Map data as co-pilot: extended scope for automated driving
The commonality is the need for sensor data to replace human senses. An autonomous car requires information about its environment to act properly. Each type of sensor has different strengths and weaknesses. Cameras give the best overview, but the images must be processed in real-time to provide a proper environment model. Radar can detect objects very well and deliver a proper environment model, but is easily confused by metal objects like drain covers. Lidar, a laser-based system, seems to be a good compromise, but it is very expensive. The answer is to combine multiple sensors to get the most likely picture.

These sensors gain range with every new hardware generation, but they still have physical limitations. They cannot see much further than 200m, cannot look around corners or see through hills or other physical barriers. A car relying only on sensor data will therefore drive like a visitor from another city, as if it has no knowledge about the turnoff lanes, crossings, or curves. The car will drive by its sensors and will probably perform maneuvers very late when it finally recognizes the right path to take. Additional map data is the only way to avoid this. Such data can be thought of as being like an assistant rally driver. It improves driving by enabling smooth transitions between different conditions. For example: a predictive active cruise control would sharply brake the car when the sensors spot a speed limit. With the map information, the speed can be reduced in advance. Special conditions like sharp curves, dangerous inclines, or other obstacles are also known in advance. A detailed and up-to-date map is vital to provide this information. Such a map contains two kinds of information: Static and dynamic.

Figure 1: ADAS applications using map data
Map data as co-pilot: extended scope for automated driving

Dynamic content changes all the time. Such content includes data about accidents, hazard warnings, traffic jams, or weather conditions. It needs to be reported, processed, and distributed immediately. Otherwise, it would be simply outdated and lead to incorrect action. This information correlates to the traffic news on the radio.

Static content does not change that often. Examples are slopes, curvature, lane markings, or signposts. This data can be collected gradually, and the processing and distribution may take days or even weeks. This correlates with the information on paper maps.

**Static vs. dynamic content**

Dynamic content changes all the time. Such content includes data about accidents, hazard warnings, traffic jams, or weather conditions. It needs to be reported, processed, and distributed immediately. Otherwise, it would be simply outdated and lead to incorrect action. This information correlates to the traffic news on the radio.

Static content does not change that often. Examples are slopes, curvature, lane markings, or signposts. This data can be collected gradually, and the processing and distribution may take days or even weeks. This correlates with the information on paper maps.

**Uploading feedback**

A feedback loop must be established to provide such a map. The vehicle reports potential map information, which is collected and added to a map, and then distributed back to the vehicle. The feedback loop must be as fast as possible for dynamic content if it is to be useful. In contrast, static content is still valuable after a long time, so it could be buffered in the vehicle when an upload connection is unavailable.

The feedback loop is fed continuously with the objects or situations recognized by all participants in both cases. Timestamp and position data is added to the sent messages. Some kind of authentication is needed to avoid misuse of the upload portal. This authentication allows a vehicle to connect to the cloud services and prevents uploading of malicious messages by unknown senders. It usually allows the receiver of the message to determine the sender. A decision must be made whether to permanently keep the information about the sender to allow further evaluations, or to discard it during the processing chain to ensure privacy.

The uploaded information must be validated. Every record is checked for accuracy, and invalid records are removed immediately. For example: if a GPS signal is not available, the location data for latitude 0 and longitude 0 in the Atlantic Ocean might be sent. This error is detected easily when it is received. Besides the position data, timestamps should also be treated with care, since the local time of a vehicle is not necessarily correct. This might be due to wrong time settings or a low quality clock in older vehicles. Newer vehicles might be set to the wrong time zone during travel. Since the dynamic content must be sent immediately, the arrival time of the message on the server can be used. This can be trickier for buffered static messages when the sent time stamp cannot be trusted and the message might have been kept in the buffer for an undefined time. The static content fortunately is not as time critical as the dynamic content, so the precision of the time stamp is not as relevant.

**Compiling new maps**

The information received from all vehicles is collected in a large database. Dynamic data should be distributed as quickly as possible. It makes perfect sense, however, to collect additional information about the same item. For example: the position of spilled oil spotted by a vehicle is always inaccurate. This is because the sensors and algorithms in a car are optimized for correct recognition rather than for accurate location. Recognition of whether or not it is oil may be wrong in itself. This may be due to a broken sensor, bad weather conditions, or because the algorithm was stimulated to recognize something completely different as an oil puddle. The only solution is to aggregate multiple information about the same item. It is only a mathematical challenge to determine how many cars need to report
the same attribute “slippery road” at roughly the same place to determine the correct location and to trust the information.

