The Use of Spatial Memory for Advanced Driver Assistance Systems: Preventing Stationary ACC False Alarms

Hendrik Deusch1, Regine Graf1, Martin Fritzsche2 and Klaus Dietmayer1

Abstract—This paper presents a self-learning spatial memory approach for advanced driver assistance systems. The storage concept for this memory is based on an object relational database with support for spatial queries. This memory component is applied to the problem of stationary false alarms of an adaptive cruise control system. Test results which demonstrate the practicality of this approach are also provided. Furthermore, an evaluation for different weather conditions is presented.

I. INTRODUCTION

The steady increase in processing power in very little space has already lead to the realization of more and more advanced driver assistance systems. Most likely, this progress will not come to an end in the foreseeable future. Therefore, it is only reasonable to think about more complex systems, although they may not yet be possibly implemented on an electronic control unit. One such system is a self-learning spatial memory (SLSM).

In contrast to a digital map, which is the basis of every navigation system, a SLSM has the ability to grow continuously and, what might be even more important, to forget or replace invalid information. Having such a system would allow for completely new functionalities and it would also bring multiple advantages to the already present driver assistance systems. On the one hand, a SLSM could be used to store information about road characteristics of repeatedly driven routes (e.g. slope and curvature) as presented by Piegsa and Reuss [9]. This way very fuel efficient automatic gear shifting would be viable and, as a side effect, errors in the data provided by the navigation system could be corrected. On the other hand, these characteristics could be used to improve already present systems like the adaptive headlight in order to create a predictive headlight [6]. Also, creating a completely new kind of navigation system which operates differently depending on the observed driver’s behavior would be possible as proposed in [3]. Furthermore, a SLSM facilitates the eradication of certain errors of other driver assistance systems. As long as a malfunction can be detected and as long as it is dependent on the global position of the vehicle, information about it can be stored in the spatial memory. This kind of information may be used to suppress a faulty behavior of the system. One example is the problem of stationary false alarms of adaptive cruise control (ACC) systems.

The remainder of this paper is organized as follows: In section II an overview of the storage concept for the SLSM is provided. Section III concerns itself with the application of the memory vehicle concept to the problem of stationary ACC false alarms. A presentation of preliminary results closes this section. Section IV finally sums up all findings.

II. DATA STORAGE CONCEPT

The foundation of the data storage concept is the object-relational database management system (ORDBMS) PostgreSQL and its extension PostGIS, which adds support for geographic objects. The advantages of using a geo-spatially enabled ORDBMS are manifold: The most important aspect is that data can be indexed easily, and thus accessed by a global position. Since the envisioned spatial memory should not only be applicable to storing one single type of object, the ability to store a variety of different types of objects is also crucial.

All types of objects are stored in one table which only holds information about the global geometry, the type of object, a confidence level, as well as a link to the binary data which in turn contains all further information of the stored object (see Table I).

<table>
<thead>
<tr>
<th>ID</th>
<th>position</th>
<th>p&lt;sub&gt;conf&lt;/sub&gt;</th>
<th>type</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>POINT(9.944, 48.42)</td>
<td>0.84</td>
<td>ACC_FA</td>
<td>0110...</td>
</tr>
</tbody>
</table>

The geometry of the object is in the simplest of all cases just a globally defined point, but it may as well be a line representing a piece of a route. The confidence level of an object p<sub>conf</sub> represents the probability that the stored information is valid. It depends on the type of object and thus has to be calculated by the actual object itself. If the object e.g. represents a piece of a route with a certain slope, for which the optimal points for gear shifting should be determined, p<sub>conf</sub> would mainly depend on the measurement uncertainty of the height sensor and on the number of times the route has been driven. Objects also have to have the ability to delete themselves from the data base, in case the information they provide is outdated – e.g. when road works have been completed and thus the course of the road has changed. An easy way to find out if this is the case is to observe whether the object information does or does not correspond to the sensor data during several passings of their position.

