Automatic checking and correction of localization signs in HD maps with onboard perception systems

M Desjardins, R Nguyen, R Huet, Ph Bonnifait, M.-A Mittet

Automatic checking and correction of localization signs in HD maps with onboard perception systems

M. Desjardins[1], R. Nguyen[1], R. Huet[2], Ph. Bonnifait[2], and M.-A. Mittet[3]

[1]Université de technologie de Compiègne
[2]Université de technologie de Compiègne, CNRS Heudiasyc, France
[3]Ampere Software Technology

Abstract: Navigation maps are subject to various errors coming from the mapping process, or arising over time from environment changes. To check the quality of map data before using it, in particular in the localization layer of a High-Definition (HD) map, the information given by the sensors can be processed in real-time while the vehicle is moving during a first passage. Two methods are proposed and compared to address this problem and experimental results are reported.

Keywords: ADAS; HD Map; Diagnosis; Sensor Fusion.

1. Introduction

In order to operate safely, autonomous vehicles and ADAS systems may need an accurate and reliable localization. This can be achieved using an HD map with a dedicated layer which contains landmarks that are observable with onboard sensors like cameras or LiDARs. However, such maps suffer from errors that may come from the mapping process, or from the modification of the environment in time [1]. Due to these errors, maps cannot be considered as a fully reliable source of information, and the data they contain must be updated frequently and checked online by the vehicles [2].

This problem of map map completeness and freshness has already lead to some research into diagnostic and update methods. In [3], the authors propose a method to detect missing roundabouts in Standard Definition (SD) maps based on graphical pattern recognition methods, thus enabling to correct some topology errors. Other approaches propose to detect errors in HD maps by leveraging on onboard sensors and using bayesian filtering [4], smoothing [5] or other fault detection techniques based on sequential statistical tests [6]. Some other use deep-learning algorithms coupled to a database of map geometry and sensor information to classify the map reliability with several sensors [7]. As for map update, several methods where proposed to incrementally increase the quality of a map using mapping vehicles or crowdsourced data [2], [8]–[11].

In this paper, we focus on road signs, that are common landmarks for localization using vision or LiDAR sensors. They are indeed easy to detect and are widely present in urban scenarios where GNSS-based localisation demonstrates the worse capability. We propose a comparison of two methods for automatic correction of features maps, which are both able to determine whether the map is reliable enough to be used for localization, or if the map is still subject to modification. The first approach is based on a 2D grid mapping of the environment. The observations are stacked in the grid until sufficient evidence of the existence of a sign is reached. The second one uses Random Finite Sets (RFS) Bayesian filtering to solve this problem with a Probability Hypothesis Density (PHD) filter that estimates simultaneously the number of signs and their locations. Both methods use the same prior erroneous map. We provide an experimental comparison of the two methods by highlighting their advantages and drawbacks.

2. Problem statement

2.1 Context

Consider a vehicle equipped with onboard sensors navigating a road surrounded by various types of road signs. A map describing the location of the signs is provided to the vehicle. The signs that serve as landmarks may suffer from different type of error. They can be very poorly located, non-existent in reality (false positives), some of the signs on the map no longer exist (false negatives) and some can have wrong signification (semantic error). In this article, the proposed approaches focus on signs location and existence.

During a learning phase in manual driving, the onboard system corrects and improves the map. As long as data integrity is uncertain, the vehicle remains

driven by a human. During this phase, the vehicle has to correct the map, i.e. identify precisely located signs, remove or identify as bad any faulty ones, correct those that are flawed and add to the map those that have been detected and are not on the map. Once it is confident in the data on its map, the on-board system automatically suggests switching to autonomous mode and driving the vehicle.

This approach aims to enhance the accuracy of localization and improve autonomous navigation reliability. The overall objective is to study a system that automatically performs the correction of the map without any human intervention.

2.2 Experimental Setup

The article is based on an experimental study. The test track consists in two roundabouts with a straight line between them (see for example fig. 5). 19 standard compliant road signs were placed around the track. A reference map was given for evaluation, while the a priori map given to the system was generated by removing signs from the reference, adding fake ones and spatial noise to the landmarks.

