Paper Code: F02I131

MAP-AIDED POSITIONING SYSTEM

Forssell, Urban[1]
Hall, Peter[1]
Ahlqvist, Stefan[1]
Gustafsson, Fredrik[2]
[1]NIRA Dynamics AB, Sweden; [2]Linköpings universitet, Sweden

Keywords
Positioning; Navigation; Geographical Information Systems; Global Positioning System; Sensor Fusion

Abstract
This paper describes the vehicle positioning system MAP (Map-Aided Positioning) developed by NIRA Dynamics AB. MAP uses sensor fusion to combine relative position information from the wheel speed sensors with digital map information, and is capable of computing an accurate estimate of a vehicle's absolute position without support from GPS or other external positioning service. MAP can also be combined with GPS, in which case a very robust and accurate positioning system is obtained. MAP is available as a software module suitable for integration in a PDA or similar hardware platform.

1. Introduction
This paper describes NIRA Dynamics AB's vehicle positioning system MAP; MAP for Map-Aided Positioning. MAP combines information from available position sensors (relative and absolute) and a digital map in a statistically optimal way and produces accurate positioning information that e.g. can be used for navigation purposes.

Provided with a rough initial guess of the true position, MAP is able to calculate an estimate of the true, absolute position using only relative position information from the wheel speed sensors and digital map information. MAP can therefore run as an autonomous positioning system without GPS support and still yield similar or better functionality (as GPS). MAP can also be combined with standard GPS-based solutions to improve the availability and positioning accuracy.

Next we will review the technology area and give some background information to why MAP is interesting. However, before continuing a brief note on the nomenclature used: In our view, *positioning* is about determining where you are relative to some fixed reference, while *navigation* is about finding the optimal (e.g. nearest or fastest) path from a position A to a position B.

1. 1 Building Blocks of a Modern Vehicle Navigation System
Most modern vehicle navigation systems can be described using the following three-layer model, c.f. Figure 1.

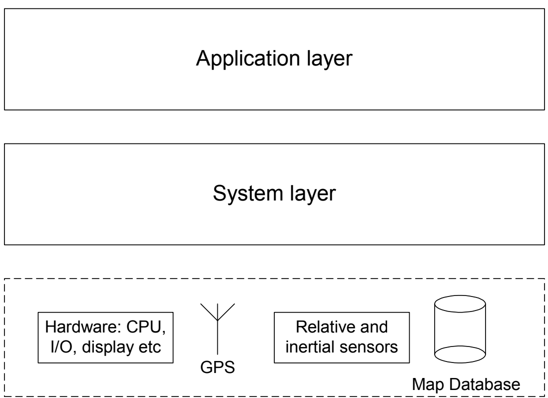

Figure 1. Three-layer model illustrating the structure of standard navigation systems.

Starting from the top we have an application layer featuring for example

1. Navigation functionality such as route planning and route guidance.
2. Advanced HMI features such as voice control and guidance and different graphical presentation features.
3. Other services and functionality including SOS alarm functions, vehicle tracking functions (anti-theft), and various so called location based services functions like directed advertisement and travel services including customized directions to hotels, points of interest, gas stations, repair shops etc.

The system layer consists of low-level applications such as

1. Positioning functions that calculate an estimate of the true position using the available sensor information from GPS and typically also relative and/or inertial sensors.
2. Functions for efficient map database handling.
3. HMI functionality such as menu systems and presentation of position information (including map-matching functions).

The third layer consists of the physical hardware and sensors:

1. Hardware for interfacing various sensors, for running various software applications, and for displaying the navigation information.
2. GPS antenna and receiver, which gives position information.
3. Relative position sensors (e.g. wheel speed and steering wheel sensors) and inertial sensors (e.g. yaw rate gyros and accelerometers).
4. A map database, often stored in a read-only-memory such as CD ROM or DVD ROM

In summary we thus see that almost all vehicle navigation systems rely on GPS as the main position information source and employ maps stored on a CD or DVD ROM. Most systems also use relative position information from e.g. inertial sensors to correct the GPS information.

1.2 Disadvantages With Today's Navigation Systems

The navigation systems of today which rely solely on GPS are not ideal, especially not in urban areas with high buildings, multi-level highways, and tunnels where satellite coverage may be poor and/or where the GPS signals are corrupted by multi-path and fading. To enhance performance under these circumstances, the systems therefore frequently feature dead reckoning functionality, i.e. the systems use relative positioning information obtained from an inertial navigation system (typically based on wheel speed information and yaw rate information) to predict future positions. However, despite these efforts to improve the positioning accuracy of the GPS based systems it is sometimes surprisingly poor. And GPS based systems typically show the worst performance when they are needed the most: in dense urban areas.

