncertainties. As opposed to the standard IMM implementation [20], evolution of mode states is not determined by sojourn-time independent transition probability matrix but evolves dynamically over time. Adaptability is achieved by estimating the rate of chance of TPM elements by observing target originated measurements alongside with contextual information. In order to prove the benefits of context on tracking, we developed a simulation scenario (5.4.1) consisting of synthetically generated data, on which we tested the IMM target tracking algorithm, context inference and learning, and recursive TPM estimation. Generation of synthetic measurements was a necessity as the real-world datasets, which consist from continuous stream of sensory measurements and context information, are merely non-existent.

5.4.1 Simulation Design

Goal-driven context-aware object tracking relies on the assumption that target navigating throughout the network (Fig. 5.1) seeks a certain goal which can be represented by a sequence of decisions $\mathbf{s}_k$, i.e. modes of the tracking filter, from a start $k = 1$ to an end $k = K$. We perceive these events as a sequence of contextual observations $\mathbf{c}_{1:K}$, which can be interpreted as certain habits associated with individual targets, but their relation to the mode sequence $\mathbf{s}_k$ remains hidden from the observer. Relationship between a single context observation or event c^j_k, where $j \in \{$*straight*, *stop&go*, *right*, *sharp right*, *left*, *sharp left*$\}$, and an mode state s_k^i is $i \in \{CV^1, CV^2, CTR^1, CTR^2, CTL^1, CTL^2\}$, is represented by a likelihood function $p(\mathbf{c}_k|\mathbf{s}_k)$. The objective of the HMM is to infer the likelihood of observed sequence for each possible state $p(\mathbf{c}_k|s_k^i)$ by computing the forward probabilities of mode states $\mathbf{s}_k$, i.e. to solve (5.40) by using Alg. 4. In other words, conditional likelihood $p(\mathbf{c}_k|s_k^i)$ is inferred from the distribution $\sum_{c=1}^{N_s}\sum_{i=1}^{N_m} p(s_k^i, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1}) = p(\mathbf{C}_k)$, evaluated as (5.47), given the most probable hidden state sequence $\hat{\boldsymbol{s}}_k = \arg\max_s p(\mathbf{C}_k|\boldsymbol{\theta}_{k|k-1})$. Here, $\boldsymbol{\theta}_{k|k-1}$ is an emission parameter for an observation c_k^j associated with state s_k^i. Such inference is only feasible when the topology of hidden Markov model is a priori know (Fig. 5.6). Therefore, the behavior model (Fig. 5.3) for each target $\mathbf{x}^i$, i.e. the sequence of decisions $\mathbf{s}_{1:k}$, needs to be learned in a form of HMM parameters (Sec. 5.3.4), that is the number of possible mode states $\sum_{i=1}^{6} s_k^i$ where $i \in \{CV^1, CV^2, CTR^1, CTR^2, CTL^1, CTL^2\}$ (types of nodes • in Fig. 5.6), the number of possible context events $\sum_{j=1}^{6} c_k^j$, where $j \in \{$*straight*, *stop&go*, *right*, *sharp right*, *left*, *sharp left*$\}$ (types of context ○ in Fig. 5.6), the context event sequence $\mathbf{c}_{1:k}$ of target $\mathbf{x}_k^i$ from a start $k = 1$ to an end $k = K$ (length of arks ○ in Fig. 5.6), state transition probabilities $\boldsymbol{\pi}_{k|k-1}$ (edges or connections between •) and emission probabilities $\boldsymbol{\theta}_{k|k-1}$ (arks or connections from ○ to •). As illustrated in the Fig. 5.6, the inference is performed at each road intersection at the time k when the vehicle enters a validity region, i.e. a spatio-temporal context neighborhood of s_k^i. Probability that a vehicle will perform a maneuver s_k^i, in this case a turn left $i \in CTL^1$, is inferred by the Bayes theorem (5.46) on the basis of a context likelihood $p(s_k^i, s_{k-1}^j|\mathbf{c}_k, \boldsymbol{\theta}_{k|k-1})$ and a priori state transition count $p(s_{k-1}^i, s_{k-2}^j, \mathbf{c}_k|\boldsymbol{\theta}_{k|k-1})$. It is not uncommon in the field of plan recognition, that probabilities on target actions s_k^i are inferred at every time step k, not only at the times when a vehicle enters some validity region of the context c_k^j, which lead to development of large variety of dynamic Bayesian networks (DBN) implementations. Reasoning schemes capable of estimating the start and end times of events, e.g. Abstract Hidden Markov Models [22], at different layers of abstraction, e.g. Hierarchical Hidden Markov Models [24], are indeed very appealing, however, the usage of HMM is well justified for cases were the area of event occurrence, i.e. location of a road intersection in the global coordinates, is precisely defined.

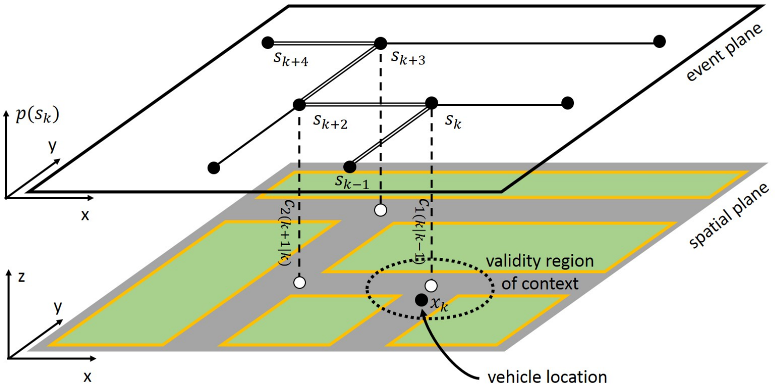


Figure 5.6: Hidden Markov model topology and spatio-temporal context neighborhood.

Till this point we have only discussed the inference of a semantical meaning of the context. However, in order to utilize CI for the purpose of transition probability matrix (TPM) estimation (5.20), likelihood $p(\mathbf{c}_k|\mathbf{s}_k)$ has to be mapped from an event plane into a spatio-temporal plane (Fig. 5.6) to form $p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1})$, i.e. context feature space needs to be extended for an attribute of location. Observe, that $p(\mathbf{c}_k|s_k^i)$, which is in fact an independent mode observation likelihood $p(\mathbf{c}_k|m_k^i)$, can be directly applied in the mode prediction update (5.9) of the IMM filter. However, this strategy is very risky, and thus not recommended, as any uncertainties in a HMM mode estimate will result in track losses and will significantly degrade the IMM performance due the presence of a mode bias. Our goal is to find the mapping of a form

$$p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1}) = f(p(\mathbf{c}_k|\mathbf{s}_k), \mathbf{x}_k). \tag{5.48}$$

For clarity, consider a context validity region (Fig. 5.6) depicted on the Fig. 5.7 for an expected value of mode likelihood $p(c_k^{straight}|s_k^{CV^1})$ (blue) and $p(c_k^{left}|s_k^{CT^1})$ (red). The Scenario represents a state

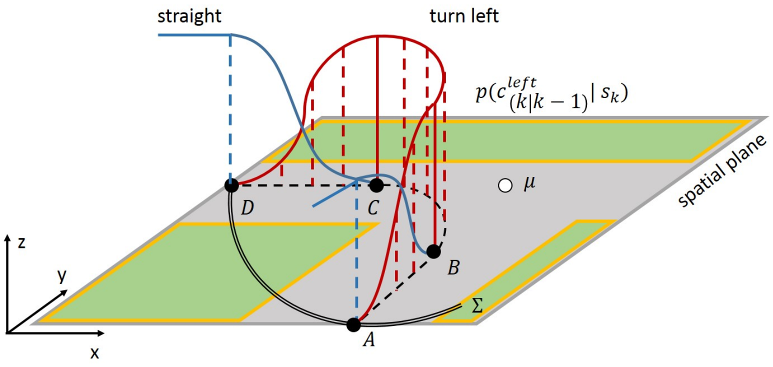


Figure 5.7: Illustration of the context relevance function during the mode transition.

transition sequence $s^i_{1:3}$ for $i \in \{CV^1, CTL^1, CV^1\}$ which is inferred on basis of a contextual likelihood $p(c^j_k|s^i_k)$ corresponding to event elements $c^j_{1:3}$ for $j \in \{straight, left, straight\}$. In order to find the mapping (5.48), we assumed that the relevance of the context event should be a quadratic function of the distance. It is further assumed, that context is observed from the target $\mathbf{x}_k$ perspective and only valid within some spatio-temporal neighborhood of the context focal point. Arguably, the simplest model of the relevance is a Gaussian distribution defined by a focal point μ and by standard deviation Σ of a spatio-temporal neighborhood. For the purpose of this study, the likelihood $p(c^j_k|s^i_k)$ for context $j = left$ is constituted from three segments: leading edge $|AB|$, steady state $|BC|$, and trailing edge $|CD|$. Context relevance for $f(p(\mathbf{c}_k|\mathbf{s}_k), \mathbf{x}_k)$ is defined as a weighting factor $\mathbf{w}_k$ as

$$\mathbf{w}_k \propto \begin{cases} \mathcal{N}\big(\mathbf{x}_k; \mu, (-3\Sigma, 0.7\Sigma) \cap (0.7\Sigma, 3\Sigma)\big) & for \quad |AB| \; and \; |CD|, \\ k\mathbf{x}_k, \;\; k = 0.9 & for \quad |BC|, \end{cases} \tag{5.49}$$

where Σ is a design parameter (in our experiments is chosen to be $\Sigma = 8^2$), depending on the size of an intersection. The choice not to model segment $|BC|$ as a Gaussian distribution is to avoid huge spikes in the likelihood. Modeling the transition from continuous velocity model $m^{CV^1}_{k-1}$ to coordinated turn model $m^{CT^1}_k$ for segment $|AB|$ and vice versa for $|CD|$ is of crucial importance in the real tracking scenarios. If the mode transition probability $p(m^{CT^1}_k|m^{CV^1}_{k-1})$ of the jump from $m^{CV^1}_{k-1}$ to $m^{CT^1}_k$ is too low, i.e. $\boldsymbol{\pi}^{ji}_{k-1|k-1} = p(m^{CT^1}_k|m^{CV^1}_{k-1}) \approx 0$, the IMM filter will very unlikely perform such a jump. In such a case, the chance of a track loss is high. The underlaying principle of any adaptive Transition Probability Matrix (TPM) estimation is based upon the count of transitions between the modes. Unsurprisingly, in real tracking scenarios certain transitions can be quite frequent and others very sparse. In anticipation for these rare cases, it is better to increase the mode ambiguity prior to an event thus ‘preparing’ the filter for such unlikely mode transition, i.e. gradually decrease and increase the probability of $m^{CV^1}_{k-1}$ and $m^{CT^1}_k$ (Fig. 5.7), respectively. In conclusion, the context likelihood model used in this study is formulated on the basis of (5.48) and (5.49) as

$$p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1}) = \mathbf{w}_k \, p(\mathbf{c}_k|\mathbf{s}_k). \tag{5.50}$$

Two types of the synthetic detests will be considered in the evaluation of the proposed methodology: the Markovian Jump Linear System process (MJLS) and more realistic semi-Markov process depicted in the Fig. 5.4. Arguably, the MJLS (5.51) is considered to be the benchmark dataset for performance evaluation of the TPM estimators [68], [110].

$$\begin{aligned} \mathbf{x}_k =& \mathbf{A}_k(m_k)\mathbf{x}_{k-1} + \mathbf{B}_k(m_k)\mathbf{u}_k + \mathbf{v}_k, \\ \mathbf{y}_k =& \mathbf{C}_k(m_k)\mathbf{x}_k + \mathbf{D}_k(m_k)\mathbf{u}_k + \mathbf{w}_k. \end{aligned} \tag{5.51}$$

The parameters of the model are defined on basis of (2.1, 2.2) and the mode state m_k is a single element of a Markov chain sequence $\mathbf{m}^i_k \in \{m^i_1, \ldots, m^i_K\}$ of length K for $\forall i \in 1, \ldots, N^i$ with the number of modes N^i. The Markov sequence is uniquely defined by a transition probability matrix $\boldsymbol{\Pi}_k = \boldsymbol{\pi}^{ji}_{k-1|k-1}$.

The hidden Markov model, e.g. Fig. 5.3, was employed in the process of semi-Markov sequence generation $\mathbf{m}_k^i \in \{m_1^i, \dots, m_K^i\}$ of length K for $\forall i \in 1, \dots, N^i$. A target specific semi-Markov sequence is uniquely defined by a transition $\boldsymbol{\pi}_{k-1|k-1}^{ji}$ and emission probability matrix $\boldsymbol{\theta}_{k|k-1}^{ji}$. As opposed to MJLS, the HMM sequence is defined only at each intersection Fig. 5.6, as the mode state $m_k^i = s_k^i$ can be inferred only on the basis of contextual information c_k^j (5.46), and therefore it is significantly shorter and with the less amount of jumps. The resulting dataset is a sequence of trajectory segments, where each segment is generated in accordance with the MJLS (5.51) process. Consider a generic trajectory segment (Fig. 5.8). Each segment is $20s$ long and constitutes from a single context event c_k^j, where $j \in \{$*straight*, *stop&go*, *right*, *sharp right*, *left*, *sharp left*$\}$. At the beginning, i.e. at the time k_1, it is assumed that vehicle is moving according to CV model, i.e. $s_k^{CV^1}$, which is reflected by a context observation $c_k^{straight}$. At random time $k_2 \in \{2, 15\}$ a context event occurs c_k^j which lasts $k_3 + K_m$ long, where K_m is random such that $k_3 \leq 18s$ at most, before the event turns back to straight mode again. The resulting trajectory is a semi-Markov process with mode specific HMM sequence which tries to mimique a real target behavior in a real environment.

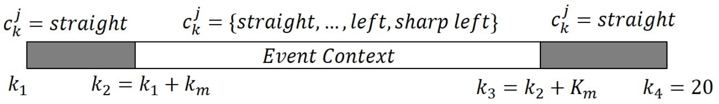


Figure 5.8: Trajectory segment.

A hypothetical scalar jump Markov linear system with three models was used in the initial set of experiments in order to compare the convergence of the Quasi-Bayesian estimator with existing works [67] and [110]. The parameters for the scalar system (5.51) read as follows

$$\begin{aligned} \mathbf{A}_k(1) =& 0.8, \quad \mathbf{A}_k(2) = 0.9, \quad \mathbf{A}_k(3) = 1.0; \\ \mathbf{B}_k(i) =& 1.0, \quad \text{for } i = 1, 2, 3; \\ \mathbf{C}_k(1) =& 1.0, \quad \mathbf{C}_k(2) = 2.0, \quad \mathbf{C}_k(3) = 4.0; \\ \mathbf{D}_k(i) =& 1.0, \quad \text{for } i = 1, 2, 3. \end{aligned} \tag{5.52}$$

Noise properties entering the scalar state space model (5.52) are defined as the Gaussian noises of system uncertainty $v_k \approx \mathcal{N}(v_k; 0, 2^2)$, measurement uncertainty $w_k \approx \mathcal{N}(w_k; 0, 2^2)$ and initial state ambiguity $x_0 \approx \mathcal{N}(x_0; 0, 1)$. The true TPM $\boldsymbol{\Pi}_k$ of mode sequence $\mathbf{m}_k^i$ and the initial TPM estimate $\hat{\boldsymbol{\Pi}}_0$ are selected as

$$\boldsymbol{\Pi}_k = \begin{bmatrix} 0.20 & 0.40 & 0.40 \\ 0.25 & 0.50 & 0.25 \\ 0.10 & 0.10 & 0.80 \end{bmatrix}, \quad \hat{\boldsymbol{\Pi}}_0 = \begin{bmatrix} 0.33 & 0.33 & 0.34 \\ 0.33 & 0.33 & 0.34 \\ 0.33 & 0.33 & 0.34 \end{bmatrix}. \tag{5.53}$$

In the second set of experiments, nearly constant velocity (5.54) and coordinated turn (5.55) models were employed in the synthetic dataset generation and the estimation process. 2D state space models are defined in the local level frame and read as follows

$$\begin{bmatrix} p_{k+1}^x \\ p_{k+1}^y \\ v_{k+1}^x \\ v_{k+1}^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}, \tag{5.54}$$

and

$$\begin{bmatrix} p_{k+1}^x \\ p_{k+1}^y \\ v_{k+1}^x \\ v_{k+1}^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & \sin\left(\dfrac{\omega_k T}{\omega_k}\right) & -\dfrac{1-\cos(\omega_k T)}{\omega_k} \\ 0 & 1 & \dfrac{1-\cos(\omega_k T)}{\omega_k} & \sin\left(\dfrac{\omega_k T}{\omega_k}\right) \\ 0 & 0 & \cos(\omega_k T) & -\sin(\omega_k T) \\ 0 & 0 & \sin(\omega_k T) & \cos(\omega_k T) \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}. \tag{5.55}$$

Uncertainties in the state space model (5.54) and (5.55) are generated by a zero mean Gaussian distributed acceleration noise $\mathbf{a}_k = \mathcal{N}(0, \sigma_a^2)$. Variance σ_a is for low and high noise nearly constant velocity models CV^1 and CV^2 set to $\sigma_a = 0.4ms^{-2}$ and $\sigma_a = 4ms^{-2}$, respectively. Four coordinated turn models for low CTX^1 or high kinematic motion CTX^2 and right CTR^x or left hand side turns CTL^x are considered in the simulation (Fig. 5.3). While coordinated turn models share the same noise acceleration variance $\sigma_a = 0.04rads^{-1}$ the parameter ω_k is used for differentiation between individual modes, i.e. $\omega_k(CTR^1) = -0.08rads^{-1}$, $\omega_k(CTR^2) = -0.20rads^{-1}$, $\omega_k(CTL^1) = 0.08rads^{-1}$ and $\omega_k(CTL^2) = 0.08rads^{-1}$. It is assumed, that ω_k is a priori known parameter and therefore angular velocity does not need to be estimated as a state. By keeping the size of the state vector constant across all models, a nontrivial problem of state estimation within variable structures spaces was avoided at the expanse of an increased number of modes. Initial conditions for the system are set to $\mathbf{x}_0 = [5km\ 5km\ 25ms^{-1}\ 25ms^{-1}]^T$. A simplified sensor model consisting directly from position measurements is utilized in the study.

$$\begin{bmatrix} y_k^x \\ y_k^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_k^x \\ p_k^y \\ v_k^x \\ v_k^y \end{bmatrix} + \begin{bmatrix} w_k^x \\ w_k^y \end{bmatrix} \tag{5.56}$$

Uncertainties in observations are modeled by a white noise $\mathbf{w}_k \approx \mathcal{N}(0, \sigma_w = 10m)$ both for x and y components. This model is used for generating the target originated measurements with detection probability $P_D = 0.9$. Generated measurements are added to 200 sets of clutter by selecting a random target start time t_s distributed uniformly over the interval $t_s \in (0.50)$ seconds. Clutter measurements are Poisson distributed with the rate $\beta_{FA} = 1.10^{-7}$. The spatial distribution of the false alarms is uniform in the region $x, y \in (0, 10km)$.

The implemented tracking algorithm follows the logic and advices published in [13] summarized in fundamentals (Sec. 2.6). Rectangular gates are used for scalar scenario with $\kappa \geq 3$ defined as follows

$$(y_k^x - \hat{y}_{k|k-1}^x) \lesseqgtr \kappa\sigma_{k|k-1}^x \qquad (y_k^y - \hat{y}_{k|k-1}^y) \lesseqgtr \kappa\sigma_{k|k-1}^y. \tag{5.57}$$

Ellipsoidal gating of the measurements $\mathbf{y}_k$ with the current initiators $\mathbf{y}_{k|k-1}$ is evaluated against gating threshold γ_G.

$$(\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1})^T \mathbf{S}_{k|k-1}^{-1} (\mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1}) \tag{5.58}$$

Here the $\mathbf{S}_{k|k-1}^{-1}$ is an innovation covariance and γ_G is χ^2 distributed given the probability of gating $P_G = 0.9$ and number of measurements $\mathbf{y}_k$. In the cases when measurements fall into the gate $\{y_k^i\}_{i=1}^{n_y}$ we evaluate them against hypotheses θ_0 and θ_i.

$$\begin{aligned} \theta_0 &= \text{All measurments } \{y_k^i\}_{i=1}^{n_y} \text{ are FA} \\ \theta_i &= \text{Measurment } \{y_k^i\}_{i=1}^{n_y} \text{ belongs to the target, rest are FA} \end{aligned}$$

Probabilities $p(\theta_i | \{y_{0:k}^i\}_{i=1}^{n_y})$ associated with hypothesis θ_i are computed as

$$p(\theta_i | \{y_k^i\}_{i=1}^{n_y}) \propto \begin{cases} (1 - P_D P_G)\beta_{FA}^{n_y} & \text{if} \quad i = 0, \\ p(y_k^i | \{y_{0:k-1}^i\}_{i=1}^{n_y}) P_D \beta_{FA}^{n_y} & \text{otherwise.} \end{cases} \tag{5.59}$$

Here the probability $p(y_k^i | \{y_{0:k-1}^i\}_{i=1}^{n_y})$ is defined as $\mathcal{N}(\mathbf{y}_k; \mathbf{y}_{k|k-1}, \mathbf{S}_{k|k-1})$.

5.4.2 Results and Discussion

This section demonstrates the performance of the quasi-Bayesian estimator on three simulated scenarios in the cases of absence or presence of contextual information. The first scenario was directly adopted from [110], where the hypothetical scalar jump Markov linear system was considered, i.e. (5.52) and (5.53). The JMLS, defined by (5.51), generates a process where mode changes occur at each time step k. For this reason, the JMLS process is quite challenging to track with conventional IMM techniques, but makes it ideal for testing the performance of TPM estimators. Furthermore, scalar JMLS scenario helps in establishing of the performance boundaries of on-line TPM estimators. In addition, convergence properties of a suboptimal quasi-Bayesian technique [68] and a maximum likelihood estimator [110] will be further discussed.

Note, that the convergence of TPM estimators is highly dependent on the amount of jumps observed between individual modes. Also note, that contextual information was not used in the first set of experiments. In order to provide a reader with an intuition of the first set of experiments, a single $100s$ long example of the JMLS process, modeled by (5.51, 5.52, 5.53), with corresponding measurements is presented on the Fig. 5.9. The tracking performance of the standard IMM filter (Alg.: 3) and the IMM augmented with an on-line mode transition probability matrix (TPM) estimation, discussed in (5.3.3), is reported on the Fig. 5.10 left and the Fig. 5.10 right, respectively. Relative errors of aforementioned filtering solutions w.r.t. the true trajectory (Fig. 5.9) are shown in the Fig. 5.11. In this particular run, utilization of the estimated transition probability matrix $\hat{\mathbf{\Pi}}_k$, as opposed to the initial $\mathbf{\Pi}_0$ (5.53), yields a 15% performance gain. This is confirmed by the mean error values of the IMM $\bar{x} = 2.5964m$ and IMM-QB adaptive $\bar{x} = 2.2065m$ estimates reported from the simulation run shown in the Fig. 5.12. Most importantly, the estimate of TPM matrix (5.53) elements $\hat{\pi}^{ji}_{k-1|k-1}$ is reported on the Fig. 5.13. It can be observed from the Fig. 5.13, that some non-insignificant biases exist in the estimated probabilities like the case in [68] and [110]. These can be attributed to the approximations involved in the derivation of measurement likelihood of the TPM

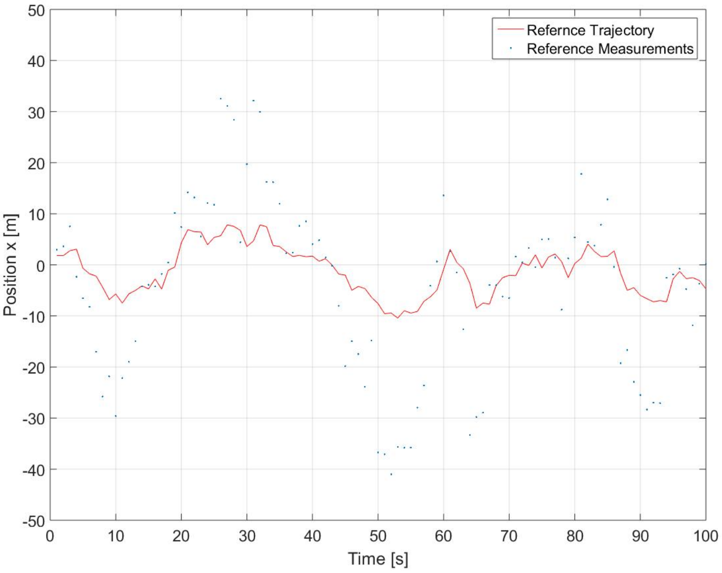


Figure 5.9: Scalar JMLS process with corresponding reference measurements.

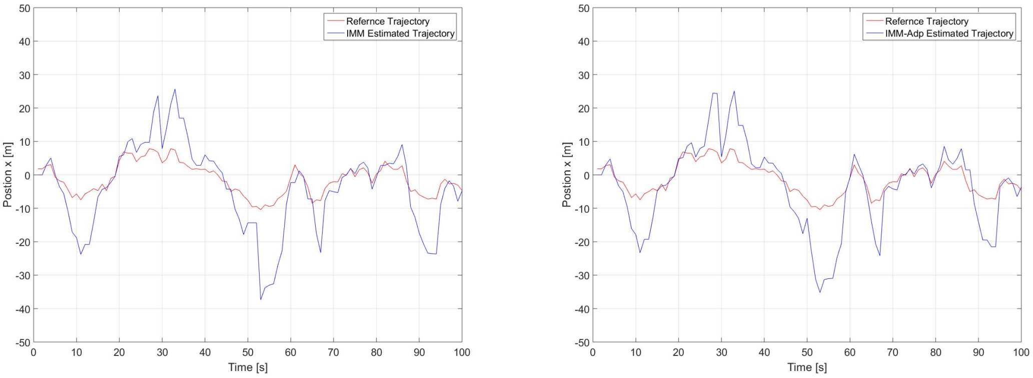


Figure 5.10: IMM (left) and IMM-QB adaptive (right) estimates of the scalar JMLS process.

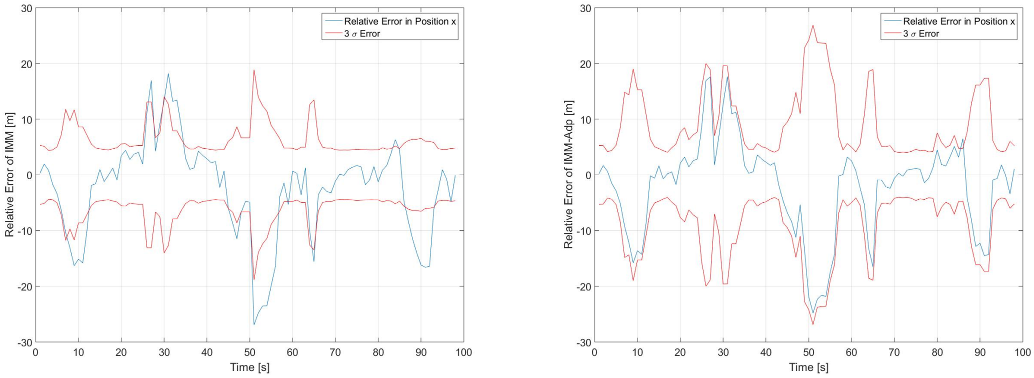


Figure 5.11: Relative errors of IMM (left) and IMM-QB adaptive (right) estimates of the scalar JMLS process.