The experimental setup consisted on a Renault Zoé car, manually driven on the test track with no other vehicle on it (fig. 1). The car localization was based on a SPAN-CPT sensor, coupling an IMU and GNSS RTK. The perception was done using a Pandora 40-layers LiDAR in dual-return mode, providing 360° perception at 10Hz.

Figure 1: The experimental vehicle on the track

A preprocessing of the LiDAR data was done to convert the raw point cloud data detected by the LiDAR sensor into a comprehensive list of signs poses, including their ground plane coordinates and orientation, for each LiDAR iteration. The first step in this process involved filtering the point cloud using an intensity threshold, leveraging the fact that the signs are reflective and thus exhibit a higher intensity compared to the surrounding environment. This allows to isolate potential sign points from the rest of the scene. Next, we applied geometric distortion corrections, which are essential to compensate for inaccuracies introduced by the LiDAR's movement, particularly due to variations in speed. Once the filtered and corrected point cloud were obtained, we performed clustering using Euclidian Clustering based on K-D trees[12] to group the points that likely belong to individual signs. Finally, Principal Component Analysis (PCA) was employed on each cluster to accurately determine the orientation of the signs, providing an accurate and structured output of their poses.

3. Grid-based approach

The first approach is described in fig. 2. It leverages a combination of clustering and graph matching techniques to accurately localize road sign observations. First, the observations labeled as road signs are mapped onto a structured grid. High-density clusters are then identified, representing likely road sign locations. These clusters are subsequently matched to the signs of the prior map. This enable to take advantage of the prior map by retaining signs that were accurately positioned. One advantage of this method is that it also allows to retain the matched points for immediate vehicle localization.

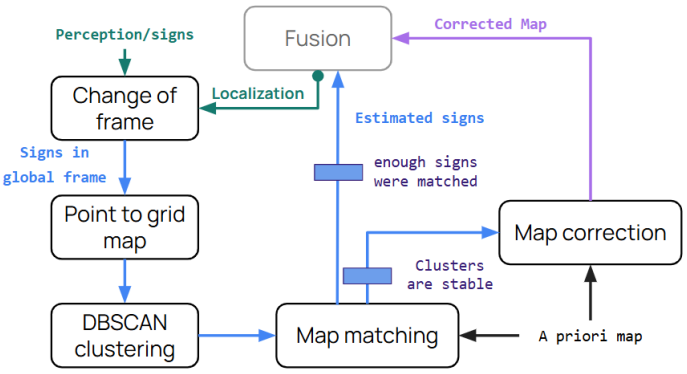

Figure 2: Grid-based approach overview

3.1 Data Aggregation into a Grid Structure

While the vehicle is driving the perception module constantly provide observations of road signs. In real time, we aggregate this data into a weighted grid map. Each grid node has a weight which is increased whenever a new observation falls in its vicinity. Such a grid efficiently stores observation data by consolidating multiple observations into specific nodes, reducing memory usage and computational time. The aggregation of the data into the grid and the clustering step take place at a relatively high frequency while driving, therefore this grid structure is essential for ensuring the system operates swiftly.

3.2 Clustering of Observations

Each time the grid is updated with new observations, the system proceeds to the clustering with DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [13] to identify dense areas where road signs

are likely located and mark noise points to be filtered out. Each cluster identified is assumed to refer to the location of a single road sign. As the vehicle gathers more data over time, these clusters become more refined and tend to stabilize. We will take advantage of this to determine the stopping conditions later on. When many imprecise observations of a single road sign are scattered between many grid nodes, the DBSCAN algorithm is able to take advantage of the weights to precisely estimate the center of the cluster.

3.3 Geometric Matching of Clusters to Map Points

One assumption of the given problem is that the majority of landmarks in the map are well located. We then want to use the information provided by the prior map to enhance the perception, by correcting a potentially existing bias. In order to do this, we use the Graph Matching via Maximum Clique (GMC) [14] algorithm, which is commonly used in simultaneous localization and mapping tasks. We feed this algorithm the centers of the clusters calculated in the clustering step, which are the observed road signs, as well as to the road signs in the prior map. The result of the GMC algorithm is a subset of road signs from the prior map that have been paired to a cluster. We will be using this information for updating the map.