TABLE I: Data Base Representation of All Objects
III. STATIONARY ACC FALSE ALARMS

The adaptive cruise control system [10] is able to take over longitudinal vehicle control by controlling the engine and the brakes of the vehicle. As long as no preceding vehicle is present, the system controls the speed similarly to a standard cruise control system. As soon as a slower lead vehicle is detected, the speed is automatically reduced in order to keep a certain distance or time gap between the vehicles.

The detection is usually based on radar data. Since radar sensors are able to measure the speed difference directly by exploiting the Doppler effect, the moving – and therefore interesting – objects can be easily detected. Neglecting stationary objects was a legitimate approach at the time the first ACC systems were developed, considering that they were designed for use on highways only. On this kind of road stationary objects on the road only exist at the tail end of traffic jams – a situation in which the driver has to take over control anyway because of the limited deceleration rates of most ACC systems. Although ACC has been continuously improved in recent years, many adaptive cruise control systems still ignore stationary objects. This system behavior may confuse the driver, if he or she is not totally aware of the abilities and limitations of the ACC system. Furthermore, drivers fully trusting their ACC systems may exhibit reduced situational awareness. To enable ACC systems for inner city scenarios, where the a-priori probability that a stationary object is relevant is much higher than on highways, the problem of stationary false alarms has to be faced.

As soon as objects for which a velocity unequal to zero has never been measured are also considered and thus tracked, a lot of objects which are not actually on the road may cause a faulty system behavior: The car may brake although there is no other car or obstacle blocking the road. Stationary objects can often be excluded from the set of relevant objects by using advanced lane detection systems. But, due to measurement uncertainties, this may not be possible for every object standing very close to the lane boundary (e.g. street lights), especially in the case of traffic islands, strong road curvature or roundabouts. Furthermore, advanced lane detection systems will also fail when no lane markings are present.

Although there already exist approaches to detect bridges[4] or gully covers with automotive radar sensors, it is neither constructive nor possible to develop a detector for each kind of irrelevant stationary object that may occur. Thus, a generic methodology is required to handle these objects. Such an approach is described in the following.

In this paper, all stationary, well reflecting (i.e. almost exclusively metallic) objects on the road or very close to it are considered potential false alarm objects, except for vehicles. On an inner city course, most of these kind of objects are either lamp posts or traffic signs.

A. Sensor Data

The main sensor used here is the Frequency Modulated Continuous Wave (FMCW) based Continental ARS 3XX automotive long range radar [2]. It is mounted above the center of the front bumper. The sensor offers radar data for the far field, that ranges from 0.25 to 200m, from −8.5 to +8.5 deg with a resolution of 1 deg as well as data for the near field, which is from 0.25 to 60m, for azimuth angles between −28 and +28 deg with a resolution of 4 deg. The sensor provides new data every 66,5 ms. Due to the limited angular resolution in the near field, only the data from the far field is evaluated here.

By preprocessing the reflection peaks, the so called targets are formed. These targets provide, along with the velocity and distance, also an estimated Radar Cross Section (RCS) for each of the objects. The RCS is a measure for the detectability of an object when using a radar sensor. The RCS value in $m^2$ corresponds to the area of an isotropically reflecting object. In this paper all RCS values are given in decibel (dB).

The RCS depends, among other factors, on the object’s size, its reflectivity, i.e. its material, the incident angle and the reflected angle. Therefore, it clearly is a property of the target, although its absolute value is not solely dependent on the target. An example for the course of the RCS value over the distance of multiple approaches towards one stationary object is given in Fig.1. This raw target data is used instead of actual data from an ACC system, where stationary objects are in most cases ignored, to allow for a statistical evaluation. To reduce the amount of data, not every stationary object is considered, but only the objects close to the road, which would potentially lead to an actual false alarm.

For the the vehicle detection (see III-B), a Prosilica EC750 camera mounted near the rear-view mirror provides images with a resolution of 752x480 px. The camera and the radar are calibrated such that the sensor data can be transformed to the vehicle fixed body frame. In addition, a GPS sensor for ego-localization is used.

