1 Introduction

1.1. Motivation

1.1.1. Automated Vehicles

Vehicle automation recently receives worldwide attention, and there have been considerable progresses and improvements in this area. Especially, German premium OEMs and non-industry members in the US currently lead this technology. With governmental authorities, they also play a pioneer role to reassess the related legal issues as the technology has progressively advanced. For a clear understanding of this technology, some definitions for automation levels are currently in use. SAE[1] international, for example, categorized and defined six levels of automation spanning from *No Automation* to *Full Automation* [1]. Similar definitions were developed by the Germany federal highway research institute (BASt) [2] and the US national highway traffic safety administration (NHTSA) [3], in which five different levels are specified instead of the six levels of SAE international. All definitions aforementioned approximately can be differentiated according to the four fundamental criteria: control (longitudinal and lateral), monitoring (driving environment), fallback (intervene capability), and scope of operation (driving modes) [1]. Using these criteria, one can say that higher levels of automation are achieved as a vehicle has more responsibilities for the control, monitoring, and fallback tasks in more variety of driving modes. Figure 1.1 provides a comparative overview of taxonomies from SAE international, BASt, and NHTSA. As shown in the figure, BASt has no *Full Automation* level of SAE international whose scope of operation includes all driving modes. On the other hand, NHTSA makes no difference between *High automation* and *Full automation* of SAE international. Anyhow, the higher the level the more automated vehicle is in all taxonomies[2]. In this context, the past and potential future evolutions of the vehicle automation are illustrated at the bottom of Figure 1.1 with corresponding sensor technologies (adapted from [4]). At the very beginning, the passive (or secondary) safety systems (e.g. airbags and seat belts) have been developed which

[1] Society of automotive engineers

[2] More detailed descriptions of each taxonomy can be found in [1]–[3]. For the purpose of this thesis, the BASt's definitions are used.