By using a geometry based matching algorithm, we are not exclusively relying on spacial proximity between the clusters and the signs from the prior map. This makes our system less sensitive to biases in the localization of the vehicle itself and in the perception module.

3.4 Stopping Condition

To determine when to stop the process, we assess whether the clusters resulting from the clustering step have changed within a specified time interval (in this case, 1 second, chosen arbitrarily for this experiment). If the centers of the detected clusters remain unchanged after a certain number of updates, we consider them stable enough to proceed with correcting the prior map. We consider that the clusters have changed if a new cluster has appeared or disappeared, and we consider that a cluster has disappeared if on an iteration of the clustering step, DBSCAN cannot find a cluster anymore near the center of an old cluster from the previous iteration. This condition tolerates that the centers of the clusters move slightly between iterations, which is to be expected as their position is being refined the more observations are collected. As soon as the clusters are deemed stable, we can proceed to the updating step.

3.5 Map Update and Correction

We proceed to the update step after the clusters have stabilized and the matching step has occurred. In order to produce the updated map, we simply retain the road signs from the prior map which were matched in the matching step (we assume they were properly positioned), plus the clusters which were not matched to any road sign in the matching step (they are the new road signs which were missing from the prior map).

4. Filtering Approach

The second approach uses a filtering method based on a Probability Hypothesis Density (PHD) filter. It is structured around three main modules (fig. 3): the PHD filter node, the decision node, and the alignment node.

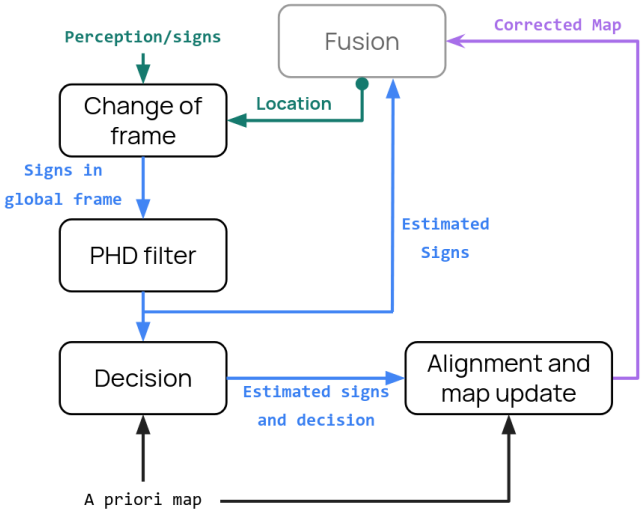

Figure 3: Filtering approach overview

To describe the method, first we will explain the operations of a PHD filter and its relevance in our case. Next, we highlight the assumptions made and the adjustments implemented. Finally, we detail the stopping conditions of the module, the evaluation of the generated map, and the post-processing steps.

4.1 Probability Hypothesis Density filter

The Probability Hypothesis Density (PHD) filter is a multiple-target tracking filter designed to handle an unknown number of targets, based on finite set statistics [15] and has already been proved for feature-based mapping and SLAM [16]. It allows for the recursive estimation of both the cardinality and individual states of a target set (X_k) based on a set of observations (Z_k). This filter operates on the principles of a Bayesian filter and utilizes the PHD function, which, when integrated over a given space, provides the count of elements within that space.

A necessary approximation for performing the computations of the filter is that its elements are Gaussian distributed. Consequently, the filter actually processes a mixture of Gaussians, both as input and output. In a mixture, a Gaussian distribution is characterized by a weight (ω), a pose (x, y, α), and a covariance matrix.

An iteration of a PHD filter consists of five steps: prediction, update, pruning, merging, and birth.

The prediction step propagates each Gaussian in the input mixture based on the previous time steps, using the system's evolution model.

The update phase incorporates the observations. All Gaussians resulting from the prediction step are considered hypotheses. For each hypothesis, its weight is multiplied by a factor of $(1 - P_D)$, where P_D is the probability of an element being detected. Let m_k be the number of observations and H_k the number of hypotheses. For each observation, H_k new Gaussians are calculated. By definition, the weight of a component calculated between an observation and a hypothesis is higher when the observation aligns closely with the hypothesis. The weights are then normalized to account for potential false detections and to ensure that each observation contributes to a total weight of 1.