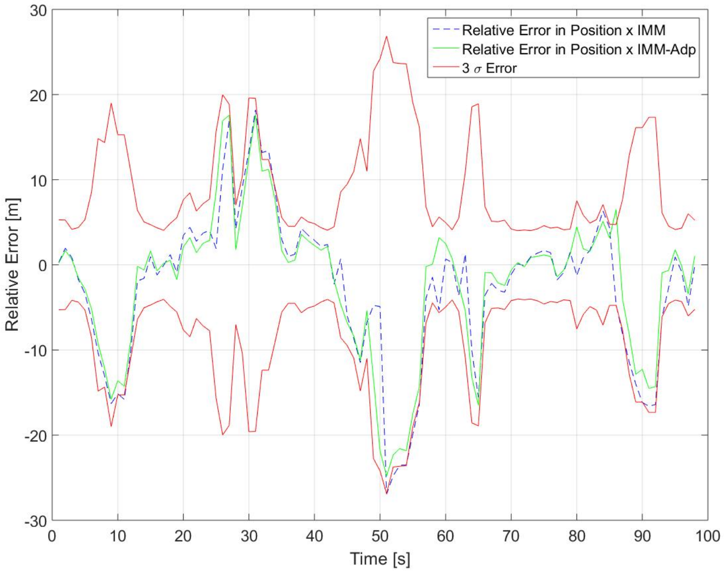


Figure 5.12: Comparison of the relative errors of IMM and IMM-QB adaptive estimates of the scalar JMLS process.

matrix $p(\mathbf{\Pi}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{c}_k, \mathbf{c}_{k-1})$ expressed by (5.29) and (5.32), mixing approximation in the IMM (5.16) and convergence properties of the quasi-Bayesian estimators [67]. As pointed out by Orguner et. al. [110], the mixing step of the IMM, which combines mixture components (5.15) into a single Gaussian (5.16) by weighting (averaging) their means (5.17) and covariances (5.18), can result in biases in the estimation by causing the position of the dominant mode in the mixture to shift slightly towards non-dominant modes. It is worth noting, that the maximum likelihood TPM estimator, proposed in [110], is an optimal solution to the JMLS estimation problem, and therefore significantly more accurate than the QB-estimator, which is regarded only as a sub-optimal but nearly $16\times$ less computationally intense [68]. Finally, the results of 100 Monte Carlo simulation runs of tracking a scalar target following the JMLS process is shown in the Fig. 5.14 and summarized in the Tab. 5.1. Based on the results, utilizing an adaptive estimate of the TPM matrix improves the performance of the IMM filter by roughly 23.19%.

The second set of experiments assumes a target that is following the hypothetical JMLS process (5.52) in a 2D scenario. Target kinematics is modeled in accordance with the nearly constant veloc-

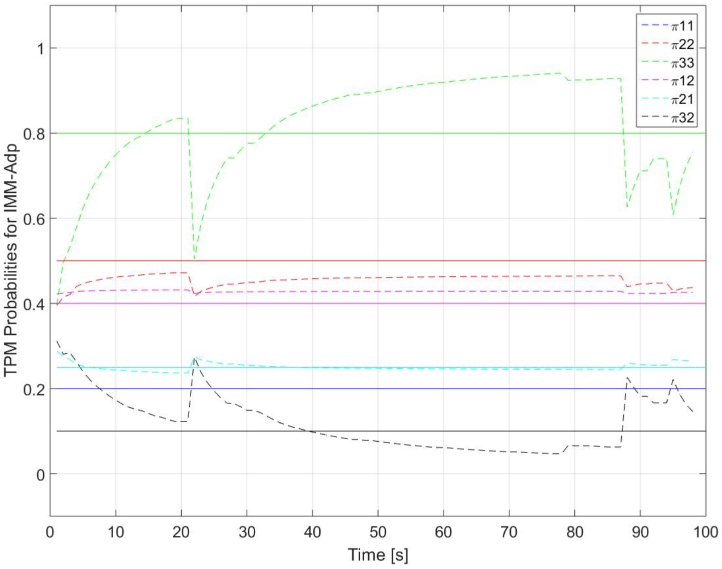


Figure 5.13: IMM-QB adaptive estimates of TPM elements $\hat{\pi}^{ji}_{k-1|k-1}$ for the scalar JMLS process.

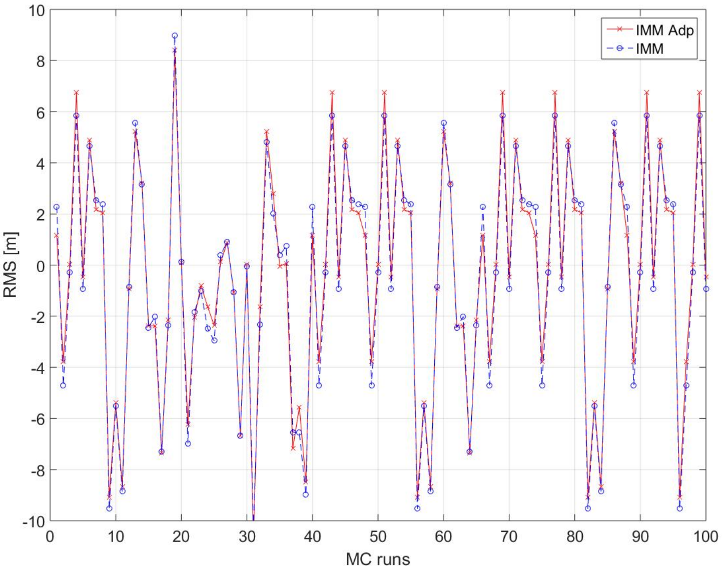


Figure 5.14: Monte Carlo simulation of the tacking of a target governed by scalar JMLS process.

Target Tracking	Relative Error Mean [m]	Relative Error Std. [m]
IMM	-0.4554	4.5588
IMM-QB Adap.	-0.3498	4.4721

Table 5.1: Comparison of the tacking of a target governed by scalar JMLS process

ity model (5.54), i.e. CV^2 with $\sigma_a = 4ms^{-2}$, and coordinated turn models (5.55) for clockwise and $\omega_k(CTR^2) = -0.20rads^{-1}$ counter-clockwise rotations $\omega_k(CTL^2) = 0.20rads^{-1}$. A simplified sensor model is considered, which consists directly from position observations (4.18) corrupted only by a white noise $\mathbf{w}_k \approx \mathcal{N}(0, \sigma_w = 10m)$. Modes of the target $\hat{s}^i_{1:k}$ are modeled as a Markov sequence $\hat{\mathbf{s}}_{1:k}$ generated on the basis of the reference $\hat{\mathbf{\Pi}}_k$, defined by (5.53). In this scenario, contextual information, in a form of a likelihood $p(\mathbf{c}_k|\mathbf{\Pi}_k, \mathbf{c}_{k-1})$ expressed as (5.50), is considered in aiding the TPM estimation process. Here, the context relevance, i.e. weighting factor $\mathbf{w}_k$, is a Gaussian distributed state dependent function $\mathcal{N}\big(\mathbf{x}_k; \mu, \Sigma\big)$ without any transients (Fig. 5.7). This is due to the fact, that mode state $\mathbf{s}_{1:k}$ of the JMLS process $\mathbf{s}_{1:k} \in \{s^i_1, \dots, s^i_k\}$, where $i \in \{CV^2, CTR^2, CTL^2\}$, jumps between modes every time step k, and thus transient models are not required. This time, however, a total number of 50 JMLS sequences $\{\mathbf{s}_{1:k}\}^{50}_1 = \mathbf{S}_{1:k}$ were generated where the true sequence $\hat{\mathbf{s}}_{1:k}$ is a subset of $\mathbf{S}_{1:k}$, i.e. $\hat{\mathbf{s}}_{1:k} \subset \mathbf{S}_{1:k}$. Set $\mathbf{S}_{1:k}$ was utilized in the event model learning. The hidden Markov model is learned (Sec. 5.3.4) form the parameters specified as: the number of possible states $\sum^3_{i=1} s^i_k$ where $i \in \{\ CV^2, CTR^2, CTL^2\}$, the number of possible context events $\sum^3_{j=1} c^j_k$ where $j \in \{\ \textit{straight}, \textit{right}, \textit{left}\ \}$, the context event sequences $\{\mathbf{c}_{1:k}\}^{50}_1 = \mathbf{C}_{1:k}$, the initial state transition matrix $\mathbf{\Pi}_0$ and the initial emission probability matrix $\mathbf{\Theta}_0$. Note, that set of context sequences $\mathbf{C}_{1:k}$ needs to be generated for the mode sequences $\mathbf{S}_{1:k}$ given the emission probability matrix $\mathbf{\Theta}_k$, in this case chosen at random. Such a mapping is computed as the probability of seeing observations $\mathbf{c}_k = \{c^j_1, c^j_2, \dots, c^j_K\}$ while being in state i at time k, i.e $\alpha^i_k = p(\mathbf{c}_k, s^i_k|\boldsymbol{\theta}_{k|k-1})$, evaluated by Alg. 4. A single sample of the 2D JMLS process (Fig. 5.15), defined by (5.52), is presented as a reference trajectory and measurements. The tracking performance of the standard IMM filter (Alg.: 3) is reported in the Fig. 5.16 left. Performance of IMM filters augmented with an on-line mode transition probabilities estimation process, for cases where contextual observations are available (5.3.3), are shown in the Fig. 5.16 right and the Fig. 5.16 bottom, respectively. Relative errors of aforementioned filtering solutions w.r.t. the true trajectory (Fig. 5.15) are reported in the Fig. 5.17. In this particular run, the IMM-QB adaptive tracker outperformed the conventional IMM by a rather small margin of 2.72% as the mean error value of the adaptive solution reached $\bar{x}^{IMM-QB} = 9.7232m$, while the IMM mean error was contained at $\bar{x}^{IMM} = 9.9959m$. However, inclusion of the contextual knowledge inferred by the HMM gave the IMM-QB-HMM adaptive estimator a significant advantage over its counterparts. A performance gain of 9.64% and 6.92% was reached over the IMM and the IMM-QB adaptive solution, respectively, while the mean error of $\bar{x}^{IMM-QB-HMM} = 9.0326m$ was observed. This performance gain can be credited to the context which boosted the convergence and reduced the steady state error of the estimate of TPM elements

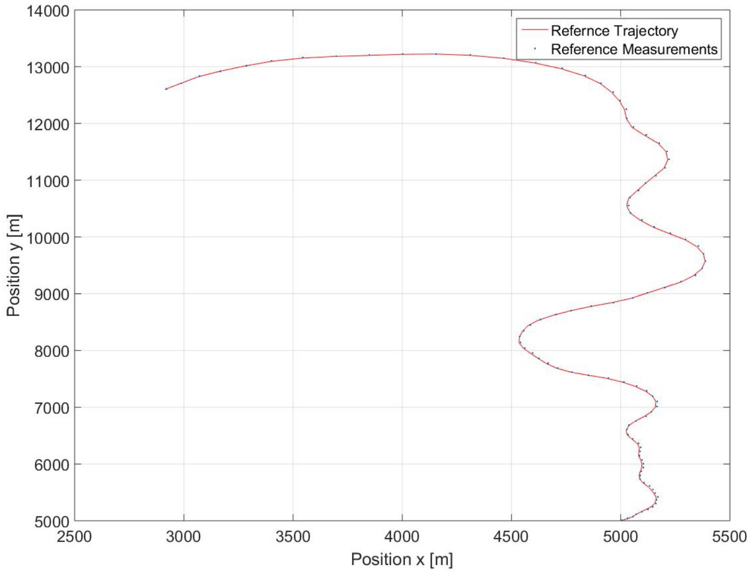


Figure 5.15: JMLS process with corresponding reference measurements.

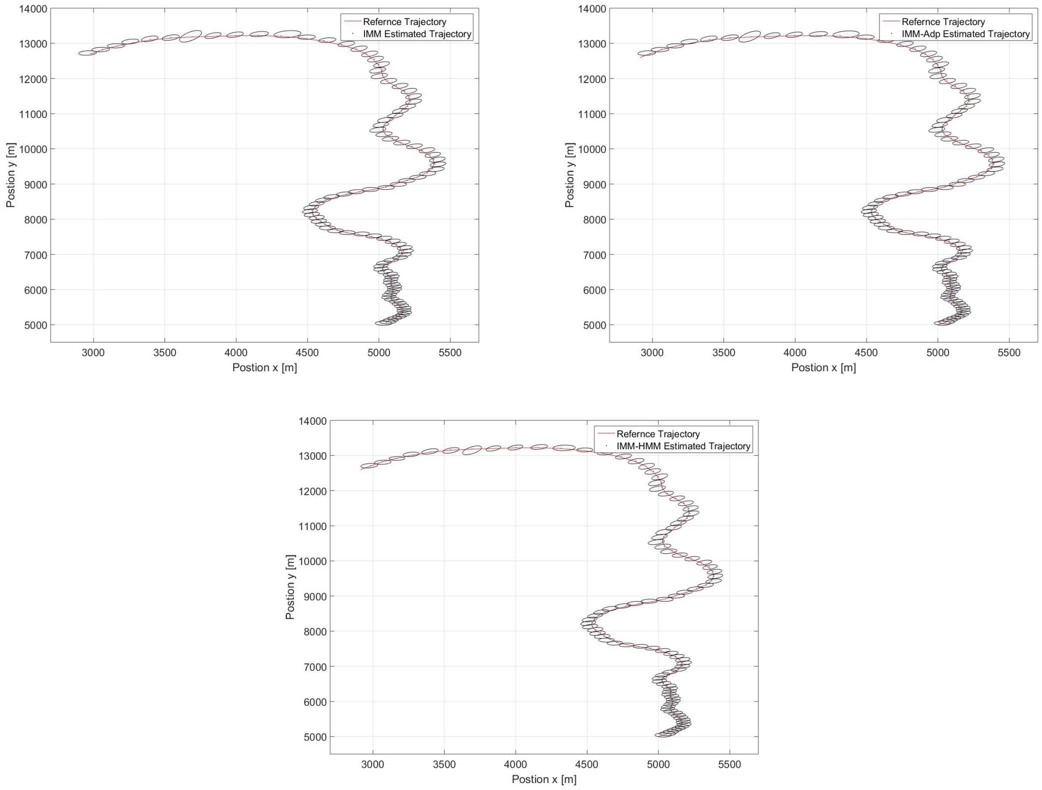


Figure 5.16: IMM (left), IMM-QB adaptive (right) and IMM-QB-HMM adaptive (bottom) estimates of the JMLS process.

(Fig. 5.19). Observe, that the informativeness of the likelihood (5.36), a conditional distribution of two independent sources $\boldsymbol{\Lambda}_k$ and $\boldsymbol{\Gamma}_k$, is a key factor in the prior probability estimation process (PPE) of any TPM matrix represented as a finite mixture. The QB estimator computes a TPM estimate $\hat{\boldsymbol{\pi}}_k^j$ recursively from a Dirichelt prior $p(\boldsymbol{\pi}_{k-1}^j)$ and a likelihood (5.36) (5.37) and (5.38). In this recursion, the combined likelihood $\boldsymbol{\Lambda}_k \boldsymbol{\Gamma}_k$ is considered as a weighting factor, and thus any increase of its informativeness has a immediate impact on the estimator performance. This is demonstrated in the Fig. 5.25, where the combination of the measurement $\boldsymbol{\Lambda}_k$ and the context likelihood $\boldsymbol{\Gamma}_k$ resulted into a more informative observation source. Similarly to the first scenario, the results of 100 Monte Carlo simulation runs of 2D tracking a target following the JMLS process is shown in the Fig. 5.20 and summarized in the Tab. 5.2. Based on the results, utilizing an adaptive estimate of the TPM matrix improves the performance of the IMM filter by roughly 13.98% while context is present and by approximately 5.46% while context is absent. This scenario was rather difficult to track, where the occurrence of track losses was quite frequent, which resulted into quite large relative errors reported in the Tab. 5.2.

Target Tracking	Relative Error Mean [m]	Relative Error Std. [m]
Kalman filter	77.8291	32.0072
IMM	58.3377	40.6348
IMM-QB Adap.	55.1503	41.5219
IMM-QB-HMM Adap.	50.1785	37.2181

Table 5.2: Comparison of the tacking of a target governed by JMLS process

In the last scenario, tracking of a target following the semi-Markov process (Fig. 5.4) is examined

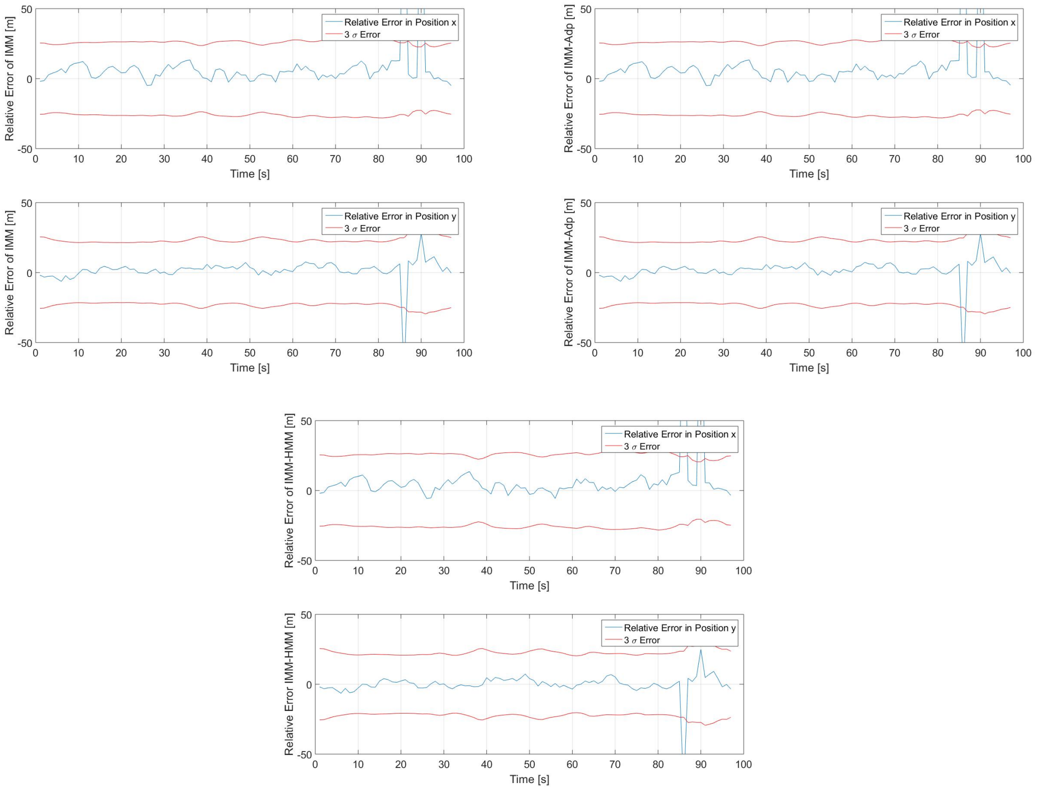


Figure 5.17: Relative errors of IMM (left), IMM-QB adaptive (right) and IMM-QB-HMM adaptive (bottom) estimates of the JMLS process.

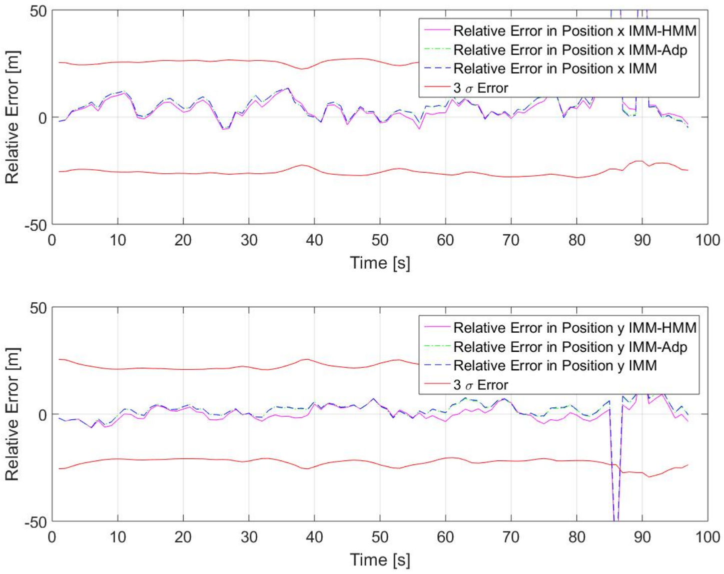


Figure 5.18: Comparison of the relative errors of IMM, IMM-QB adaptive, IMM-QB-HMM adaptive estimates of the JMLS process.

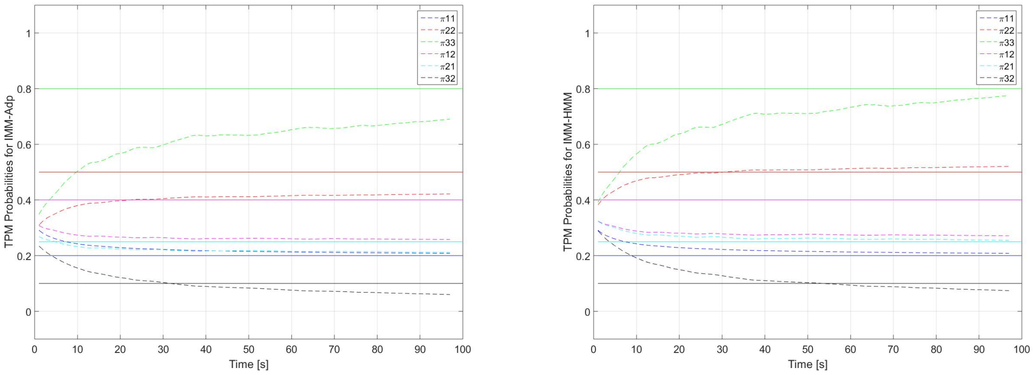


Figure 5.19: IMM-QB adaptive (left) and IMM-QB-HMM (right) estimates of TPM elements $\hat{\pi}^{ji}_{k-1|k-1}$ for the JMLS process.

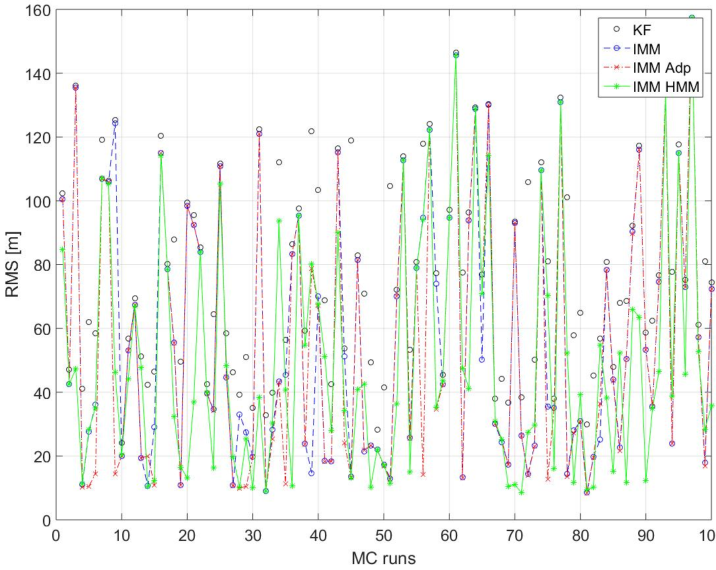


Figure 5.20: Monte Carlo simulation of the tacking of a target governed by JMLS process.

and discussed. In this approximation of the real target tracking problem we demonstrate the potential and usefulnesses of the goal-driven context-aware reasoning. The process of semi-Markov sequence generation $\mathbf{m}^i_k \in \{m^i_1, \ldots, m^i_K\}$ of length K for $\forall i \in 1, \ldots, N^i$ is governed by the hidden Markov model, depicted in the Fig. 5.3. A target specific semi-Markov sequence is uniquely defined by a transition $\boldsymbol{\pi}^{ji}_{k-1|k-1}$ and emission probability matrix $\boldsymbol{\theta}^{ji}_{k|k-1}$, chosen at random for the each Monte Carlo experiment. The resulting dataset is a sequence of trajectory segments (Fig. 5.8), where each segment is generated in accordance with the MJLS (5.51) process. Target kinematics is modeled in accordance with the nearly constant velocity models (5.54) and coordinated turn models (5.55), i.e. s^i_k where $i \in \{CV^1, CV^2, CTR^1, CTR^2, CTL^1, CTL^2\}$. Similarly to the second scenario a simplified sensor model (4.18) is utilized for the data generation. Context events c^j_k, where $j \in \{$*straight*, *stop&go*, *right*, *sharp right*, *left*, *sharp left*$\}$, are modeled with accordance to the concepts described in Fig. 5.6 and Fig. 5.7 resulting in a context likelihood model $p(\mathbf{c}_k|\boldsymbol{\Pi}_k, \mathbf{c}_{k-1}) = \mathbf{w}_k\, p(\mathbf{c}_k|\mathbf{s}_k)$ (5.50). An example of a 300s long semi-Markov trajectory where a target crossed a total of 25 intersections, i.e. 25 contextual events were observed, is depicted in the Fig. 5.21. The tracking performance of the standard IMM filter (Alg. 3) is

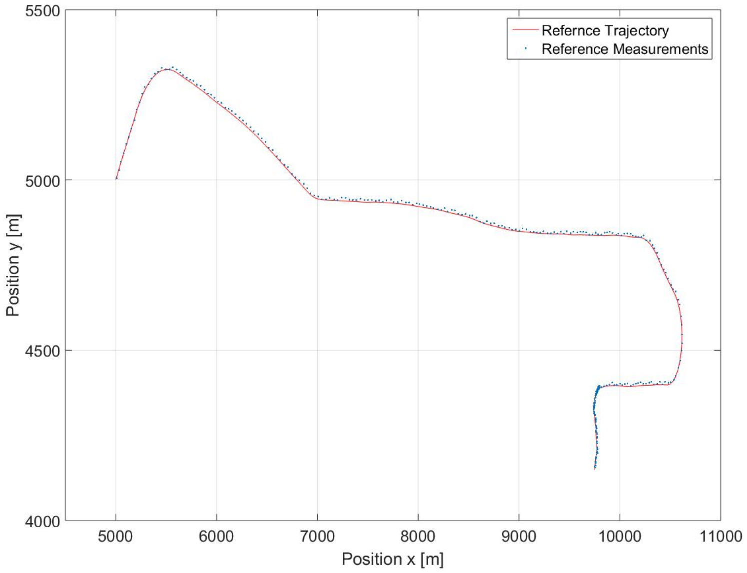


Figure 5.21: Semi-Markov process with corresponding reference measurements.

reported in the Fig. 5.22 left. Performance of IMM filters augmented with an on-line mode transition probabilities estimation process, for cases where contextual observations are denied and available (5.3.3), are shown in the Fig. 5.22 right and Fig. 5.22 bottom, respectively. Note that the time between mode m_k^i and m_{k-1}^j, i.e. the sojourn time τ^i in mode m_k^i, is selected based on the sojourn time PDF $f(\tau^{ji})$ or $\Pi(dk) = e^{\Lambda dt}$. Variable $\mathbf{\Lambda}$ is the transition density matrix of the process, defined similarly as $\mathbf{\Pi}$, with diagonal elements $\lambda^{ii} = -\frac{1}{\tau}$. By knowing that the mean sojourn time of the process, depicted in the Fig. 5.8, is approximately $20s$, i.e. $\tau \approx 20s$, the direct discrete counterparts of main diagonal elements λ^{ii} can be evaluated as $\pi^{ii} = 1 - \frac{1}{\tau}$. Thus, the best guess of the TPM for a given semi-Markov process is utilized in the IMM filter and read as follows

$$\mathbf{\Pi}_k^{IMM} = \begin{bmatrix} 0.95 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.95 & 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.95 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.95 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.95 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.95 \end{bmatrix}. \tag{5.60}$$

In cases of IMM-QB and IMM-QB-HMM adaptive filters the initial value of the TPM was chosen from an uninformative prior Π_0 (5.54). As in previous scenarios, relative errors of the IMM, the IMM-QB adaptive and the IMM-QB-HMM adaptive filters w.r.t. the true trajectory (Fig. 5.21) are shown in the Fig. 5.23. By stacking relative error plots (Fig. 5.23) over each other (Fig. 5.24), several observation can be made about filter performances. The performance difference between the IMM-QB adaptive and the IMM is rather small, when only a marginal performance gain was achieved by the adaptive algorithm 2.72%. The mean error value of the IMM-QB adaptive solution reached only $\bar{x}^{IMM-QB} = 9.7232m$ and the IMM mean error was contained at $\bar{x}^{IMM} = 9.9959m$. However, inclusion of the contextual knowledge inferred by the HMM gave the IMM-QB-HMM adaptive estimator a significant advantage over its counterparts. A performance gain of 9.64% and 6.92% was reached over the IMM and the IMM-QB adaptive solution, respectively, while the mean error of $\bar{x}^{IMM-QB-HMM} = 9.0326m$ was observed. It should be stated, that performance gain of adaptive TPM trackers relies heavily on the amount of mode changes/jumps a target perfromes. When the number of mode states is increased, a significantly longer sequences need to be observed in order to achieve a sufficient convergence rates towards the reference TPM. Inclusion of CI increases the confidence of TPM estimators that a certain mode transitions was observed as demonstrated in the Fig. 5.25. In the figure, a combination of the context likelihood and the measurement likelihood,

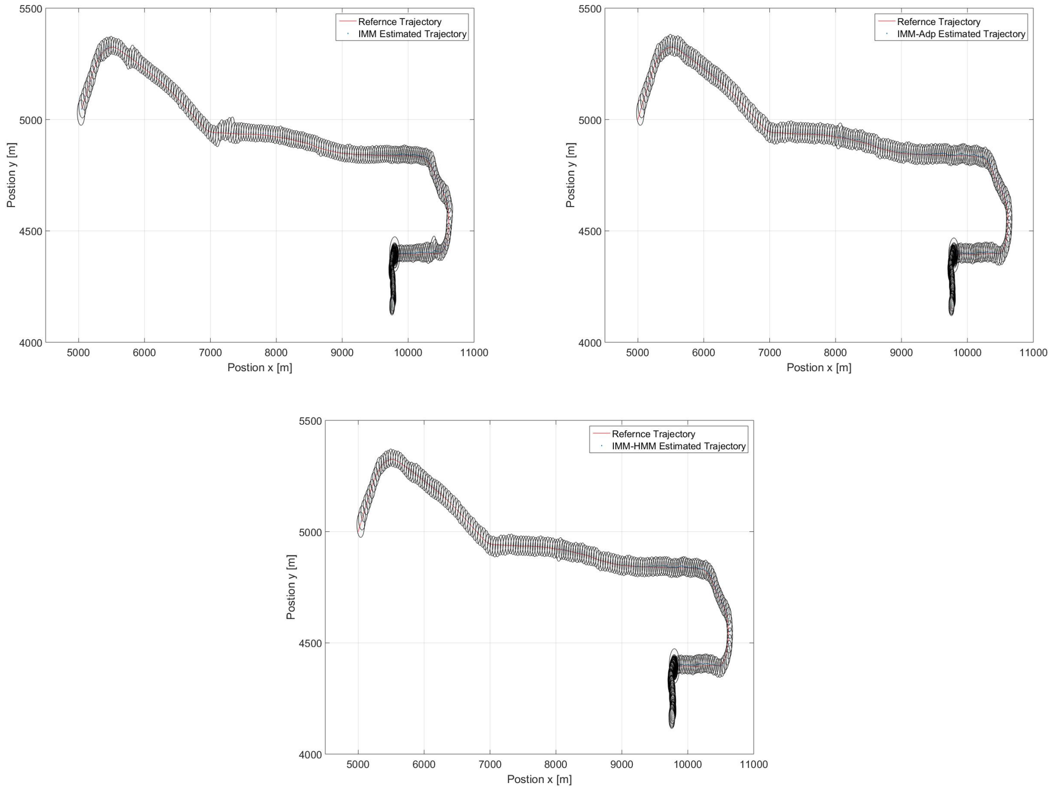


Figure 5.22: IMM (left), IMM-QB adaptive (right) and IMM-QB-HMM adaptive (bottom) estimates of the semi-Markov process.

i.e. joint likelihood (Fig. 5.25 bottom), significantly improve the informativeness of the measurement likelihood (Fig. 5.25 left). The results of 100 Monte Carlo simulation runs of 2D tracking a target following the semi-Markov process is shown in the Fig. 5.26 and summarized in the Tab. 5.3. Based on the results, utilizing an adaptive estimate of the TPM matrix improves the performance of the IMM filter by roughly 5.32% while context is present and by approximately 1.56% while context is absent.