![Fig. 1: Curve of RCS over distance for multiple approaches towards one stationary object.](image-url)
B. System Overview

The complete system for the unsupervised suppression of ACC false alarms, as depicted in Fig. 2, comprises several components. First of all, the stationary radar targets are tracked with a JIPDA filter (III-B.2). This step is necessary to provide a more stable detection of each of the potential false alarm objects. Raw detections in contrast to tracks may be lost when a relevant stationary object is occluded in some of the radar scans. This happens especially when a lead vehicle is present. Parallel to the tracking of the radar targets, a second JIPDA filtering process is running. This filter takes vehicle detections from a continuous stream of camera images (15 FPS) as well as the stationary radar targets as input and therefore tracks vehicles. Here, only measurements from the radar are allowed to create new tracks, to avoid tracking moving vehicles. This extra processing is required to exclude all motionless vehicles from the set of potential false alarm objects which is done in the subsequent “Filter” block. Then, the remaining track data is passed to the false alarm processing block. This block also receives the presently relevant false alarm objects from the spatial memory component. To be relevant, the false alarms have to be within a bounding box with an edge length of $500\text{m}$ around the current position. The position for the data base queries is provided by a GPS-receiver. To reduce the amount of data base interactions, the false alarm objects are precached. The false alarm processing block then performs a matching (see III-E) between the current tracks and the false alarm objects from the spatial memory. Matching is only performed if the difference between the estimated global position of the track and the stored position of the potential false alarm object is within a threshold (i.e. below GPS uncertainty). If the match is performed successfully, a suppression of the erroneous ACC system behavior may be triggered. Further, if the the track is also complete (i.e. the object has been passed) the memory is updated with the data from the track. If the matching of a completed track fails, a new false alarm object with a low existence probability $p_{\text{conf}}$ is created.

1) Vehicle Detection: Vehicle detection in the monocular image sequence of the Prosilica EC750 camera is executed by an implementation of the popular Boosted Cascade by Viola and Jones [11]. Based on the findings of Gabb et al. [5] the following features are used:

- Haar-like filters
- Low-level image statistics (standard deviation)
- Edge Orientation Histograms (EOH)
- Activation History Features

This detector accurately finds front and rear sections of vehicles, while the false alarm rate is kept relatively low (see the ROC curves in [5]). Thus, the subsequent tracking can rely on these detections with a high probability.

2) Joint Integrated Probabilistic Data Association Filter (JIPDA): As presented in [8], the JIPDA algorithm is able to track multiple targets despite the presence of clutter. Thus, it works well to track potential stationary false alarms, as well as motionless vehicles. The actual filter used here is closely related to the one presented by Mählisch et al., hence details on the tracking process can be found in [7]. There are two main differences between the filtering employed here and the one proposed then. First of all, since in this paper stationary objects are tracked in contrast to moving vehicles, as it was the case in [7], the filter is parameterized differently. Secondly, instead of using a laser scanner and a video camera, a radar sensor for the false alarm candidates and respectively a radar sensor and a video camera for the vehicles are utilized.

C. Data Object

In addition to their position and the confidence value $p_{\text{conf}}$, the data objects for the ACC false alarms also contain the courses of the RCS of all approaches to the stationary object. That is, for each drive towards the object, all RCS values, i.e. one every 66.66ms, have to be stored. The number of RCS values may be different for each of the passages due to different driving speeds or due to the occlusion of the object. To allow for an easier comparison of the curves (see III-E), the courses are discretized at fixed distances $d \in (0, 200]\text{m}$ between the ego vehicle and the object. To reduce the required disk space it is possible to calculate a kind of averaged curve of the RCS. The object confidence $p_{\text{conf}}$ depends on the number of times the stationary object has been passed while it was not occluded by another vehicle and of how often it actually has been matched. If $p_{\text{conf}}$ falls below a threshold, the object deletes itself from the memory. Statistical requirements for a 100 $(1-\alpha)\%$ confidence can be derived based on the desired width of the confidence interval given the dimensionality of the false alarm location (normally 2D) and the standard deviation of the measurements.