The pruning step removes components with weights deemed too low, considered insignificant while the merging step combines Gaussians that are close in position and angle. This prevents having multiple components with small weights that are spatially close, by merging them into a single Gaussian with a necessarily larger covariance.

Finally, the birth step adds one component per observation to the mixture.

4.2 Field of View Management

The PHD filter and the assumptions made are typically used for tracking moving objects that are expected to be visible when they exist. However, the problem considered here does not fit this scenario perfectly, as we are dealing with signs (static entities) with limited visibility (they may not always be visible). This has required us to find solutions to adapt the filter to the framework of the study.

Since signs are static, the prediction phase of the filter is a constant. However,a model uncertainty is added by the mean of a covariance matrix.

Regarding the visibility of the signs, since the perception method used is LiDAR, a sign is only visible when facing the vehicle. Given the track's layout and the various possible positions for the signs, they are only visible within a restricted field of view. A solution is to determine a visibility field that accommodates any sign configuration around the track.

The factors considered are the distance from the vehicle to the sign and the angle. If a sign is too close (<2 meters) or too far, it is considered not visible. The sign's angle is used to assess whether the vehicle is directly facing it. Figure 4 illustrates a possible configuration, with non-visible signs shown in red and the visible one in green, along with its visibility field.

The implementation of sign visibility has been incorporated into the PHD filter in two main ways. Firstly, it is

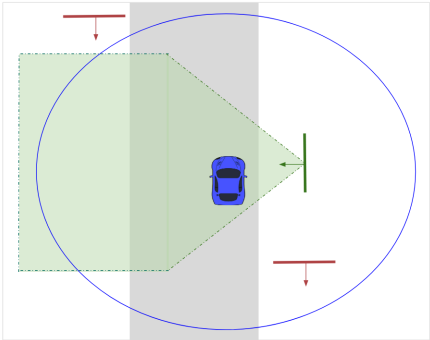

Figure 4: Visibility of a sign depending on its orientation and distance from the vehicle

used to filter observations. We only consider observations if we are reasonably certain that the vehicle can see the sign, which helps filter out lower-quality observations. However, the defined visibility field remains relatively broad. Secondly, this visibility information is applied during the update step of the filter, with a more restricted visibility area. If a Gaussian pose is considered as a non-visible sign, its detection probability P_D is set to 1. This means that when multiplying the weight by the factor $(1-P_D)$, the sign is treated as if it cannot be undetected because it cannot be seen. The second application occurs during the calculation of a new Gaussian component between an observation and an hypothesis. The formula for the weight of this component is as follows:

$$\tilde{w}_{k|k}^{i\times\mathcal{H}_k+h} \leftarrow P_D w_{k|k-1}^{h} \mathcal{N}(z_k^i; \hat{z}_{k|k-1}^{h}, S_k^h) \qquad (1)$$

where $w_{k|k-1}^{h}$ corresponds to the weight of the hypothesis, $\tilde{w}_{k|k}^{i\times\mathcal{H}k|k-1+h}$ to the weight of the new component, and z_k^i is the observation. It can be noted that if this hypothesis is not visible, $P_D = 0$, and thus the observation cannot be associated with this hypothesis.

4.3 Stopping Conditions and Map Evaluation

The decision node receives as input the poses of the signs estimated by the PHD filter. It determines when the map is stable enough to proceed to the next phase. This decision involves two metrics: the distance traveled by the vehicle since the start of the global correction node and the Generalized optimal sub-pattern assignment (GOSPA) metric [17].

Intuitively, it is easy to deduce that at least one complete lap of the track is needed to ensure all signs have been seen. Therefore, we calculate the distance traveled by the vehicle, and as long as it is less than a bit more than one lap, map correction continues.

The GOSPA metric calculates the distance, or rather the degree of dissimilarity, between two finite sets of elements. It is computed by associating points from the two sets and then calculating the metric based on the

distance of associated points, the number of false detections, and the number of missed detections. Specifically, we use the mean-GOSPA, which is the GOSPA metric divided by the number of detected elements. This adjustment accounts for the fact that with more elements to compare, there are more potential sources of error. To compute the mean-GOSPA, we use both the prior map and the map from the PHD filter. Even if the prior map data might be noisy, it contains a number of well-placed signs. The presence of incorrect or missing signs in the prior map does not prevent the mean-GOSPA from decreasing as new, well-placed signs appear in the PHD map.

