FeedMAP: Improving Safety and Comfort for In-Vehicle Applications by Map Deviation Detection and Online Map Updating

Jan Loewenau,
BMW Group Forschung und Technik, Munich, Germany
jan.loewenau@bmw.de

Sinisa Durekovic,
NAVTEQ, Germany
sinisa.durekovic@navteq.com

Bernd Thomas
NAVIGON AG, Würzburg, Germany
bernd.thomas@navigon.com

Hans-Ulrich Otto
Tele Atlas Deutschland GmbH, Harsum, Germany
hans.ulrich.otto@teleatlas.com

Abstract

Up-to-date map data is a must for current and future navigation and Advanced Driver Assistance System (ADAS) applications. Today, digital maps are normally stored on DVDs or hard disks, with periodic updates only available on replacement disks. However, new mechanisms for updating maps have been investigated and some of them already reached the market. As the real world is changing every day, detecting changes to the road network quickly and at a low cost is a challenge. Although mapmakers continuously survey the European road network for changes, map information is not always up-to-date or accurate.

Introduction

The ActMAP framework [1] provides concepts and methods for wireless distribution of incremental map updates for in-vehicle navigation and Advanced Driver Assistance Systems (ADAS) applications with the general goal to achieve highest up-to-dateness of an in-vehicle map database. Although this incremental map updating framework helps to shorten the time span between map updates significantly the basic assumption is that map deviations are detected by the map suppliers. This obviously has a disadvantage, since constantly checking wide areas of a road network is a time consuming and cost intensive process for update supplies. As a consequence road network changes in remote areas or dynamic events (e.g., road construction sites) are not detected at all or only with a very high latency.

The basic idea of the FeedMAP project [7] is to use the end customer's vehicle equipped with either a navigation system or ADAS application for the automatic detection of map deviations. Consequently a closed loop of map deviation detection and incremental map updating provides a even higher degree of map up-to-dateness for in-vehicle map databases. Thus coupling the ActMAP and FeedMAP framework is conceptually a reasonable step for the following reasons:

- Faster availability of map updates due to automatic, permanent, and global area-wide monitoring of map errors and real world road changes.
- Minimizing maintenance (data acquisition) costs for update suppliers.
- Minimizing data acquisition costs for update suppliers.
- Increasing the quality of maps in general and specifically by reliable update information given from public authorities (e.g., on road status and constructions by Swedish Road Administration) and other trustworthy LBC providers.
- Better quality of service for the end customer due to increased up-to-dateness of maps and additional services like dynamic content updates.