D. Initialization Phase

In an unknown environment, the system usually starts off with an empty data base. The system then begins to collect the false alarm data just as the host vehicle drives. Data from objects which actually trigger an ACC alarm is stored in the data base. Passing the object multiple times and successfully matching it, increases its associated confidence and therefore lets it persist in the memory to then suppress the false alarm.
in the future.
In a real world scenario, this kind of initialization phase can be annoying. A possible solution would be that the OEMs provided a proprietary server. This server could collect and merge the false alarm maps generated on the vehicles of their customers. This server could then supply an up-to-date map of the false alarms. This would either shorten the initialization phase or even supersede it, at least for much-used roads. Another approach would be to ignore stationary objects for the ACC functionality, as long as a given road has not been driven at least \( n \) times, but to still store those which would cause an alarm. As soon as that the vehicle “knows” the false alarm objects of a route it could extend the functionality to allow alarms generated by stationary objects.

E. False Alarm Matching

Figure 1 shows that the RCS curves of different approaches towards the same target seem to be similar. But a direct euclidean point to point matching of the curves is not possible since they are (at least partly) shifted and stretched on the x-axis and on the y-axis. Thus, the matching process of the potential stationary false alarm objects stored in the SLSM and the current tracks of the stationary objects is based on the so called Longest Common Sub Sequence (LCSS) [13]. Before the matching is performed, the current tracks are sampled with a fixed distance between each sample of \( d_s = 0.5m \). Only those distances, where RCS data is available from the data base as well as from the current track, are considered. The data sets can then be written as \( x = [x_0, ..., x_n] \) and \( y = [y_0, ..., y_m] \), where \( x_i \) is the RCS value of the current track at the discretized distance \( d_i \) and \( y_i \) the RCS value of the potential false alarm provided by the memory component respectively.

1) Longest Common Sub Sequence: Finding the longest common sub sequence of two sequences is a well know problem e.g. in bioinformatics where genome sequences are compared. Vlachos et al. presented in [13] the definition of the LCSS for continuous two-dimensional trajectories. LCSS represents a measure of similarity that resembles human perception of similarity.

The adapted definition for LCSS for one-dimensional curves \( x = [x_0, ..., x_n] \) and \( y = [y_0, ..., y_m] \) is as follows:

**Definition 1:** For an integer \( \delta \) and a real number \( \epsilon \), \( LCSS_{\delta,\epsilon}(x, y) \) can be defined as

\[
\begin{cases}
0 & \text{if } x \text{ or } y \text{ is empty} \\
1 + LCSS_{\delta,\epsilon}(Head(x), Head(y)), & \text{if } |x_n - y_m| < \epsilon \text{ and } |n - m| < \delta \\
\max(LCSS_{\delta,\epsilon}(Head(x), y), LCSS_{\delta,\epsilon}(x, Head(y))), & \text{otherwise}
\end{cases}
\]

with \( Head(x) \) being the sequence \( Head(x) = [x_0, ..., x_{n-1}] \).

The value of \( \delta \) controls the maximum number of samples between two points of the two curves to be considered for matching. In the original definition this is a time span, whereas here it is the maximum number of distance steps \( d_i \). The constant \( \epsilon \) can be seen as a matching threshold. It determines how big the maximum difference between two RCS values can be such that these two values are still considered a match. An example of the point to point correspondences determined by the LCSS is given in Fig 3.

As proposed in [12] it is also possible to define a similarity function based on the LCSS value:

**Definition 2:** The LCSS-Similarity \( S_{LCSS} \in [0,1] \) between two trajectories \( x \) and \( y \) is given by:

\[
S_{LCSS}(\delta, \epsilon, x, y) = \frac{LCSS_{\delta,\epsilon}(x, y)}{\min(n, m)}
\]

The resulting similarity of the curves exemplarily depicted in Fig 3 for \( \delta = 10 \) (i.e. \( \pm 5.0 m \)) and \( \epsilon = 2.0 \) is \( S_{LCSS} = 0.7722 \). This result mirrors the impression that the two curves look similar for the most part.