After more than one lap of the track, if this metric remains constant over a certain period, we consider the map to be stable and thus conclude that that diagnosis phase has converged.

4.4 Alignment

Once the decision is made to stop updating the PHD map, we plan to perform an alignment of the generated map with the prior map. This aims to correct any potential biases that may be introduced, particularly by localization errors.

To achieve this, we use an Iterative Closest Point (ICP) algorithm [18]. ICP is employed to match two datasets (often in the form of point clouds) which represent two partial views of the same object. Each view consists of a set of points, and the goal is to iteratively minimize the distance between these points.

Finally, after all these steps, the aligned map is sent to the localization system to replace the prior map.

5. Results and discussion

In this section, we present the results obtained from the two previously described approaches: the grid-based approach and the filtering approach using the PHD filter. These results allow us to evaluate the performance of each method in terms of road sign localization accuracy, map error correction, and data stability. Both approaches were tested on the same dataset to ensure a fair comparison and to identify the strengths and limitations of each method.

The environment in which these methods were tested is the ”Seville” test track at UTC. Figure 5 shows the shape of the track, the actual placement of the road signs (red), as well as the prior map (green). As a reminder, the prior map includes: a certain number of exact signs, signs that exist but with errors in position and angle, signs that do not exist, and signs that are missing compared to the ground truth.

Figure 6 shows the result of the grid-based approach after the stopping condition was met. The clustering and matching steps retained a subset of the prior map and produced a set of new signs, which together make up the updated map. As can be seen on the figure, one

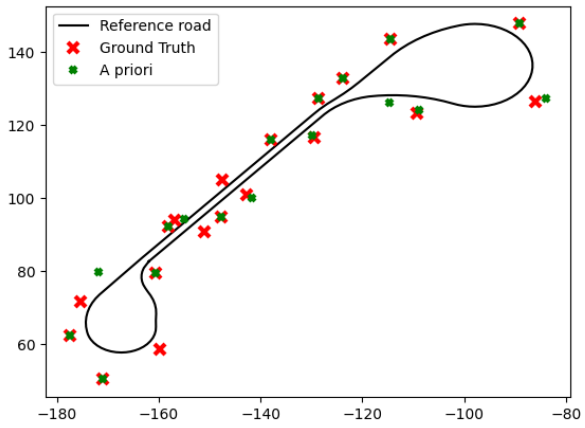

Figure 5: Map of the driving area with ground truth and prior map whose errors are clearly visible

real sign was not recognized. This was caused by the inaccuracies in the perception module which caused two nearby real road signs to be detected by DBSCAN as one single cluster. In addition to that, some of the points retained from the prior map are clearly misslocated. These errors are due to the matching step, that prevents the method to correct small localization errors.

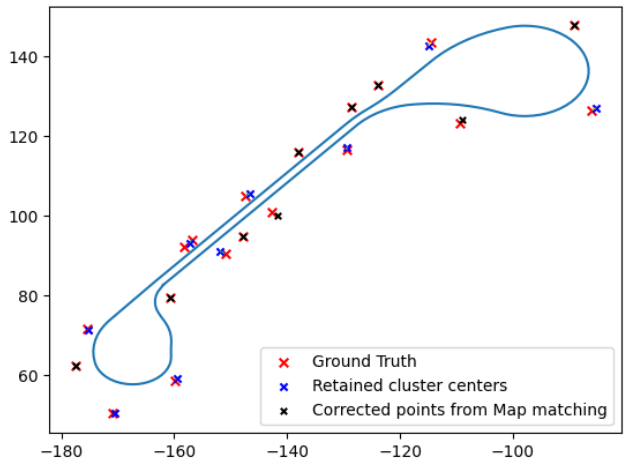

Figure 6: Comparison of the actual road sign, the centers of the clusters and their state in the matching algorithm (grid-based approach)

Figure 7 shows the counter value during the experiment to determine when to stop. Each time a LiDAR scan is obtained, the counter value increases while the clusters do not change, and is set to 0 whenever a change is detected. In this experiment the value started increasing constantly after 90 seconds, which means that no new road sign was detected after this time. It reaches a threshold of 1000 after 175 seconds, signaling that the map verification is complete and the

prior map can be updated.