A further feature of many navigation systems is some kind of map matching to enhance the precision e.g. when turning around sharp corners and following characteristic road patterns. In principle, map matching is about moving the symbol indicating the vehicle's position on the map from the position calculated by the positioning module to the nearest road. If done correctly this can reduce the positioning error and also improve the user interface. A problem in many systems, though, is that the positioning and map-matching functions are not integrated and do not support each other, which sometimes results in sub-optimal performance. As an example consider a case where the GPS signal is lost and the system relies on dead reckoning information only for a period of time. Then, due to errors accumulated in the dead reckoning system, the map-matching algorithm will have an almost futile job trying to "catch up" with the true position. This type of (poor) behavior is very common in today's navigation systems.

Another drawback with GPS based systems is the relatively high total system cost, including antenna, receiver, cables, connectors, installation etc. Due to the costs for the antenna, cables, connectors, and installation the overall cost will never drop to levels near zero. Since cost is an issue for all system integrators, the suppliers of these kinds of systems have to build in more value in their product. And the improved positioning accuracy and higher availability provided by MAP is one way to do this.

2. General Description of MAP

2.1 Basic Features of MAP

MAP is a positioning system for vehicles designed to give very accurate position information, which can be used in a navigation system of the type described above. Through the use of advanced nonlinear filtering techniques MAP fuses information from a DRS (Dead Reckoning System) and a digital map. Given an initial (rough) estimate of the true position MAP can autonomously calculate the true, absolute position of the vehicle without support from external position information sources such as GPS. Among other things this means that MAP does not have any problems in areas where the GPS accuracy is poor e.g. near high-rises and in tunnels. MAP also allows new, cheaper navigation systems to be built as the hardware requirements are reduced (no GPS antenna, no cables, no GPS receiver etc).

We are currently focusing on two main applications of the MAP technology:

1. A cheap, flexible navigation system to be used in cities.
2. An enhanced GPS navigation system.

In the first case one idea is to combine MAP with GSM positioning for initialization and support and digital maps downloaded to a read-write memory such as a "Smart Card" or similar. Here the working principle is basically as follows. When starting the system after a memory loss, so called "cold start", the system connects to a GSM positioning service offered by some mobile operator. This gives an initial estimate of the through position with an accuracy of 300-3000m nominally (the existing GSM positioning service delivers position information in form of a sector area in a mobile telephony cell). Using this information the MAP algorithm is initialized and as the vehicle moves and relative position information is calculated by the dead reckoning system, MAP can determine the true, absolute position by fusing this relative position information with the map information using the nonlinear filtering algorithm.

Our idea is to implement the above system on a PDA, which also can be used for other purposes outside the vehicle (e.g. navigating to the nearest restaurant using other GSM positioning services). It is however possible to think of a number of other ways to implement the system, one being to mimic the design of today's GPS based systems. For instance, instead of using maps downloaded using a wireless link one can use maps stored on permanent media such as CD or DVD ROM.

Instead of using GSM positioning for initialization and support it is possible to use manual start where the user inputs an area in which MAP should start the search. It is also possible to use other radio-based services such as RDS for this purpose with minor modifications.

In the second case, the idea is to utilize to best features of the GPS and MAP technologies to construct a really high-performance system that meets the markets requirements on positioning accuracy and reliability in the top-of-the-line navigation systems. In this case the MAP algorithm will fuse information from the GPS, the dead reckoning function, and from the digital map in a statistically optimal way. A nice feature of this system is that loss of GPS information is handled automatically and to calculate the position estimate the best available information is always used. This enhanced GPS system will therefore not have problems in dense urban areas, in tunnels etc. And since it combines information from several different sources in an optimal way you will always get the best possible position estimate.

Our product in this case is first and foremost a software module implementing the MAP algorithm, which takes inputs from the GPS, the dead reckoning system, and the map database and computes an accurate position estimate. The idea is that this software should be integrated in a navigation system and thus complemented with routines for graphical presentation, map matching, route planning etc.