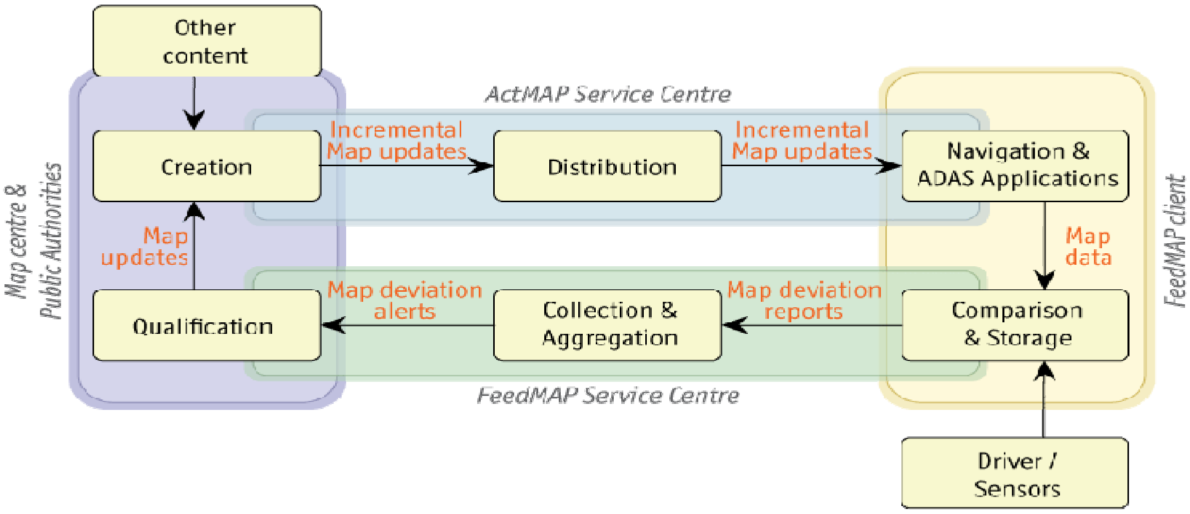

Figure 1: ActMAP - FeedMAP loop

Considering the core functionalities of the ActMAP and the FeedMAP system (Figure 1) the main roles of each system is clearly characterized as Update Distributor (Map Centre and ActMAP Service Centre) and Update Detector (FeedMAP client and FeedMAP Service Centre). Whereas update suppliers like map centers or content providers in general take over the role of an update evaluator/provider. Hence they have to decide if FeedMAP updates derivable from FeedMAP deviation alerts meet a special quality criteria, such that they can be propagated via the ActMAP services to in-vehicle applications.

Deviation Detection

The FeedMAP-ActMAP loop starts when the difference between ground truth and content of the digital map is detected. Difference may be in absence of real-world entity in the digital map, presence of digital map entity that does not exists in reality or in difference between a value of entity attribute stored in the digital map and actual real-world value of the attribute.

In the FeedMAP framework, those differences are called Map Deviations and they are described in XML-formatted data structures referred to as a Map Deviation Report (MDR). These Map Deviations are detected by FeedMAP Clients (FMC). A FeedMAP Client generally fits into two categories: 1) Car probes are FeedMAP clients equipped with sensors and algorithms that are used in deviation detection. 2) Public Authorities (PA) are FeedMAP clients that generate MDR's. However, since PA initiates or at least keeps official records about many attributes contained in the digital map, they are the reference source for that information.

Map deviation detection algorithms implemented in car probes can be generally grouped into three categories: autonomous, manual, and joint detection.

Autonomous deviation detection

Autonomous detection does not involve any conscious driver action; source of the data that indicates the deviation is only provided by different car sensors. An example of the Autonomous detection is detection of missing links. To detect roads that are not in the digital map, system can monitor behavior of the map-matching module of the navigation system. When map matching is unsuccessful, despite of good quality of sensor data such as GPS, one can conclude that vehicle is traveling along 'uncharted' street and Map Deviation Report can be generated.

Manual deviation detection

Manual deviation detection algorithms rely only on driver interventions. Detection of Scenic Routes or changes in Point-of-Interest attributes (telephone number, opening hours, etc.) are typical examples. In general, all manual detection algorithms can be automated by use of hardware sensors and by applying complex software algorithms, but in most cases such approach is not feasible.

Joint deviation detection

Joint detection algorithms are those algorithms where the system detects the deviation, but some driver confirmation is necessary. This confirmation can be explicit (when the system asks the driver for confirmation), or implicit (when driver action confirms the assumption of system).

Investigated Deviation Types

For some deviation types, different deviation detection methods can be developed. Systems can rely on more or less sophisticated hardware sensors and or more or less complex software algorithms in deviation detection. Wrong Legal Speed Limit can be explicitly reported by the driver, deducted from monitoring drivers' behavior, or image-recognition camera-based sensor can do the same automatically. Of course, more advanced sensors and more advanced algorithms will reduce the need for drivers' involvement in deviation detection. Since today's 'Driver Workplace' is already fairly complex, one should avoid manual detections and joint detections with explicit confirmations because they increase the drivers' workload.

Deviation Report Analysis

It is the task of the FeedMAP Service Centre (FMSC) to receive MDR's, to analyze them and to decide if and when a Map Deviation Alert (MDA) is to be generated and sent to the Map Centre (MC).