Vanishing information is another problem affecting reliability. An example of this is a construction site along the road. It appears someday, lasts for maybe 2 weeks and then disappears again. Such a construction site will probably affect even static information like lane markings, lane geometry, speed limits, and so on. The updated values from the cars that recognize the construction site are counted as outliers, or are unable to influence the statistical mean value of the past few years. As a result, no map update or an insufficient one will be created. The age of the information must be considered in order to address this problem. More recent information must be rated with a higher probability than older information, even if it originates from a less reliable source. A similar problem applies when a sign is removed. The sign will simply not be spotted any more, and no car will send information about it. Therefore, a check must be performed to see if there is an item in the database for which no further acknowledgements arrive. Finally, these items must be removed from the database.

Figure 3: Combine track data from different customers

Distributing map updates

The collected and aggregated information leads to a much more detailed map. This map has more comprehensive and more accurate geometry information such as lanes, exact curvature or gradients, and more details along the road like zebra crossings, sign posts, or traffic lights. These maps are too large to be downloaded every day, even though mobile networks are getting faster with every generation and the costs for mobile data are shrinking constantly. The updates must therefore be sent out in small portions that only contain the new data. Version management has to take into consideration all the information already present in the vehicle and avoid sending this information repeatedly. Nevertheless, a general update will still contain irrelevant information because a car in one city will not need to know about a traffic jam in another city. The update thus boils down to a very specific map update package for each vehicle at its given position and expected route. The update package will only contain the latest information about the current track and the side roads that can be taken next.

Minimalistic update packages have the advantage that they can be installed as well as downloaded faster. Since the vehicle may receive update packages at any time, it must be able to handle them even while driving. Either the map database must be stopped, updated, and restarted, or a live update must be performed. If the database is stopped, the vehicle loses the map as an additional source of information and reverts to purely sensor-based driving. After the restart, all the relevant data must be read from the database again and synchronized with the other sensor data. A live update guarantees continuous usage of the map, but brings with it various database synchronization problems. The update algorithm may change information in the database while the automated driving algorithm is trying to read this data. Although this may work sometimes, there is always a possibility that both algorithms will try to access the same data at the same time. Broken data sets are transmitted when this happens. Just imagine driving along a curve while the curvature is being updated. A typical database approach would be to lock the affected data and thus prevent a read operation while data is updated. But this can lead to a deadlock when each algorithm waits for the other to unlock a needed record. Eventually there has to be a compromise between the database downtime and the availability of the most recent data.
The feedback loop is closed when the most recent data becomes available in the vehicle. The car can make use of all the data collected either from its previous drive or from other vehicles. This is actually the crucial point in the whole feedback loop. It can only work properly if it has enough participants. Static map data is still collected and updated by special cars. The only purpose of these cars is to collect data. They therefore drive accordingly, i.e. slowly, carefully, and over the same road several times. These cars are equipped with highly accurate sensors and transmit very precise information. While this works perfectly to create a basic road map for navigation, it is simply not enough for a map that is to serve as a basis for automated driving. Such a map must outline the slightly shifted lanes around a construction site as well as indicating the changed speed limit. A special car cannot collect data often enough to keep the map of a whole country up to date given that a construction site may only last less than a month. This means that every car and every driver must participate to maintain a highly accurate map in the same way that Wikipedia users do to keep the world’s largest encyclopedia updated.
Map data as co-pilot: extended scope for automated driving

Notes
About EB Automotive

Elektrobit (EB) is an award-winning and visionary global supplier of embedded and connected software solutions and services for the automotive industry. A leader in automotive software with over 25 years serving the industry, EB’s software powers over 70 million vehicles and offers flexible, innovative solutions for connected car infrastructure, human machine interface (HMI) technologies, navigation, driver assistance, electronic control units (ECUs), and software engineering services. EB is a wholly owned subsidiary of Continental.

Map data as co-pilot: extended scope for automated driving

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
sales.automotive@elektrobit.com
elektrobit.com