2.2 Background

The algorithm used in MAP is a recursive Bayesian estimation algorithm. Such algorithms have become increasingly popular in academia [4,5] and some application areas, mainly military [6]. A system for terrain navigation using Bayesian estimation is known from [1]. The main result in [1] is a navigation system using terrain altitude information stored in a GIS (Geographical Information System) combined with altitude measurements from radar.

MAP combines advanced nonlinear filtering techniques such as the ones discussed in [1] with a novel use of digital road map information, thus obtaining a positioning system for vehicles with very high position accuracy without the drawbacks of today's (satellite-based) positioning/navigation systems.

2.3 System Design

This section is devoted to a general description of the MAP system. As shown in Figure 2, MAP calculates an accurate position estimate using all possible sensors and services, e.g. GPS and/or GSM positioning or

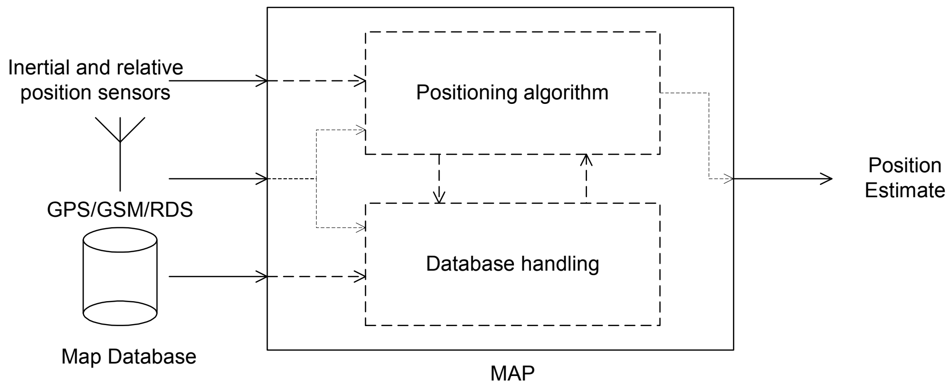

Figure 2. Block diagram description of MAP.

similar, relative sensors, inertial sensors, and digital map database information. This position estimate can be used by the navigation routines as described above.

The internal structure of the system consists of two main functions/modules: the positioning algorithm and the routines for the map database handling. The former will be described next, the latter further down. Here we can note that these two modules, although separated here for pedagogical reasons, are very closely coupled and interact in each iteration of the recursive MAP algorithm.

2.4 Conceptual Description of the Positioning Algorithm

The underlying idea in MAP springs from a statistical viewpoint, where the knowledge about the vehicles position at each time instant is completely summarized by the conditional probability density function (PDF). The PDF evolves with time (using dynamical models) and information contained in observations made on the system. Let us consider a simple example: Assume that we know that the vehicle we want to position has been traveling along a straight road for quite some time. The PDF will then tell us that the vehicle, most likely, is located somewhere on that road. The PDF is illustrated in Figure 3. When the vehicle later makes a right-hand turn, information about the movement and the spatial configuration of the map is fused into the PDF (Figure 4). The result is that the mass of the PDF is more concentrated to the true position, i.e. the uncertainty about where the vehicle is located has been reduced.

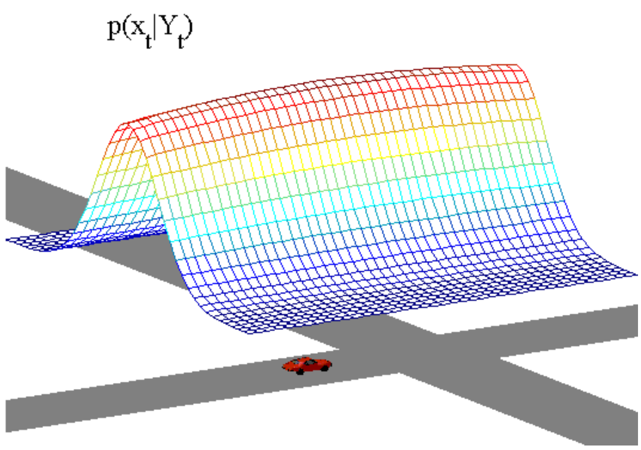

Figure 3. The PDF before the turn is distributed along the straight ahead road with no well defined peak.