Figure 2 illustrates the general processing steps of the FMSC. When a new MDR has arrived, it will be processed in up to four steps:

Step 1: Validation will check if the MDR is "good enough" to be processed. Validation includes checking of syntax and up-to-dateness of the report, completeness as well as internal consistency of the data included in the report.

Step 2: Clustering will assign the MDR to an existing or new cluster. The clustering process itself is uses Data Mining procedures. Which clustering algorithm is applied depends on the actual deviation type and location referencing method stated in the Map Deviation Reports. For instance, clustering of link-wide deviation such as Legal Speed Limit can be based on simple metrics that defines zero distance between deviation reports that refer to the same link ID and infinite distance if link identifiers are not identical.

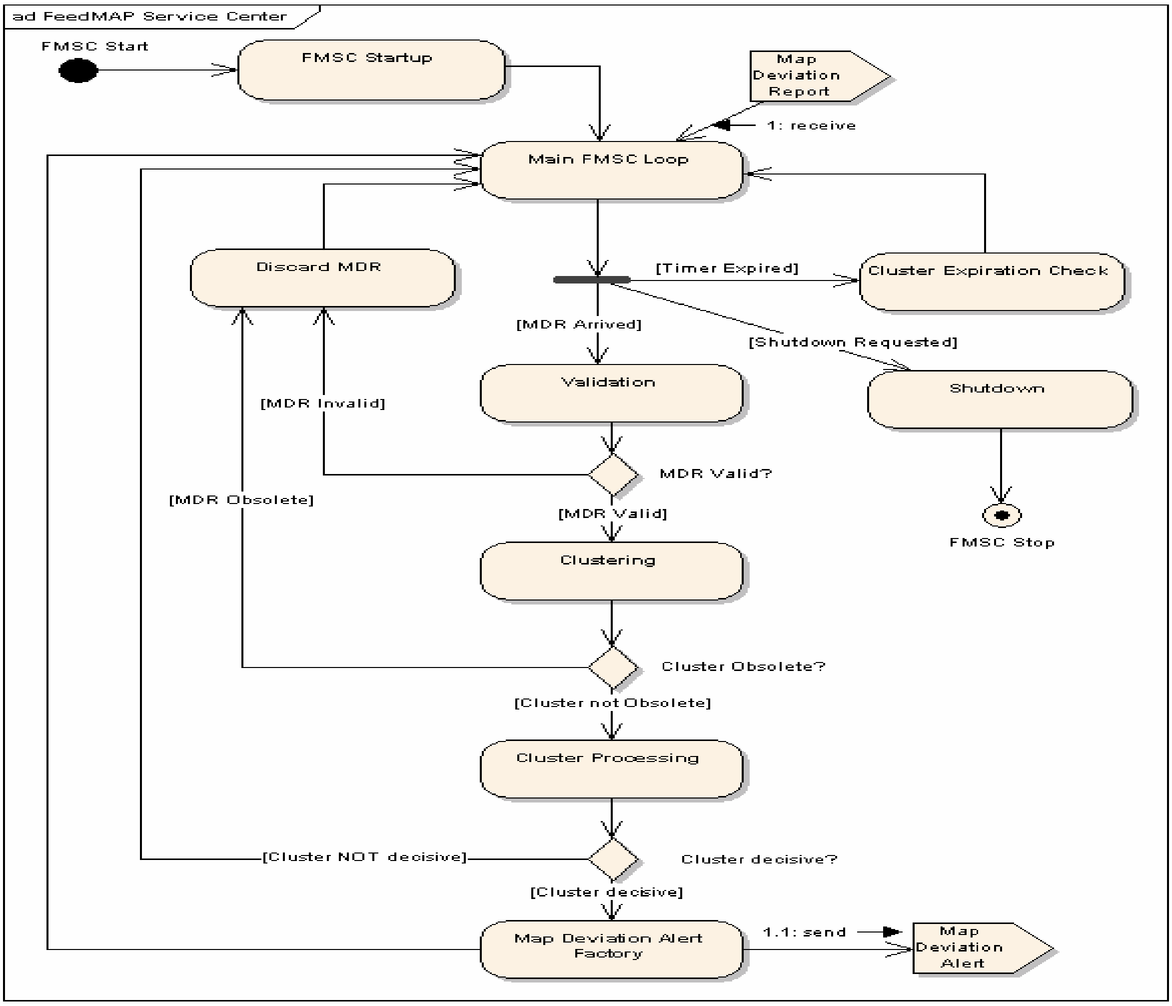

Figure 2: FMSC analysis process

Step 3: Cluster Processing will analyze updated cluster(s). Analysis will be focused on two aspects of cluster data: the cluster ‘Location Centroid’, representative location of the deviation, and on ‘Data Centroid’, which is the most probable corrected value of the deviation. Both Location Centroid and Data Centroid are calculated from Map Deviation Reports taking into account reported data quality as well as confidence to the user that reported the deviation.

Step 4: A Cluster becomes ‘decisive’ if there are enough MDRs in it, and if the quality of its Location Centroid indicates that the location of the deviation is known with enough accuracy. In addition, it may be requested that the quality of Data Centroids are above predefined thresholds. The MDA Factory generates out of ‘decisive’ clusters Map Deviation Alerts.

Applications and FeedMAP-ADAS Test Scenarios

Within the FeedMAP project different implementations of the FeedMAP Clients, Service Centers and applications are developed. In this paper we focus on ADAS [5] related applications and ADAS development platforms based on the concept of the ADAS Horizon. The ADAS Horizon comprises a solution for providing digital map information about the most probable path the vehicle will take to the vehicles CAN-bus.