Target Tracking	Relative Error Mean [m]	Relative Error Std. [m]
Kalman filter	14.2412	6.6749
IMM	8.1645	1.4268
IMM-QB Adap.	8.0408	1.3650
IMM-QB-HMM Adap.	7.7301	1.2878

Table 5.3: Comparison of the tacking of a target governed by semi-Markov process

5.5 Conclusion

An object tracking framework that views target actions as a Hidden Markov Model with a relevant spatial and event-temporal context associated with each node is proposed and evaluated on synthetic datasets. Each Markov sequence represents an unique goal that the target seeks to reach within a certain graph topology. As the target progresses through the network, the target's future actions are inferred based on a current evidence and a priori knowledge. The belief that a vehicle seeks an objective is used to aid the tracking

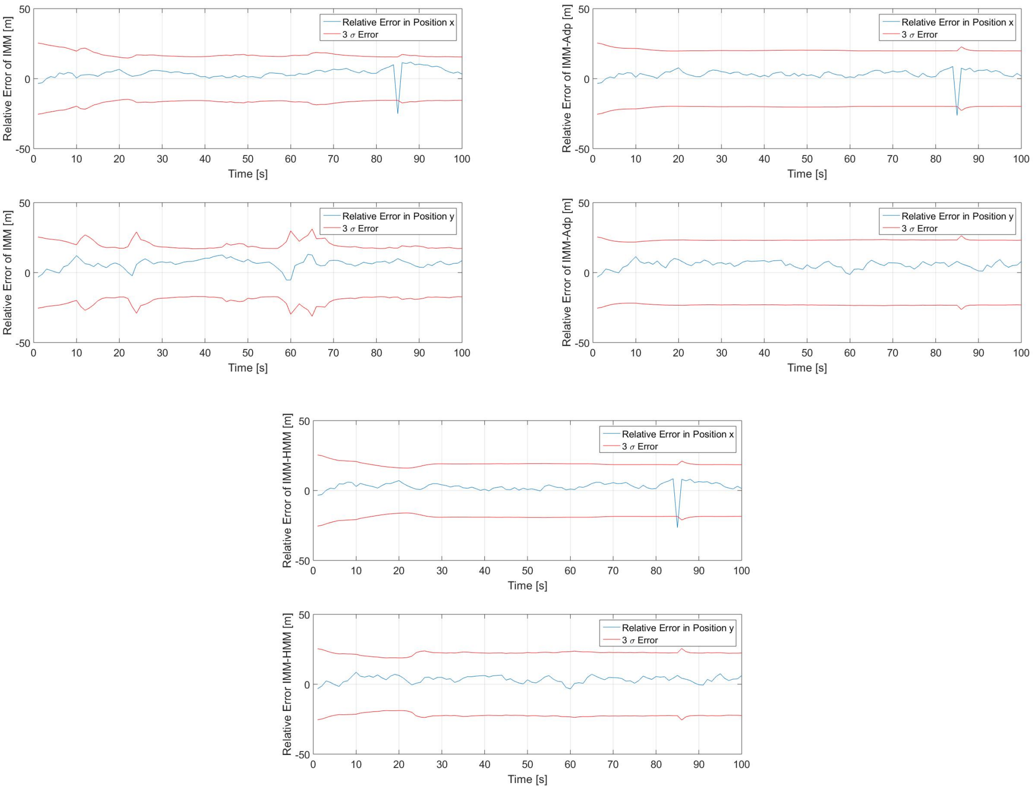


Figure 5.23: Relative errors of IMM (left), IMM-QB adaptive (right) and IMM-QB-HMM adaptive (bottom) estimates of the semi-Markov process.

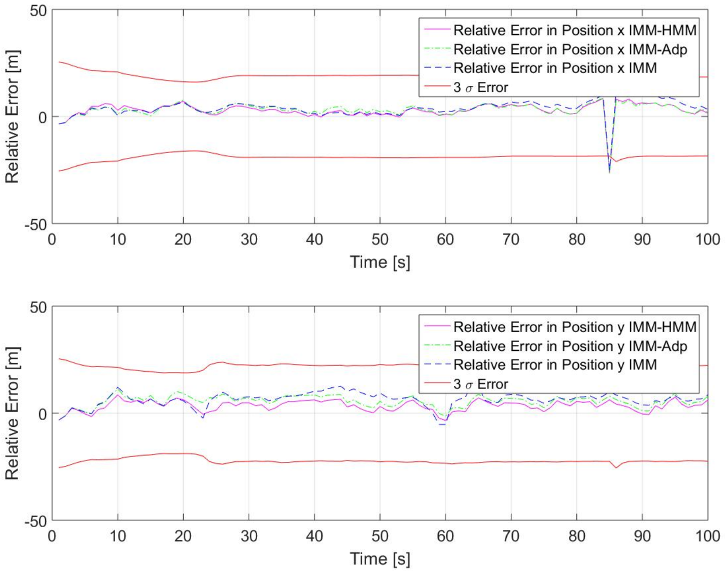


Figure 5.24: Comparison of the relative errors of IMM, IMM-QB adaptive, IMM-QB-HMM adaptive estimates of the semi-Markov process.

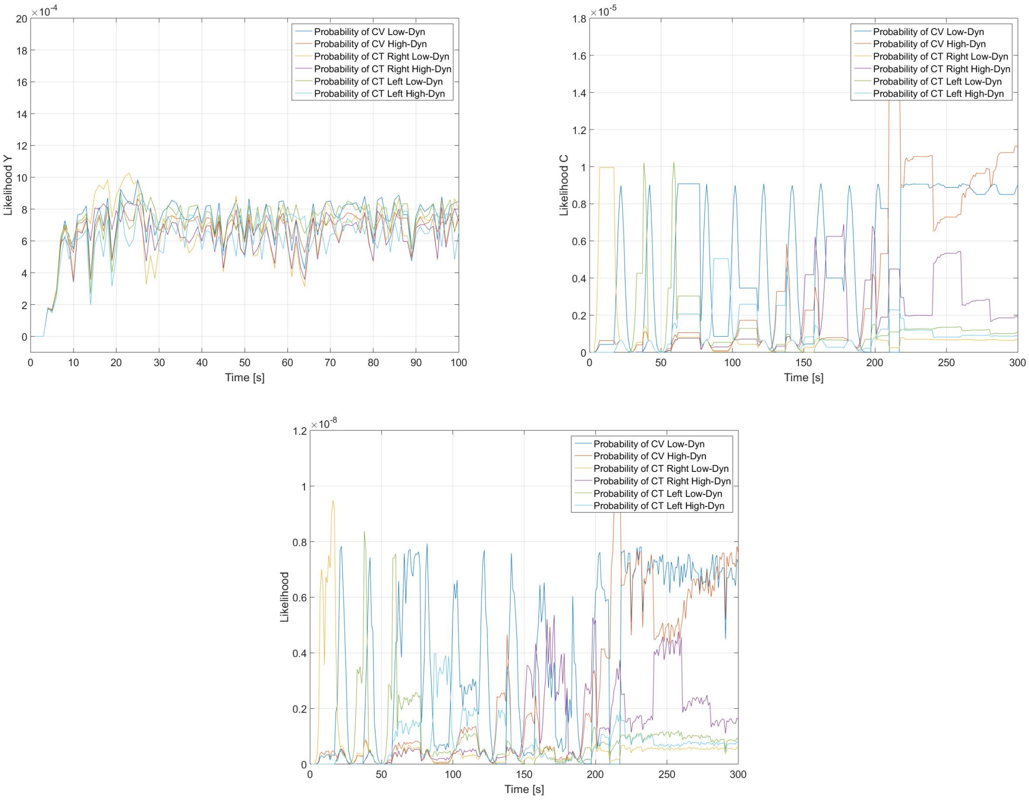


Figure 5.25: Examples of measurement (left), context (right) and joint likelihood (bottom) functions utilized during by the IMM-QB-HMM adaptive estimator.

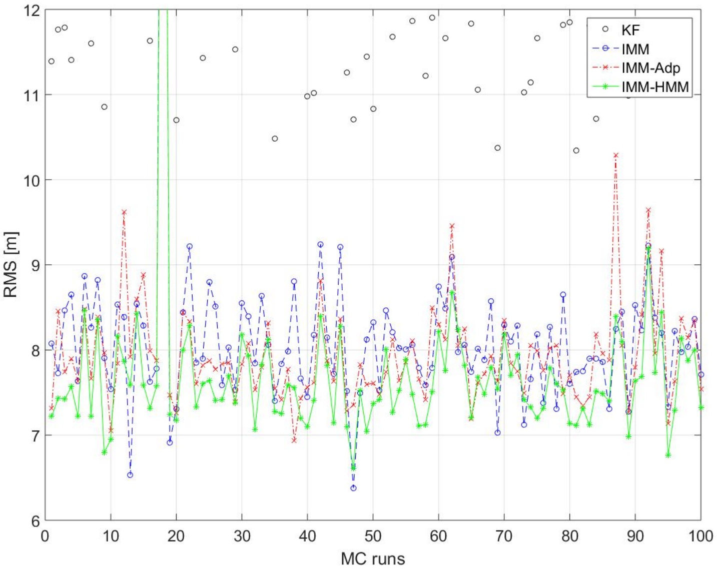


Figure 5.26: Monte Carlo simulation of the tacking of a target governed by semi-Markov process.

process by adjusting the mode transition probabilities in the Interactive Multiple Models estimation process. The posterior elements of the TPM matrix are evaluated recursively, by a quasi-Bayesian framework, in terms of the multiple models mode probabilities and the joint likelihood of contextual information and measurements.

The sub-optimality of quasi-Bayesian estimators give rise to steady state errors observed in the transition probabilities estimate. A plausible reasons behind these biases are due to the approximations involved in the derivation of measurement likelihood of the TPM matrix and due to the mixing process of combining mixture components into a single Gaussian by averaging their means and covariances. A plausible reasons behind these biases are due to the approximations involved in the derivation of measurement likelihood of the TPM matrix and due to the mixing process of combining mixture components into a single Gaussian by averaging their means and covariances. It is worth noting, that the convergence of the QB estimators is highly dependent on the amount of jumps observed between individual modes. For this reason, the informativeness of the likelihood is a key performance factor in the recursive estimation of any TPM matrix represented as a finite mixture.

The underlaying role of the context in a process of the adaptive TPM estimation is twofold. First, context significantly improves the convergence and reduces the steady state error of the estimated mode transition probabilities. Second, prior to an event CI adjusts the mode ambiguity of the IMM in such a way, that occurrence of tacks losses is significantly reduced in cases when the target performs an unlikely maneuver. Model of the contextual likelihood constitutes from the inferred target mode and the relevance weighting, that scales with a distance to the event. When the outcome of the event is uncertain or misjudged, the progressive context weighting gives the IMM filter enough time to reconsider the mode observation gains, i.e. to prioritize the measurement over context, and avoid track losses.

Based on the outcome of Monte Carlo simulations, utilization of the adaptive estimate of the TPM matrix improved the performance of the IMM filter by roughly 23.19% in a scalar jump Markov linear process tracking (JMLS). When the simulation complexity was extended to 2D JMLS tracking, the adaptive solutions were able to maintain the advantage over the IMM filter by roughly 13.98% while context was present and by approximately 5.46% while context was absent. In the simulation of the ground object tracking with airborne sensors, the knowledge of the road topology and target behaviors improved the track accuracy by roughly 5.32%. We argue, that performance gain of the IMM-QB-HMM filter can be further increased over the time as the length of the event sequence, i.e. the amount and type of mode transitions, is risen. It was further observed that the observability of mode states decreases with the increased complexity. This performance drop can be credited to the sub-optimality of the QB estimators and approximations used in the derivation of the contextual TPM likelihood from the joint prior (5.20).

6

Occupancy Grids Fusion and Environment Mapping

Grid-based environment mapping and obstacle detection becomes increasingly more challenging when the sensors readings are highly contrasting. As opposed to feature-based approaches, occupancy-grid techniques do not utilize predefinitions of map features thus completely avoid the data association problem. Without a measurement prediction, commonly used approaches to the grid fusion (Bayesian occupancy filter or Dempster-Shafer theory) weight the measurement grids equally unless specified otherwise by the user. Empirically adjusted sensor weights are tailored only for a certain scenarios and not at all suited for a general purpose mapping. It therefore becomes apparent, that sensor weights needs to be adjusted recursively during the fusion process. We show, that discrepancies between the grid maps can be exploited in such a manner where fusion of contradicting information will be less susceptible to the sensor weighting and the accuracy of mapped environment can be further improved. We present a realization of such a conflict insensitive occupancy grid mapping, which combines advantages of grid-based mapping and situation assessment in a holistic approach.

6.1 Introduction

Situational awareness and environment mapping are the fundamental tasks of any autonomous system. The design of a robust on-road situation aware system relies on the fusion of multiple complementing sensors in order to overcame challenges associated with a high variation of observed environments, both in terms of obstacles' characteristics and ever changing weather conditions [71]. Most approaches to the mapping use the simultaneous localization and mapping (SLAM) scheme for representing both the environment and for successful navigation through the newly build environmental map. Widely utilized feature-based SLAM refers to the problem of determining ones pose (position and orientation) using observation to features of unknown position. Features can represent any well-defined point in the world as corners, lines or markers. The autonomous system is tasked to incrementally build a map of its surrounding environment and at the same time use this map to compute its own location [38]. Mapping the features or obstacle positions is thus an integrated part of the SLAM problem. On the contrary, the occupancy-grid SLAM techniques do not require predefinitions of map features thus completely avoid the data association problem. Instead, they model arbitrary environments using a dense field of binary (occupied or vacant) random variables, arranged in an evenly spaced grid. As a matter of fact, localization and mapping are inverse problems, where mapping considers the map of an environment as an unknown variable, and localization is trying to find the state of the vehicle w.r.t the world.

Map building techniques are typically either specialized for structured environments and depend entirely on such structure being present in their surroundings i.e. feature-based models, or are tailored for unstructured environments and ignore any structure that may exist i.e. occupancy grids. The highway and most of urban and rural sections of the road can be regarded as structural environments. When the environment to be mapped is an intersection, a busy urban center, or an off-road scenario many essential parts of the environment will be incorrectly classified due to the high variation in model-based descriptors. For these environments, the natural choice is to employ occupancy grid mapping [78], [80], [108].

In this work, we are concerned about grid-based SLAM applications where the location of the vehicle is known and the objective is to map the surrounding environment for possible obstacles and threats. The idea is to utilize visual sensors (Fig. 6.1a), LiDAR ranging (Fig. 6.1b) and contextual information (Fig. 6.1c) in order to determine drivable spaces (Fig. 6.1d) by employing the occupancy-grid mapping techniques. Grid maps, in comparison to feature [62] or graph based maps [76], are sensor independent and capable to represent any kind of information with a relative ease. The accuracy of the grid map depends primarily on the resolution, based on which, the environment is discretized into a binary mesh. Example of the occupancy grid for visual detector, LiDAR, context map and drivable spaces, i.e. a fused grid, are shown in the Fig. 6.1a, b, c and d, respectively. Visual sensors are extremely rich source of the environmental features which are naturally modeled as n-dimensional vectors of coordinates and attributes best encaptured as the feature-based maps. In order to utilize these feature spaces in the occupancy grid framework (Fig. 6.1a), each feature coordinate needs to be projected into the 2D ground plane [79], [108], [109], e.g. by assuming that a word is flat, and clustered into objects of an interest, such as road lanes and static or dynamic objects. Sensors relying on the time of arrival principle, such as LiDAR (Fig. 6.1b), are invaluable for reliable obstacle detection and commonly utilized in the grid mapping [64], [80]. The exploitation of contextual information for the data fusion and its applicability for the environment mapping caught a lot of attention in recent years [134], [18]. While exploiting the map context is a well established in the literature [79], [83], utilization and representation of any arbitrary context is still an understudied problem. We believe that CI can be intuitively represented in a form of likelihood functions which can be conveniently represented as grid maps [153]. The challenge of multi-sensor grid mapping is to combine all pieces of evidence (Fig. 6.1a, b, c) in order to reliably and accurately depict the surrounding environment (Fig. 6.1d).

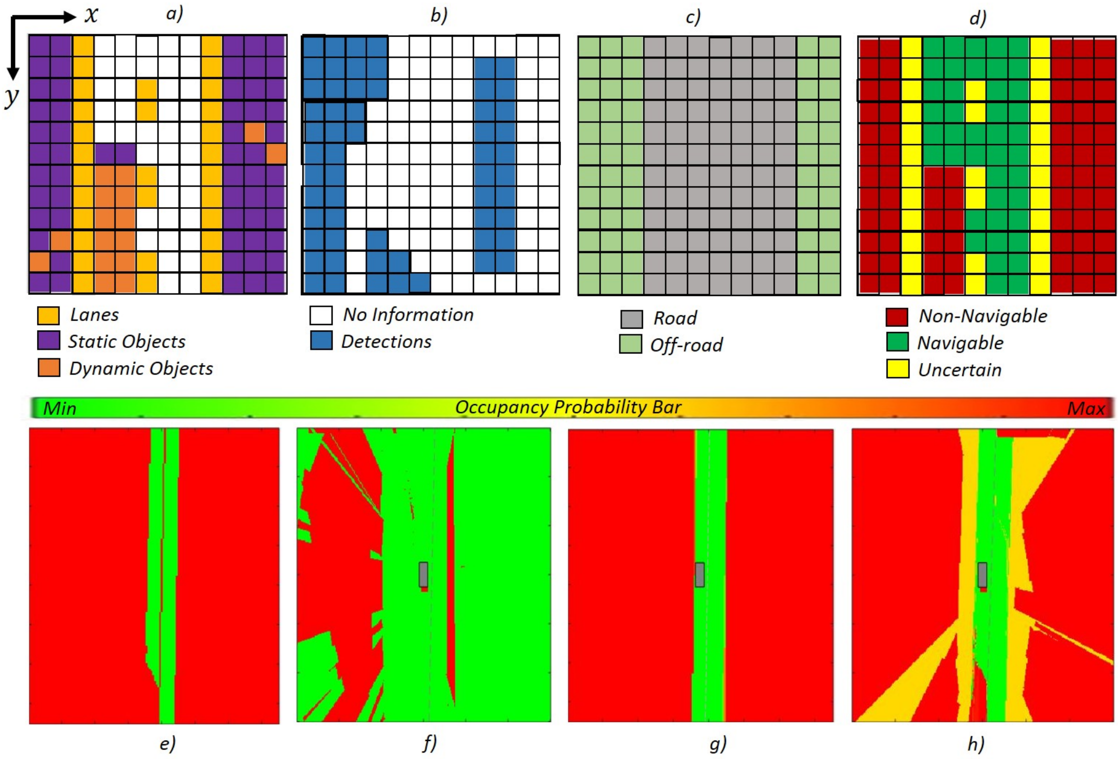
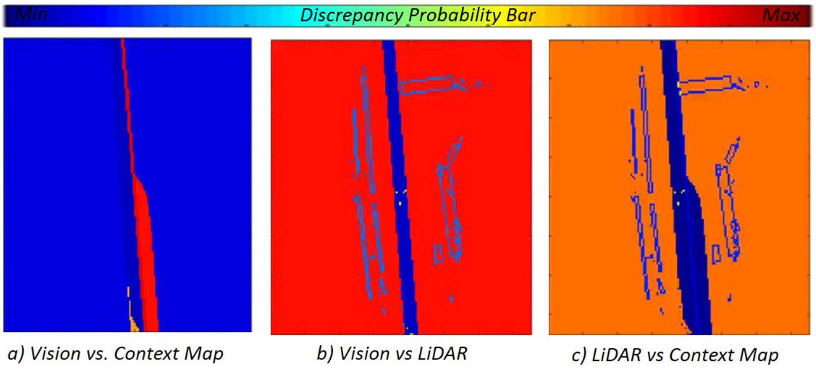


Figure 6.1: Illustration of occupancy grids and drivable spaces.

As opposed to single sensor mapping, grid maps built as a result of the multi-sensor fusion relies heavily on sensor measurement weighting [122]. Standard approaches to the mapping, i.e. the Binary Bayes filter [43] or the Dempster-Shafer theory [33], [63], weight the measurements equally unless specified by the user. On the other hand, empirically adjusted sensor weights are tailored only for a certain scenarios and produce less reliable map in situations where the sensors readings are contradictory. It therefore becomes apparent, that the sensor weights need to be adaptive in multi-sensor mapping applications. Adaptability of these weights should be governed by the estimation process in an iterative manner. We are going to explore the possibility of conflict resolution for cases of the Bayesian occupancy filtering (BOF) and the Dempster-Shafer theory (DST) of evidence.

In a Bayesian framework, we are taking advantage from discrepancies naturally occurring between each source of information (Fig. 6.2). We argue, that the discrepancies can be reasoned in a logic-based manner and the outcome of this reasoning process, i.e. a siltation assessment, can aid the fusion solution in conflict detection and resolution. In contrast, the Dempster-Shafer theory is considered as one of the major paradigm shifts for the grid fusion under uncertainty. However, evaluation and performance analyses of the multi-sensor DST fusion scenarios under real world conditions are quite rare and hard to find. For this reason, we are presenting a comparative study where three Bayesian occupancy fusion algorithms, i.e. linear opinion pool, logarithmic rule, BOF with situation assessment, and the classical DST and the DST with a proportional conflict redistribution (PCR5) are compared, evaluated and discussed.



a) Vision vs. Context Map

b) Vision vs LiDAR

c) LiDAR vs Context Map

Figure 6.2: Illustration of discrepancies between occupancy grids.

6.1.1 Overview of the Framework

Nowadays approaches to the environment mapping are based either on feature recognition and tracking techniques, i.e. feature-based models ([36], [45], [121], [156], [80]), or operate on the occupancy based map building principle ([62], [76], [79], [108], [109]). The two approaches are by no means mutually exclusive, but they in fact supplement each others weaknesses. The capability to model free spaces makes occupancy maps well suited for path planning and obstacle avoidance tasks. However, occupancy maps are less desirable for localization purposes, where relative pose of the objects is required for accurate estimation of own position and orientation. Consequently, feature-based methods have been successfully applied to the numerous localization and mapping tasks [8]. However, feature-based SLAM techniques fail to discern unknown types of obstacles and to operate reliably in cluttered environments. Our objective is to create an accurate representation of the environment and determine the drivable spaces in such a way, that an autonomous system capable of estimating its own pose will be able to safety navigate through the newly built map. The mapping algorithm is tasked to represent the environment by using the data originated from multiple complementary sensors and context information sources alike, i.e. cameras, LiDAR and Geographic Information System (GIS) map. Therefore, by examining existing map building strategies surveyed in [62] and [76], the occupancy grid mapping was selected as the solution of choice for the task of determining the drivable spaces.