F. Results

1) Evaluation for constant weather conditions: The proposed system has been evaluated on an inner city course of approximately 20 kilometers length. The course has been driven eight times during daytime while the weather conditions were good. In doing so, several intersections were passed multiple times, so that one stationary object could possibly be tracked more than eight times. Only tracks with a minimum length of \( l = 30m \) were evaluated.

Each of the tracks was manually labeled as one of the following:

- irrelevant objects
- stationary false alarm candidates
- still standing vehicles

Furthermore, all stationary objects which have been tracked less than five times were discarded. This led to a total of 152 potential stationary false alarm objects, which have been tracked \( 6.395 \pm 1.608 \) times on average. Also, 188 tracks of different stationary vehicles were collected which in contrast...
to the other objects should obviously not be stored in the SLSM.

For two curves to be considered matching, a minimum LCSS-Similarity of $S_{\text{LCSS}}^{\text{min}}(\delta = 10, \epsilon = 2.0) = 0.53$ has been determined. This threshold was determined by finding the optimal compromise between a high matching rate for two RCS curves of the same object (true positive) and a low matching rate for RCS curves of stationary objects and those of vehicles (false positive), see Fig. 4. The resulting true positive rate is 80.70% and the resulting false positive rate is 16.44%.

Comparing the signals of vehicles to those of potential false alarm objects is meaningful since it can be considered the worst case scenario: A vehicle is standing close to a false alarm object and the braking of the ACC system would be suppressed if the signals stored in the SLMS were matched to the ones of the car. Furthermore, this comparison would not make sense if the RCS values of potential false alarm objects and stationary vehicles were totally different. However, the RCS values’ mean of all the potential stationary false alarms (16.70 ± 6.363) and the RCS values’ mean of the vehicles (15.52 ± 6.027) are alike.

A significantly higher true positive rate seems to be unachievable, since the curves that are not matched actually are different. This may have many reasons. First of all, while driving towards a stationary target, the radar signal must not always be reflected by the same section of the object. Secondly, the ego vehicle was not always driving exactly the same path (e.g. on another lane), such that the approach angle between the sensor and the object was not the same for all passings. Also objects which were very close to the actual object may have influenced the RCS estimation. A second evaluation shows that the presented approach is also able to distinguish between different false alarm objects in most cases. When trying to match the signals of approaches towards different potential false alarm objects to each other, only 21.00% of the comparisons yield in a match, i.e. result in a false positive, see 4 (right). Many of these false positives can be explained by the fact that although only objects located at different positions were compared, it was not taken into account that it might actually have been similar objects, i.e. approaching two different light posts on a similar trajectory will in most cases lead to a similar RCS curve.

2) Evaluation for mixed weather conditions: As already described in [1] water or snow on the surface of an automotive radar sensor causes performance limitations. This effect is also investigated here. Figure 5 exemplarily shows the mean curve of the RCS of multiple approaches towards one stationary object for different weather conditions. The RCS values during snowfall are lower than during foggy weather and that the RCS values recorded during sunshine are the highest. This degradation of the radar signal is problematic for the matching process proposed here, since a match can only be found within the window defined by the LCSS parameters $\epsilon$ and $\delta$.

This problem manifests itself in the matching results, see Table II. The average LCSS-Similarity $S_{\text{LCSS}}$ and therefore the matching rate – during sunshine is much higher than for the adverse weather conditions. When trying to match RCS curves of one weather condition to the curves of another one, the performance deteriorates even more. These results were obtained on a course of 4 kilometers length, which has been driven ten times during each of the three weather conditions. The radar sensor was not cleaned between the rides. Thus, it is very likely that the thickness of water film was not constant (i.e. it most likely increased) for all rides during foggy conditions or while it was snowing.

Better results can be obtained when the mean RCS of the complete track is removed from all values, see Table III. When subtracting the mean value, the height of the matching window has to be changed to $\epsilon = 1.75$, if a similar (78.22%) true positive rate on the data from the inner city course is to be achieved. This in return has a negative effect on the false positive rate, which increases to 27.61%.