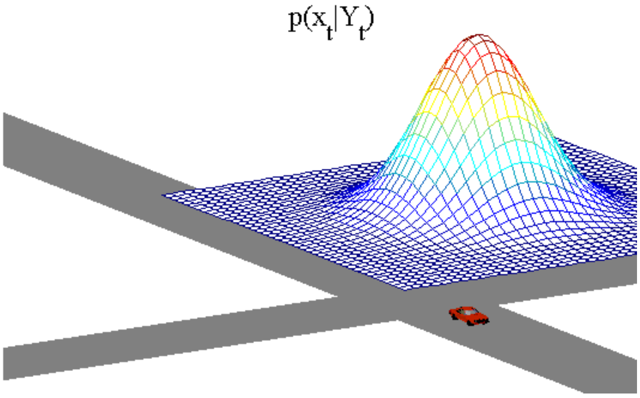

Figure 4. After the right-hand turn the PDF is concentrated with a significant peak at the true position.

The MAP approach is to recursively estimate the conditional PDF using information from vehicle-mounted sensors measuring relative movements (e.g. wheel-speed sensors) and a digital road map. This type of signal processing problem is often referred to as *sensor fusion* and is usually tackled by statistical methods. Here, the recursive estimation of the conditional probability for the vehicle's position will be formulated using a Bayesian framework, aiming at a non-linear filtering algorithm.

Sensors, which provide relative position information, e.g. inertial sensors (accelerometers, yaw rate gyros etc.) and wheel speed sensors, will in the sequel be referred to as a dead reckoning system (DRS).

3. The Positioning Algorithm

3.1 Idea

Now we will describe the positioning algorithm presented briefly above using a mathematical approach. To illustrate the basic ideas, the following simple state space model can be used:

$$x_{t+1} = x_t + u_t + w_t \tag{1}$$

$$y_t = h(x_t) + e_t \tag{2}$$

The state x_t represents the vehicle's position on the map. At each iteration, this is updated using u_t, which is the relative movement obtained from the DRS. Drift in the DRS is modeled by additive i.i.d. (independent identically distributed) noise, w_t, with probability distribution $p_{w_t}(\cdot)$. The measurement, y_t, consists of the nonlinear function $h(\cdot)$, evaluated at the current position, plus i.i.d. additive measurement noise, e_t, with probability distribution $p_{e_t}(\cdot)$. e_t is assumed independent of w_t. It should be noted that $h(\cdot)$ is used to introduce map information in the model. Since it is assumed that the vehicle is driving on a road in the road network, $h(\cdot)$ can e.g. represent the shortest distance to the nearest road at time t, while y_t represents a fictitious measurement of this distance at time t. This measurement of course always equals zero, according to the assumption made above.

Let $f_{x_t|Y_t}(x)$ denote the conditional probability density function (PDF) for the state x_t, given the measurements up to the time t. In a Bayesian framework the PDF can be recursively updated in two steps [2]:

Measurement update:

$$f_{x_t|Y_t}(x) = \frac{1}{c} f_{e_t}(y_t - h(x)) f_{x_t|Y_{t-1}}(x) \tag{3}$$

Time update:

$$f_{x_{t+1}|Y_t}(x) = \int_{\Re^2} f_{x_t|Y_t}(\chi) f_{w_t}(x - \chi - u_t) d\chi \tag{4}$$

However, due to the non-linear nature of the estimation problem, these expressions cannot be evaluated analytically. Therefore, some kind of discretization of the state space is necessary. Below we will outline two possible implementations: one *point mass filter* (PMF) and one sequential Monte Carlo filter. The latter is often also referred to as a *particle filter* (PF). In both cases the algorithm has a recursive structure as illustrated in Figure 5.

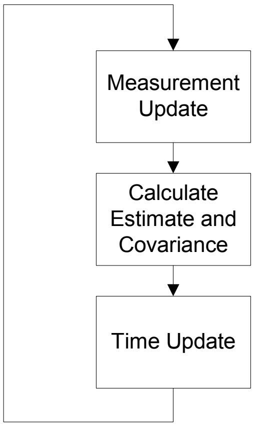

Figure 5. The recursive structure of the nonlinear filter

The algorithm consists of two basic steps: A time update step, in which the solution is propagated according to the state transition equation (1), and a measurement update step, in which new information is fused into the solution according to (2). The solution to the estimation problem is completely specified by the probability density function (PDF). Our primary interest, however, is to estimate the position (the state x_t). Thus a calculation of such an estimate along with the associated error covariance forms the third step in the iteration.