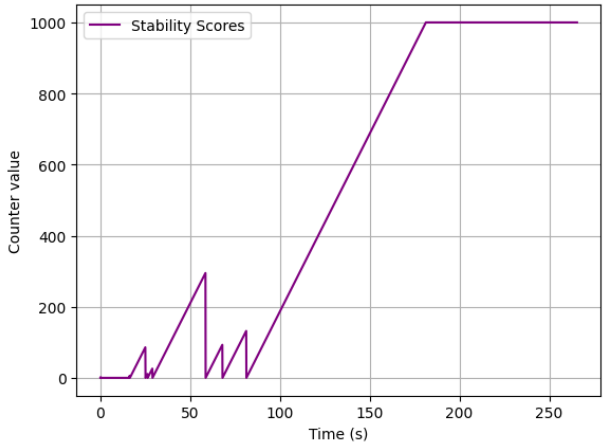

Figure 7: Mapping stability indicator as a function of time (grid-based approach)

Concerning the PHD filter, the chosen value for the detection probability P_D is set at 0.8, reflecting a high confidence level in the system's ability to detect panels. Additionally, the weight assigned to new observations during the birth step of the PHD filter is initialized at 0.01, allowing for a balanced contribution of fresh data without overwhelming the existing hypotheses. The covariance matrix is initialized based on the uncertainty associated with the data sources, particularly considering the inherent uncertainties of the LiDAR measurements. This careful initialization of the covariance matrix is crucial for accurately modeling the positional uncertainties and enhancing the overall effectiveness of the filtering process. The result of the filter-based approach is shown in fig. 8.

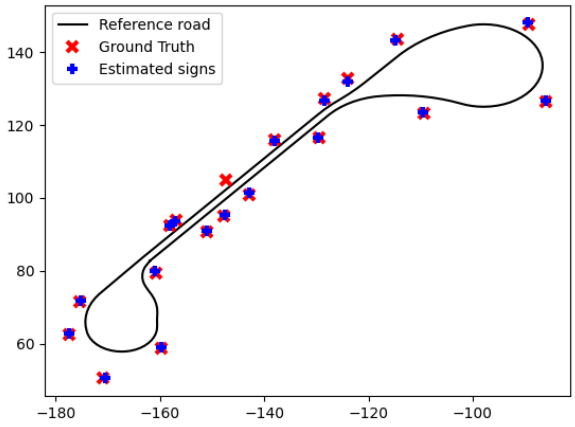

Figure 8: Map obtained through the filtering approach

As mentioned previously, it is essential to detect the convergence of this approach. Figure 9 shows the mean-GOSPA between the prior map and the estimated map using the PHD filter as a function of the distance traveled by the vehicle. The curve initially decreases and then begins to stabilize around a asymptotic value. This value cannot be zero because this test compares the map with the prior map, which contains errors, missing signs, or extra signs (the lower the mean-GOSPA, the better the correspondence between the two sets). The stabilization of the mean-GOSPA indicates that there are no more significant changes in the mapping. The stabilization of this indicator is therefore a good indicator to stop the diagnosis of the prior map.

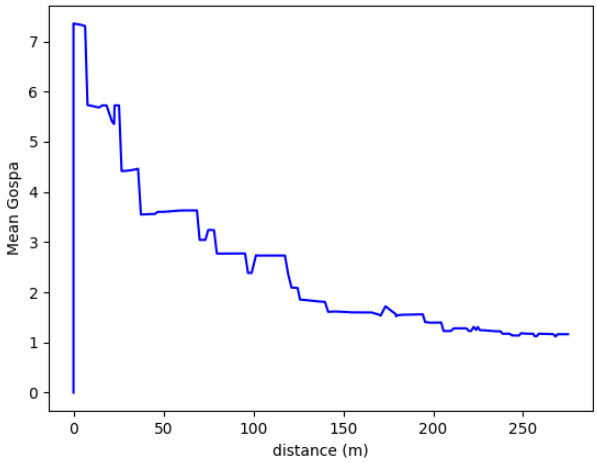

Figure 9: Mean-GOSPA as a function of the distance traveled by the vehicle (filtering approach)

Figure 10 shows that the number of signs does not increase monotonically. This behavior can be attributed to factors such as missed detections and variations in the weights of the Gaussians, which can temporarily lead to the absence of certain signs. Consequently, while the overall trend may show an increase in the number of detected signs, fluctuations occur due to these uncertainties in the detection process. This sensitivity to missed detections stems from the fact that the PHD filter only accounts for the number of detected elements without considering the uncertainty associated with that count. As a result, fluctuations in the detected number of signs may occur, indicating the need for methods that better incorporate uncertainty for more robust tracking.