Arguably, the Bayesian multi-sensor data-fusion is the most popular approach to the occupancy map building, as demonstrated by ever increasing number of articles, e.g. [66], [71] and [43] to name a few. The underlining idea behind the occupancy grids [39] is to represent the surrounding environment as a 2D grid map $\mathbf{m}$ with fixed dimensions and spatial resolution (Fig. 6.1). The occupancy of each cell m^{ij} is represented by a random variable within interval $m^{ij} \in \{0, 1\}$, where $i \in \{1, \ldots, N_i\}$ corresponds to number of rows and $j \in \{1, \ldots, N_j\}$ represent number of columns. When the occupancy grids are being updated by sensor measurements, each cell m^{ij} within the grid $\mathbf{m}$ is associated with an occupancy probability

$p(m^{ij}) \in \{0,1\}$ as opposed to a binary value, i.e. occupied state $p(m^{ij}) = 1$ or free state $p(m^{ij}) = 0$. The map building can be formulated as an optimization problem (6.5) that maximize the probability that particular cell is in fact occupied (or free) $\hat{m}^{ij}$ given the prior knowledge of the environment, vehicle poses and sensor measurements, i.e. to solve $\hat{m}^{ij} = \arg\max_{\mathbf{m}_{1:k}} p(m_k^{ij}|m_{1:k-1}^{ij}, \mathbf{x}_{1:k}, \mathbf{y}_{1:k})$. The recursive solution to this optimization problem [151] is commonly referred to the Bayesian occupancy filter or the temporal grid fusion (Fig. 6.3). By following the block diagram from left to right, given the know pose of

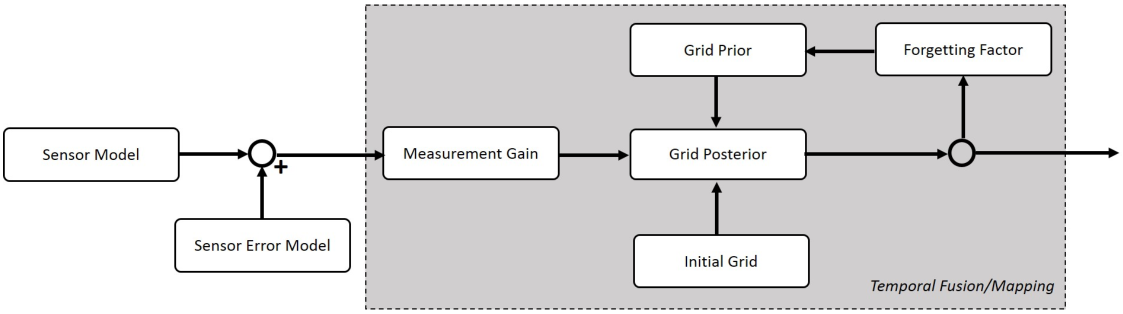


Figure 6.3: Architecture for single sensor temporal grid fusion.

the vehicle $\mathbf{x}_k$ and sensor readings $\mathbf{y}_k$, an input in a form of the likelihood $p(\mathbf{y}_k|\mathbf{x}_k)$, is fed into the temporal fusion block. As stated by the Bayes theorem, probability that particular cell $p(m^{ij})$ is occupied (or free), i.e. $p(m_k^{ij}|\mathbf{x}_k, \mathbf{y}_k)$ (6.8), is proportional to the prior grid $p(m_{k-1}^{ij}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$ and the likelihood $p(\mathbf{y}_k|m_k^{ij}, \mathbf{x}_k)$. In general, the surrounding environment is assumed to be static, and thus there is no need to calculate evolution of the map state $p(m_k^{ij}) \propto p(m_k^{ij}|m_{k-1}^{ij})$, i.e. Chapman-Kolmogorov equation. As a consequence of $p(m_k^{ij}|m_{k-1}^{ij}) = 0$, the Bayesian gain $f\big(p(m_k^{ij}|\mathbf{x}_k, \mathbf{y}_k) - p(m_k^{ij}|m_{k-1}^{ij})\big)$ is computed solely on bases of the measurements as indicated by the block scheme (Fig. 6.3). Term $p(m_k^{ij}|\mathbf{x}_k, \mathbf{y}_k)$ in the measurement gain function is defined as the inverse likelihood model. The forgetting factor is a design parameter purpose of which is to increase or decrease the weight of newly obtain evidence on the prior map. Note, the temporal fusion architecture (Fig. 6.3) is tailored for a single sensor occupancy grid building.

Two strategies to the multi-sensor occupancy grid mapping can be distinguished based on a stage where the temporal fusion is performed [71], [80]. Centralized architectures combine the multiple sensor readings into a single measurement input before the temporal occupancy mapping takes place. On the contrary, distributed architectures (Fig. 6.4), perform the temporal fusion update for each individual sensor first, and only after then the sensor fusion routine is executed. We argue, that distributed occupancy mapping is more effective while fusing measurements of heterogeneous origin as grids can represent any information by an occupancy probability and completely avoid nontrivial modeling of physical relationships existing between the states $\mathbf{x}_k$, $\mathbf{y}_k$ and the map $\mathbf{m}_k$. Grid fusion architecture depicted in the Fig. 6.4 was adopted as the general purpose solution for environment mapping and conflict resolution utilized in this work.

In accordance with the block scheme (Fig. 6.4), the knowledge about the drivable spaces is obtained from the sensor measurements in three distinct stages: temporal fusion, multi-sensor fusion and conflict resolution. The raw sensor readings are processed into individual sensor grids by the Bayesian occupancy filter (Fig. 6.3). Sensor grid represent the observed scene by an occupancy probability $p(m_k^{ij}) \in \{0,1\}$ specified for each cell m_k^{ij} of the matrix grid $\mathbf{m}_k$. At the multi-sensor fusion level, individual sensor grids are fused together in order to provide truthful representation of the environment expressed in the fusion grid. Here, the choice of a fusion strategy is not straightforward as various formulas tent to emphasize different pieces of information, and thus resulting in alternative representations of the environment. Perhaps the most pronounced solutions to the grid fusion are based on Bayesian inference and Dempster-Shafer theory (DST) of evidence. As for the Bayesian fusion rules basic fusion formula (BFF), linear option pool (LOP) [1] and the maximum log-probability of occupancy (ML), were considered for the purpose of this study. The basic fusion formula utilizes the knowledge of sensor weights in a form of proportional gain applied to the grids as a weighted sum. The linear option pool strategy is merely identical to the basic fusion formula, but in addition to the BFF option pools utilize the knowledge of the prior grid $\mathbf{m}_{k-1}$. In general, the LOP generates a lot of conflicts which need to be resolved by a further processing, i.e. by

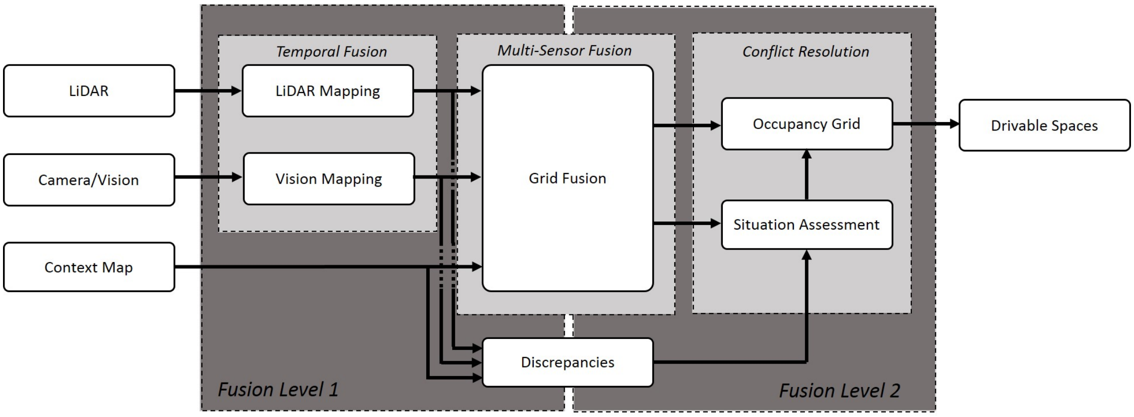
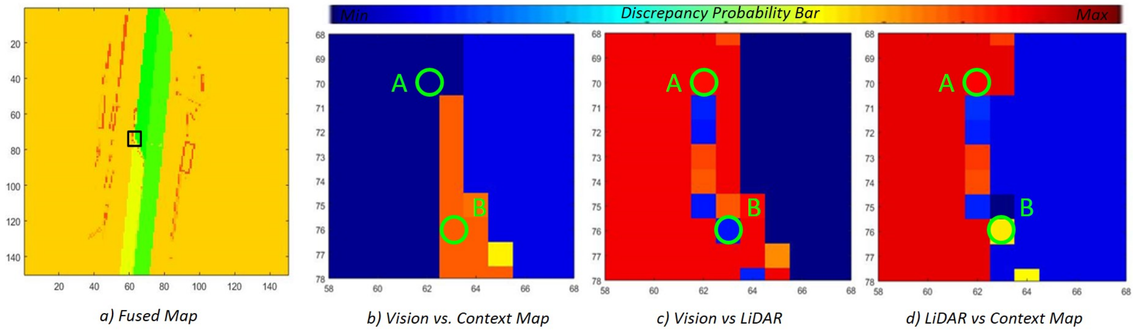


Figure 6.4: Architecture for multi-sensor occupancy grid fusion.

a conflict resolution procedure. The fusion based on the maximum probability of occupancy is the most conservative one. Similarly to the linear option pool, the prior knowledge of the world is utilized in a ML grid update step. However, the solution suffers from a huge amount of false negative alarms which are assumed to be true.

Unless specified a priori by the user, the distinction of drivable spaces is primarily dependent on the result of equality weighted measurement grids. In classical filtering, the truthfulness of an individual sensor reading is evaluated against that measurement prediction in the form of a likelihood gain. However, the grid-based approaches do not have the option to predict the behavior of the objects, as this will require implementation of feature recognition and data association techniques, and thus solely rely on measurement weighting. Empirically adjusted sensor weights are tailored only for certain scenarios and not at all suitable for a general purpose mapping. It therefore becomes apparent, that the sensor weights need to be adjusted recursively during the fusion process [125], [32]. Even though it is feasible to fine tune the weights for each individual measurement grid such that the drivable spaces can be reliably determined for the presented scenario, we opt for development of more robust approach based on situation assessment strategies.

Conflict resolution module, based on the station assessment (SA), was introduced into the grid fusion architecture in order to improve the fidelity of the drivable spaces [47], [11]. Generally, the situation assessment could be based on probabilistic models or alternatively based on encoded rules and logical formulas. The later was encoded into aforementioned architecture (Fig. 6.4) as only the combination of three sources of information were considered in the solution making an empirical approaches viable. However, for large scale datasets with extensive variation in sensor reading the rule-based reasoning might become inadequate and employing the machine learning techniques necessary [95]. As explained on the illustration below (Fig. 6.5), the discrepancies between grid maps can be exploited in such a manner where the fusion of contradicting information will be less susceptible to the sensor weighting and the accuracy of mapped environment can be further improved. On the Fig. 6.5a, a single frame of the fused grid map is shown while the area under investigation is bounded by a black box. It can be observed, that wast majority of the grid cells are painted by shades of yellow, which corresponds to highly conflicting or uncertain information about the drivable spaces, i.e. $p(m_k^{ij} \approx 0.5)$. In order to tip the occupancy of ambiguous cells towards either occupied $p(m_k^{ij} >> 0.5)$ or free $p(m_k^{ij} << 0.5)$ state the following discrepancy resolution logic was adopted. Assume that points A and B are both marked as ambiguous, that is $p(m_k^A \approx 0.5)$ and $p(m_k^B \approx 0.5)$, respectively. If point A is a grid cell corresponding to the road edge, this information will be confirmed by a visual sensor as a lane and by context map as a round boundary. However, LiDAR will likely classify the cell $p(m_k^A)$ as free, because the majority of light beam energy will be dispersed into the surrounding environment. Therefore, the discrepancy between visual sensor and context map (Fig. 6.5b) will be very small as the measurements from both sensors confirm each other. On the contrary, discrepancies of visual sensor vs. LiDAR (Fig. 6.5c) and contextual map vs. LiDAR (Fig. 6.5d) will be rather high as the measurements from both sensor pairs contradict each other. Therefore, the cell $p(m_k^A)$ will be marked as occupied since $(|\text{Context Map} - \text{Vision}| < 0.5) \wedge (|\text{LiDAR} - \text{Vision}| >= 0.5) \wedge (|\text{LiDAR} - \text{Context Map}| >= 0.5)$. Let us assume, that point B belongs to a grid cell corresponding

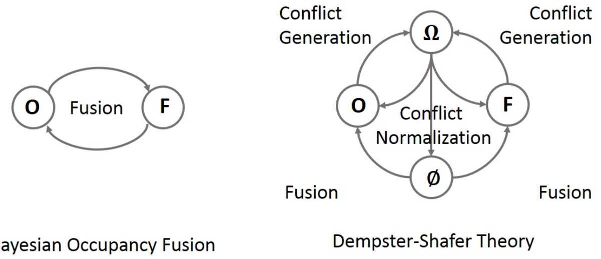


a) Fused Map b) Vision vs. Context Map c) Vision vs LiDAR d) LiDAR vs Context Map

Figure 6.5: Exploitation of discrepancies between occupancy grids.

to the new jersey. The presence of the new jersey on the road will be detected by visual sensor and LiDAR as an obstacle, but not at all by a context map (GIS). For this reason, the discrepancy between visual sensor and LiDAR (Fig. 6.5c) will be very small as the measurements from both sensors confirm each other. On the contrary, the discrepancies of visual sensor vs. context map (Fig. 6.5a) and LiDAR vs. contextual map (Fig. 6.5d) will be rather high as the measurements from both sensor pairs contradict each other. Therefore, the cell $p(m_k^B)$ will be marked as occupied since $(|\text{Context Map} - \text{Vision}| > 0.5) \wedge (|\text{LiDAR} - \text{Vision}| < 0.5) \wedge (|\text{LiDAR} - \text{Context Map}| < 0.5)$. By assuming three independent information sources N_y for which three discrepancy grids are needed to represent, the conflict between each pair $N_{\delta y}$ a total of ${N_{\delta y}}^{(N_y - 1)}$ rules need to be formulated in order to represent the scenario. We have implemented these rules into the weighted linear opinion pool fusion through a situation assessment feedback and successfully demonstrated the potential of such a coupling during the mapping. Such a reasoning is not uncommon in the occupancy grid fusion as numerous theories were introduced for the conflict resolution over past decades.

The most cited alternative to the Bayesian theory of fusion was formulated by Dempster and Shafer as a set of combination rules known as the Dempster-Shafer theory of evidence (DST). DST was developed to address the uncertainties occurring between occupancy grids by utilization of combination rules, i.e. the theory of evidence, applied on evidential grids. In the evidential grids, the uncertainties are modeled as belief functions. The belief about the environment occupancy is classified by either occupied O or free F, which results into a set $\mathcal{X} = \{O, F\}$. The set of all possible subset of $\mathcal{X}$, i.e. the frame of discernment (FOD), is denoted as $2^{\mathcal{X}} = \{occupied(O), free(F), conflict(\Omega), unknown(\emptyset)\}$. In the Bayes theorem, the occupancy of a cell was evaluated based on single probability per cell, while in DST the decision about occupancy is made based on the belief function of four masses i.e. occupied O, free F, conflict Ω and unknown $\emptyset$. Furthermore, the belief transition in the Bayes theorem is possible only between two states and it is symmetrically restricted (Fig. 6.6 left). In the evidential approach, the belief can be transferred among four states, in which each of these transitions has a different meaning, dynamic and importance (Fig. 6.6 right). These comparisons illustrate the applicability and relevance of the DST approach for grid fusion, where the DST models higher level of confidence in cells than just a single probability value. Despite being capable of combining the independent pieces of evidence, the DST has been also strongly



Bayesian Occupancy Fusion Dempster-Shafer Theory

Figure 6.6: State transitions occurring in the Bayesian occupancy framework and in the Dempster-Shafer theory field of discernment.

criticized for its unexpected behavior and by providing counter-intuitive results when combining highly conflicting information. It is worth noting, that basic DST formula is in complex fusion systems further altered by a human expert knowledge in order to improve quality and reliability of the solutions. In order to cope with a lack of expert knowledge or with an unknown and unpredictable evidence, the DST was extended to new more flexible theories such as transferable belief models (TBM), adaptive combination rules (ACR) or proportional conflict redistribution (PCR). The PCR formula is particularly interesting as it redistributes the partial conflicting masses to the elements involved in the conflicts. We are going to evaluate the DST alongside with PCR in their capability to provide the reliable information about drivable spaces under measurement uncertainties.

6.1.2 Main Contributions

The contribution of this work are twofold:

- Discrepancy-grid approach for conflict redistribution and automatic sensor weight adjustment is introduced for the Bayesian occupancy (BOF) mapping.
- Comparative study of grid based approaches, namely BOF linear opinion pool, BOF maximum log-likelihood, BOF with discrepancies resolution, Dempster-Shafer theory (DST), DST with PCR5, to mapping are discussed, developed and evaluated on the real world scenario.

6.1.3 Relations to the Context Exploitation Framework

Unlike the context exploitation frameworks discussed in Sec. 4.1.3 and Sec. 5.1.3, the map building techniques are focused on acquiring the information about possible threats and dangerous areas from the surrounding environment rather than using this a priori knowledge to enhance the SE process. The fusion node, as a core module in the context-adaptive framework (Fig. 3.3), is in light of the grid-mapping architecture (Fig. 6.4) only consisting from the state estimation process, which is divided into a temporal fusion and a multi-sensor grid fusion. The contextual information is represented as a discrepancies between the individual sensor grids (Fig. 6.5) and/or the GIS map (Fig. 6.1c). The GIS map contexts is used alongside the measurements in the fusion process governed by Bayesian occupancy filter. On the other hand, the discrepancies are processed in the context reasoning module, i.e. the context middleware, which are later used to adjust cell probabilities of the resulting occupancy map. Different from the Bayesian approaches, the Demster-Shafer theory models the uncertainties between the grids within a specific frame of discernment (Fig. 6.6). DST allows for the extension of FOD for the contextual states within a unified and well established framework, which does not require implementations of the context management functions.

6.2 Related Work

Nowadays approaches for robotics and autonomous vehicle mobility rely almost exclusively on the knowledge of spatial uncertainties which are observed in the surrounding environment. In the past, this need has been circumvented by use of highly accurate sensors and by the use of fixtures and calibration points. These approaches avoid the need of representing uncertainties in the environment by utilizing expansive high grade sensors with the minimal drift. Another approaches suggest to use multiple, overlapping, lower resolution sensors and to combine the spatial information (including uncertainties) from all data sources to obtain the best spatial estimate. However, the problem with usage of low grade sensors leads to accumulation of uncertainties. Modeling uncertainties pose a challenge for design of situation-aware sensor fusion solutions capable of keeping the system divergence within tolerable margins and to ensure reliable detection of possible threats. A comprehensive review of the current challenges and approaches was recently presented in [71]. According to survey, the main difficulties in design of situation-aware sensor fusion algorithms arise from spatio-temporal registration of the sensor readings, data association, management of conflicting information and especially heterogeneity of sensors. Khaleghi et al. [71] classifies approaches to the situation-aware sensor fusion based on sensor layouts, operating environments and the level of abstraction at which the sensor fusion is performed.

Arguably, the most numerous approaches to the situation-aware sensor fusion assume that environment can be represented by feature-based models and depend entirely on such features being present in their surroundings. In conclusion of the seminal developments in autonomous robotics [132], the problem of simultaneous localization and mapping (SLAM) gained widespread interest, with numerous potential applications ranging from robotic planetary exploration to intelligent surveillance [36], [105], [158], [100]. The goal of an autonomous vehicle performing SLAM is to start from an unknown location in an unknown environment and build a map (consisting of environmental features) of its environment incrementally by using the uncertain information extracted from its sensors, whilst simultaneously using that map to localize itself with respect to a reference coordinate frame and navigate in real time. The most commonly used sensors for autonomous vehicle performing SLAM can be categorized into laser, sonar, and vision-based systems. Additional sensors can be used to better perceive platform state information and the outside world. These are not restricted to inertial measurement units (IMU), magnetometers, wheel encoders, infrared technologies and global navigation satellite systems (GNSS). However, all these sensors carry certain errors, often referred to as measurement noise, and also have several range limitations e.g. light and sound cannot penetrate walls. SALM approaches falls into two distinct categories based on sensor layout [8]: an active and a passive.

- Laser ranging systems are accurate active sensors operating based on the time of flight principle. Sonar-based systems are fast and provide measurements and recognition capacities with amounts of information similar to vision, but with the lack of appearance data. Active system are usually largely dependent on inertial sensors where a small error can have large effects on later position estimates.

- Vision systems are passive, they have long range and high resolution, but the computation cost is considerably high and good visual features are more difficult to extract and match. Vision is used to estimate the 3D structure, feature location and robot pose, for instance by means of stereo camera pairs or monocular cameras with structure from motion recovery.

Further classification on SLAM approaches can be made in terms of working environment, for instance, ground-indoor, ground-outdoor, air-borne or underwater. According to [8], majority of SLAM approaches are designed for ground-based vehicles operating in indoor-environments [31], [40], [89]. There have been a growing interest in SLAM for underwater scenarios [159], in which vision plays an important role [88], [120], in most cases combined with other sensory systems to acquire both depth and appearance information of the scene, for instance, acoustic or inertial sensors. Among airborne applications, the work of Kim et. al. [73], [74] has become the most renowned among the scientific community. SLAM approaches employing visual sensors (monocular or stereo cameras) require a sufficiently accurate motion model in order to make predictions on the robot pose before projecting the image features from the previous onto the current frame. Typically a constant velocity model accompanied with wheel encoder provides accurate estimate as long as the camera motion stays within the bound of the error covariance. Random sampling consensus (RANSAC) [86] offers the possibility to construct a proposal distribution from the observed image features, which contains both inliers and outliers, and thus offer reliable and robust solution.

In the context of autonomous systems [43], the vehicle location is in majority of cases considered to be know, i.e. observed by on-board sensors. For this reason, the objective of situation-aware sensor fusion is to map the surrounding environment for possible obstacles and threats. A long-range obstacle detection system, based on stereo-vision and a laser scanner, was recently proposed in [45] and [121]. Authors show, that false-alarm rates and miss-detections can be significantly reduced by utilizing simultaneous multi-sensors tracking and obstacles detection approaches. This concepts was further extended by Wang et al. [156], who proposed to fuse a millimeter-wave radar and a monocular camera for on-road obstacle detection and tracking. Feature extraction, i.e. high-level track representation, was performed for both the radar and the camera, and only matching tracks from both sensors were considered as valid. Kubertschack et al. [80] formulated the problem of static environments mapping into a situation-aware multi-sensor fusion architecture. The main difficulty associated with feature-based approaches to the mapping, or SLAM in general, are related to the data association. Commonly used techniques to this nontrivial problem are not restricted to the Global Nearest Neighbor (GNN) [36], Joint Probability Data Association (JPDA) [13], Joint Compatibility Branch and Bound (JCBB) [107], Multiple Hypothesis Tracking (MHT) [100] and [118], or Maximum-Likelihood Data Association (MLDA) [158], to mention a few. Data association process is an integral part of the object tracking, for which the common solution is to employ Kalman or particle filters or any of their variants for each detection. Another potential drawback of feature-based sensor fusion is related to the signal strength, i.e. signal to noise ratio. Weak signals, if not confirmed by each individual sensor, can be easily rejected and prevent the start of track initialization process. In these scenarios, grid-based SLAM approaches are preferred.

The main advantage of occupancy-grid SLAM over the feature-based techniques is that they are not feature dependent and thus they completely avoid the data association problem. Instead, they model arbitrary environments using a dense field of binary (occupied or vacant) random variables, arranged in an evenly spaced grid [39]. A generic grid mapping approach suitable for mapping of huge areas was presented in [78]. Authors defined a map based on so called grid patches which makes the proposed mapping real time capable. The grid cells of this map can hold arbitrary data which is calculated based on arbitrary sensors. A multi-layer 2D grid mapping based on image sequences was proposed in [84]. 3D occupancy grid mapping for automotive radar was presented in [81], [32]. Moreover, authors presented a simplistic way of updating the grid by using a fast trilinear interpolation in the measurement domain, in which the grid spacing is uniform to relax the grid independence assumption. Baig and Aycard [9] apply the Gaussian mixtures to extract moving objects from the video stream. Resulting grid is a linear combination of LASER range-findings and camera originated measurements. The main drawback of this approach is the requirement of a robot to successfully perform background subtraction. An efficient computation of occupancy grid maps with laser range-finders and radar sensors were proposed [64]. The approach utilizes the graphics processing unit to overcome the limitations of classical occupancy grid computation in automotive environments. Moreover, the authors show that accuracy of a lower resolution radar sensor could be improved by applying super-resolution algorithms for data processing. Authors also discussed a novel histogram based approach for road boundary detection with LiDAR and RADAR sensors. Recently, more attention was given to the fusion of sensors operating on the time-of-arrival principle with stereo caners [70], [109]. Since both instruments can provide the measure of depth these techniques are arguably more useful for situation-aware mapping. An extension to the classical occupancy grid framework for dynamic environments was proposed in [113], [1]. Introduction of dynamic grids greatly improves robustness in case of faulty or spurious measurements from any of the sensors and when they provide conflicting information. As shown in [47], LiDAR and RADAR operate at different frequencies of the electromagnetic spectrum, which made them an ideal pair for fusion where their complementary characteristics can be fully exploited. SLAM approach based on occupancy grid mapping was presented in [54]. Presented approach uses a particle filter in which each particle carries an individual map of the environment. In this paper, authors presented an adaptive technique for reducing this number in a Rao-Blackwellized particle filter for learning grid maps. This significantly decreases the uncertainty about the robots pose in the prediction step of the filter. A quasi-occupancy grid approach was presented by Groover et al. [55], where the RADAR and monocular camera measurements are clustered into blobs, which are afterwards matched in the fusion process.

By surveying the approaches to grid-based mapping [66], [71], [43] and [63], the Bayesian occupancy filtering is the most widely utilized algorithm for determining the state, i.e. probability of occupancy, for each cell in the grid. A commonly used alternative to the Bayesian filters is formulated as the Dempster-Shafer Theory (DST) of fusion. DST is considered as one of major paradigm shifts for reasoning under uncertainty. Despite being capable of combining the independent pieces of evidence, the DST has been also strongly criticized because of its unexpected behavior and by providing counter-intuitive results when combining highly conflicting information. However, it is genuinely agreed that Dempsters rule provides valid results in scenarios where the initial conditions are respected and the problem is well modeled. For this reason, the conflict solving in DST fusion is intensively studied problem as indicated by ever increasing number of publications, such as [35], [98], [122], [85], [65], to name a few. While there are numerous cases where DST was implemented for the navigation purposes, the articles addressing the multi-sensor data fusion cases are nearly nonexistent. It is worth noting, that basic DST formula is in complex fusion systems further altered by a human expert knowledge in order to improve quality and reliability of the solution [160], [137]. In order to cope with a luck of expert knowledge or with an unknown and unpredictable evidence, the DST was extended to new more flexible theories such as:

- Transferable Belief Model (TBM) [33], refutes the constraint on the frame of discernment and the underlying probability model, which allows to allocate belief to the elements of the empty set.

- Dezert-Smarandache Theory (DSmT), extends the DST to allow usage of hybrid and dynamic models and solves numerical issues which originates while combining highly conflicting pieces of evidence.

- Adaptive Combination Rule (ACR), which maximizes the conjunctive and the disjunctive rules based on the distribution of the conflict according to a new choice of weighting coefficients.

- Proportional Conflict Redistribution (PCR) [130], [44], redistributes the partial conflicting masses to the elements involved in the partial conflicts only, considering the conjunctive normal form of the partial conflicts.

Hierarchical or multi-level approaches to situation-aware sensor fusion are quite rare in the literature. Three-level architecture for environmental mapping was proposed in [72]. Infrared and ultrasonic sensors were fused at the signal level through an occupancy grid mapping. At the object assessment level, visual features observed by cameras were fused with infrared and ultrasonic grids. At the situation assessment level the results from previews two layers were further reasoned in a probabilistic way. Results show that multi-layered approaches can significantly improve the efficiency and accuracy of the obtained map. A design of pre-crash safety system was proposed in [95], where information from a multi-layer laser scanner and radar were fused in order to improve the vehicle detection. More recently, Nuss et al. [108] combined occupancy grid maps, digital road maps and multi-object tracking for a rich and robust environmental perception.

6.3 Model Formulation

6.3.1 Bayesian Occupancy Filter

Generic mapping could be represented by a kinematic state space model consisting of three parts: a kinematic motion model (6.1), a measurement model (6.2) and a map (6.3).