Adaptive Speed Recommendation and FeedMAP

The BMW application “Adaptive Speed Recommendation” (ASR) is a typical example of ADAS applications whose usability strongly depends on the correctness of the map data. The Adaptive Speed Recommendation Info function provides additional, detailed recommended speed information on the stretch of road the driver is currently covering (see Figure 3). To provide this helpful support, a traffic sign graphics in the instrument cluster – head up display and navigation display - in the cockpit informs the driver of the speed limit at his current location. Adaptive Speed Recommendation is extended with Map Deviation Detection, Reporting and Dynamic Map Update capabilities and it is now active part of the ActMAP-FeedMAP chain.

Figure 3 BMW Vehicle used in the FeedMAP framework

To calculate recommended speed at each moment, the Most Probable path of the Electronic Horizon [3][4][6] is analyzed; the following road characteristics are taken into account: Curves, Legal Speed Limits, Crossings, and Roundabouts. As such, ASR system combines curve Information, Speed Limit Information and Crossing Information ADAS functions in one application. It will warn the user of the need to slow down before the vehicle reaches the point where speed must be reduced. For instance, the ASR may display the information about speed limit 50-300 meters ahead of the actual Speed Limit traffic sign. The exact distance depends on several factors such as current vehicle speed, vehicle braking acceleration (deceleration), driver reaction time, etc. Of course, if the system calculates that the present speed does not violate current or future legal speed limit, no information will be shown.

Since the quality of ASR results heavily on quality of Digital Map Data, it is obvious that this application will benefit from up-to-date map content provided by FeedMAP loop. The Opposite is true as well: ASR plays a critical role in implementations of many car-speed related map deviations. Simplest example is the FeedMAP algorithm for detection of issues with Legal Speed Limit data in the Digital Map. Figure 4 shows a block diagram of the ADASRP[1] platform connected to the BMW sensor CAN box w.r. to the vehicle data.