Metric	Grid-based method	Filtering Approach
Missed detections	1	1
Clutter	0	0
Mean absolute error	0.702m	0.3751 m
Standard Deviation	0.285m	0.1403 m
Mean error Angles	unknown	9.7754°
Standard Deviation Angles	unknown	7.4703°

Table 1: Comparison of the two methods

Table 1 provides a comparison of the approaches using both qualitative and quantitative data. Both methods

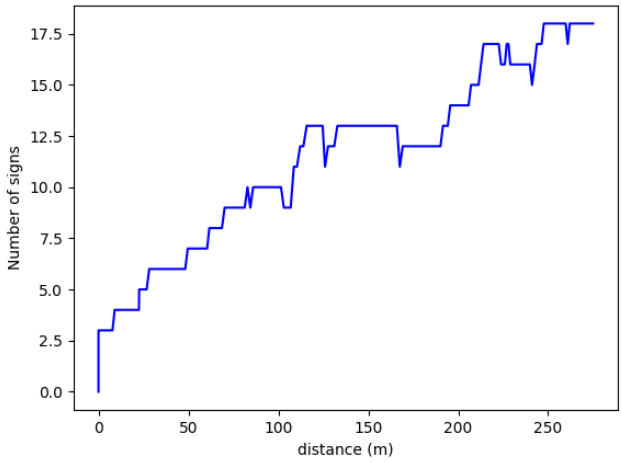

Figure 10: Number of signs estimated by the PHD filter

experience missed detections, albeit for different reasons. The grid-based method may fail to detect signs if the grid or sensor resolution is insufficient, leading to situations where closely spaced signs are perceived as a single entity. In contrast, the PHD filter is highly sensitive to missed detections; if the perception module fails to identify a sign during one or a few iterations, the corresponding Gaussian weight, representing the probability of the sign being located at that position, decreases rapidly and may even disappear. Regarding the accuracy of the methods, both perform well, with notably better results for the second approach. Furthermore, the filter-based method also provides information about the orientation of the signs.

Regarding the stopping conditions, two proposed indicators are relevant. For the PHD filter, the convergence of the mean-GOSPA effectively is clearly visible. An automatic stopping condition based on this metric was not implemented in this work but this opens the door to a wide range of solutions. For the grid-based method, the stopping condition is met as soon as the vehicle is not making new observations during a period of time, regardless of the accuracy of the clusters and the vehicle's trajectory. The tuning of this duration needs more tests in different conditions.

6. Conclusion

This paper addressed the challenge of ensuring accurate and up-to-date HD maps for autonomous vehicle navigation. Two automatic methods for detecting and correcting localization signs were proposed, implemented and compared.

While the PHD filter approach allows for more accurate positioning of signs, the total number of detected signs heavily relies on the quality of the perception module. Conversely, the grid method is more robust, albeit with slightly lower accuracy.

Indeed, the grid-based approach was able to extract the localization of most road signs, even though it was notably less accurate than the filtering method. By taking advantage of the matching process, it can localize road signs from the prior map while they remain unconfirmed. Some improvements could be made to the way the grid accumulates observations. The current implementation of our system does not take into account how clutter detections could accumulate on the grid over long periods of time, causing DBSCAN to eventually falsely detect clusters made of them. A potential solution could be to create multiple smaller grid maps instead of one, each with different clustering parameters based on how long the associated physical location is visible by the perception module. We also believe significant improvements in accuracy could be obtained by further tuning the proximity parameters of the GMC algorithm.