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{v}_k \quad \text{or} \quad p(\mathbf{x}_{k+1}|\mathbf{x}_k) \tag{6.1}$$

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{m}_k) + \mathbf{w}_k \quad \text{or} \quad p(\mathbf{y}_k|\mathbf{x}_k, \mathbf{m}_k) \tag{6.2}$$

$$\mathbf{m}_k = \mathbf{g}_k(\mathbf{y}_k, \mathbf{x}_k) \quad \text{or} \quad p(\mathbf{m}_k|\mathbf{x}_k, \mathbf{x}_k) \tag{6.3}$$

In above equations, $\mathbf{f}_k$ represents a function of the target state vector $\mathbf{x}_k$ the vector of input signals $\mathbf{u}_k$ and process noise $\mathbf{v}_k$ in time k. The state evolution of the vehicle kinematics $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is considered to be know during the mapping, thus (6.1) is mentioned here only for completeness of the formalism and for comparison to the localization and mapping problems. Variable $\mathbf{h}_k$ represents a functional relationship between sensor output $\mathbf{y}_k$, target state vector $\mathbf{x}_k$ and system inputs $\mathbf{u}_k$ affected by a measurement noise $\mathbf{w}_k$. The map of the environment, i.e. the occupancy grid map, is represented by multi-dimensional matrix $\mathbf{m}_k$ (6.4) conditioned on the vehicle state $\mathbf{x}_k$ and sensor observations $\mathbf{y}_k$.

$$\mathbf{m}_k = \begin{bmatrix} m_k^{11} & m_k^{12} & \cdots & m_k^{1j} \\ m_k^{21} & m_k^{22} & \cdots & m_k^{2j} \\ \vdots & \vdots & \ddots & \vdots \\ m_k^{i1} & m_k^{i2} & \cdots & m_k^{ij} \end{bmatrix} \tag{6.4}$$

Each element of the grid matrix m_k^{ij} holds information about the probability that corresponding location on the map is occupied with a probability $p(m_k^{ij})$ ranging between $\{0, 1\}$. A cell of the grid map is denoted by m_k^{ij} where $i = \{1, \ldots, N_i\}$ represents the number of grid rows N_i and $j = \{1, \ldots, N_j\}$ number of grid columns N_j. The map building step (6.3) can be formulated as an optimization problem that maximize the probability that particular cell is in fact occupied (or free) $\hat{\mathbf{m}}_{1:k}$ given the prior knowledge of the environment $\mathbf{m}_{1:k-1}$, vehicle poses $\mathbf{x}_{1:k}$ and sensor measurements $\mathbf{y}_{1:k}$. In other words, $\hat{\mathbf{m}}_{1:k}$ can be computed as the maximum a posteriori prediction of a function

$$\hat{\mathbf{m}}_{1:k} = \arg\max_{\mathbf{m}_{1:k}} p(\mathbf{m}_k|\mathbf{m}_{1:k-1}, \mathbf{x}_{1:k}, \mathbf{y}_{1:k}). \tag{6.5}$$

The map $\hat{\mathbf{m}}_{1:k}$ in (6.5) represents the environment by fine-grained metric grids of cells and estimates the probability of any cell for being occupied depending on the sensor readings. Unlike a target tracking, the mapping do not only contain information about the presence of objects, but also about their absence (free spaces). Two components are required to find the grid occupancy probability $p(\mathbf{m}_k|\mathbf{m}_{1:k-1}, \mathbf{x}_{1:k}, \mathbf{y}_{1:k})$: a vehicle state and a measurement model. There is no explicit motion model defined in the case of mapping (6.1) since it is genuinely assumed that the maps do not change with time and therefore calculations (6.5) are only done on the basis of measurements. Generally speaking, it is not possible to directly estimate $\hat{\mathbf{m}}_{1:k}$ from the probability $p(\mathbf{m}_k|\mathbf{m}_{1:k-1}, \mathbf{x}_{1:k}, \mathbf{y}_{1:k})$ since there are $2^{[N_i \times N_j]}$ possible states to be estimated. Therefore, it is assumed that each cell is independent to each other, which allow to reduce the complexity to $2[N_i \times N_j]$ by reconstructing the map from products of the map's marginal probabilities

$$p(\mathbf{m}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \prod_{i}^{N_i} \prod_{j}^{N_j} p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}). \tag{6.6}$$

The probability that a single cell $p(m^{ij}_{1:k})$ is filled given the measurements $\mathbf{y}_{1:k}$ and poses $\mathbf{x}_{1:k}$ is computed by the Bayesian formula

$$p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|m^{ij}_{1:k}, \mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})\; p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1})}. \tag{6.7}$$

The terms in above equation (6.7) are denoted as the map estimate $p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k})$, the measurement likelihood $p(\mathbf{y}_k|m^{ij}_{1:k}, \mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$, the map prior $p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$ and the evidence or normalize factor $p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \mathbf{x}_{1:k})$. By assuming that current readings are independent of all previous states, e.g. by applying the 1^{st} order Markov assumption and by knowing the map prior, the probability of map occupancy (6.7) can be simplified to

$$p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|m^{ij}_{1:k}, \mathbf{x}_k)\; p(m^{ij}_{1:k}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1})}. \tag{6.8}$$

The Markov assumption requires $m^{ij}_{1:k}$ to hold the information about the entire map and time histories $\mathbf{m}_{1:k}$. However, this is not necessary true as $m^{ij}_{1:k}$ only represents the state of a single grid cell. There is no guarantee that an observation $\mathbf{y}_k$ is conditionally independent of all prior observations given only the state of a single cell, since standard sensors, such as LiDAR, necessarily couples observations passing through multiple grid cells. In regions close to the sensor origin, the number of LiDAR beams that contribute to the probability of a single cell could be quite large. In contrast, far from the origin a single beam could overlap multiple cells. For this reason, the $m^{ij}_{1:k}$ in equation (6.8) needs to be able to hold information about entire grid map $\mathbf{m}_{1:k}$. This is achieved by applying Bayesian rule on measurement likelihood term $p(\mathbf{y}_k|m^{ij}_{1:k}, \mathbf{x}_k)$ which leads to further factorization

$$p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k}) = \frac{p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)\; p(\mathbf{y}_k|\mathbf{x}_k)}{p(m^{ij}_0)} \frac{p(m^{ij}_{1:k}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1})}. \tag{6.9}$$

The initial knowledge for a cell ij for a map $\mathbf{m}_k$ is denoted by subscript 0, i.e. $p(m^{ij}_0)$. The factor $p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)$ is called the inverse sensor model in the literature. The probability that a particular cell is occupied is given by $p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k})$ defined by (6.9) while the probability that cell is free is $1 - p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k})$. Given these two probabilities the odds that a particular grid $p(m^{ij}_{1:k})$ is being occupied is given by

$$p(m^{ij}_{1:k}) = \frac{p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k})}{1 - p(m^{ij}_{1:k}|\mathbf{x}_{1:k}, \mathbf{y}_{1:k})}. \tag{6.10}$$

Substituting (6.9) to (6.10) leads to

$$p(m^{ij}_{1:k}) = \frac{p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)}{1 - p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)} \frac{1 - p(m^{ij}_0)}{p(m^{ij}_0)} \frac{p(m^{ij}_{1:k}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}{1 - p(m^{ij}_{1:k}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})}. \tag{6.11}$$

The odds ratio of a single hypothesis ensures that an occupancy probability will converge to $p(m^{ij}_k = 0.5)$, i.e. to the unknown state, when there is no evidence which will support the hypothesis. By applying logarithm to the occupancy grid computation (6.10) numerical errors from multiplying minuscule floating point numbers can be significantly reduced.

$$\log p(m^{ij}_{1:k}) = \log \frac{p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)}{1 - p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)} + \log \frac{1 - p(m^{ij}_0)}{p(m^{ij}_0)} + \log p(m^{ij}_{1:k-1}). \tag{6.12}$$

For this update rule, one needs only to specify $p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)$ e.g. the inverse sensor model, initial map $p(m^{ij}_0)$, and prior occupancy probability of a given cell $p(m^{ij}_{1:k-1})$. The iterative Bayesian filter update for computing the new cell estimate $p(m^{ij}_{1:k})$ can be recovered from the log odds representation (6.12) by the following equation

$$p(m^{ij}_k) = 1 - \frac{1}{1 + \exp^{\log p(m^{ij}_{1:k-1})}}. \tag{6.13}$$

6.3.2 Discrepancies Resolution

Commonly used approaches to grid fusion, i.e. Bayesian occupancy filter and Dempster-Shafer theory, are not able to judge the truthfulness of sensors readings. As a consequence, all measurement grids contribute equally to the final result, unless specified otherwise by the user. Empirically adjusted sensor weights are tailored only for certain scenarios and not at all suited for a general purpose mapping. Therefore, researchers devoted a significant effort into development of real time assessment strategies which allow to determine the sensor quality measure. The majority of these approaches recognize the Dempster-Shafer theory, which will be analyzed in detail in Sec. 6.3.3, as the main paradigm for reasoning under uncertainty [82], [109] and only few strategies were developed for Bayesian occupancy frameworks [165], [82]. Zhou et. al. [165] proposed a sensor fusion framework based on a linear combination of grids. The weights are computed as an optimization problem that minimizes the entropy of the fused grids by taking into account empirical data from sensors. Kumar et. al. [82] extended the baseline Bayesian formulation for an additional state, that represents the probability that a measurement is spurious. Whenever the measurements are inconsistent with each other the variance of the sensor distribution is increased, thus having a smaller contribution in the fusion process.

Our vision for automatic sensor weighting, depicted on the Fig. 6.4, is based on the assessment of grid discrepancies. The discrepancies (Fig. 6.2) are the measure of disagreement between a pair of sensors denoted as

$$\begin{aligned} p(\mathbf{m}_{k,vis.\ vs.\ con.}) &= p(\mathbf{m}_{k,vision}) - p(\mathbf{m}_{k,context}),\\ p(\mathbf{m}_{k,vis.\ vs.\ LiD.}) &= p(\mathbf{m}_{k,vision}) - p(\mathbf{m}_{k,LiDAR}),\\ p(\mathbf{m}_{k,LiD.\ vs.\ con.}) &= p(\mathbf{m}_{k,LiDAR}) - p(\mathbf{m}_{k,context}). \end{aligned} \tag{6.14}$$

The knowledge of discrepancies can be used to tip the probability of $uncertain$ cells, i.e. $p(m^{ij}_k) \approx 0.5$, to either *occupied* $p(m^{ij}_k >> 0.5)$ or $free$ $p(m^{ij}_k << 0.5)$ state via the situation assessment feedback. As outlined in the introductory Sec. 6.1.1, we utilize our expert knowledge of the environment to formulate the set of rules which are unique to the sensor setup, but general enough to be valid for wide range of on-road scenarios. We are particularity concerned about the detection of obstacles (new jerseys) which might occur on the road and lanes which determine the road boundaries. These objects are likely source of uncertainties as their existence is in conflict with at least one sensor reading at any given time. By assuming that new jerseys and road lanes might exist in the environment, respective rules (6.15), evaluated on bases of discrepancies (6.14) depicted in the Fig. 6.5, affirm their existence.

$$p(m^{ij}_k) = \begin{cases} occupied, & if \quad (p(\mathbf{m}_{k,vis.\ vs.\ con.}) < 0.5) \wedge (p(\mathbf{m}_{k,vis.\ vs.\ LiD.}) >= 0.5) \wedge \\ \text{(new-jersey)} & \qquad (p(\mathbf{m}_{k,LiD.\ vs.\ con.}) >= 0.5), \\ occupied, & elseif \quad (p(\mathbf{m}_{k,vis.\ vs.\ con.}) > 0.5) \wedge (p(\mathbf{m}_{k,vis.\ vs.\ LiD.}) < 0.5) \wedge \\ \text{(road lane)} & \qquad (p(\mathbf{m}_{k,LiD.\ vs.\ con.}) < 0.5), \\ free, & else. \end{cases} \tag{6.15}$$

Above rules (6.15) are applied to every cell whose probability is $p(m^{ij}_k) \approx 0.5$, i.e. $uncertain$ state in such a way, that initial probability of a cell $p(m^{ij}_k)$ will be increased above the uncertainty threshold $p(m^{ij}_k) > (0.5 + \delta)$ if *occupied* or decreased below the occupancy threshold $p(m^{ij}_k) < (0.5 - \delta)$ if $free$. Parameter δ specifies the interval of uncertainty as stated by (6.49), where $2\delta = 0.1$. The rule based reasoning is not uncommon in the occupancy grid conflict resolution as application of probabilistic reasoning techniques require implementation of image processing techniques on the grids. Recognizing shapes from the occupancy grids is challenging due to high variations of shapes caused by interchanging cell probabilities [95].

6.3.3 Dempster-Shafer Theory of Evidence

In the occupancy grid mapping problem every cell is assumed to be either $occupied$ (O) or $free$ (F). For every cell m_k^{ij} of grid map $\mathbf{m}_k$ the frame of discernment (FOD) (Fig. 6.6 right) $m_k(\mathcal{X}) = \{O, F\}$ and its power set 2^{Ω} are defined as

$$2^{\mathcal{X}} = \{\emptyset, O, F, \Omega\}. \tag{6.16}$$

The sets $\emptyset$ and Ω represent the $null$ set and the $conflict$ set, respectively. The former will always have a mass function equal to zero $m_k(\emptyset) = 0$, since a cell must be in any of the states $m_k \in \{O, F, \Omega\}$ defined in the FOD (6.17). The latter set is especially interesting since it represents the status of a cell being neither free F nor occupied O but in conflict Ω. In the Bayesian framework, one can model this situation by assigning $p(m_k^{ij}) = 0.5$, but there is no measure of confidence in that statement. On the other hand, the Dempster-Shafer framework is capable to express uncertainties, that the cell might have an unknown $\emptyset$ or conflicting state Ω, with a certain probability. The implementation of the Dempster-Shafer framework for a set $\{\emptyset, O, F, \Omega\}$ requires to maintain only two grids $m_k(O)$ and $m_k(F)$ associated to the occupied O and free F states of a cell for each sensor reading $\{m_{k(1)}(O), m_{k(2)}(O), \ldots, m_{k(n)}(O)\}$ and $\{m_{k(1)}(F), m_{k(2)}(F), \ldots\, m_{k(n)}(F)\}$, respectively. Even though the power set $2^{\mathcal{X}}$ is composed of four propositions, it can be seen from the basic probability assignment (BPA) theorem that masses $m_k(O)$ and $m_k(F)$ are sufficient to fully describe the whole FOD (6.16).

$$\begin{aligned} \sum_{A \in 2^{\mathcal{X}}} m_k(A) = m_k(\emptyset) + m_k(O) + m_k(F) + m_k(\Omega) = 1, \\ m(\Omega) = m_k(O \cup F) = 1 - m_k(O) - m_k(F), \\ m_k(\emptyset) = 0. \end{aligned} \tag{6.17}$$

The belief X of FOD $2^{\mathcal{X}}$ is a resulting probability that accounts for all the evidence that supports the proposition X. It represents the degree to which X, in our case occupied or free $X \in \{O, F\}$, is believed to be true.

$$Bel(X) = \sum_{A \subseteq X} m(A). \tag{6.18}$$

On the other hand, the plausibility of set $X \in \{O, F\}$ of FOD $2^{\mathcal{X}}$ considerate the evidence that does not provide knowledge about the preposition X. It therefore represents the degree to which X is believed not to be false.

$$Pl(X) = 1 - \sum_{A \cap X = \mathcal{X}, A \in 2^{\mathcal{X}}} m(A). \tag{6.19}$$

According to the DST, two sources of information $m_{k(1)}$ and $m_{k(2)}$ are combined into a fused belief function $m_{k(12)}$ through the operator $\oplus$, which could represent either conjunction $\cap$ or disjunction $\cup$ formula, in accordance with $X \neq \emptyset$ (6.20) and $X = \emptyset$ (6.21)

$$m_{k(12)}(X) = (m_{k(1)} \oplus m_{k(2)})(X) = \frac{(m_{k(1)} \cap m_{k(2)})(X)}{1 - (m_{k(1)} \cap m_{k(2)})(\emptyset)}, \tag{6.20}$$

$$m_{k(12)}(\emptyset) = (m_{k(1)} \oplus m_{k(2)})(\emptyset) = 0. \tag{6.21}$$

The $(m_{k(1)} \cap m_{k(2)})(X)$ is the conjunctive combination rule such that

$$(m_{k(1)} \cap m_{k(2)})(X) = \sum_{A \cap B = X; A, B \in 2^{\mathcal{X}}} m_{k(1)}(A)\, m_{k(2)}(B). \tag{6.22}$$

The denominator of (6.20) is a normalization factor where the $(m_{k(1)} \cap m_{k(2)})(\emptyset)$ is the measure of a conflict between sources of information. If combining information is highly contradicting, term $(m_{k(1)} \cap m_{k(2)})(\emptyset)$ becomes close to 1 and denominator of (6.20) become close to zero. That is the fundamental weakens in the DST theory and an underlaying topic of criticism. Furthermore, it can be shown that the Dempster-Shafer combination rule is commutative and associative. Therefore, the fusion of n sensors $\{m_{k(1)}, m_{k(2)}, \ldots, m_{k(n)}\}$ can be performed sequentially, as follows

$$m_{k(12n)}(X) = ((m_{k(1)}(X) \oplus m_{k(2)}(X)) \oplus \ldots) \oplus m_{k(n)}(X). \tag{6.23}$$

The DST formula (6.20) for a grid $m_{k(1)}(X)$ and $m_{k(2)}(X)$ can be interpreted for a belief $Bel(X = O)$ and $Bel(X = F)$ as stated by formulas (6.24) and (6.25), respectively.

$$m_{k(12)}(O) = \frac{m_{k(1)}(O)m_{k(2)}(O) + m_{k(1)}(O)m_{k(2)}(\Omega) + m_{k(1)}(\Omega)m_{k(2)}(O)}{1 - m_{k(1)}(O)m_{k(2)}(F) - m_{k(1)}(F)m_{k(2)}(O)} \tag{6.24}$$

$$m_{k(12)}(F) = \frac{m_{k(1)}(F)m_{k(2)}(F) + m_{k(1)}(F)m_{k(2)}(\Omega) + m_{k(1)}(\Omega)m_{k(2)}(F)}{1 - m_{k(1)}(O)m_{k(2)}(F) - m_{k(1)}(F)m_{k(2)}(O)} \tag{6.25}$$

The conflict masses $m_{k(1)}(\Omega)$ and $m_{k(1)}(\Omega)$ are expressed as

$$\begin{aligned} m_{k(1)}(\Omega) &= m_{k(1)}(O \cup F) = 1 - m_{k(1)}(O) - m_{k(1)}(F), \\ m_{k(2)}(\Omega) &= m_{k(2)}(O \cup F) = 1 - m_{k(2)}(O) - m_{k(2)}(F). \end{aligned} \tag{6.26}$$

6.3.4 Proportional Conflict Redistribution PCR5

The idea behind the proportional conflict redistribution (PCR) is to transfer conflicting masses of FOD $2^{\mathcal{X}}$ proportionally to non-empty sets according to all integrity constraints [130], [44], [65]. For instance, Dempster's rule distributes the conflicting mass equally according to the total conflicting mass through the normalization step. The DSmH rule transfers the partial conflicts onto partial uncertainties. The combination formula for PCR5 rule for fusion of measurement grids $m_{k(1)}(X)$ with $m_{k(2)}(X)$ is defined as follows

$$m_{k(12)}(X) = m_{k(12)}(X) + \sum_{Y \in 2^{\mathcal{X}} \setminus \{X\}; c(O \cap F) = \Omega} \frac{m_{k(1)}(X)^2\, m_{k(2)}(Y)}{m_{k(1)}(X) + m_{k(2)}(Y)} + \frac{m_{k(2)}(X)^2\, m_{k(1)}(Y)}{m_{k(2)}(X) + m_{k(1)}(Y)} \tag{6.27}$$

where $m_{k(12)}(X)$ corresponds to the conjunctive consensus on X between the two sources and where all denominators are different from zero and $c(X)$ is the canonical form of X. Canonical term can be in its simplest form expressed for example $X = (O \cap F) \cap (O \cup F \cup \Omega)$ as $c(X) = O \cap F)$. If a denominator in one of the terms (6.27) is zero, that fraction is discarded. Mathematically, PCR5 does a better redistribution of the conflicting mass than Dempsters rule and other rules since PCR5 goes backwards on the tracks of the conjunctive rule and redistributes the partial conflicting masses only to the sets involved in the conflict and proportionally to their masses put in the conflict, considering the conjunctive normal form of the partial conflict. PCR5 is quasi-associative and preserves the neutral impact of the vacuous belief assignment.

A single step of the PCR5 algorithm for three independent sensor readings starts with evaluation of the conjunctive rules of the DST belief masses $m_{k(12)}(O)$, $m_{k(12)}(F)$ and $m_{k(12)}(\Omega) = m_{k(12)}(O \cup F)$ according to formulas (6.24), (6.25) and (6.26), respectively. Based on this formulas, a total conflicting mass can be obtained

$$\begin{aligned} m_{k(123)}(O \cap F) =& [m_{k(12)}(O)\, m_{k(3)}(F) + m_{k(3)}(O)\, m_{k(12)}(F)] + [m_{k(3)}(O)\, m_{k(12)}(O \cap F) + \\ & + m_{k(3)}(F)\, m_{k(12)}(O \cap F) + m_{k(3)}(O \cup F)\, m_{k(12)}(O \cup F)]. \end{aligned} \tag{6.28}$$

Partial conflicting mass $m_{k(12)}(O \cap F)$ is calculated as

$$m_{k(12)}(O \cap F) = 1 - m_{k(12)}(O) - m_{k(12)}(F) - m_{k(12)}(O \cup F), \tag{6.29}$$

where $m_{k(12)}(O \cup F) = m_{k(1)}(O \cup F)\, m_{k(2)}(O \cup F)$ and individual conflicts are computed as in (6.26). The partial conflicting mass gains are evaluated based on formulas (6.30) and (6.31).

$$\hat{m}_{k(1)}(O \cap F) = \hat{m}_{k(1)}(\Omega) = \frac{m_{k(1)}(\Omega)}{m_{k(1)}(O) + m_{k(2)}(F)} \tag{6.30}$$

$$\hat{m}_{k(2)}(O \cap F) = \hat{m}_{k(2)}(\Omega) = \frac{m_{k(2)}(\Omega)}{m_{k(2)}(O) + m_{k(1)}(F)} \tag{6.31}$$

The final step of the PCR5 algorithm [130] is to redistribute the conflicting masses proportionally on non-empty sets (6.32), (6.33) and (6.34),

$$m_{k(123)}(O) = m_{k(12)}(O) + \hat{m}_{k(11)}(\Omega) + \hat{m}_{k(12)}(\Omega) + m_{k(3)}(O) + m_{k(12)}(O \cap F), \tag{6.32}$$

$$m_{k(123)}(F) = m_{k(12)}(F) + \hat{m}_{k(21)}(\Omega) + \hat{m}_{k(22)}(\Omega) + m_{k(3)}(F) + m_{k(12)}(O \cap F), \tag{6.33}$$

$$m_{k(123)}(O \cup F) = m_{k(12)}(O \cup F) + m_{k(3)}(O \cap F)\, m_{k(12)}(O \cap F), \tag{6.34}$$

involved in the model according to all integrity constraints (6.35) and (6.36), respectively.

$$\hat{m}_{k(11)}(\Omega) = m_{k(1)}(O) + \hat{m}_{k(1)}(\Omega) \qquad \hat{m}_{k(12)}(\Omega) = m_{k(2)}(O) + \hat{m}_{k(2)}(\Omega) \tag{6.35}$$

$$\hat{m}_{k(21)}(\Omega) = m_{k(1)}(F) + \hat{m}_{k(1)}(\Omega) \qquad \hat{m}_{k(22)}(\Omega) = m_{k(2)}(F) + \hat{m}_{k(2)}(\Omega) \tag{6.36}$$

6.4 Experimental Results

Information about drivable spaces is, based on the decentralized multi-layer multi-grid fusion architecture (Fig. 6.4), obtained in three distinct stages: temporal fusion, multiple-grid fusion and conflict resolution. The significance of these steps for overall map accuracy was discussed in the introductory Sec. 6.1.1, and therefore we will turn our attraction towards implementation and evaluation aspects of the architecture. At the temporal fusion level, sensor readings are transformed into the occupancy grids by the Bayesian update algorithm. The BOF operates under assumption that surrounding environment is static (Sec. 6.3.1). However, this assumption could be easily violated when relative velocities between the observer and detected objects are not negligible, e.g. when the area of interest is mapped by a moving vehicle. Forgetting factor, a design parameter, was introduced to the mapping process (Fig. 6.3) in order to mitigate the effects of non-static environments on temporal fusion. The main focus of this work is on the multiple grid fusion and conflict resolution. For this reason, we seek to fuse grids which provide different perspective on the environment, which are not necessarily supportive, yet complementary. A good example of these grids is shown in the Fig. 6.1 and consequently in the Fig. 6.9, where detections perceived by a pair, i.e. vision sensor and LiDAR observe new-jerseys on the road, will be not confirmed by the third one, i.e. context map (GIS) which does not include the information about road reconstruction. However, the 2 vs.1 sensor rule does not guarantee the truthfulness of the map and thus independent sensor weighting should be considered. We will show, that empirically adjusted weights are feasible for solutions when detection of certain objects are preferred over others, e.g. new-jerseys deployed on the road. Methods for automatic grid weighting require, to a certain extend, implementation of model based predictors, thus leaning towards feature-based mapping implementations. Instead of estimating the sensor weights, nowadays approaches to the occupancy mapping primarily focus on the conflict resolution based on the measure of discrepancy between each sensor pair. The later strategy is adopted for the purpose of this work. Furthermore, we will present simple, but efficient, set of rules that can significantly improve the accuracy of built maps via situation assessment feedback. However, the applicability of these rules is limited to the specific sensor setup, i.e visual detector, LiDAR and context map. Furthermore, we will present a comparative study between Bayesian occupancy filtering (BOF) approaches and Dempster-Shafer theories of evidence (DST) in an automotive driving scenario. The presented scenario (Fig. 6.9) poses multiple challenges for the mapping algorithm as the vehicle is tasked to detect obstacles and successfully avoid these obstacles by entering the opposing lane, while at the same time the map needs to correctly update drivable spaces from the contradicting senor readings. The accuracy of occupancy grid mapping solutions was evaluated on the dataset consisting from LiDAR observations, lane detections and geodetic reference points of all observed features. In the following Sec. 6.4.1 and Sec. 6.4.2 we will summarize the implementation of the grid mapping architecture (Fig. 6.4) and evaluate the accuracy of each fusion method, respectively.

6.4.1 Simulation Design

By following the block diagram (Fig. 6.4) from left to the right, the process of map building starts with definition of inputs. Three types of sensors were considered for the task in hand: LiDAR, visual detectors (camera) and contextual (GIS) map. As stated by the temporal fusion formula (6.9), occupancy grid $\mathbf{m}_k$ is built on the bases of the inverse measurement likelihood $p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)$ and the prior map $p(m^{ij}_{1:k-1})$ or $p(m^{ij}_{1:k}|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1})$. Each sensor likelihood $p(m^{ij}_{1:k}|\mathbf{x}_k, \mathbf{y}_k)$ for time k carries the knowledge about the vehicle position $\mathbf{x} = [p^x\ p^y]^T$, defined in local-level frame (ground truth) g, and sensor originated measurements $\mathbf{y} = [y^x\ y^y]^T$, defined in the vehicle frame b. Information about vectors $\mathbf{x}_k$ and $\mathbf{y}_k$ associated with each sensor type at time stamps $1 : k$ was presented to us in a form of dataset. The dataset, intellectual property of Magneti Marelli S.p.A., was generated by a multi-sensor data fusion solution deployed on the autonomous driving vehicle operating in the geo-referenced environment. Format of the vehicle state vector and the measurements for every detected object $o \in \{1,\ldots,N_o\}$ and feature $f \in \{1,\ldots,N_f\}$ is defined as follows

$$\begin{aligned} \mathbf{y}^{b,\{1:N_o\}}_{k,LiDAR} &= \big[k, \mathbf{p}(A)^{b,o}_k, \mathbf{p}(B)^{b,o}_k, \mathbf{p}(C)^{b,o}_k, \mathbf{p}(D)^{b,o}_k, \text{ID}\{o\}\big], \\ \mathbf{y}^{b,\{1:N_f\}}_{k,camera} &= \big[k, \mathbf{p}(A)^{b,f}_k, \mathbf{p}(B)^{b,f}_k, \mathbf{p}(C)^{b,f}_k, \mathbf{p}(D)^{b,f}_k, \text{ID}\{f\}, \text{class}\{f\}\big], \\ \mathbf{p}^{g,\{1:N_f\}}_{k,reference} &= \big[k, p^{gx,f}_k, p^{gy,f}_k, \text{lane width}\{f\}, \text{lane curviture}\{f\}, \text{lane ID}\{f\}\big], \\ \mathbf{x}^g_k &= \big[k, p^{gx}_k, p^{gy}_k, v^{gx}_k, v^{gy}_k, \theta(yaw)\big]. \end{aligned} \tag{6.37}$$

In above equations, points $\mathbf{p}(A,\ldots,D)^{b,o}_k$ and $\mathbf{p}(A,\ldots,D)^{b,f}_k$ define the surface area of objects and features detected by LiDARs and cameras observed in body frame b, respectively. Road lanes are subset of all visible features, position of which is precisely located with respect to the geo-referenced map $\mathbf{p}^{g,\{1:N_f\}}_{k,reference}$. We are not allowed to disclosure the details about sensor fusion solution, neither discuss sensor deployment and their characteristics, based on which the dataset was built. However, we argue that knowledge of these characteristics is not crucial for grid mapping evaluation, as we are more concern about appearance of objects within the grid map rather than their exact location w.r.t. ground truth, which will always be depended on the grid scale.