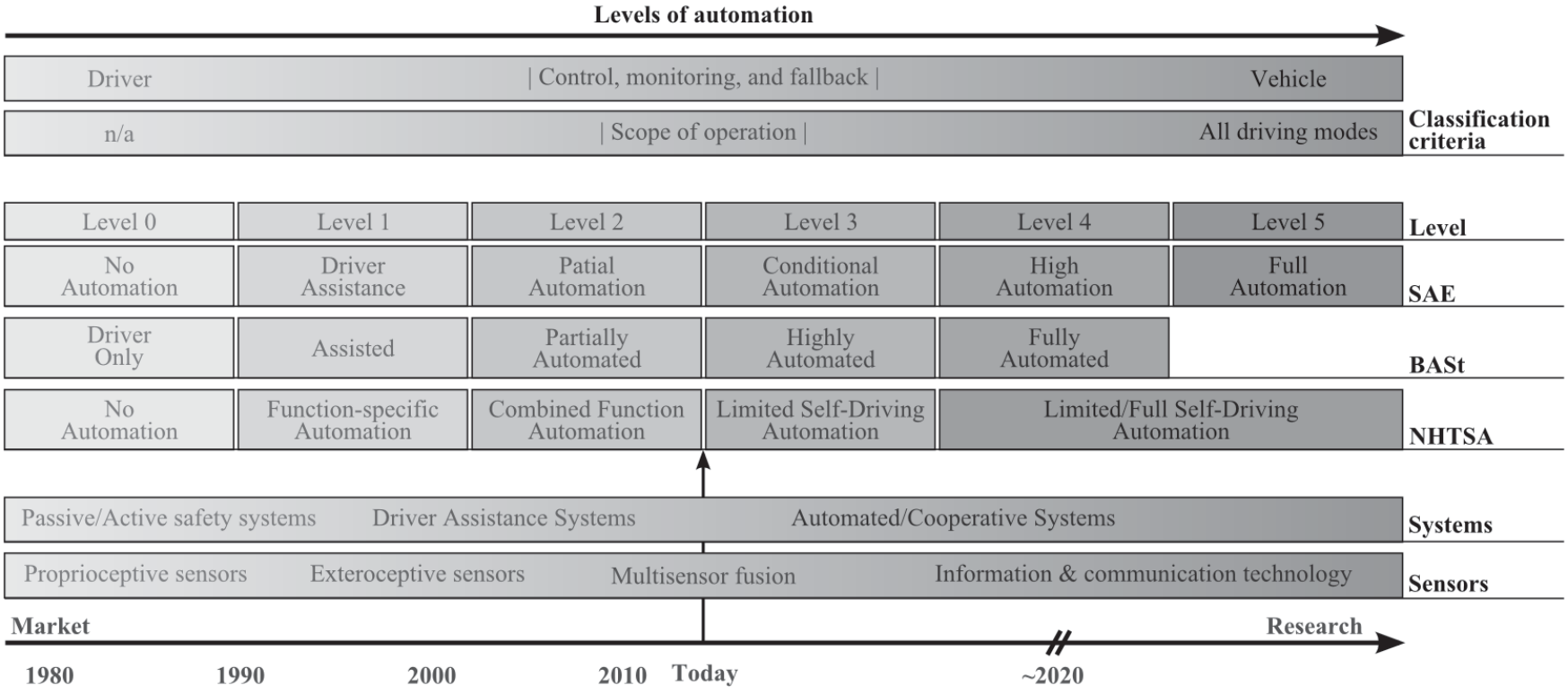

Figure 1.1.: A comparative overview of automation taxonomies with corresponding system and sensor technologies, adapted from [1]–[4].

correspond to the *Driver Only* level (i.e. do not control the vehicle directly). These systems are active during an accident to reduce the accident effects. On the other hand, the active (or primary) safety systems are active prior to an accident in which the systems enable the dynamic control for the vehicle to follow the trajectory requested by the driver in the best possible way. These systems are based on the proprioceptive sensors (e.g. odometry and inertial sensors) and correspond to the level of *Assisted* (i.e. do perform part of vehicle control, but not all). The most popular active safety systems are anti-lock braking system (ABS), traction control system (TCS), and electronic stability control (ESC). The passive and active safety systems have been increasingly mandatory on newly produced cars in many countries. Another example of the *Assisted* level is the advanced driver assistance systems (ADAS) such as adaptive cruise control (ACC), lane keeping assist (LKS), park assist, etc. These systems have been made possible with information acquired by exteroceptive sensors (e.g. ultrasonic, radar, lidar, and camera), and become increasingly common in the market. The *Partially Automated* level can be achieved by the system and sensor fusion technologies. All control tasks in this level are performed by the vehicle, but the driver is still an active supervisor regarding the monitoring and fallback tasks. ACC with LKS and traffic jam assist (TJA) are the representative examples of this category. Some products of this level are currently under development or already available in the market. However, the *Highly Automated* (e.g. motorway chauffeur, platooning, etc.) and *Fully Automated* systems (e.g. motorway pilot, automated emergency stop, etc.) have not been launched to the market yet. For these higher levels of automation, the vehicle has to have the ability to monitor the environment and decide when the vehicle is approaching the end of its operating capabilities. While the driver's fallback role is still included in the *Highly Automated* level, the vehicle can achieve a minimal risk condition in the *Fully Automated* level. In this context, one of major challenges for higher levels of automation is to make decisions regarding safety critical issues and act appropriately against them. For this reason, it would be essential in the future to exchange data between traffic participants or with the road infrastructures [5], [6].

1.1.2. Functional System Architecture

Vehicle automation obviously leads to the increased driving comfort, safety, and efficiency. Some industry leaders already announced their road map to achieve the final goal of fully automated vehicles, and expect that vehicles equipped with automated driving features will enter the market around the year 2020 [7]. However, the reliable performance must be firstly guaranteed to make higher levels of automation appear on the market because a failure could be very dangerous to vehicle occupants and other road

users. The vehicle has to perceive its environment reliably and decide where is safe and desirable to move, and finally appropriate control strategies have to be realized according to these decisions. In order to develop these complex systems efficiently, it is needed to subdivide the entire task into smaller subtasks. Then, the resulting functional modules and their dependencies can be described by the so-called *functional system architecture*. In this context, many system architectures have been proposed in the literature [8]–[10], and our working group also developed a novel functional system architecture in the research project *Stadtpilot*[3]. It was developed in a top-down approach based on the definition of the functional requirements for an automated vehicle and explicitly combines perception-based and localization-based approaches [13], [14] (see Figure 1.2).

This system architecture has a three-level design in a row direction: *strategical*, *tactical*, and *operational*. These three levels differ (among other characteristics) in their resolution, horizon and accuracy (concerning time and space), relevant environmental features, tasks, and cycle times[4].

In an orthogonal direction to these three levels, the system architecture is constructed with four columns: *absolute global localization*, *external data*, *perception* (consisting of environmental perception and self-perception), and *mission accomplishment*. The columns *perception* and *mission accomplishment* are typically part of a vehicle-referenced view, which means that the environment is described in relation to the vehicle. An absolute global localization is not necessary in this case. On the contrary, the columns *absolute global localization* and *external data* describe the environment in an absolute global reference frame, and simultaneously determine the pose of the vehicle in relation to the environment.