ASR and Detection of Incorrect Legal Speed Limits

To detect wrong Legal Speed Limits, the system monitors the speed recommended by the ASR system and actual driven speed. Let's assume that a driver drives faster than the ASR recommendation on a particular road segment. In that situation, ASR issues a warning, but this warning is ignored by the driver and he/she does not slow down. Since Legal Speed Limit is part of ASR calculation, one can conclude that actual Legal Speed Limit is higher than the one stored in the digital map: driver ignored ASR speed warning because he knows that the actual speed limit is different from the speed limit stored in the database and used by the system.

A more complex case is if, along a single segment, a driver is significantly slower than the legal speed limit. First of all, the output from ACC radar can be checked to find out if a slow vehicle is present in the front. If this is the case, no conclusion can be reached. If, however, no cars are in front, the driver may simply follow the ASR recommended speed that tells him there is a crossing or sharp curve ahead. Again, digital-map legal speed limit may be correct. When ASR calculates that a driver could be faster, and if there are no cars in front, the most probable reason why the driver is driving slowly is because he saw the actual legal speed limit sign. Under this assumption, the FeedMAP client using ASR will generate a Map Deviation Report about probable wrong Legal Speed Limit.

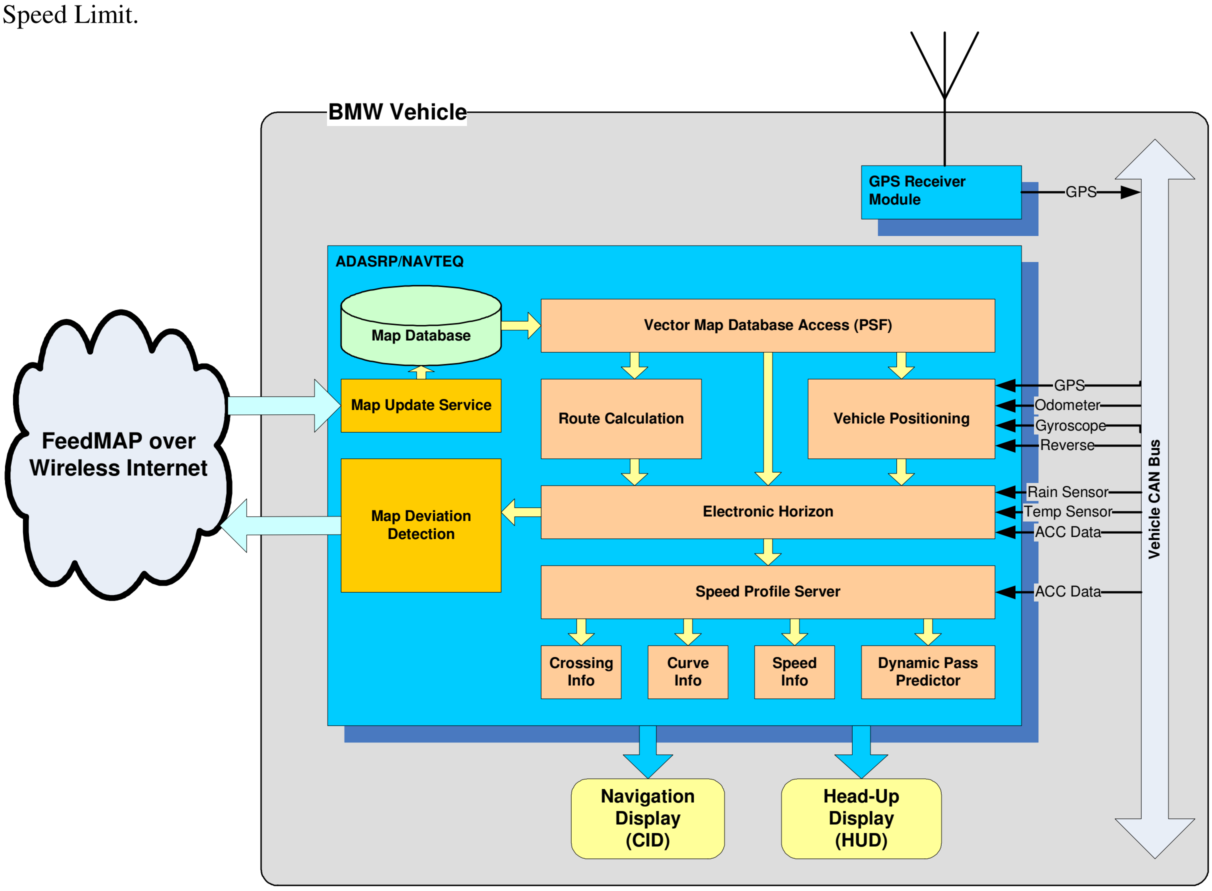

Figure 4 Hardware and software modules used in the BMW prototype test vehicle

Updating the ADAS Horizon

NAVIGON extended within the FeedMAP project their ADAS Horizon Provider solution (MapSensor) with FeedMAP client functionality. This application (FeedMAPSensor) is evaluated in a joint test-site with Volvo Technology (VTEC), and Tele Atlas. The FeedMAPSensor is installed on a VTEC truck, which is equipped with sensors connected to the truck's CAN bus. Such sensors comprise an image lane detection unit, ACC radar, and slope unit. The sensors' information is used to assist the deviation detection algorithms in autonomous detection mode.