By using a grid scale of $1:4$ one meter of the real world plane is represented by 4 evenly spaced grid cells. The total size of the area represented by a single grid map $\mathbf{m}_k$ is $120m \times 120m = 14400m^2$, which is equivalent to a grid of 480×480 cells (Fig. 6.7). The cell of the grid map is denoted by m^{ij}_k where $i = \{1,\ldots,N_i\}$ with represents the number of grid rows $N_i = 480$ and $j = \{1,\ldots,N_j\}$ with number of grid columns $N_j = 480$. The vehicle is positioned in the center of the mesh allowing sensors to scan an area approximately (the vehicle itself occupies a volume in the grid) up to $60m$ from the vehicle in all 4 directions. Relationships between vehicle and grid frames are visualized in the Fig. 6.7 below. If the surface area of an object extends across multiple cells, those under the object's surface will be declared as occupied.

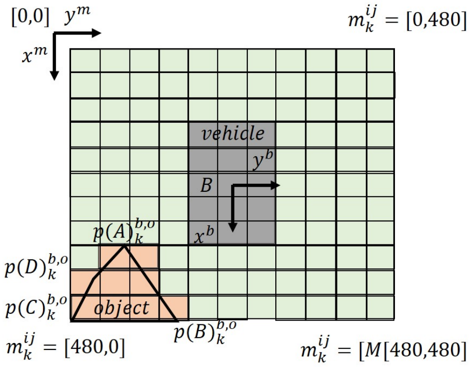


Figure 6.7: Visualization of the occupancy map.

LiDAR and camera readings (6.37) are translated into the grid (Fig. 6.7) by the following process. First, all data points $\mathbf{p}(A,\ldots,D)^{b,o}_k$ at time k are translated $\mathbf{T}^m_b$, offset by the grid origin, and scaled $\mathbf{S}^m_b = 0.4$ from the body frame b to grid frame m, i.e.

$$\mathbf{p}(A,\ldots,D)^{b,m}_k = \mathbf{T}^m_b + \mathbf{S}^m_b \mathbf{p}(A,\ldots,D)^{b,o}_k \tag{6.38}$$

is applied for $\forall\{o\}$ at time k. Body and map frames are aligned to each other thus no rotations are necessary. Second, points belonging to a single object $\mathbf{p}(A,\ldots,D)^{b,m}_k$ occupy certain cells in the grid, thus $\big[p(m^{\mathbf{p}(A)}_k),\ldots,p(m^{\mathbf{p}(D)}_k)\big] = 1$ is assumed to be true. In order to represent the shape of an object in the grid space $\mathbf{m}_k$, we apply the Bresenham algorithm to mark the cell that form a close approximation to a straight line between two points, that is $m^{\mathbf{p}(A)}_k \xrightarrow{\text{Bresenham}} m^{\mathbf{p}(B)}_k$,..., $m^{\mathbf{p}(C)}_k \xrightarrow{\text{Bresenham}} m^{\mathbf{p}(D)}_k$, $\forall\{o\}$ at time k. Third, the surface area defined by points $m_k^{\mathbf{p}(A)}, \dots, m_k^{\mathbf{p}(D)}$ is assumed to be occupied, and for this reason the flood fill algorithm was applied to mark appropriate grid cells under the surface integral $p\big(m_k^{\iint_s o(x,y)ds}\big) = 1$ for $\forall\{o\}$ object surfaces at time k. The process of transforming features f, i.e. feature points $\mathbf{p}(A,\dots,D)_k^{b,f}$, into visual grids follows the same process as defined for the objects o with one exception. The feature vectors also hold information about object classes, which in our case are classified as road lanes, static or dynamic objects (Fig. 6.1a). The distinction of classes is only important during the performance evaluation, and therefore all visual detections are set to be occupied with the maximum probability, i.e. $p\big(m_k^{\iint_s f(x,y)ds}\big) = 1$ for $\forall\{f\}$ feature surfaces at time k.

The context grid (Fig. 6.1c) was built artificially from the road reference points $\mathbf{p}_{k,ref.}^{g,\{1:N_f\}}$ defined in the ground coordinates (6.37). These data points are for $\forall\{f\}$ at time k translated $\mathbf{T}_g^i$ and scaled $\mathbf{S}_g^i$ from the ground frame g to the intermediate grid frame i by

$$\mathbf{p}_k^{i,f} = \mathbf{T}_g^i + \mathbf{S}_g^i \mathbf{p}_k^{g,f}. \tag{6.39}$$

After, a rotation $\mathbf{R}_i^m$ around the heading angle θ from intermediate to map frame (6.40) was performed.

$$\mathbf{p}_k^{m,f} = \mathbf{R}_i^m \mathbf{p}_k^{i,f} \tag{6.40}$$

Once completed, corresponding grid points $\mathbf{p}_k^{m,f}$ were connected by the Bresenham line algorithm and the off-road spaces classified as occupied by the flood fill process.

A temporal fusion step (Fig. 6.3), in other words the Bayesian occupancy filtering (BOF), is identical for all three measurement sources and follows the derivations established in the Sec. 6.3.1. In short, the log-odds representation of the BOF map update $p(\mathbf{m}_k)$, expressed by (6.12), consist of three logarithmic functions corresponding to the inverse sensor likelihood function $\log \frac{p(\mathbf{m}_k|\mathbf{x}_k,\mathbf{y}_k)}{1-p(\mathbf{m}_k|\mathbf{x}_k,\mathbf{y}_k)}$, the initial map $\frac{1-p(\mathbf{m}_0)}{p(\mathbf{m}_0)}$ and the prior $\log p(\mathbf{m}_{k-1})$. Generation of inverse likelihood grids $p(\mathbf{m}_k|\mathbf{x}_k,\mathbf{y}_k)$ was discussed in above paragraphs. Occupancy probabilities of the initial grid frame $p(\mathbf{m}_0)$ for each temporal fusion block (Fig. 6.4) corresponds to those of the contextual (GIS) $p(\mathbf{m}_{k,context})$ map at $k = 1$. The prior grid $p(\mathbf{m}_{k-1})$ is initialized as uninformative occupancy probability matrix with elements $p(m_{k-1}^{ij}) = 0.5$. The forgetting factor is modeled as an exponential time delay for each element

$$p(\mathbf{m}_k) = (p(\mathbf{m}_k) - 0.5)^{(-\frac{dk}{\tau}+0.5)}, \tag{6.41}$$

where the time period is equal to the measurement update $dk = 10^{-6}s$ and the time delay was chosen to be $\tau = 10^{-4}s$. By increasing or decreasing the parameter τ one increases or decreases speed the probability of a cell is updated by new measurements.

The focal element of the grid fusion architecture (Fig. 6.4) is a fusion process itself. Two predominant approaches, the Bayesian occupancy filtering (BOF) and Demspter-Shafer theory of evidence (DST), were employed for the task of the multi-sensor occupancy grid mapping. Until this point, BOF (Sec. 6.3.1) was presented as the solution of choice for a single sensor grid building. However, the likelihood in the Bayesian update formula (6.9) can be easily exchanged for a function of multiple measurements combined with a specific fusion formula. More specifically, linear opinion pool (LOP) [1]

$$\begin{aligned} p(\mathbf{m}_k) = \mathbf{w}_k^{-1}\big(\mathbf{w}_{k,LiDAR}\; p(\mathbf{m}_{k,LiDAR}) + \mathbf{w}_{k,vision}\; p(\mathbf{m}_{k,vision}) + \mathbf{w}_{k,context}\; p(\mathbf{m}_{k,context})\big) \\ \mathbf{w}_k = \mathbf{w}_{k,LiDAR} + \mathbf{w}_{k,vision} + \mathbf{w}_{k,context} \end{aligned} \tag{6.42}$$

and, very conservative, maximum logarithm of the occupancy (ML)

$$p(\mathbf{m}_k) = \max f\big(p(\mathbf{m}_{k,LiDAR}), p(\mathbf{m}_{k,vision}), p(\mathbf{m}_{k,context})\big) \tag{6.43}$$

were tested in the evaluation scenario. In above equations, term $\mathbf{w}_{k,sensor}$ represents the weight of an individual sensor and $\mathbf{w}_k$ is the normalization factor. Here, the sensor weights are chosen empirically based on the expert knowledge of the scenario. It has been concluded from the experimentation, that LiDAR should be slightly preferred over visual sensors, and that both sensors should be trusted more than contextual map, in order to emphasize detections not included in the map. Thus, the weights were set up in a ratio $\mathbf{w}_{k,LiDAR} : \mathbf{w}_{k,vision} : \mathbf{w}_{k,context} = 2 : 1.7 : 1$. Design strategies that can analyze the performance of each sensor in real time and assign some quality measure to it are especially interesting for grid fusion. Our vision (Fig. 6.4) to the automatic sensor weighting rely on conflict resolution and assessment techniques, rather than sensor entropies optimization [165] or inclusion of probabilities that a particular measurement is being spurious [82]. The process of a conflict resolution starts with the determination of discrepancies (Fig. 6.2), i.e. by solving the differences (6.14). The knowledge of discrepancies is used to generate hypothesis (6.15) about a particular cell being occupied or free. Cells classified as the unknown are evaluated by the hypothesis and their probability is adjusted to either occupied or empty state (6.49). Combination of the linear opinion pool (6.42) with the conflict assessment (6.15) is presented as the third formula (LOP-SA) for the Bayesian occupancy fusion.

Alternative to BOF, the Dempster-Shafer theory (Sec. 6.3.3) offers the possibility to express states and uncertainties in the form of belief functions (6.17, 6.18) and model transitions between these beliefs (Fig. 6.6). The initial assignment to the $occupied$, $free$ and $uncertian$ masses is performed for each sensor according to the formulas (6.44), (6.45) and (6.46), respectively.

$$m_k^{ij}(O) = \begin{cases} p(m_k^{ij}), & if \quad p(m_k^{ij}) > 0.55, \\ 0, & otherwise \end{cases} \tag{6.44}$$

$$m_k^{ij}(F) = \begin{cases} p(m_k^{ij}), & if \quad p(m_k^{ij}) < 0.45, \\ 0, & otherwise \end{cases} \tag{6.45}$$

$$m_k^{ij}(\Omega) = \begin{cases} p(m_k^{ij}), & if \quad 0.45 \leq p(m_k^{ij}) \leq 0.55, \\ 0, & otherwise \end{cases} \tag{6.46}$$

The Demspter-Shafer combination rule (6.23) is commutative and associative which allows to combine LiDAR, vision and context measurements by computing the following masses

$$\begin{aligned} m_k(O) &= (m_{k(LiDAR)}(O) \oplus m_{k(vision)}(O)) \oplus m_{k(context)}(O), \\ m_k(F) &= (m_{k(LiDAR)}(F) \oplus m_{k(vision)}(F)) \oplus m_{k(context)}(F), \\ m_k(\Omega) &= (m_{k(LiDAR)}(\Omega) \oplus m_{k(vision)}(\Omega)) \oplus m_{k(context)}(\Omega). \end{aligned} \tag{6.47}$$

For instance, the DST fusion formula (6.20) for a grid $m_{k(LiDAR)}(X)$ and $m_{k(vision)}(X)$ can be interpreted for a belief $Bel(X = O)$ and $Bel(X = F)$ as (6.24) and (6.25), respectively. For example, the fusion of LiDAR and vision sensor reads as

$$\begin{aligned} m_{k(LiD.\oplus Vis.)}(O) =& \frac{m_{k(LiD.)}(O)m_{k(vis.)}(O) + m_{k(LiD.)}(O)m_{k(vis.)}(\Omega) + m_{k(LiD.)}(\Omega)m_{k(vis.)}(O)}{1 - m_{k(LiD.)}(O)m_{k(vis.)}(F) - m_{k(LiD.)}(F)m_{k(vis.)}(O)}, \\ m_{k(LiD.\oplus Vis.)}(F) =& \frac{m_{k(LiD.)}(F)m_{k(vis.)}(F) + m_{k(LiD.)}(F)m_{k(vis.)}(\Omega) + m_{k(LiD.)}(\Omega)m_{k(vis.)}(F)}{1 - m_{k(LiD.)}(O)m_{k(vis.)}(F) - m_{k(LiD.)}(F)m_{k(vis.)}(O)}, \\ m_{k(LiD.)}(\Omega) =& m_{k(LiD.)}(O \cup F) = 1 - m_{k(LiD.)}(O) - m_{k(LiD.)}(F), \\ m_{k(vis.)}(\Omega) =& m_{k(vis.)}(O \cup F) = 1 - m_{k(vis.)}(O) - m_{k(vis.)}(F). \end{aligned} \tag{6.48}$$

Aforementioned fusion process can be visualized by the schematics (Fig. 6.8). When readings from a pair of sensors, i.e. LiDAR and visual detectors, are highly conflicting, denumerators in (6.48) could become close to zero which could lead to the misinterpretation of the environment and to divisions by a zero. In order to prevent potential issues we have enhanced the DST formula with the proportional conflict redistribution algorithm (PCR5) (Sec. 6.3.4). In comparison to the DST, PCR evaluates total conflicting mass of all sensors (6.28) as opposed to the partial masses occurring between the pairs. Transfer of the conflicting mass within FOD $2^{\mathcal{X}}$ follows derivations (6.30 - 6.36), which proportionally reassigns conflicting mass $m_k(\Omega)$ to either occupied $m_k(O)$ or free $m_k(F)$ state in accordance to the individual sensor contribution.

The ultimate goal of the occupancy grid mapping is to determine the presence of obstacles on the road. Therefore, a decision must be made to translate a probabilistic formulation into a logical state (6.49). The state of each cell m_k^{ij} can be one of the following: free, occupied or unknown.

$$m_k^{ij} = \begin{cases} free, & if \quad p(m_k^{ij}) < (0.5 - \delta), \\ unknown, & if \quad (0.5 - \delta) \leq p(m_k^{ij}) \leq (0.5 + \delta), \\ occupied, & if \quad p(m_k^{ij}) > (0.5 + \delta). \end{cases} \tag{6.49}$$

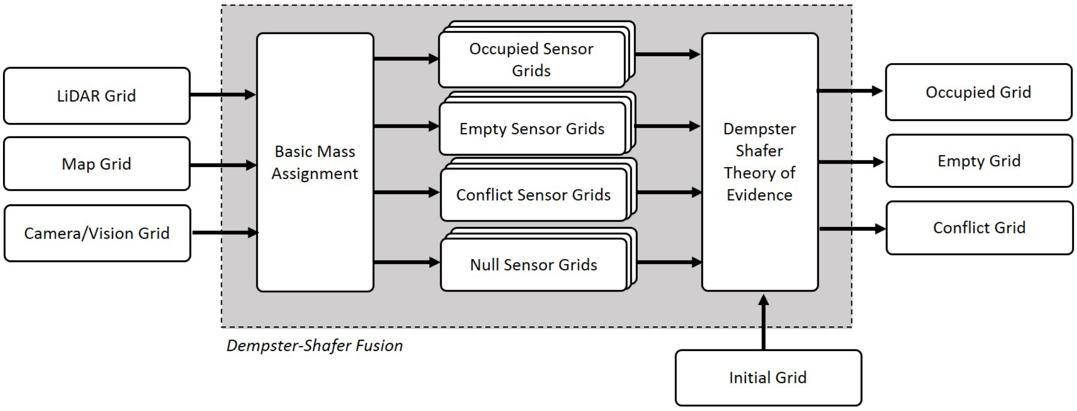


Figure 6.8: Visualization of the Dempster-Shafer fusion.

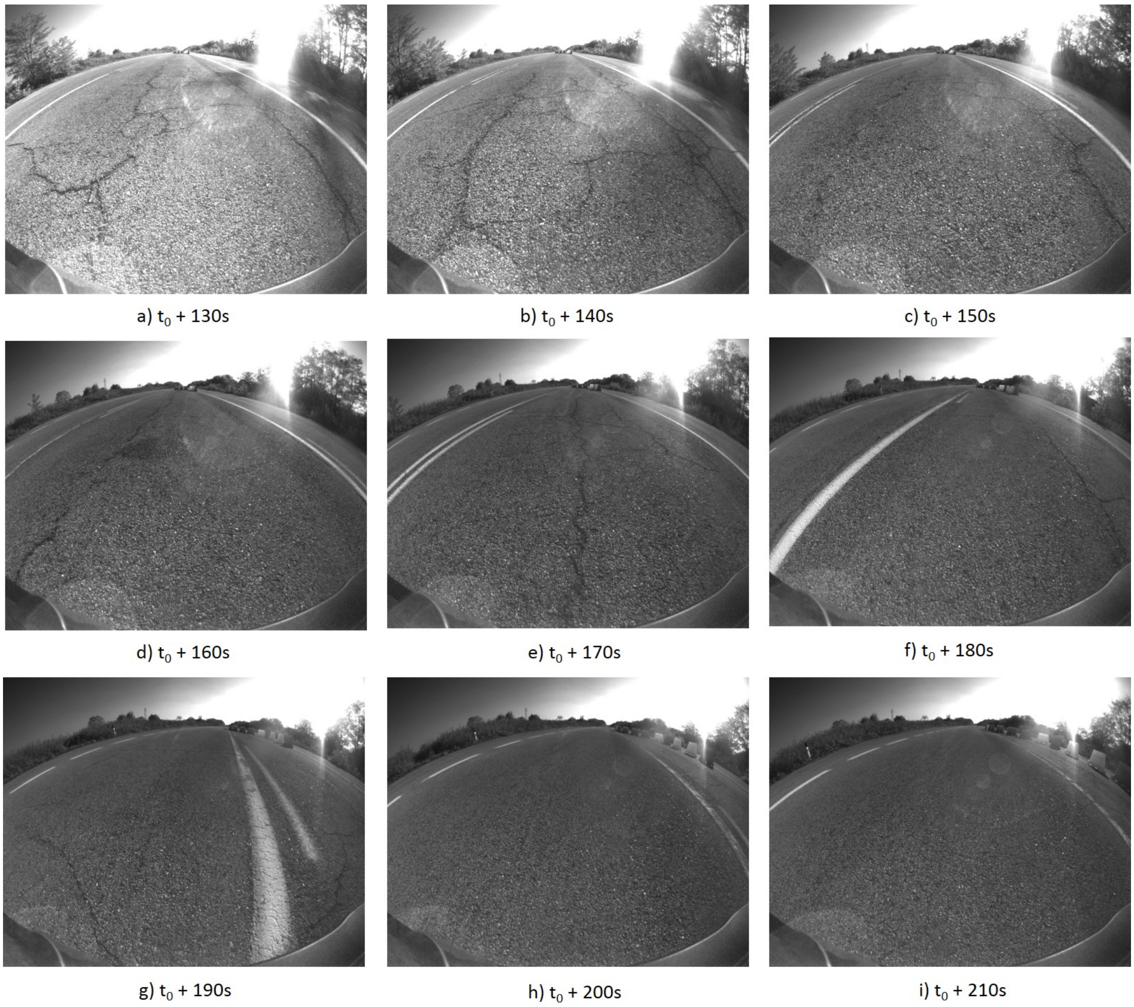
A small margin around $unknown$ state of size $2\delta = 0.1$ was chosen to address two common problems. First, probabilities rarely equals exactly 0.5, which is the definition of $unknown$. Second, a cell is classified as $free$ or $occupied$ only if the probability is high enough, so there is a reasonably high confidence. Choosing $\delta = 0.05$ is generally a fair assumption. If probability of occupancy reaches the value of 0 or 1, then the log-likelihood in (6.12) tend to converge to $\pm\infty$. For this reason, new sensor measurements would not contribute to occupancy grid update. Therefore, whenever an update is performed, it necessary to avoid the numerical instabilities by offset the probability of cells, e.g $p(m_k^{ij}) \in \{p(min) = 10^{-4}, p(max) = 1 - 10^{-4}\}$.

6.4.2 Results and Discussion

A scenario which is used for evaluation of Bayesian and Dempster-Shafer approaches to the occupancy grid mapping represents a typical road reconstruction challenge. Autonomous vehicle moving in the right-lane is tasked to detect the obstacles in front, i.e new jerseys, and avoid them by entering the opposing lane. As visualized on the frame sequence (Fig. 6.9) from time t_0+130s till t_0+160s vehicle is approaching the new-jerseys. From time $t_0 + 170s$ till $t_0 + 200s$ vehicle is crossing the middle lane. From time $t_0 + 210s$ onwards the vehicle is moving alongside the obstacle. Sections of the trajectory which are particularly interesting to analyze occur at times

- t_0+170s, when the vehicle is approaching the obstacles. The grid cells belonging to the new-jerseys should be classified as occupied with the high confidence.
- $t_0 + 190s$, when the vehicle is about to perform an evasive maneuver. The grid cells belonging to the new-jerseys and area behind them should be continuously classified as occupied. Furthermore, the cells describing the opposing lane should be classified as free.
- t_0+210s, when the vehicle is driving alongside the obstacle. Similarly to the conditions at t_0+190s, the grid cells belonging to the new-jerseys and area behind them should continuously classified as occupied. The area in front of the vehicle, now driving in the left lane, should be classified as free.

Towards the goal, we first analyze the temporal fusion grids for each individual sensor at the time stamps t_0+170s, t_0+190s and t_0+210s depicted in Fig. 6.10, Fig. 6.11 and Fig. 6.12, respectively. Starting with the time stamp $t_0 + 170s$, snapshots of the ground reference (Fig. 6.10a) and results of temporal fusion for context map (Fig. 6.10b), vision detector (Fig. 6.10c) and LiDAR (Fig. 6.10d,e) are presented. The ground truth is a surf plot of LiDAR reference points corresponding to the obstacles, yellow rectangles, and plot of the references associated with road lanes expressed in the vehicle body frame b. In the figure three type of lanes can be recognized: side lanes painted by a dark blue color, middle lane indicated as cyan, and marking of obscured area are depicted as red. The context map (Fig. 6.10b) indicates the drivable area only and the rest of the grid map is marked as not accessible. The context map does not pose the knowledge about road reconstruction and prevents usage of the opposing lane. All detected visual features


a) t_0 + 130s

b) t_0 + 140s

c) t_0 + 150s

d) t_0 + 160s

e) t_0 + 170s

f) t_0 + 180s

g) t_0 + 190s

h) t_0 + 200s

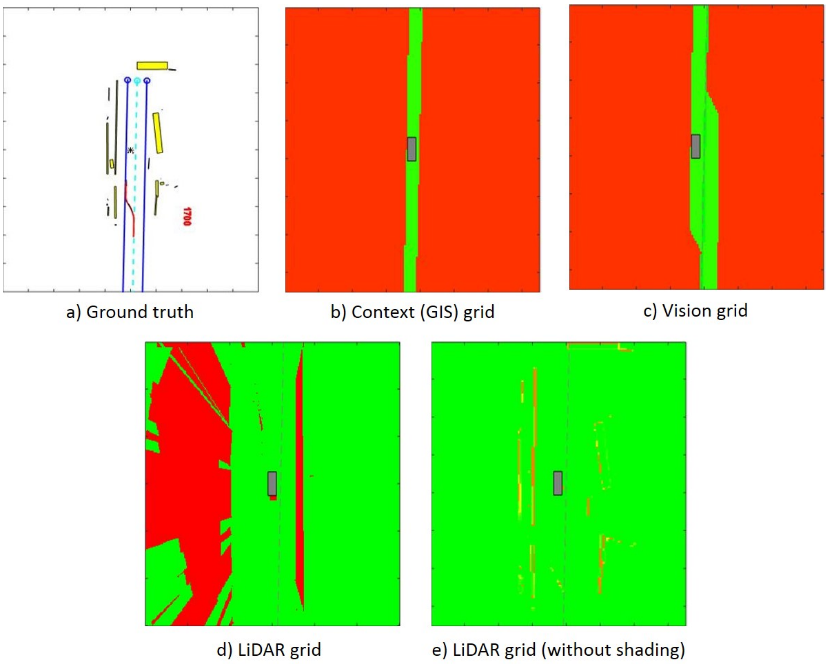
i) t_0 + 210s

Figure 6.9: Visualization of the evaluation scenario.

(Fig. 6.10c), which can be classified as road lanes, static or dynamic objected are represented as occupied with exception of drivable spaces. LiDAR detections are shown in the Fig. 6.10d and Fig. 6.10e, were new jerseys and off-road objects (trees and bushes) are represented by occupancy grids with and without shadow cones, respectively. Removing the cones of vision allows for easier comparisons of LiDAR detections with the ground truth. Analogous description can be adopted for sensors grids at time stamps $t_0 + 190s$ and $t_0 + 210s$.

Having understood sensor grids, which serve as inputs to the fusion module, three alternatives to the Bayesian occupancy filtering and two Dempster-Shafer theories will be evaluated based on the following criteria:

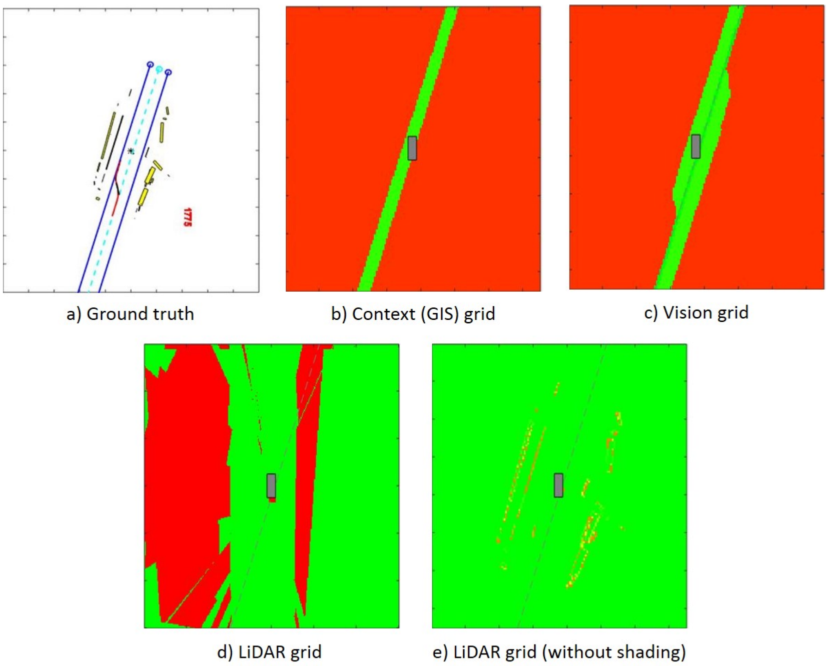
- Criterion 1: new-jerseys located in the front of the car in the traffic lane, observed by LiDAR and visual detectors, should be classified as occupied;
- Criterion 2: obscured areas detected by a lane detector behind the new-jerseys should be classified as occupied;
- Criterion 3: opposing lane in front of the car should be classified as free;
- Criterion 4: off-road areas should be classified as occupied;
- Criterion 5: miss detections should be sparse or not occur at all.



a) Ground truth b) Context (GIS) grid c) Vision grid

d) LiDAR grid e) LiDAR grid (without shading)

Figure 6.10: Measurement grids at the time stamp $t_0 + 170s$

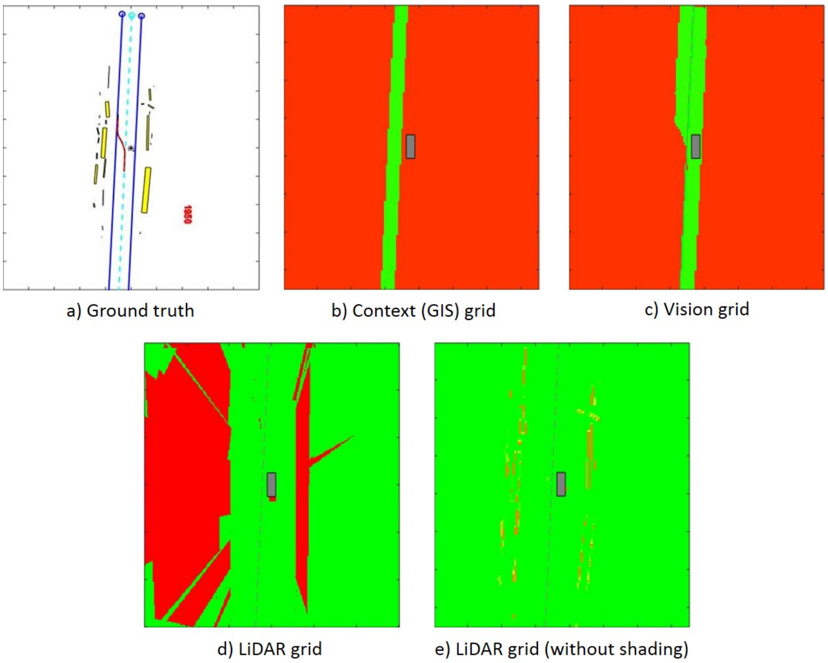


a) Ground truth b) Context (GIS) grid c) Vision grid

d) LiDAR grid e) LiDAR grid (without shading)

Figure 6.11: Measurement grids at the time stamp $t_0 + 190s$

The capability of a particular fusion solution to truthfully represent the environment at time stamps $t_0 + 170s$, $t_0 + 190s$, $t_0 + 210s$ is visualized on Fig. 6.13, Fig. 6.14, Fig. 6.16) and commented in Tab. 6.1, Tab. 6.2, Tab. 6.3, respectively. The ability of a certain fusion solution to reflect on the criteria either $successfully$ comply to them, or $fail$ them, or be $uncetain$ about the criterion. Sensor grids were weighted in a ratio $\mathbf{w}_{k,LiDAR} : \mathbf{w}_{k,vision} : \mathbf{w}_{k,context} = 2 : 1.7 : 1$ for full length of the experiment.