In this representation, the work in this thesis corresponds to the *operational* level in the three-level design. We do not touch the *strategical* and *tactical* levels, but rather provide useful information for tasks in those levels. On the other hand, our work corresponds to *perception* in the column direction. We do not perform the global localization, and planning&control tasks. Instead, our work covers the local pose of the vehicle, the map of stationary objects, the pose and dynamics of movable objects. Here, the localization is based neither on the absolute position sensor nor on the predefined map data. Only the vehicle pose relative to the starting point is continuously estimated. This pose may contain long-term drift, but it is quasi-continuous regarding successive poses. Therefore, the purpose of our localization task is to stabilize the vehicle pose error when global position information is neither reliable nor available.

[3] More information about the project is available from [11], [12] and the project website: https://www.tu-braunschweig.de/stadtpilot

[4] More detailed descriptions of the three levels can be found in Table 5.1 and 5.2 of [14].

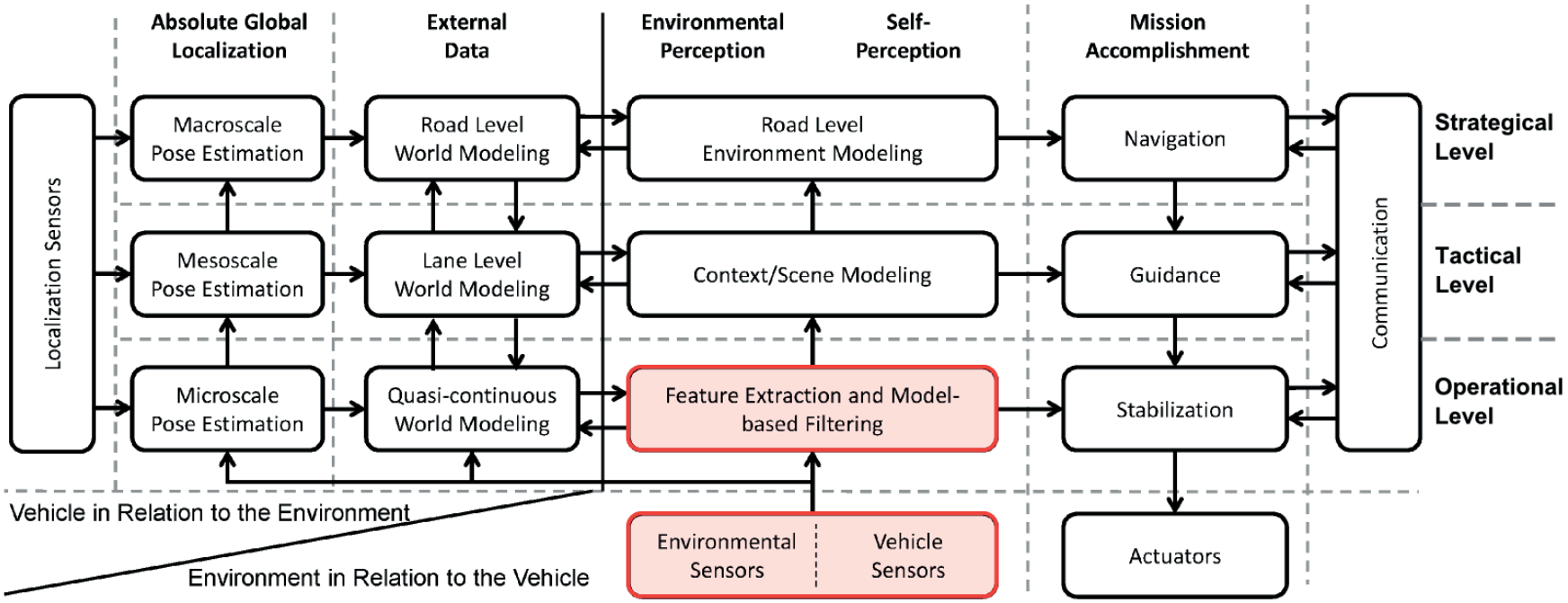

Figure 1.2.: Functional system architecture for an autonomous on-road motor vehicle in public road traffic proposed by Matthaei and Maurer [13]. The work in this thesis corresponds to tasks in red colored boxes.