[1] NAVTEQ Advanced Driver Assistance System Research Platform is Windows-based framework application for prototyping various ADAS solutions.

The FeedMAPSensor also extends the ADAS horizon and the information on the most probable path by detected deviations (Figure 5). Additionally the FeedMAPSensor is capable of receiving incremental map updates from an ActMAP service centre (Figure 1). Information of the incremental map updates are also used to extend the ADAS horizon.

Consequently ADAS applications based on the ADAS horizon information provided by the FeedMAPSensor to the vehicle's CAN-bus benefit from the FeedMAP concept. Such update and detection information available on the vehicle's CAN-bus can be used to alert the driver about map changes and/or directly can be used by ADAS systems for improved applications. Hence the overall driving safety is increased, because of up to date information.

The FeedMAPSensor comprises the manual detection of 4 deviation types and automatic detection of 5 deviation types. For manual deviation detection a GUI is used that allows the user by "point, click, and select" operations on the map to report: Road Works, Point of Interest, Speed Limit, and Traffic Sign deviations. Although manual deviations detection is of some importance for cases where automatic detection is very complex, the main focus of the FeedMAP project is clearly on the automatic detection of deviations, since this reduces workload and minimizes the risk of disturbing the driver.

FeedMAPSensor automatically detects wrong road geometry, missing road, speed limit, slope and lane info deviations. The detection of wrong road and missing road deviations are solely based on GPS sensor data. The speed limit deviation detection can be performed in two different modes, either automatic or manual. Automatic detection is based on speed information given by the GPS receiver and additional (optional) radar information about the speed of vehicles in front of the truck.