3.2 Implementation

Here we will describe two ways of implementing the MAP algorithm, first using a PMF and second using a PF. The PMF will be described only briefly, while focus will be on the PF.

3.2.1 Point Mass Filter (PMF)

With the PMF one discretizes the state space (the map) using a grid, see Figure 6.

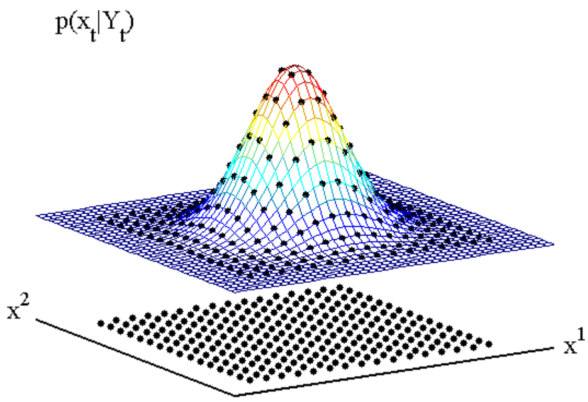

Figure 6. The PDF discretized by a homogenous grid (deterministic sampling).

The Bayesian solution, obtained in Section 3.1, is then applied to each grid point, i.e. the PDF is represented by a set of point-masses, or weights.

Measurement update

The measurement update re-computes the weights using the information in the new measurements according to (3). The normalizing constant is obtained by calculating the sum of all new weights.

Time update

The time update consists of a translation of the grid points plus convolution of the probability density function with the uncertainty in the relative movement (obtained from dynamic sensor models). Mathematically this is a result of numerical integration of (4).

More details about the PMF can be found in [1].

3.2.2 Particle Filter (PF)

In both the PMF and the PF the Bayesian problem is tackled using quantization of the state space. In the PF case this means that the PDF is represented by a number of i.i.d. (independent identically distributed) samples, referred to as *particles* (see e.g. [1],[2]).

The sampling technique used in the PF is called Monte Carlo (MC) sampling. The main advantage of this method is that the samples of the PDF are automatically chosen in parts of the state space that are important for the integration result, i.e. more samples are drawn from regions containing most of the PDF mass. This is illustrated in Figure 7.

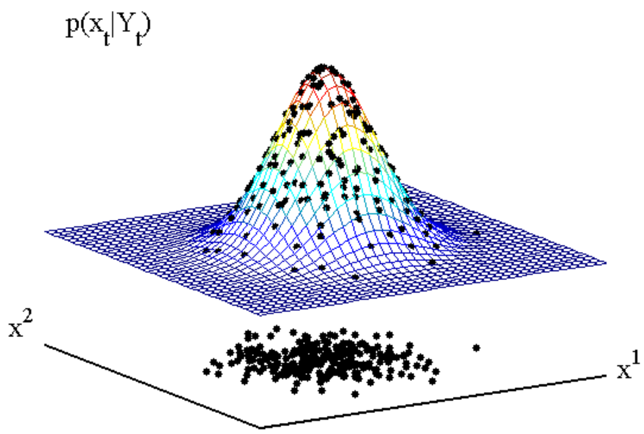

Figure 7. The PDF sampled by Monte Carlo simulation.

The basic algorithm consists of two steps: A time update step, in which the solution is propagated according to the state transition equation, and a measurement update step, in which new information is fused into the solution.

Time update

In the time update the particles (which can be considered as candidates to the true position) are translated using relative displacement, measured by the DRS, and realizations of the process noise.

Measurement update

The measurement update calculates a new, so called *importance weight* for each particle based on the outcome of the measurement equations, when applied to the current particle. The weight can be considered as a sampled value of the posterior PDF. In order to keep the calculations sound in a probabilistic sense, the weights need to be normalized.

Resampling (Bootstrap)

To make sure that the particles remain i.i.d. samples from the PDF, resampling has to be performed on a regular basis. This is done by drawing samples with replacement until a certain number of new particles are obtained. The probability of resampling a specific particle at each draw is equal to its weight.

Calculation of estimates and confidence parameters

The (approximate) solution to the estimation problem is completely specified by the particle swarm. However, the primary interest is usually to obtain various estimates and confidence parameters. There are mainly two ways of calculating estimates, through *expectation* or through *maximum a posteriori* (m.a.p.). Expectation is easily performed by calculating a weighted sum over all particles. The m.a.p. estimate is also very straightforward – just pick the particle with the largest weight.