1.1.3. Environmental Perception

The reliable environmental perception is probably first and one of the most important parts for automated vehicles. One of difficult issues in the environmental perception is that a sensor (or a vehicle) is usually a part of the whole system being perceived and certainly moves over time within the environment. Therefore, the vehicle should firstly know its current pose to correctly interpret the sensor measurements. This task is generally referred to as *localization*. Localization should provide robust results even in the absence of a global navigation satellite system (GNSS) since absolute position information are not always available due to missing satellite connections or multi-path effects [15], [16]. The vehicle should also sense its whole surrounding and provide the robust detection (or classification) of stationary and movable objects to utilize them separately in subsequent algorithms. This task is known as *movable object detection (MOD)*. The stationary objects provide the spatial information around the vehicle. Here, the stationary objects can be represented with any type of maps, and this task is generally referred to as *mapping*. In the mobile robotics literature, mapping is often dealt with localization simultaneously. For the movable objects, their states and dimensions are estimated to obtain their spatial and temporal information, and this task is known as *movable object tracking (MOT)*. This allows the vehicle to predict dynamic properties of the surrounding movable objects, which is very useful information for automotive applications such as the ACC and pre-crash safety (PCS) systems. Only when all tasks are carried out properly, we can expect safe operations of the automated vehicles.

All environmental perception tasks aforementioned are conducted based on the information (or measurement data) from perception sensors, e.g. camera, radar (radio detection and ranging), or lidar (light detection and ranging) sensor. Each sensor technology has both strengths and weaknesses to apply. The camera sensor, for example, has advantages of lower costs and rich information including the color and shape of objects, whereas it is vulnerable to illumination variances and weather conditions. The radar sensor is more robust to weather conditions and provide accurate distance information, but it gives poor information about the objects' shape and velocity. The lidar sensor provides accurate shape and distance information. Moreover, its performance is independent of illumination variances. However, the lidar sensor is very expensive, and its moving parts for obtaining a sequential measurement data (leading to the term *laser scanner*) require additional processing algorithms. Since all sensors have different limitations, there have been approaches using multiple sensors (i.e. sensor fusion) to improve the quality of the measurement data and to cover all surroundings of the vehicle [17], [18]. However, these approaches are also challenging because each sensors' outputs are related to the other sensor outputs in a complicated manner.

In this work, we use a laser scanner as the primary perception sensor of the vehicle. Since the price of the laser scanner continues to decrease, and a high-end laser scanner such as a Velodyne laser scanner covers whole surrounding of the vehicle with a high data rate, the laser scanner has been popular within the automated vehicle society, e.g. DARPA[5] challenges [19], [20] and test cars from several manufacturers. In this context, we mainly focus on issues of using the laser scanner to achieve the goal of robust environmental perception.

The environmental perception tasks previously mentioned can be separated into two technical problems: simultaneous localization and mapping (SLAM), and detection and tracking of movable objects (DATMO). The SLAM algorithm is responsible for the localization and mapping tasks, while the DATMO algorithm is responsible for the MOD and MOT tasks. In other words, the SLAM algorithm provides an estimated pose of the vehicle and a map of stationary objects, whereas the DATMO algorithm is referred to as a process of detection and dynamic state estimation of movable objects. Obviously, both SLAM and DATMO algorithms are very essential and critical to the automated vehicles. In this thesis, we propose a novel framework to tackle both SLAM and DATMO problems in a more accurate and efficient way so that the vehicle can operate correctly in any dynamic and populated environments. This work is conducted as a part of the *Stadtpilot* project in which we have extended our work of the urban challenge [21] by automated driving in public traffic on a piece of the Braunschweig city ring road. The successful results of this work are supposed to serve as important information to the subsequent applications.

1.2. State of the Art

In most applications, the SLAM and DATMO problems are considered in isolation. However, they can be solved successfully only when the environment is exactly split into stationary and movable parts. The SLAM algorithm assumes the environment as stationary, and therefore the estimation results are not guaranteed unless movable objects are properly filtered out. On the other hand, stationary objects are considered background in the DATMO algorithm. However, when driving in the complex urban environment composed of the stationary and movable objects, neither of them has sufficient performance in isolation. In this context, Wang et al. [22] showed that the SLAM and DATMO algorithms are mutually beneficial and therefore should be considered concurrently. They introduced a mathematical framework which decomposes the overall joint posterior into two separate components with the independence assumption between sta-

[5] Defense advanced research projects agency

tionary and movable measurements. This framework helps to reduce the complexity (or dimension) of the general SLAM and DATMO problem, and more promising results of both algorithms are expected by sharing the information from each other. Following the work of Wang et al. [22], Vu et al. [23] introduced a similar framework of the environmental perception with an affordable 2D laser scanner even though details of their implementation are little bit different.