Although the results for each of the conditions are now
similar to the results of the longer inner city course, the matching of mixed conditions still yields much worse results. This may be due to the fact that the data was recorded over a stretch of several months and thus it is possible that some of the stationary objects have changed. This again would not be a problem for the overall system, since it would create a new potential false alarm object at the same position. Then the old one would be deleted, if it was not matched again during future passings of the position – otherwise both instances would coexist.

IV. SUMMARY

This paper proposed an approach to learning potential stationary ACC false alarms. A spatial data base was used to store the geographically referenced objects. This approach is extensible and makes it also possible to let the system forget invalid information easily. For matching stored radar data of the potential false alarms to accruing tracks, the LCSS-Similarity has been used. The evaluation of the matching mechanism showed that it is a promising approach for solving the described problem. Successfully matching radar data of difficult weather conditions is also possible if the mean value is removed.

One aspect that still has to be investigated is how to find the minimum track length which is required for reliable matching. Also, research has to be done on how the system can be sure that information stored in the SLSM is trustworthy or that data is invalid and can be forgotten. Furthermore, the proposed system will be used to implement a localization algorithm. This algorithm will be based on matching sensor data (video as well as radar) to the data of various stationary objects stored in the SLSM, that can be reliably detected.

TABLE II: Matching for different weather conditions

<table>
<thead>
<tr>
<th>Weather Condition</th>
<th>$\widehat{SS}<em>{L</em>{CSS}}$</th>
<th>Matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snow</td>
<td>0.3879 ± 0.2561</td>
<td>36.52%</td>
</tr>
<tr>
<td>Fog</td>
<td>0.5570 ± 0.2097</td>
<td>65.73%</td>
</tr>
<tr>
<td>Mix(Snow,Fog)</td>
<td>0.3576 ± 0.2239</td>
<td>32.99%</td>
</tr>
<tr>
<td>Sun</td>
<td>0.7419 ± 0.1562</td>
<td>94.17%</td>
</tr>
<tr>
<td>Fog</td>
<td>0.5753 ± 0.2092</td>
<td>68.13%</td>
</tr>
<tr>
<td>Mix(Sun,Fog)</td>
<td>0.4664 ± 0.1725</td>
<td>46.21%</td>
</tr>
<tr>
<td>Snow</td>
<td>0.4014 ± 0.2439</td>
<td>37.56%</td>
</tr>
<tr>
<td>Sun</td>
<td>0.7194 ± 0.1662</td>
<td>90.77%</td>
</tr>
<tr>
<td>Mix(Snow,Sun)</td>
<td>0.2663 ± 0.1956</td>
<td>15.21%</td>
</tr>
</tbody>
</table>

TABLE III: Matching for different weather conditions with removed mean

<table>
<thead>
<tr>
<th>Weather Condition</th>
<th>$\bar{S}<em>{\text{mean}}^{(SL</em>{CSS})}$</th>
<th>Matched $\bar{S}<em>{\text{mean}}^{(SL</em>{CSS})}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snow</td>
<td>0.5767 ± 0.1728</td>
<td>74.25%</td>
</tr>
<tr>
<td>Fog</td>
<td>0.5612 ± 0.2097</td>
<td>71.16%</td>
</tr>
<tr>
<td>Mix(Snow,Fog)</td>
<td>0.4701 ± 0.1714</td>
<td>50.68%</td>
</tr>
<tr>
<td>Sun</td>
<td>0.6902 ± 0.1705</td>
<td>89.32%</td>
</tr>
<tr>
<td>Fog</td>
<td>0.6413 ± 0.1759</td>
<td>83.24%</td>
</tr>
<tr>
<td>Mix(Sun,Fog)</td>
<td>0.5452 ± 0.1420</td>
<td>68.60%</td>
</tr>
<tr>
<td>Snow</td>
<td>0.5807 ± 0.1725</td>
<td>73.48%</td>
</tr>
<tr>
<td>Sun</td>
<td>0.6782 ± 0.1956</td>
<td>81.54%</td>
</tr>
<tr>
<td>Mix(Snow,Sun)</td>
<td>0.4490 ± 0.1813</td>
<td>46.65%</td>
</tr>
</tbody>
</table>

REFERENCES