The *covariance matrix* is obtained by calculating the second moment, if the estimate was obtained by expectation. The same procedure applied to a m.a.p. estimate yields a *correlation matrix*, since the m.a.p. estimate is not necessarily unbiased.

For a more detailed presentation of the PF see [1],[2].

4. Example

In order to illustrate the PF functionality we will look at an example where wheel speed measurements were collected during a test drive in a suburban area. Four snapshots showing the propagating particle cloud for this particular scenario are depicted in Figure 8. Initially the particles are randomly distributed along the streets in the area. When the movements of the vehicle (i.e. the wheel speed signals) are inserted into the filter the particles start revealing the actual position of the vehicle. The speed of convergence of the filter is of course dependent on the information contents in the measurements and the street network. If the vehicle trajectory includes several heading angle changes (turns) this will improve the convergence, with respect to the spatial configuration of the map.

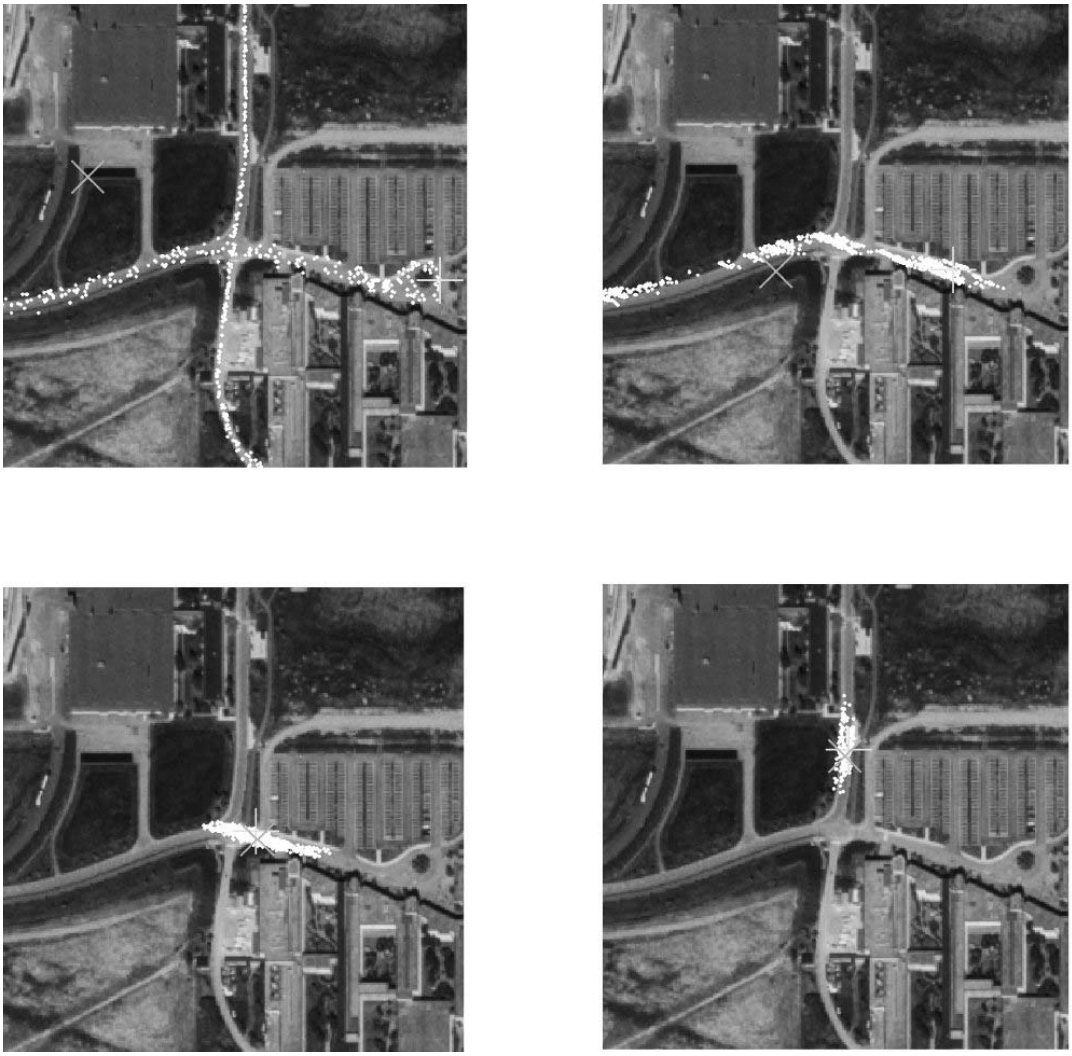

Figure 8. Four snapshots showing the propagation of the particle cloud from an authentic test drive. At first the particles are distributed on the major roads. Then, as the vehicle moves, the particles start tracking the true position.