This work was also inspired by the pioneer work of Wang et al. [22]. We firstly isolate MOD from DATMO for a better understanding of the overall system structures. In our definition, the MOD algorithm is a common (or preceding) part of the SLAM and MOT algorithms. It classifies measurements from perception sensors into stationary and movable objects which are inputs of the SLAM and MOT algorithms, respectively. On the other hand, it utilizes outputs from both SLAM and MOT algorithms to produce robust classification results. In this context, the MOD algorithm seems to act like a bridge between both algorithms. Figure 1.3 shows the relationship between the MOD, SLAM, and MOT algorithms. The MOD algorithm is performed essentially based on the stationary map from the SLAM algorithm. However, the predicted target information from the MOT algorithm can be additionally used to enhance the performance of the MOD algorithm. The map quality of the SLAM algorithm is then improved with well classified stationary objects, and this accordingly leads to the more accurate correction of the vehicle pose predicted by measurements from motion sensors during the SLAM process. On the contrary, the good quality of the map and accurate vehicle pose from the SLAM algorithm make the MOT algorithm to be performed more reliably.

Based on this framework, we focus on these three problems, i.e. SLAM, MOD and MOT, in this thesis. The following subsections give a brief overview of existing methods to tackle these problems.

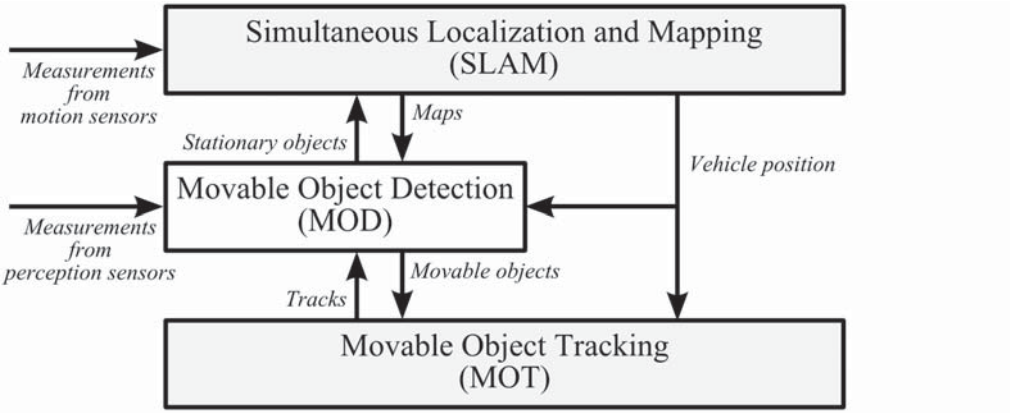

Figure 1.3.: The relationship between the MOD, SLAM, and MOT algorithms.

1.2.1. Simultaneous Localization and Mapping

The SLAM algorithm is performed based on on-board sensors such as perception and odometry sensors. In this technique, the vehicle builds the map of the environment, and simultaneously uses this map to localize itself. This technique is very useful especially when absolute position information are inaccurate or unavailable. However, the mutual dependency between localization and mapping makes the problem difficult. In the literature, various techniques have been introduced to solve this problem, and they can be generally classified in terms of the environment model and the underlying estimation technique.

The environment model (or map) is basically necessary for general vehicle navigation and guidance. In this context, the environment model should be able to represent occupied, free and unknown regions properly. However, its probabilistic form is also crucial because it is incorporated with the pose and measurement uncertainties during the SLAM process. This is the reason why direct representation approaches [24], e.g. a point cloud map representation, are generally not used in SLAM in spite of their simple way to implement. The most commonly used environment models in the field of SLAM are the feature map [25], [26], grid map [27]–[30], and topological map representations [31], [32]. Here, the topological map representation is generally used to embed two other approaches on top of them for enhancing scalability of the map. Therefore, we narrowed the selection for the environment model to the feature and grid map representations in this work. Which environment model we select critically influences the overall performance of the SLAM algorithm. The key properties of each model can be summarized as follows:

- *Feature map representation:* This representation supposes that the environment consists of a set of isolated features which reflect the contents of the environment. In general, the features are represented with geometric primitives, e.g. points, lines, circles, polygons, etc. This representation is very attractive because a compact representation of the environment is possible so that the memory consumption and the processing time are considerably reduced. However, the key questions of this representation are what type of features to use and how to detect and identify them reliably. These tasks generally require considerable efforts, and this limits the application of this approach. Another issue of this representation is that a set of features is useful for localization, but does not give any spatial information about the environment which is important for the vehicle navigation and guidance.
- *Grid map representation:* In this approach, the environment is represented by discrete grid cells. Each cell contains information about the area it covers. It

can provide specific and detailed geometric information about arbitrary objects because it makes no assumption about the types of objects during the mapping process. Especially, Moravec and Elfes [33] firstly introduced the occupancy grid map in which each cell of the grid map has a probability indicating that the cell is occupied or not. The advantage of this representation is that the spatial information of the environment can be obtained properly in a probabilistic form without predefined features. In addition, measurement uncertainties from multiple sensors can be efficiently represented in the map so that the sensor fusion problem can be handled properly[6]. As a result, this approach has become the most popular way among map representation methods recently. However, it typically requires a huge amount of memory and computational resources, and also suffers from discretization errors. When the range of the sensor is short or the environment is a large open space, localization using this method can be problematic as well.