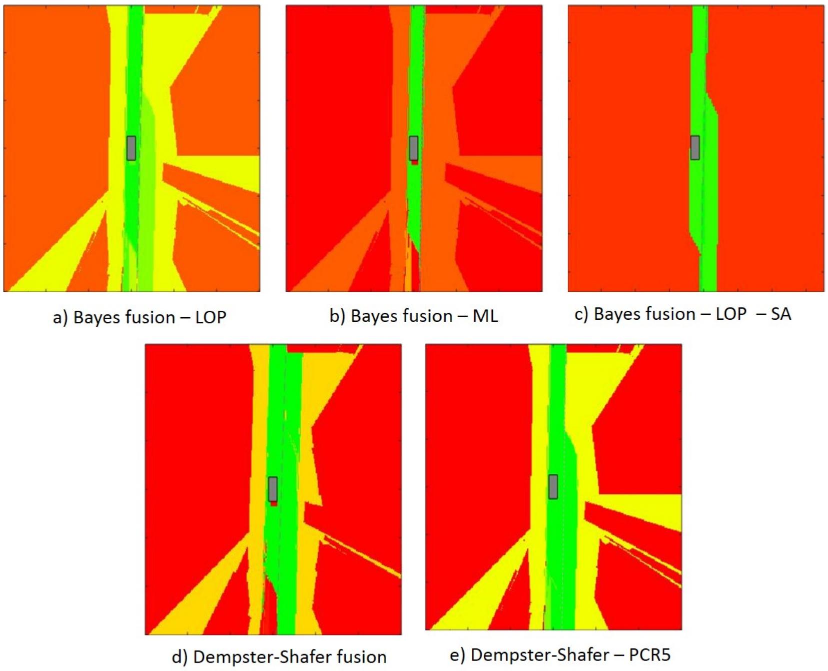


a) Ground truth b) Context (GIS) grid c) Vision grid

d) LiDAR grid e) LiDAR grid (without shading)

Figure 6.12: Measurement grids at the time stamp $t_0 + 210s$

By analyzing the grids (Fig. 6.13) at time $t_0 + 170s$ following conclusions could be made about the fusion solutions (Tab. 6.1). The Bayesian algorithm with linear opinion pool fusion formula (Fig. 6.13a) tempt to produce a quite $uncertain$ representation of the scene. This is due to the fact that information about road reconstruction is not updated in the context map but only observed by visual detectors and



a) Bayes fusion – LOP b) Bayes fusion – ML c) Bayes fusion – LOP – SA

d) Dempster-Shafer fusion e) Dempster-Shafer – PCR5

Figure 6.13: Fused grids at the time stamp $t_0 + 170s$

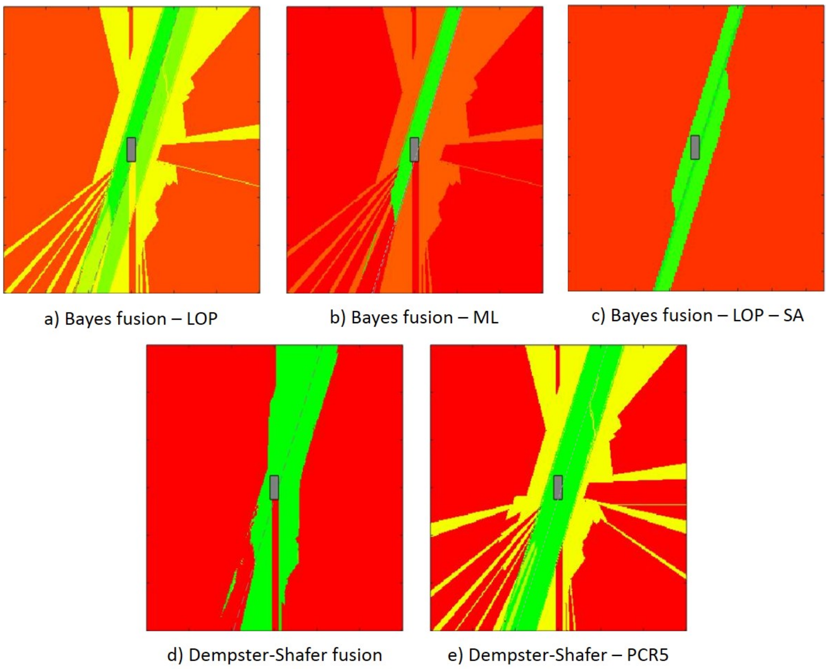
LiDAR. Furthermore, both sensors observe only partial occupancy of the grids at the new-jerseys location as they are position at the top of the vehicle and can see above these obstacles. By employing the situation assessment feedback into the LOP mapping (Fig. 6.13c), areas previously classified as $unknown$ are now recognized as either $occupied$ or $free$. Utilization of the SA module for mapping proved to be invaluable as the $BF-LOP-SA$ approach produces the most accurate map of the scenario among all tested solutions. On the other hand, the $BF-ML$ mapping is the most conservative approach to map building as a cell of the output grid is the one with the highest probability of occupancy from all available inputs 6.13b). Solution is clearly not able to recognize a possibility of driving in the opposing lane. The Dempster-Shafer fusion 6.13d) utilizes the conjunction of three masses in order to determine the drivable spaces. Resulting occupancy map is surprisingly accurate given the amount of conflict which occurs between senors grid. Finally, the $DST-PCRR5$ is the only solution which is automatically adjusting the sensor weights. Despite providing rather mediocre result, it will be shown in further time instances that the confidence, that obstacles are in fact present, will increase as time progresses.

Fusion Method	Criterion 1	Criterion 2	Criterion 3	Criterion 4	Criterion 5
Bayes Filter LOP	*Fail* Occupancy probability is is very low.	*Uncertain* Detection with probability close to 0.5.	*Success* Detection with probability close to uncertain.	*Uncertain* Occupancy prob. is ambiguous in areas LiDAR sees as free and high in the rest.	*Uncertain* Miss detections are frequent but classified as as free and uncertain at most.
Bayes Filter ML	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Fail* Occupancy probability is very high.	*Success* Occupancy probability is high.	*Fail* Miss detections are frequent and classified as occupied.
Bayes Filter LOP - SA	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is low.	*Success* Occupancy probability is high.	*Success* Negligible amount of miss detections are present.
Dempster-Shafer	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is low.	*Success* Occupancy prob. is moderately high in areas LiDAR sees as free and high in the rest.	*Fail* Miss detections are frequent and classified as occupied.
Dempster-Shafer PCR5	*Uncertain* Detection with probability close to 0.5.	*Success* Occupancy probability is high.	*Success* Occupancy probability is low.	*Uncertain* Detection with probability is close to 0.5.	*Success* Negligible amount of miss detections are present.

Table 6.1: Evaluation of fused grids at the time stamp $t_0 + 170s$.

At time $t_0 + 190s$ to the simulation, the vehicle is above to start an evasive maneuver. In this moment, the possibility of driving in the opposing lane should be recognized and the presence of new-jerseys should be affirmed by the mapping algorithm (Fig. 6.14). Unsurprisingly, the Bayesian occupancy filtering enhanced with the conflict resolution capability $(BOF-LOP-SA)$ resolved any uncertainties in the $BF-LOP$ solution (Fig. 6.14a) and deviled the most accurate map (Fig. 6.14c). A fusion formula based on the maximum logarithm of occupancy proved to be unfeasible for the maneuver which follows (Fig. 6.14b). The reliability of the map, built by DST formula (Fig. 6.14d), starts to produce some contra-intuitive results as the conflict between masses rises. On the other hand, the proportional conflict redistribution rule (Fig. 6.14e), with automatic weights adjustment, starts to deliver results compatible to $BF-LOP$ mapping. In order to illustrate how conflicting the information from sensor grid appears in a DST fusion a 4D plot (Fig. 6.15) of occupancy map and the conflicting masses is presented. In the figure, XY plane represents the result of $PCR5$ occupancy grid mapping with probabilities ranging between $0 < p(m_k^{ij}) < 1$ represented by a color bar from green to red. The conflict mass is represented by a Z plane. It can be observed from the Fig. 6.15, that driving in the opposing lane was declared feasible by $PCR5$ despite conflicting mass is particularly high in the area. Another highly conflicting cells can be identified under the new-jerseys and corresponding area in the left size of the road. For this reason, the DST started to provide contra-intuitive results (6.8). The final comments on the scenario $t_0 + 190s$ are presented in the Tab. 6.2.

In the last data sample (Fig. 6.16), we looking at the persistence of detected drivable space in the


a) Bayes fusion – LOP

b) Bayes fusion – ML

c) Bayes fusion – LOP – SA

d) Dempster-Shafer fusion

e) Dempster-Shafer – PCR5

Figure 6.14: Fused grids at time stamp $t_0 + 190s$

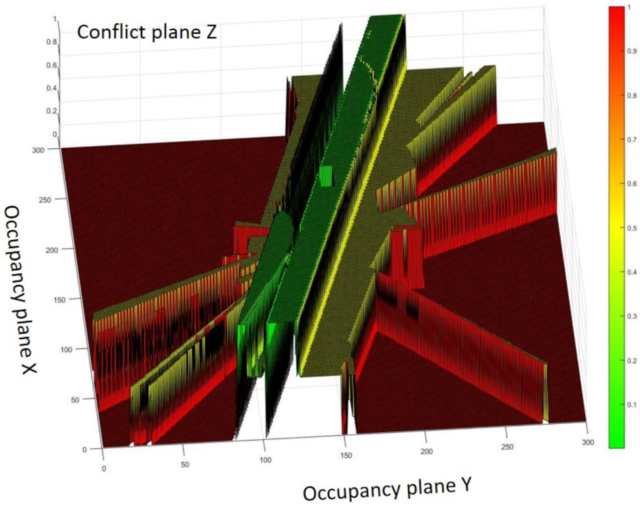


Figure 6.15: 4D Occupancy grids at time stamp $t_0 + 190s$

opposing lane and the consistency of obstacles located in the driving lane. The performance of fusion solutions in $t_0 + 210s$ is very similar to the one observed in the previous $t_0 + 190s$ snapshot, which one difference. The $PCR5$ fusion is gaining the confidence in way how the conflict mass is redistributed and thus continuously increasing its performance. The important observations are collected in the Tab. 6.3.

Fusion Method	Criterion 1	Criterion 2	Criterion 3	Criterion 4	Criterion 5
Bayes Filter LOP	*Fail* Occupancy probability is is very low.	*Uncertain* Detection with probability close to 0.5 or lower.	*Uncertain* Detection with probability close to 0.5 or declared as free	*Success* Detection with probability close to uncertain.	*Uncertain* Miss detections are frequent but classified as uncertain at maximum.
Bayes Filter ML	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is moderately high.	*Fail* Occupancy probability is very high.	*Fail* Miss detections are frequent and classified as occupied.
Bayes Filter LOP - SA	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy prob. is moderately high in areas LiDAR considers as free and high in the rest.	*Success* Miss detections are very rare.
Dempster-Shafer	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Fail* Occupancy probability is very low.	*Success* Occupancy probability is low.	*Fail* Miss detections are frequent and classified as occupied.
Dempster-Shafer PCR5	*Uncertain* Detection with probability close to 0.5.	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is low.	*Success* Miss detections are very rare.

Table 6.2: Evaluation of fused grids at the time stamp $t_0 + 190s$.

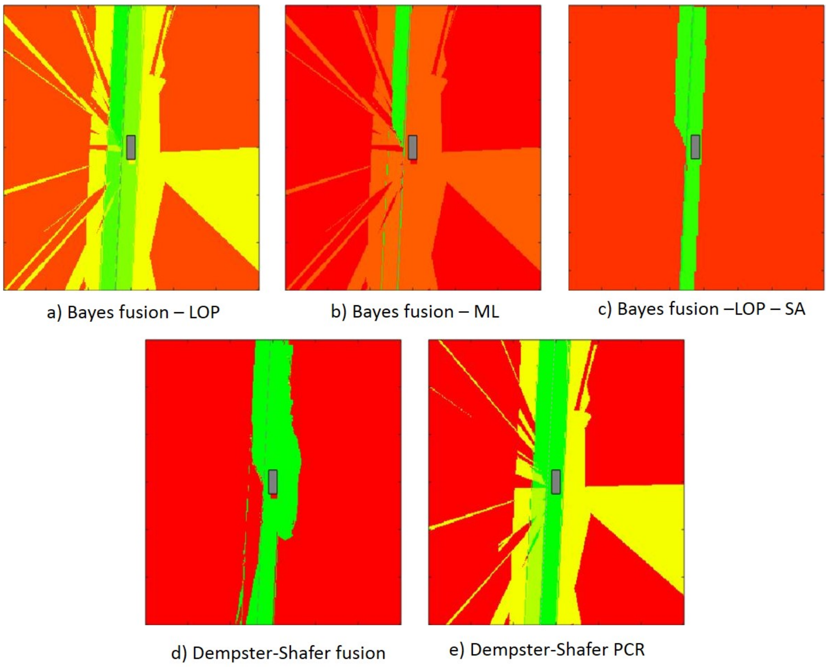

a) Bayes fusion – LOP
b) Bayes fusion – ML
c) Bayes fusion –LOP – SA
d) Dempster-Shafer fusion
e) Dempster-Shafer PCR

Figure 6.16: Fused grids at time stamp $t_0 + 210s$

6.5 Conclusion

We evaluated two most common algorithms for occupancy mapping, i.e. the Bayesian occupancy filter and the Demspter-Shafer theory of evidence, in the roadworks scenario. The mapping algorithm was tasked to build an occupancy grid map of the scenario by combining the readings from LiDAR, visual detectors and the context (GIS) map. Building maps from multiple grids proved to be challenging in cases when sensor readings are highly contradicting. Baseline approaches threats measurements equally, which in case of Bayesian solution leads to large amount of grids cell being classified as uncertain or in case of

Fusion Method	Criterion 1	Criterion 2	Criterion 3	Criterion 4	Criterion 5
Bayes Filter LOP	*Fail* Occupancy probability is is very low.	*Uncertain* Detection with probability close to 0.5 or lower.	*Uncertain* Detection with probability close to 0.5 or declared as free	*Success* Detection with probability close to uncertain.	*Uncertain* Miss detections are frequent but classified as uncertain at maximum.
Bayes Filter ML	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is moderately high.	*Fail* Occupancy probability is very high.	*Fail* Miss detections are common and classified as occupied.
Bayes Filter LOP - SA	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is low.	*Success* Miss detections are very rare.
Dempster-Shafer	*Success* Occupancy probability is high.	*Success* Occupancy probability is high.	*Fail* Occupancy probability is very low.	*Success* Occupancy probability is low.	*Fail* Miss detections are frequent and classified as occupied.
Dempster-Shafer PCR5	*Uncertain* Detection with probability close to 0.5.	*Uncertain* Occupancy probability is high.	*Success* Occupancy probability is high.	*Success* Occupancy probability is low.	*Success* Miss detections are very rare.

Table 6.3: Evaluation of fused grids at the time stamp $t_0 + 210s$.

the Dempster-Shafer theory to an unintuitive results. It therefore become apparent, that sensor weights should be adjusted either empirically or resolved by means of alternative fusion formulas and reasoning techniques.

By assessing the Bayesian fusion rules, the conservative maximum logarithmic rule did not meet the requirements as the autonomous vehicle would have stopped upon the obstacles were reached. Linear opinion pool showed, that empirical sensor weights are extremely hard to tune for whole simulation run. As a consequence large portions of the map were classified as uncertain. For this reason, we have developed the situation assessment logic, which manage to tip the probability to either occupied or free state. We proved, that simple yet efficient set of rules can significantly reduce amount of ambiguities in the map. Introduced, rule set works exceptionally well for the given sensor setup, i.e. LiDAR, camera, GIS map, for majority of on-road scenarios.

Both DST solutions were exposed to highly conflicting evidences, which implies that the PCR solution should have deliver more reliable information about drivable spaces. However the results shows, that baseline DST outperformed the PCR by a significant margin in capability to detect obstacles. However, the DST solution was prone to miss-classifications which hindered its performance. PCR was very close to the Bayesian fusion results and provided slightly higher confidence of obstacle detection. The choice of sensor weights have significant impact on the fusion performance and further analysis on optimal sensor weight should be performed. However, we argue that PCR increased its performance over time and its capability of determining grids weights independently is highly desirable. The solutions were tested with and without the prior grid knowledge. As opposed to Bayesian fusion, involvement of the posterior grid in the fusion process tends to degrade the performance of both approaches and appears to be unnecessary.

Conclusion

In this thesis, several concepts and issues to be taken into account when developing a context-aware information fusion system are discussed and analyzed. More specifically, an architecture capable of dynamical context exploitation at object and situation assessment fusion levels has been proposed and evaluated. The concept has been applied to ground tracking scenarios with airborne radars and to the autonomous system localization and mapping problem.

In the first ground target tracking scenario, the problem of a multi-level context representation and exploitation is addressed. The presented approach represents different types of spatial context in the form of likelihood functions. The likelihoods are applied as constrains into the particle filter measurement update. Given the scenario, the accuracy of a target estimate can be improved by approximately 27% w.r.t the conventional tracking solutions. Despite achieving promising results, there are multiple factors and limitations which need to be considered during algorithm design. First, the generation of accurate and information rich likelihood masks needs to be addressed. Employing the Bayes classifiers to cluster and partition contextual data proved to be very versatile by providing: the ability to couch different types of knowledge, a principled functional representation exploitable as likelihood in the tracking process, the ability to dynamically update contextual likelihoods by re-training. By enhancing the classification accuracy and robustness further performance gains can be achieved. Constraining particles in the measurement update could lead to large amount of particles with small or vanishing weights. By reducing the size of the likelihood function the Particle filter becomes less efficient and eventually lead to track losses. In order to address this issue, marginalized particle filters, i.e. the Rao-Blackwellized PF, should be considered for cases when the constrains become highly nonlinear.

In the second ground target tracking scenario, the target's actions are represented as a Hidden Markov model with relevant spatial and event-temporal context associated with each node and arc, respectively. Each Markov sequence represents an unique goal that a target seeks to reach within a certain graph topology. As the target progresses through the network, the target's future actions are inferred based on current evidence and a priori knowledge. The belief that a vehicle seeks an objective is used to aid the tracking process by adjusting the mode transition probabilities in the interacting multiple models (IMM) estimation process. The posterior elements of the TPM matrix are evaluated recursively, by a quasi-Bayesian estimator, in terms of the multiple models mode probabilities and the joint likelihood of contextual information and measurements. The convergence of the QB estimators is highly dependent on the target mode excitation, i.e. an amount of jumps observed between individual modes. For this reason, the informativeness of the likelihood is a key performance contributor in the recursive TPM estimation. The underlaying role of the context is twofold. First, the presence of CI significantly improves the convergence and reduces the steady state error of the estimated mode transition probabilities. Second, prior to an event CI adjusts the mode ambiguity of the IMM in such a way that occurrence of tacks losses is significantly reduced in cases when the target performs an unlikely maneuver. When the outcome of the event is uncertain or misjudged, the progressive context weighting gives the IMM filter enough time to reconsider the mode observation gains, i.e. to prioritize the measurement over context, and avoid track losses. By assessing the outcome of Monte Carlo simulations, the utilization of the adaptive TPM estimate improved the performance of the IMM filter by roughly 23.19% in a scalar jump Markov linear process tracking (JMLS). When the simulation complexity was extended to a 2D JMLS tracking, the adaptive solutions were able to maintain the advantage over the IMM filter by roughly 13.98% while context was present and by approximately 5.46% while context was absent. In the simulation of the ground object tracking with airborne sensors, the knowledge of road topology and target habits improved the track accuracy and continuity by roughly 5.32%. We argue, that the performance gain of the IMM-QB-HMM filter will improve over the time as the length of the event sequence, i.e. the amount and type of mode transients, increases. Arguably, accuracy and convergence speed of the TPM estimation process can be improved by numerical gradient methods, such as a maximal likelihood estimator, in exchange for the computational power. Probabilistic reasoning techniques based on HMM are quite limited in their ability to repent the context-temporal relationships. For this reason, events models are better represented as the dynamic Bayesian networks (DBN) or the conditional random fields (CRF). In such cases, it is recommended to swap the Kalman filter for the particle filter, which is better suited for inference in DBN networks and in propagating a non-linear event modes in the IMM framework.

Before discussing the subject of context-aware mapping, it is worth noting that event-context and event-temporal context can be also exploited in the process of data association while consolidating hypothesis about the measurement origin. Alternatively, the modal context combined with the kinematic measurements can be exploited in a form of the joint probability density of the likelihoods in so called, image-enhanced IE-IMM filter. According to the Bayesian recursion rule, updates of the hybrid state densities results a mixture of Gaussians with an exponentially increasing number of components. The challenge here is to develop a tractable approximate estimators that retain the informativeness of the joint likelihood and decrease the computation time. Alternatively, a Gaussian wavelet estimator (GWE), that retains the growth of the Gaussian mixture components by a hypothesis merging techniques, could be also considered.

In field of the autonomous system localization and mapping, grid-SLAM approaches utilizing the Bayesian occupancy filter and the Demspter-Shafer theory of evidence are evaluated in the road reconstruction scenario. Mapping algorithms are tasked to build an occupancy grid map of the scenario by combining the readings from LiDAR, visual detectors and context (GIS) map. Building maps from multiple grids proved to be challenging in the cases when sensor readings are highly contradicting. Baseline grid mapping approaches threats measurements equally, which in case of Bayesian solutions lead to large amount of grids cell being classified as an uncertain or in case of the Dempster-Shafer theory to an unintuitive results. It therefore become apparent that sensor weights should be adjusted either empirically or resolved by means of alternative fusion formulas and/or reasoning techniques. For these reasons, the situation assessment logic is proposed, which managed to tip the probability of uncertain grid elements to either occupied or free state. It was proved that even a simple yet efficient set of rules can significantly reduce the amount of ambiguities in the built map. However, the baseline DST outperformed the proportional conflict redistribution (PCR) by a small margin in the capability to detect obstacles. On the other hand the DST solution was prone to miss-classifications which hindered its performance. The PCR solution was very close to the Bayesian fusion results augmented with the situation assessment module and shows more confidence in detecting the obstacles. The choice of sensor weights has a significant impact on the fusion performance. Therefore, further analysis on the optimal sensor weight should be performed. However, it can be argued that the PCR increases its performance over time and its capability of determining the grids weights without an expert knowledge is highly desirable.

In order to improve the fidelity of the mapping process, the compilation of multiple mapping techniques should be considered, such as combination of feature and grid SLAM or multiple model DST rule frameworks. Extending the field of discernment in a DST framework for contextual states should be also considered.

Bibliography

[1] J. D. Adarve, M. Perrollaz, A. Makris, and C. Laugier. Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools. In *Transaction on Robotics and Automation*, pages 5–10, St Paul, Minnesota, 2012.

[2] S. C. Agate and J. K. Sullivan. Road-constrained target tracking and identification a particle filter. *Proceedings of SPIE*, 5204(805):532–543, 2003.

[3] V. Akman and M. Surav. The Use of Situation Theory in Context Modeling. *Computational Intelligence*, 13(3):427–438, aug 1997.

[4] J. Albusac, D. Vallejo, and J. J. Castro-Schez. Monitoring Complex Environments Using a Knowledge-Driven Approach Based on Intelligent Agents. In *Intelligent Systems,*, pages 24–31, 2010.

[5] S. Allam, F. Dufour, and P. Bertrand. Discrete-time estimation of a Markov chain with marked point process observations. Application to markovian jump filtering. *Transactions on Automatic Control*, 46(6):903–908, 2001.

[6] D. Angelova. Sequential Monte Carlo Algorithms for Joint Target Tracking and Classification Using Kinematic Radar Information. In *Proceedings of the 7th International Conference on Information Fusion*, pages 709–716. Stockholm, Sweden, 2004.

[7] S. M. Arulampalam, N. Gordon, M. Orton, and B. Ristic. A variable structure multiple model particle filter for GMTI tracking. In *Interantional Conference on Information Fusion*, pages 928–934. Annapolis, USA, 2002.

[8] J. Aulinas, Y. Petillot, J. Salvi, and X. Llado. The SLAM problem: A survey. In *Proceedings on Artificial Intelligence Research and Development*, pages 363–371, Amsterdam, The Netherlands, 2008. IOS Press.

[9] Q. Baig and O. Aycard. Low level data fusion of laser and monocular color camera using occupancy grid framework. In *11th International Conference on Control, Automation, Robotics and Vision*, number December, pages 905–910, Singapore, 2010.

[10] G. Battistello and M. Ulmke. Exploitation of a priori information for tracking maritime intermittent data sources. In *14th International Conference on Information Fusion*, pages 1–8, Chicago, Illinois, 2011.

[11] E. Bauer, F. Lotz, M. Pfromm, M. Schreier, B. Abendroth, S. Cieler, A. Eckert, A. Hohm, and P. U. Rieth. PRORETA 3 : An Integrated Approach to Collision Avoidance and Vehicle Automation. *Automatisierungstechnik*, 60(12):755–765, 2016.

[12] B. M. Bell, J. V. Burke, and G. Pillonetto. An inequality constrained nonlinear Kalman-Bucy smoother by interior point likelihood maximization. *Automatica*, 45(1):25–33, 2009.

[13] S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, Boston, 1999. ISBN 978-1580530064.

[14] E. Blasch, J. Schubert, K. B. Laskey, G. W. Ng, R. Nagi, and D. Stampouli. Issues of Uncertainty Analysis in High-Level Information Fusion. In *15th International Conference on Information Fusion*, Singapore, 2012.

[15] E. Blasch, P. Valin, A.-L. Jousselme, and D. Lambert. Top Ten Trends in High-Level Information Fusion. In *15th International Conference on Information Fusion*, pages 2323–2330, Singapore, 2012.

[16] E. Blasch, J. Garcia Herrero, L. Snidaro, J. Llinas, G. Seetharaman, and K. Palaniappan. Overview of contextual tracking approaches in information fusion. In *Proceedings of the SPIE*, page 87470B, Baltimore, may 2013.

[17] E. Blasch, A. N. Steinberg, S. Das, J. Llinas, O. Kessler, E. Waltz, and F. White. Revisiting the JDL Model for Information Exploitation. In *16th International Conference on Information Fusion*, pages 129–136, Istanbul, 2013.

[18] E. Blasch, C. Yang, J. Garciia, L. Snidaro, and J. Llinas. Contextual Tracking Approaches in Information Fusion. In *Context-Enhanced Information Fusion: Boosting Real-World Performance with Domain Knowledge*, pages 73–97. Springer International Publishing, Cham, 2016.

[19] E. P. Blasch and C. Yang. Ten Methods to Fuse GMTI and HRRR Measurements For Joint Tracking and Identification. In *7th International Conference on Information Fusion*, pages 1006–1013, Stockholm, Sweden, 2004.