5. Map Database Handling

Since the map database is one of the most fundamental components in map-aided positioning, the map data handling is a vital part of the system. These routines should be able to extract the information needed by the positioning algorithm from the database as quickly as possible. The measurement equations in the somewhat simplified model presented in Section 3 use a single quantity provided by the map: The shortest distance from an arbitrary point to the road network. This can be obtained through a limiting search operation and minimization. Figure 9 illustrates how map information is retrieved in the measurement update step.

Apart from performing the data access operations discussed above the map handling routines should handle the update of actual map database. It is obvious that the size of the map that is stored in the internal memory and operated on by the algorithm is limited by several factors. Therefore the system should be able to tell when the current map needs a refresh and perform such an update without disturbing the real-time performance.

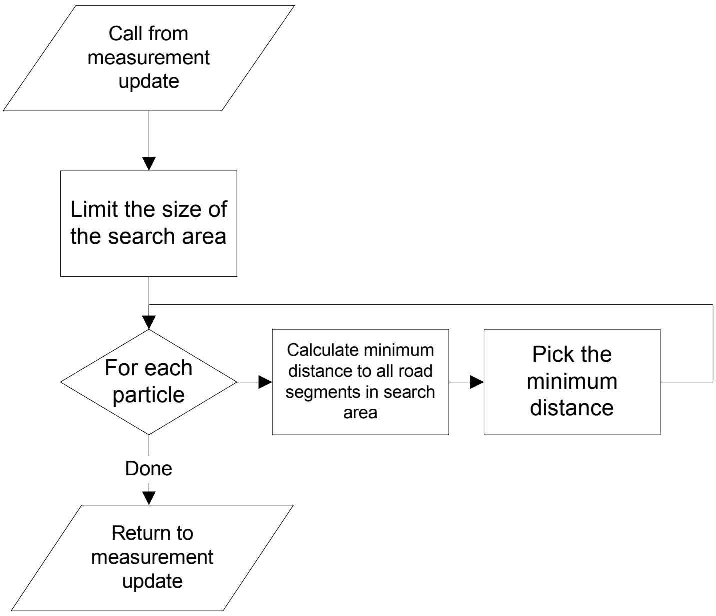

Figure 9. The map data update

7. Extensions

7.1 Heading Angle Estimation

The model studied above does not include estimation of the heading angle of the vehicle. However, in order to make the MAP system autonomous, that has to be handled by the algorithm. Details how the model should be extended to solve this can be found in [2].

7.2 Sensor Error Modeling

The sensors used in the MAP system are associated with several kinds of errors. Here, these have been considered as white noise processes. This is, however, not an appropriate assumption, and the error models might need to be refined in order to achieve decent performance. We will not present any details here, but better error models can be inserted into the filter structure discussed so far, or, since errors often are additive, it is also possible to estimate these parameters using conventional linear filters, such as the Kalman Filter.

8. Summary and Conclusions

The MAP system implements an autonomous positioning technique, utilizing existing vehicle mounted sensors and digital road maps. It also elegantly solves the map-matching problem by adopting sensor fusion ideas. The MAP system can be integrated in conventional satellite-based systems in order to improve the accuracy and reliability.

References

[1] N. Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD Thesis. Linköping University. May 1999.

[2] P. Hall. *A Bayesian Approach to Map-Aided Vehicle Positioning.* Master Thesis. Linköping University. January 2001.

[3] N. Gordon, D. Salmond, A. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation.* IEE Proceedings on Radar and Signal Processing, 140:107-113, 1993.

[4] P. Fearnhead. *Sequential Monte Carlo Methods in filter theory.* PhD thesis. University of Oxford. 1998.

[5] M. Pitt and N. Shephard. *Filtering via simulation: Auxiliary particle filter.* Journal of the American Statistical Association, 94(446):590-599.

[6] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P.J. Nordlund. *Particle filters for positioning, navigation and tracking.* IEEE Transactions on Signal Processing. February 2002.