For the underlying estimation technique, the probabilistic approaches such as the Kalman filter [37]–[40], expectation maximization (EM) [41], [42], and particle filter [25], [26], [30] are dominant techniques because they can deal with the system uncertainties and sensor noises in an efficient way. Their mathematical frameworks are derived from the recursive Bayes rule, and utilized in the SLAM algorithm with following characteristics:

- *Kalman filter:* This technique assumes that the control and measurement models are linear with additive Gaussian noises, and that the initial posterior is also Gaussian. Then, Gaussian random variables are propagated analytically through the linear models. For the nonlinear models, the extended Kalman filter (EKF) or unscented Kalman filter (UKF) technique can accommodate the nonlinearities. While EKF linearizes the models via Taylor series expansions, UKF uses the so-called *uncented transform* instead of linearization [43]. The strength of the Kalman filter and its variants is to represent a full posterior over the vehicle pose and map with a small number of parameters, e.g. mean and covariance. In addition, it provides optimal minimum mean-square error estimates of the state, and its covariance matrix seems to converge strongly in most cases. However, the Gaussian noise assumption restricts the adaptability of this technique. This technique only works with the feature map in which the filter is likely to diverge if the features are not uniquely identified (data association problem). Moreover, since each feature location error is correlated with the vehicle pose error and errors in other

[6] Separate maps for each sensor or data types are usually maintained, and then fused into a single layer [34]–[36].

features, the matrix operation is generally intractable as the number of features increases.

- *Expectation maximization:* This technique iterates two steps: an expectation step (E-step), where the posterior over the vehicle poses is calculated for a given map, and a maximization step (M-step), in which the most likely map is calculated given the pose expectations. Through multiple iterations, a series of increasingly accurate maps and vehicle poses can be obtained. A key advantage of the EM algorithm over the Kalman filter lies in the fact that it solves the correspondence problem (data association problem) surprisingly well. However, this technique needs to process the same measurement data repeatedly, and this makes the technique is not suitable for real-time applications. Therefore, only the M-step of this technique is used to map the environment, while the E-step is simplified or eliminated. This is generally referred to as the *incremental maximum likelihood* method [42]. The main idea of this method is to incrementally build a single map of the environment as the measurement data arrive without keeping track of any residual uncertainties of the map. Although both vehicle pose and map are still optimized simultaneously, their errors may grow without bounds because their estimates are frozen at each time step and can never be changed in the future. This seems to be problematic when closing a loop. To overcome this limitation, the EM algorithm is usually combined with the particle filter. The EM algorithm (only M-step) is used to build the map, while the localization is done by using the particle filter.

- *Particle filter:* In the Kalman filter and its variants, the noise is assumed to be Gaussian, and the control and measurement model have to be linearized. However, this assumption is generally invalid in the real world. On the contrary, the particle filter does not assume a particular distribution, but rather represents the posterior distribution by a set of weighted samples (called particles). Especially, the Rao-Blackwellized particle filter (RBPF) is generally used in the field of SLAM because the high dimensional state-space of the SLAM problem makes direct application of particle filter computationally infeasible. This algorithm is known as *FastSLAM* in which the SLAM problem is decomposed into a vehicle localization problem and a corresponding map building problem given the vehicle pose. This is possible under the assumption that the mapping problem is conditionally independent given the vehicle pose. This framework enables that the memory consumption and the processing time are considerably reduced compared with the standard Kalman filter technique. Moreover, this technique maintains the full posterior, and therefore it can properly handle the loop closing problem as well.

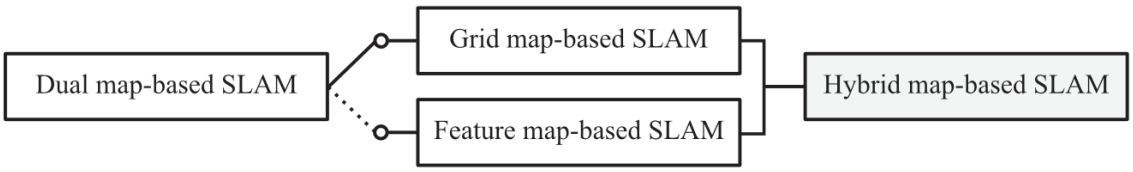

Figure 1.4.: Variants of RBPF-SLAM depending on the environment models.