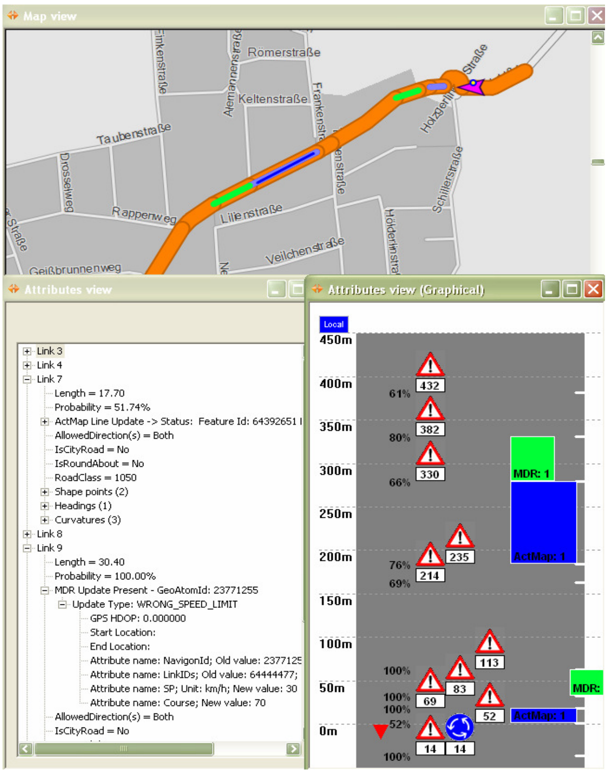

Figure 5 ActMAP Updates (blue), MDR (green), Horizon (orange), attribute and graphical view of ADAS Horizon (left and right

Since the measured speed information solely based on the GPS information might lead to false assumptions on possible speed limit changes, due to possible congestions or illegal speeding by the driver, the radar information is used to improve the speed limit detection heuristic. The vehicle's speed is compared with speed limit attributes attached to the map and based on different computational models the radar information read from the CAN-bus is taken into consideration for estimating a new speed limit.

The detection of Missing Roads is also supported and requires a close interaction with map matching components of the core system. Information about the new road's geometry is collected and reported to the FeedMAP Service Centre. Missing road and wrong geometry detection is mainly based on thresholds for the distance between map matched position and GPS position, number of succeeding "spurious" position samples, and maximal distance between sample points, see in Figure 6. Wrong and Missing Slope information is detected by the use of a slope assessment unit based on the truck's power train control units. This sensor information is compared to slope information when special ADAS maps are used and reported respectively.

For detecting Lane Information deviations a image processing unit by VTEC is used. The vision system provides real time information on the numbers of lanes and lane markers to the truck's CAN-bus. It is capable of detecting the lane width, lane marking types, and information on the neighbour lanes. This data provided by the sensor system plus radar information is used to build a lane model. The detected lane information (markings, number of lanes) is compared to existing ADAS map attributes. In case some lane information is not present for comparison in the digital map (no ADAS attributes) the sensor data is also used to report a MDR containing missing lane information to the FeedMAP Service Centre.

Conclusion

During first field tests some critical issues have been identified, which are briefly described in the following.

A great variance in positioning information given by different GPS receivers was observed. A variance of up to 30m was detected while estimating the position at the same time and same location with 5 different GPS receivers. This has a direct impact on the reported deviation quality, robustness of algorithms used by the FMSC.

The detection in parallel of different deviation types might lead to contradictory MDR's, e.g., for same deviation spots wrong geometry and missing roads have been reported. One explanation can be found in the standard software components which are tailored for navigation tasks and only adapted for deviation detection tasks (e.g., map matcher component). The use of standard map matchers provide only partially good results for detecting deviations, since their general goal is to match the vehicles position on street segments taking into consideration additional map attribute information (e.g., direction of traffic flow). Hence the use of additional modified map matchers seems reasonable. This objective significantly differs from the FeedMAP requirements to detect deviations (aka errors) in the map. In particular the following issues have been observed:

- ‚Bypasses in the network' (parallel roads with connections) lead to ‚valid' map-matching with little map-matching quality reduction hence no error can be detected
- map-matching without turn/direction restrictions finds much more MM solutions – especially in inner-city/junction situations, hence detection can be flawed (especially missing link)
- „The more ‚robust' the map-matching behaviour – the more problems with error detection…" the map-matching configuration needs to be optimised for ‚detection purpose' vs. optimal locating robustness. An increased certainty for map matched position after map-matching interruption necessitates longer history
- The conclusion regarding the use of standard map matching components is that a FeedMAP client requires at least a different map-matching configuration if not even a different implementation. In general the tested approaches based on certain threshold (e.g., distance between map matched and GPS position) are good indicators for existing deviations, but it also became apparent that detected deviations like for example a missing road requires complex and sophisticated processing algorithms on the FMSC side.