[20] H. A. P. Blom and Y. Bar-Shalom. Interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33:780–783, 1988.

[21] C. L. Bowman. Process assessment and process management for intelligent data fusion & resource management systems. In *AIAA Space Conference & Exposion*. Pasadena, 2012.

[22] H. H. Bui. A general model for online probabilistic plan recognition. In *International Joint Conference on Artificial Intelligence*, pages 1309–1315, 2003.

[23] H. H. Bui, G. Venkatesh, and G. West. Policy Recognition in the Abstract Hidden Markov Model. *Journal of Artificial Intelligence Research*, 17(1):451–499, 2002.

[24] H. H. Bui, D. Q. Phung, and S. Venkatesh. Hierarchical hidden Markov models with general state hierarchy. In *Proceedings of the 19th national conference on Artifical intelligence*, number July, pages 324–329, 2004.

[25] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni. Static Force Field Representation of Environments Based on Agents' Nonlinear Motions. *Advances in Signal Processing*, 2017(1):13, 2017.

[26] L. Campo, Y. Bar-Shalom, and P. Mookerjee. State Estimation for Systems with Sojourn-Time-Dependent Markov Model Switching. *IEEE Transactions on Automatic Control*, 36(2):238–243, 1991.

[27] S. Challa and N. Bergman. Target tracking incorporating flight envelope information. In *Proceedings of the 3rd International Conference on Information Fusion*, volume 2, pages 22–27. IEEE, 2000.

[28] Y.-T. Chiang, L.-S. Wang, and F.-R. Chang. Filtering method for nonlinear systems with constraints. In *Proceedings on Control Theory and Applications*, volume 149, pages 525–531. IEE, 2002.

[29] L. Cuiping, S. Jinping, M. Shiyi, and L. Desheng. Tracking ground targets with road constraint using multiple hypotheses tracking. In *Proceedings of the 2nd International Conference on Signal Processing Systems*, volume 2, pages 265–269, Dalian, China, 2010.

[30] S. Das. Agent-based information fusion. *Information Fusion*, 11(3):216–219, 2010.

[31] A. J. Davison and D. W. Murray. Simultaneous Localization and Map-Building Using Active Vision. *Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.

[32] J. Degerman, T. Pernst, and K. Alenljung. 3D occupancy grid mapping using statistical radar models. In *Intelligent Vehicles Symposium*, number IV, pages 902–908, Gothenburg, Sweden, 2016. IEEE.

[33] M. Delafosse, L. Delahoche, A. Clerentin, V. Ricquebourg, and A. M. Jolly-Desodt. A Dempster-Shafer fusion architecture for the incremental mapping paradigm. In *10th International Conference on Information Fusion*, Quebec, Canada, 2007.

[34] A. K. Dey. *Providing Architectural Support for Building Context-Aware Applications Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, Georgia Institute of Technology, 2000.

[35] J. Dezert, F. Smarandache, A. Tchamova, and P. Konstantinova. Target type tracking with PCR5 and Dempster's rules: A comparative analysis. In *9th International Conference on Information Fusion*, pages 1–10, Florence, Italy, 2006.

[36] M. W. M. G. Dissanayake, P. Newman, S. Clark, and M. Csorba. A Solution to the Simultaneous Localisation and Map Building (SLAM) Problem. *Transactions on Robotics and Automation*, 17: 229–241, 2001.

[37] E. O. Drummond. Feature, Attribute, and Classification Aided Target Tracking. In *Procedings on Society of Photo-Optical Instrumentation Engineers*, page 542, San Diego, California, 2001. SPIE.

[38] F. H. Durrant-Whyte and T. Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *Robotics and Automation*, 2:1–9, 2006.

[39] A. Elfes. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer*, 22(6): 46–57, 1989.

[40] C. Estrada and J. D. Tard. Hierarchical SLAM : real-time accurate mapping of large environments. *Transactions on Robotics*, 21(4):588 – 596, 2005.

[41] S. J. Evans and J. R. Evans. Image-Enhanced Multiple Model Tracking. *Automatica*, 35(11):1769–1786, 1999.

[42] H. Fargetton and J.-G. Siedler. Control of multi sensor system based on anomaly mapping and expert system. Technical report, Berlin, Germany, 2011.

[43] P. G. C. Fernandez. *Grid-Based Multi-Sensor Fusion for On-Road Obstacle Detection: Application to Autonomous Driving*. PhD thesis, KTH Royal Institute of Technology, Stockholm, 2015.

[44] M. C. Florea, J. Dezert, P. Valin, F. Smarandache, and A.-L. Jousselme. Adaptative Combination Rule and Proportional Conflict Redistribution Rule for Information Fusion. In *COGnitive Systems with Interactive Sensors*, Paris, 2006.

[45] N. Floudas, A. Polychronopoulos, O. Aycard, J. Burlet, and M. Ahrholdt. High Level Sensor Data Fusion Approaches For Object Recognition In Road Environment. In *2007 IEEE Intelligent Vehicles Symposium*, pages 136–141, Istanbul, Turkey, 2007.

[46] J. Garcia, J. Gomez-Romero, M. Patricio, J. M. Molina, and G. Rogova. On the representation and exploitation of context knowledge in a harbor surveillance scenario. In *14th International Conference on Information Fusion*, Chicago, Illinois, 2011. IEEE.

[47] R. Garcia, O. Aycard, T.-D. Vu, and M. Ahrholdt. High level sensor data fusion for automotive applications using occupancy grids. In *10th International Conference on Control, Automation, Robotics and Vision*, number December, pages 530–535, Hanoi, Vietnam, 2008.

[48] F. Giunchiglia. Contextual Reasoning. In *Epistemologica - Special Issue on I Linguaggi e le Macchine*, pages 345–364. Trento, Italy, 1993.

[49] J. Gomez-Romero, J. Garcia, M. Kandefer, J. Llinas, J. M. Molina, M. A. Patricio, M. Prentice, and S. C. Shapiro. Strategies and Techniques for Use and Exploitation of Contextual Information in High-Level Fusion Architectures. In *13th International Conference on Information Fusion*, pages 1 – 8, Edinburg, 2010.

[50] J. Gómez-Romero, J. García, M. A. Patricio, J. M. Molina, and J. Llinas. High-Level Information Fusion in Visual Sensor Networks. In *Visual Information Processing in Wireless Sensor Networks*, pages 197–223. IGI Global, 2012.

[51] J. Gomez-Romero, A. M. Serrano, J. Garcia, M. J. Molina, and G. Rogova. Context-based multi-level information fusion for harbor surveillance. *Information Fusion*, 21(1):173–186, 2015.

[52] L. Gong. Contextual modeling and applications. In *International Conference on Systems, Man and Cybernetics*. IEEE, Waikoloa, Hawaii, 2005.

[53] P. Grays and D. Salber. Modeing and using sensed context information in the design of interactive applications. In *Engineering for Human-Computer Interaction*. Springer, Berlin, Heidelberg, 2001.

[54] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[55] R. Grover, G. Brooker, and F. H. Durrant-Whyte. A Low-level Fusion of Millimetre-Wave Radar and Nightvision Imaging for Enhanced Characterisation of a Cluttered Environment. In *Australian Conference for Robotics and Automation*, number November 2001, pages 14–15, Sidney, Australia, 2001.

[56] N. Gupta and R. Hauser. Kalman Filtering with Equality and Inequality State Constraints. Technical Report 07, 2007.

[57] F. Gustafsson. Particle filter theory and practice with positioning applications. *IEEE Aerospace and Electronic Systems Magazine*, 25(7):53–82, jul 2010.

[58] F. Gustafsson. *Statistical Sensor Fusion*. Holmbergs, Lund, 2010. ISBN 978-91-44-05489-6.

[59] F. Gustafsson, U. Orguner, T. B. Schon, P. Skoglar, and R. Karlsson. *Handbook of Intelligent Vehicles*. Springer London, London, 2012. ISBN 978-0-85729-084-7.

[60] D. L. Hall and J. Llinas. *Multisensor Data Fusion, Second Edition - 2 Volume Set*. CRC Press, London, 2001. ISBN 9780849323799.

[61] S. K. Hautaniemi and J. P. P. Saarinen. Multitarget tracking with the IMM and Bayesian networks : Empirical studies. In *Sensor Fusion: Architectures, Algorithms, and Applications V*, pages 47–57, Mar. 2001.

[62] S. Hofmann and C. Brenner. Quality assessment of automatically generated feature maps for future driver assistance systems. In *17th International Conference on Advances in Geographic Information Systems*, number C, pages 500–503, Washington, D. C., 2009.

[63] A. Hogger. *Dempster Shafer Sensor Fusion for Autonomously Driving Vehicles*. PhD thesis, KTH Royal Insitute of Technology, Stockholm, 2016.

[64] F. Homm, N. Kaempchen, J. Ota, and D. Burschka. Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 1006–1013, San Diego, California, 2010.

[65] R. Ilin and E. Blasch. Information Fusion with Belief Functions : A comparison of Proportional Conflict Redistribution PCR5 and PCR6 rules for Networked Sensors. volume 6, pages 2084–2091, Washington, DC, 2015.

[66] E. Ivanjko, I. Petrovic, and M. Brezak. Experimental Comparison of Sonar Based Occupancy Grid Mapping Methods. *Automatika*, 50:65–79, 2009.

[67] V. P. Jilkov and X. R. Li. Adaptation of Transition Probability Matrix for Multiple Model Estimators. In *Interantional Conference on Information Fusion*, pages 1–3, Montreal, Canada, 2001. IEEE.

[68] V. P. Jilkov and X. R. Li. Online Bayesian Estimation of Transition Probabilities for Markovian Jump Systems. *Transactions on Signal Processing*, 52(6):1620–1630, 2004.

[69] A.-L. Jousselme, K. Bryan, and M. Ilteris. Selection of use cases to support maritime situation awareness algorithmic development. Technical report, NATO Centre for Maritime Research and Experimentation (CMRE), La Spezia, Italy, 2015.

[70] M. Kang, S. Hur, W. Jeong, and Y. Park. Map Building Based on Sensor Fusion for Autonomous Vehicle. In *11th International Conference on Information Technology*, pages 490–495, Las Vegas, Nevada, 2014.

[71] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, 14(1):28–44, 2013.

[72] G. W. Kim and B. H. Lee. Hierarchical Sensor Fusion for Building an Occupancy Grid Map Using Active Sensor Modules. In *2006 SICE-ICASE International Joint Conference*, pages 2600–2605, Busan, South Korea, 2006.

[73] J. Kim and S. Sukkarieh. Airborne Simultaneous Localisation and Map Building. In *Proceedings on Robotics and Automation*, pages 406–411. IEEE, 2003.

[74] J. Kim and S. Sukkarieh. Autonomous Airborne Navigation in Unknown Terrain Environments. *Transactions on Aerospace and Electronic Systems*, 40(3):1031–1045, 2004.

[75] E. K. Kjaer. A Survey of Context-Aware Middleware. In *Procedings on Software Engineering*, pages 148–155, Anaheim, California, 2007.

[76] J. Knaup and K. Homeier. RoadGraph - Graph based environmental modelling and function independent situation analysis for driver assistance systems. In *13th International Conference on Intelligent Transportation Systems*, pages 428–432, Funchal, Portugal, 2010.

[77] S. Ko and R. R. Bitmead. State estimation for linear systems with state equality constraints. *Automatica*, 43(8):1363–1368, 2007.

[78] M. Konrad, M. Szczot, F. Schüle, and K. Dietmayer. Generic grid mapping for road course estimation. In *Intelligent Vehicles Symposium, Proceedings*, number 4, pages 851–856, Baden-Baden, Germany, 2011.

[79] M. Konrad, D. Nuss, and K. Dietmayer. Localization in Digital Maps for Road Course Estimation using Grid Maps. In *Intelligent Vehicles Symposium*, pages 88–92, Alcalá de Henares, Spain, 2012. IEEE.

[80] T. Kubertschak, M. Maehlisch, and H.-J. Wuensche. Towards a unified architecture for mapping static environments. In *17th International Conference on Information Fusion*, pages 1–8, Salamanca, Spain, 2014.

[81] M. Kumar, P. D. Garg, and R. Zachery. Multi-sensor fusion strategy to obtain 3-D occupancy profile. In *Proceedings on Industrial Electronics Conference*, volume 2005, pages 2083–2088, Raleigh, North Carolaina, 2005.

[82] M. Kumar, P. D. Garg, and A. R. Zachery. A method for judicious fusion of inconsistent multiple sensor data. *IEEE Sensors Journal*, 7(5):723–733, 2007.

[83] M. Kurdej, J. Moras, and P. Bonnifait. Map-Aided Evidential Grids for Driving Scene Understanding. *Intelligent Transportation Systems*, 7(1):33–41, 2015.

[84] H. Lategahn, W. Derendarz, T. Graf, B. Kitt, and J. Effertz. Occupancy grid computation from dense stereo and sparse structure and motion points for automotive applications. In *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 819–824, San Diego, California, 2010.

[85] J.-p. Lauffenburger. Conflict Management in Multi-sensor Dempster-Shafer Fusion for Speed Limit Determination. In *Strategy*, number IV, pages 985–990, Baden-Baden, Germany, 2011.

[86] G. H. Lee, F. Fraundorfer, and M. Pollefeys. RS-SLAM: RANSAC Sampling for Visual FastSLAM. In *International Conference on Intelligent Robots and Systems*, pages 1655–1660. IEEE, 2011.

[87] Y. Lemeret, E. Lefevre, and D. Jolly. Improvement of an association algorithm for obstacle tracking. *Information Fusion*, 9(2):234–245, 2008.

[88] J. Leonard, M. Walter, and R. Ballard. Visually Navigating the RMS Titanic with SLAM Information Filters. *Transactions on Robotics*, 22(6):1100–1114, 2006.

[89] J. J. Leonard and F. H. Durrant-Whyte. Mobile Robot Localization by Tracking Geometric Beacons. *Transactions on Robotics and Automation*, 7(3):376 – 382, 1991.

[90] R. Lherbier, B. Jida, J. C. Noyer, and M. Wahl. Use of contextual information by Bayesian Networks for multi-object tracking in scanning laser range data. *Proceedings on 9th International Conference onntelligent on Transport Systems Telecommunications*, 3(1):97–102, 2009.

[91] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target Tracking Part II: Ballistic Target Models. *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, (August):1–23, 2001.

[92] X. R. Li and V. P. Jilkov. A Survey of Maneuvering Target Tracking Part III : Measurement Models. *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, (August):1–24, 2001.

[93] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. Part V: Multiple-model methods. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4):1255–1321, 2005.

[94] L. Liao, D. Fox, and H. A. Kautz. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *The International Journal of Robotics Research*, 26(1):119–134, jan 2007.

[95] P. Lindner and G. Wanielik. Multi level fusion for an automotive pre-crash safety system. In *nternational Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 143–146, Seoul, South Korea, 2008.

[96] J. Llinas. A Survey and Analysis of Frameworks and Framework Issues for Information Fusion Applications. In *Hybrid Artificial Intelligence Systems*, volume 6077, pages 14–23. Springer Berlin Heidelberg, 2010.

[97] J. Llinas, C. L. Bowman, G. Rogova, A. Steinberg, E. Waltz, and F. White. *Revisiting the JDL Data Fusion Model II.* San Diego, 2004.

[98] V. Lohweg and U. Mönks. Sensor fusion by Two-Layer Conflict Solving. In *2nd International Workshop on Cognitive Information Processing*, number 5, pages 370–375, Naregno, Elba Island, Italy, 2010.

[99] E. Maggio and A. Cavallaro. Learning Scene Context for Multiple Object Tracking. *Transactions on Image Processing*, 18(8):1873–1884, 2009.

[100] D. Maksarov and F. H. Durrant-Whyte. Mobile vehicle navigation in unknown environments : a multiple hypothesis approach. *Control Theory and Applications*, 142(4):385 – 400, 1995.

[101] M. Mallick, S. Maskell, T. Kirubarajan, and N. Gordon. Littoral tracking using particle filter. In *Proceedings of the 5th International Conference on Information Fusion*, volume 2, pages 935–942, Annapolis, Maryland, 2002.

[102] E. D. Mart, J. Garcia, and J. L. Crassidis. Improving Multiple-Model Context-Aided Tracking through an Autocorrelation Approach. In *15th International Conference on Information Fusion*, number July, pages 1822–1829, Singapore, 2012.

[103] J. McCarthy. Notes on Formalizing Context. In *Proceedings of the 13th international joint conference on Artifical intelligence*, pages 555–560, Chambery, France, 1993.

[104] M. Mertens and M. Ulmke. Ground Moving Target Tracking with Context Information and a Refined Sensor Model. In *11th International Conference on Information Fusion*, volume 1, pages 1497–1504, Cologne, 2008.

[105] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.

[106] J. A. Montoya-Zegarra, J. D. Wegner, L. Ladicky, and K. Schindler. Mind the Gap: Modeling Local and Global Context in (Road) Networks. *German Conference on Pattern Recognition*, 8753: 212–223, 2014.

[107] J. Neira and J. D. Tardos. Data Association in Stochastic Mapping Using the Joint Compatibility Test. *Transactions on Robotics and Automation*, 17(6):890–897, 2001.

[108] D. Nuss, M. Stuebler, and K. Dietmayer. Consistent environmental modeling by use of occupancy grid maps, digital road maps, and multi-object tracking. In *IEEE Intelligent Vehicles Symposium, Proceedings*, number IV, pages 1371–1377, Dearborn, Michigan, 2014.

[109] D. Nuss, M. Thom, A. Danzer, and K. Dietmayer. Fusion of laser and monocular camera data in object grid maps for vehicle environment perception. In *11th International Conference on Information Fusion*, pages 1–8, Salamanca, Spain, 2014.

[110] U. Orguner and M. Demirekler. Maximum Likelihood Estimation of Transition Probabilities of Jump Markov Linear Systems. *IEEE Transactions on Signal Processing*, 56(10):5093–5108, 2008.

[111] G. Pallotta, M. Vespe, and K. Bryan. Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. *Entropy*, 15(6):2218–2245, 2013.

[112] F. Papi, M. Podt, and Y. Boers. On constraints exploitation for particle filtering based target tracking. In *15th International Conference on Information Fusion*, pages 455–462, Singapore, 2012.

[113] E. I. Paromtchik, M. Perrollaz, and C. Laugier. Fusion of telemetric and visual data from road scenes with a lexus experimental platform. In *Intelligent Vehicles Symposium*, number IV, pages 746–751, Baden-Baden, Germany, 2011.

[114] V. Pavlovic, J. M. Rehg, and T. Cham. A Dynamic Bayesian Network Approach to Tracking Using Learned Switching Dynamic Models. In *Hybrid Systems: Computation and Control*, number 617, pages 366–380. Springer Berlin Heidelberg, 2000.

[115] A. Plotnik and S. Rock. *Robotics Research*, volume 28 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-48110-2.

[116] C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimationa moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.

[117] C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *Transactions on Automatic Control*, 48(2): 246–258, 2003.

[118] D. B. Reid. An Algorithm for Tracking Multiple Targets. *Transactions on Automatic Control*, AC-24 (6):843–854, 1979.

[119] G. Rogova, M. Hadzagic, M. St-Hilaire, M. C. Florea, and P. Valin. Context-based information quality for sequential decision making. In *International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, pages 16–21, San Diego, feb 2013.

[120] M. J. Saez, A. Hogue, F. Escolando, and M. Jenkin. Underwater 3D SLAM through Entropy Minimization. In *Proiceedings on Robotics and Automation*, pages 3562–3567. IEEE, 2006.

[121] K. Sahba, E. K. Alameh, and L. C. Smith. Long Range Obstacle detection and spectral discrimination using multi-wavelength motionless wide angle laser scanning. *Optics express*, 16(8):5822–5831, 2008.

[122] J. Schubert. Conflict management in Dempster-Shafer theory using the degree of falsity. *International Journal of Approximate Reasoning*, 52(3):449–460, 2011.

[123] R. Schubert. Unifying Bayesian Networks and IMM Filtering for Improved Multiple Model Estimation. In *12th International Conference on Information Fusion*, pages 810–817, Seattle, WA, 2009.

[124] R. Schubert and G. Wanielik. A unified bayesian approach for tracking and situation assessment. In *2010 IEEE Intelligent Vehicles Symposium*, pages 738–745, June 2010.

[125] C. Seeger, M. Manz, P. Matters, and J. Hornegger. Locally Adaptive Discounting in Multi Sensor Occupancy Grid Fusion. In *Intelligent Vehicles Symposium*, number IV, pages 266–271, Gothenburg, Sweden, 2016. IEEE.

[126] E. Shahbazian, J. M. Deweert, and G. Rogova. *Findings of the NATO Workshop on Data Fusion Technologies for Harbour Protection*, pages 337–351. Springer Netherlands, Dordrecht, 2009. ISBN 978-1-4020-8883-4.

[127] D. Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303, 2010.

[128] D. Simon and T. L. I. Chia. Kalman filtering with state equality constraints. *Transactions on Aerospace and Electronic Systems*, 38(1):128–136, 2002.

[129] D. Simon and D. L. Simon. Constrained Kalman filtering via density function truncation for turbofan engine health estimation. *International Journal of Systems Science*, 41(2):159–171, 2010.

[130] F. Smarandache, J. Dezert, and O. D. Ied. Proportional Conflict Redistribution Rules for Information Fusion. In *8th International Conference on Information Fusion*, number 2, Philadelphia, 2005.

[131] M. F. A. Smith and E. U. Makov. A Quasi-Bayes Sequential Procedure for Mixtures. *Journal of the Royal Statistical Society*, 40(1):106–112, 1978.

[132] R. C. Smith, M. Self, and P. Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. *Autonomous Robot Vehicles*, pages 167–193, 1990.

[133] L. Snidaro, I. Visentini, J. Llinas, and G. L. Foresti. Context in fusion: some considerations in a JDL perspective. In *16th International Conference on Information Fusion*, pages 115–120. Istanbul, Tukey, 2013.

[134] L. Snidaro, J. García, and J. Llinas. Context-based information fusion: a survey and discussion. *Information Fusion*, 25:16–31, 2015.

[135] L. Snidaro, L. Vaci, J. Garcia, E. D. Marti, A.-L. Jousselme, K. Bryan, D. D. Bloisi, and D. Nardi. A framework for dynamic context exploitation. In *18th International Conference on Information Fusion*, volume 1, pages 1–18, Washington, DC, 2015. IEEE.

[136] L. Snidaro, I. Visentini, and K. Bryan. Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks. *Information Fusion*, 21:159–172, 2015.

[137] S. Soleimanpour, S. S. Ghidary, and K. Meshgi. Sensor fusion in Robot localization using DS-Evidence Theory with conflict detection using Mahalanobis distance. In *7th IEEE International Conference on Cybernetic Intelligent Systems*, pages 1–6, London, Unted Kingdom, 2008.

[138] T. Stang and C. Linnhoff-Popien. A context modeling survay. In *First International Workshop on Advanced Context Modeling, Reasoning and Management*. Nottingham, UK, 2004.

[139] A. N. Steinberg. Revisions to the JDL Data Fusion Model.pdf. In *Missile Sciences Conference*, Monerey, California, 1998.

[140] A. N. Steinberg. Context-Sensitive Data Fusion using Structural Equation Modeling. In *12th International Conference on Information Fusion*, pages 725–731, Seattle, 2009.

[141] A. N. Steinberg and C. L. Bowman. Adaptive Context Discovery and Exploitation, 2013.

[142] A. N. Steinberg and G. Rogova. Situation and Context in Data Fusion and Natural Language Understanding. In *11th International Conference on Information Fusion*, pages 1–8, Cologne, 2008.

[143] D. Streller. Road map assisted ground target tracking. In IEEE, editor, *Proceedings of the 11th International Conference on Information Fusion*, pages 1162–1168, Cologne, Germany, 2008.

[144] C. S. Stubberud, S. Member, and A. K. Kramer. Fuzzy Logic Based Data Association with Target/Sensor Soft Constraints. In *22nd International Symposium on Intelligent Control*, pages 620–625, Singapore, Singapore, 2007. IEEE.

[145] G. Suarez-Tangil, E. Palomar, A. Ribagorda, and I. Sanz. Providing SIEM Systems with Self-adaptation. *Information Fusion*, 21:145–158, 2015.

[146] C. Sutton, C. T. Morrison, P. R. Cohen, J. Moody, and J. Adibi. A Bayesian Blackboard for Information Fusion. In *Proceedings of the 7th International Conference on Information Fusion*, 2004.

[147] D. D. Sworder and J. E. Boyd. <title>Maneuver sequence identification</title>. *Signal Processing and Information Fusion*, 5204:155–164, 2004.

[148] D. D. Sworder, P. F. Singer, D. Doria, and R. G. Hutchins. Image-enhanced estimation methods. *Proceedings of the IEEE*, 81(6):797–814, jun 1993.

[149] K. Sycara, R. Glinton, B. Yu, J. Giampapa, S. Owens, M. Lewis, and L. T. C. C. Grindle. An integrated approach to high-level information fusion. *Information Fusion*, 10(1):25–50, 2009.

[150] M. Tahk and J. L. Speyer. Target Tracking Problems Subject to Kinematic Constraints. *Transactions on Automatic Control*, 35(3):324–326, 1990.

[151] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. MIT Press, 2005. ISBN 9788578110796.

[152] M. Ulmke and W. Koch. Road-map assisted ground moving target tracking. *Transactions on Aerospace and Electronic Systems*, 42(4):1264–1274, 2006.

[153] L. Vaci, L. Snidaro, and G. L. Foresti. Encoding context likelihood functions as classifiers in particle filters for target tracking. In *Proceedings on Multisensor Fusion and Integration*, pages 310–315, Baden-Baden, Germany, 2016. IEEE.

[154] D. L. Vail, M. M. Veloso, and J. D. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, volume 5, page 1, New York, 2007.

[155] I. Visentini and L. Snidaro. Integration of contextual information for tracking refinement. In *14th International Conference on Information Fusion*, pages 2000–2007, Chicago, Illinois, 2011.

[156] X. Wang, L. Xu, H. Sun, J. Xin, and N. Zheng. Bionic vision inspired on-road obstacle detection and tracking using radar and visual information. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 39–44, Salamanca, Spain, 2014.

[157] W. Wen and H. F. Durrant-Whyte. Model-based multi-sensor data fusion. In *International Conference on Robotics and Automation*, pages 1720–1726, San Diego, California, 1992. IEEE.

[158] W. S. Wijesoma, L. D. L. Perera, and M. D. Adams. Toward Multidimensional Assignment Data Association in Robot Localization and Mapping. *Transaction on Robotics*, 22(2):350–365, 2006.

[159] B. S. Williams, P. Newman, M. W. M. G. Dissanayake, and F. H. Durrant-Whyte. Autonomous Underwater Simultaneous Localisation and Map Building. In *Proceedings on Robotics and Automation*, pages 1145–1150, San Francisco, USA, 2000.

[160] Z. Xiang and U. Ozguner. Environmental perception and multi-sensor data fusion for off-road autonomous vehicles. In *5th International Conference on Intelligent Transportation Systems, Proceedings*, volume 2005, pages 584–589, Vienna, Austria, 2005.

[161] R. Xu and W. Donald. Survey of Clustering Algorithms. *Annals of Data Science*, 16(3):645–678, 2005.

[162] C. Yang and E. Blasch. Kalman Filtering with Nonlinear State Constraints. *ransaction on Aerospace and Electronic Systems*, 45(1):70–84, 2009.

[163] C. Yang and E. P. Blasch. Pose Angular-Aiding for Maneuvering Target Tracking. In *8th International Conference on Information Fusion*, pages 32–39, Philadelphia, Pennsylvania, 2005. IEEE.

[164] C. Yang, Y. Bar-Shalom, and C. F. Lin. Discrete-Time Point Process Filter for Mode Estimation. *Transaction on Automatic Copntrol*, 37(11):1812–1816, 1992.

[165] Y. Zhou and H. Leung. Minimum Entropy Approach for Multisensor Data Fusion. In *Signal Processing Workshop on Higher-Order Statistics*, pages 336–339, Banff, Canada, 1997. IEEE.

[166] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury. Context-Aware Activity Modeling Using Hierarchical Conditional Random Fields. *Transactions on Pattern Analysis and Machine Intelligence*, 37 (7):1360–1372, 2015.