Since the RBPF based approach has many advantages over the others, it has been widely used as an efficient estimation algorithm, and we also adopt this technique in this work. This technique not only improves the precision of an estimation, but also reduces the overall computational complexity. Once RBPF is chosen as the underlying estimation technique, the variants of SLAM are decided by the aforementioned environment models (see Figure 1.4). Grid map-based SLAM was first introduced by Murphy [27] and Doucet et al. [44]. Subsequently, Montemerlo et al. [25] extended it to the so-called FastSLAM which is one of the feature map-based SLAM approaches. Additionally, there were attempts to combine both maps. Wurm et al. [45], for example, proposed an approach to combine feature map-based SLAM with grid map-based SLAM. We denote this approach as dual map-based SLAM in this thesis. Their approach maintains both maps and selects one of them which is currently the best one to map the surroundings of the vehicle. The model selection process is obtained in an adaptive way using reinforcement learning. However, they did not mention the details of the actual feature detection algorithm. Another example of the combined approach is the so-called *DenseSLAM* proposed by Nieto et al. [46] (see also [47], [48]). They combined feature maps with dense metric sensory information. However, the dense map (e.g. grid map) does not provide any information to the localization and feature maps, but rather provide much information only for the navigation.

In the context of the RBPF framework, the efficiency of the algorithm can be measured in terms of the number of particles. In fact, we need as many particles as possible to make the filter converge to the correct posterior. However, in practice, only a finite number of particles can be used because of implementation issues, e.g. computational resources. This is the reason why approaches using RBPF try to make an improved proposal distribution. By choosing a more accurate proposal distribution, the number of required particles can be significantly reduced. In the literature, there have been attempts to incorporate the measurement data into the sampling process. Montemerlo et al. [26] tried to compute the improved proposal based not only on a motion model, but also on the most recent sensor measurement in the algorithm called FastSLAM 2.0. On the other hand, Hähnel et al. [29] performed scan matching as a preprocessing step for grid map-based SLAM. Grisetti et al. [30] extended this approach to compute an informed

proposal on a per particle basis on the fly instead of using a fixed proposal distribution. The idea to compute the proposal is similar to FastSLAM 2.0 [26]. In this way, the improved proposal can be obtained and it requires fewer samples by generating more samples around the region of high probability so that samples are not wasted.

All approaches aforementioned use only one type of map to compute the improved proposal distribution even in the case of maintaining both maps. In contrast, we propose the so-called *hybrid map-based SLAM* algorithm in this thesis. We represent the environments with the hybrid map which consist of the feature and grid map (see Figure 1.4). This basically allows us to obtain an equivalent performance of approaches mentioned previously. However, we derived a novel sampling formula with measurements in both maps. Since this reduces the uncertainty of the predicted vehicle pose, the accuracy and the efficiency of the algorithm are significantly improved. In addition, we extend the 2D grid representation to 3D one, and we denote this environment model as the *volumetric hybrid map* in this thesis. This representation provides not only more contextual information of the environment, but also more accurate measurement likelihood for the map update.

One of the simplest ways to represent the 3D environment is a regular voxel representation [49], [50] in which the entire space is divided into a set of voxels (cubic volume elements). In this approach, information about free and occupied space can be represented explicitly, but its actual implementation is limited by the large memory requirement. The extent of the mapped area has to be predetermined, and the memory usage significantly increases to represent the large-scale outdoor environment with fine resolutions. This problem can be mitigated by simplifying the vertical information within the typical 2D grid map. Herbert et al. [51], for example, used 2.5D elevation maps, and Triebel et al. [52] introduced multi-level surface maps. By contrast, Ryde and Hu [53] proposed the volumetric representation with multi-resolution occupied voxel lists. These approaches are more efficient than the regular voxel representation, but none of them has a way to update the map in a probabilistic form because free and unknown areas are not represented or differentiated as in the regular voxel representation. Therefore, they are not suitable for the vehicle localization or navigation. Consequently, tree-based representations such as octrees [54]–[56] have been proposed as alternatives to overcome this problem. In these approaches, the extent of the mapped environment does not need to be predetermined, and the map can be updated in a probabilistic form because the environment is represented in a volumetric fashion. Our algorithm in this work is also based on these flexible and efficient tree-based representations.