Regarding the filtering approach, the results obtained from the test data presented in this article demonstrate the real potential of this method. Its structure allows it to achieve good results and it can also correct biases present in a sensor. The main area for improvement remains the issue of miss-detections, which significantly impact the filter's performance. An idea can be accumulating observations to improve the results. Another potential solution can be implementing a Cardinalized Probability Hypothesis Density (CPHD) filter, which retains uncertainty about the number of elements. This filter is much more robust but also needs a management of the visibility field of view.

Finally, as both methods were tested with a highly accurate localization system and an HD precise map, even if it contains some error, it would be interesting to see how a less accurate localization or map system could behave on the presented approaches.

Acknowledgment

This work was partially carried out by some students of the université de technologie de Compiègne who would like to thank all the other students that followed this course and the professors who helped them to manage this project, especially A. Lima.

References

[1] J.-H. Pauls, T. Strauss, C. Hasberg, M. Lauer, and C. Stiller, "Can we trust our maps? an evaluation of road changes and a dataset for map validation," in 21st International Conference on Intelligent Transportation Systems (ITSC), Nov. 2018, pp. 2639–2644.

[2] D. Pannen, M. Liebner, W. Hempel, and W. Burgard, "How to keep HD maps for automated driving up to date," in IEEE Int. Conf. on Robotics and Automation, May 2020, pp. 2288–2294.

[3] C. Zinoune, P. Bonnifait, and J. Ibañez-Guzmán, "Detection of missing roundabouts in maps for driving assistance systems," in IEEE Intelligent Vehicles Symposium, Jun. 2012, pp. 123–128.

[4] D. Pannen, M. Liebner, and W. Burgard, “HD map change detection with a boosted particle filter,” in IEEE Int. Conf. on Robotics and Automation, May 2019, pp. 2561–2567.

[5] A. Welte, P. Xu, P. Bonnifait, and C. Zinoune, “HD map errors detection using smoothing and multiple drives,” in IEEE Intelligent Vehicles Symposium Workshops, Jul. 2021, pp. 37–42.

[6] C. Zinoune, P. Bonnifait, and J. Ibañez-Guzmán, “Sequential FDIA for autonomous integrity monitoring of navigation maps on board vehicles,” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 1, pp. 143–155, Jan. 2016.

[7] O. Hartmann, M. Gabb, R. Schweiger, and K. Dietmayer, “Towards autonomous self-assessment of digital maps,” in IEEE Intelligent Vehicles Symposium Proceedings, Jun. 2014, pp. 89–95.

[8] Y. Liu, M. Song, and Y. Guo, “An incremental fusing method for high-definition map updating,” in IEEE International Conference on Systems, Man and Cybernetics (SMC), Oct. 2019, pp. 4251–4256.

[9] M. Stübler, S. Reuter, and K. Dietmayer, “A continuously learning feature-based map using a bernoulli filtering approach,” in Sensor Data Fusion: Trends, Solutions, Applications (SDF), Oct. 2017, pp. 1–6.

[10] K. Kim, S. Cho, and W. Chung, “HD map update for autonomous driving with crowdsourced data,” IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 1895–1901, Apr. 2021.

[11] K. Jo, C. Kim, and M. Sunwoo, “Simultaneous localization and map change update for the high definition map-based autonomous driving car,” Sensors, vol. 18, no. 9, p. 3145, Sep. 2018.

[12] R. B. Rusu, “Semantic 3d object maps for everyday manipulation in human living environments,” KI - Künstliche Intelligenz, vol. 24, pp. 345–348, 2010.

[13] D. Deng, “Dbscan clustering algorithm based on density,” in 7th International Forum on Electrical Engineering and Automation, 2020.

[14] T. Bailey, “Mobile Robot Localisation and Mapping in Extensive Outdoor Environments,” Ph.D. dissertation, The University of Sydney, 2002.

[15] R. Mahler, “Multitarget bayes filtering via first-order multitarget moments,” IEEE Trans. Aerosp. Electron. Syst., vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[16] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, Random Finite Sets for Robot Mapping and SLAM (Springer Tracts in Advanced Robotics). Berlin, Heidelberg: Springer, 2011, vol. 72.

[17] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, “Generalized optimal sub-pattern assignment metric,” in 2017 20th International Conference on Information Fusion (Fusion), 2017, pp. 1–8.

[18] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, pp. 239–256, 1992.