One group of problems observed at the FeedMAP service centre side concerns the geometry of the received MDRs:

- geometry included double coordinate pairs, this can be filtered out, but
- when filtering, dimension of a linear location can change into a point location
- Start and end location contain too few points or are too short to properly cluster
- Start & end locations are too far away from deviation location
- Points are occasionally in reversed order

Several of the problems are the result of the map-matching findings as listed above and will only disappear if the map-matching is improved. Generally, this problem causes problems in the clustering and may be the reason for insufficient or even incorrect results when creating MDAs.

Also the deviation types WRONG_ROAD_GEOM & MISSING_ROAD are sometimes conflicting. Where to draw the border between them, i.e. when is a deviation a new road vs. a changed road? How to deal with different client configurations resulting in different thresholds? How to cluster reports of the same situation but different length (cause again by map-matching)? Inner-city situations are maybe often ambiguous. Finally, severe changes of a whole sub network are hardly fully covered by a single MDR.

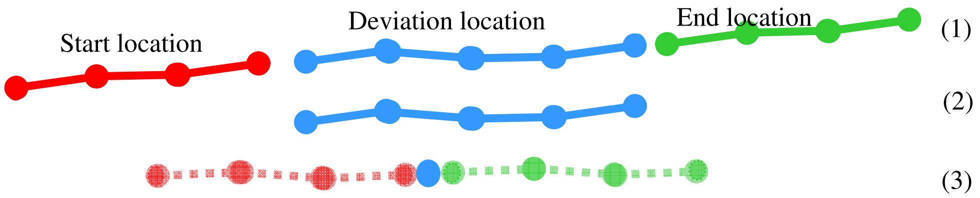

Figure 6: Possible Start, Deviation, End Locations in an MDR

Based on these experiences, the following rules (best practises) for the creation of MDRs are proposed:

- Points are ordered ascending according to the time they are driven by the client
- Start & end locations are required for WRONG_ROAD_GEOM & MISSING_ROAD
- Last position of start location is the last position before deviation has been detected
- First position of end location is the position, where deviation is not anymore observed
- Start, deviation & end locations shall not overlap
- Start & end locations are always of dimension LINE
- Deviation locations can be of dimension POINT or LINE
- Each location of dimension LINE has minimum of 3 points
- Start and & locations shall have a minimum length of 20 meters

Although the deviation detection in general is a sophisticated and complex process it can be stated that the automatic detection of map deviations and the combination with incremental map updating techniques are building one reasonable basis for in-vehicle ADAS applications to improve the up-to-dateness and quality of digital maps and finally the safety and comfort of driving.

REFERENCES

[1] Loewenau, J. et al (2002), "Preliminary map actualisation requirements", IST-2001-34141 ActMAP project. Brussels

[2] Thomas B. et al (2007), "ActMAP White Paper and Interfaces to the FeedMAP framework", D3.2, IST-2005-027325 FeedMAP project[2]. Brussels

[3] Loewenau, J. (2006) editor, D12.81 Interface Test Results, MAPS&ADAS project[3], Deliverable D12.81, Brussels

[4] Otto H.U, (2005) editor, "Data Requirements" D1, MAPS & ADAS Consortium and ADASIS Forum[4].

[5] Mezger K, (2005) editor, "ADASIS Functional Architecture". D2, MAPS & ADAS Consortium and ADASIS Forum

[6] Angenvoort J, (2005) editor, "Draft Specifications". Deliverable D3, MAPS & ADAS Consortium and ADASIS Forum

[7] Visintainer F. et al (2006), "Preliminary requirements and strategies for map feedback", D2.1, IST-2005-027325 FeedMAP project. Brussels

[2] http://www.ertico.com/en/subprojects/feedmap/papers__presentations/

[3] http://www.prevent-ip.org/en/prevent_subprojects/horizontal_activities/maps_-adas/maps_adas_papers_presentations.htm

[4] http://www.ertico.com/adasis