1.2.2. Movable Object Detection

In order to reliably conduct SLAM and MOT, we have to decide whether objects are stationary or movable. Then, the stationary parts of the environment are modeled with a map, and some filtering techniques are applied to track movable objects. The presence of movable objects causes errors of the vehicle pose estimate and degrades the overall quality of the map during the SLAM process. On the contrary, the number of false alarms and the computing time in the MOT algorithm can be reduced only when stationary objects are properly filtered out. For this reason, a number of approaches for detecting movable objects have been proposed in the literature.

Track-before-detection (TBD) performs both detection and tracking simultaneously [57]–[59]. This approach tracks all objects extracted from a sensor first, and then decides whether tracked objects are movable or not based on their dynamic properties, e.g. life time, position, velocity, etc. However, the high computational requirements of this approach makes the real-time implementation unfeasible in the populated environment. In addition, it gives a high rate of false tracks which cause the tracker to suffer from the wrong data association. Especially, the stationary objects next to the road always remain problematic.

An alternative method is to detect movable objects based on their appearances (when cameras are used) or geometric shapes (when laser scanners are used) instead of their dynamic properties. This approach requires prior knowledge of movable objects. However, there are many kind of movable objects in the complex urban environment, e.g. cars, buses, trucks, bicycles, motorcycles, pedestrians and so on. Moreover, their appearances and geometric shapes in a sensor's view are frequently changed and occlusions are also problematic. Therefore, it is generally difficult to characterize them separately. To ensure the good performance of this approach, a sufficiently large number of training examples is required [60]–[63].

Hähnel et al. [64] introduced another approach using the EM algorithm. They included additional variables in a likelihood function to characterize whether the measurements were caused by stationary objects or not. In the expectation step, this approach estimates which measurements might correspond to stationary objects. In the maximization step, the most likely map and robot pose are calculated by using estimates from the expectation step. This process is iterated until no further improvement can be achieved. While this algorithm does not require predefined features, the real-time implementations are limited by the computational cost, and therefore it is considered as the off-line method.

Recently, a consistency-based approach seems to be more widely used for detecting movable objects. Hähnel et al. [61] presented a technique for identifying movable objects based on the difference (or inconsistency) between consecutive sensor readings. Biswas et al. [65] took an occupancy grid map representation in which movable objects are identified by the map differencing technique. Wolf and Sukhatme [66] proposed a model that maintains two separate occupancy grid maps, one for the stationary parts and the other for the movable parts of the environment. However, a critical problem of the general consistency-based approach is the poor signal-to-noise ratio (SNR). For low SNR objects, the detection threshold must be low enough to allow sufficient probability of the movable object detection. However, a low threshold also gives a high rate of false detection. Too slow and long objects, for example, have the dominant measurement data classified as stationary, and therefore it is difficult to decide their dynamic properties in the typical consistency-based approaches. This problem can be generally solved in two ways: by reducing the sample rate and by restricting the decision area. Wang et al. [22], for example, proposed an approach which sample the measurement data at a low rate so that the signal is prevalent over the noise. Matthaei et al. [67] also proposed the usage of a specialized cell type to model the timing behavior of the grid cell (see also [68], [69]). On the contrary, Petrovskaya and Thrun [70] proposed the motion evidence which restricts the decision area into the front and back of the object. Besides, additional dynamic information can be used to improve the detection performance. Matthaei et al. [67], for example, used information from velocity-measuring sensors, e.g. radar sensor. The work presented by Wang and Thorpe [71] is an example which utilizes track data. They detect potential movable objects which are in areas that were previously occupied by stable tracks. Rieken et al. [36] also proposed a similar approach to produce unambiguous conditions for the so-called *significant inconsistency* state in their approach.

Our approach here is essentially established based on the consistency-based approach. For the poor SNR, we adopt the method proposed by Petrovskaya and Thrun [70] since the method reducing the sample rate requires more time for the detection[7]. The stable track list from the MOT algorithm is also used to improve the detection performance. In order to mitigate the relatively high probability of false negative detections especially in a low velocity range, we implement an exceptional logic for the MOT algorithm which stationary objects can be used for the measurement update (but not for the track initialization) of too slow but stable tracks. This does not affect the performance of the SLAM algorithm, but rather improves the continuity of the track in the MOT algorithm. Besides, we introduce the target selection algorithms in addition to the MOD algorithm. In our definition, the term *target* represents the selected movable object which will be

[7] More refined algorithms considering all aspects mentioned in this section have been developed by our working group. The interested reader is referred to [36], [67]–[69]