Journal of VLSI Signal Processing 45, 111–122, 2006

DOI: 10.1007/s11265-006-9775-4

Fusion of Map and Sensor Data in a Modern Car Navigation System

DRAGAN OBRADOVIC AND HENNING LENZ
Siemens AG, Corporate Technology, Information and Communications, CT IC4, Otto-Hahn-Ring 6, 81730 Munich, Germany

MARKUS SCHUPFNER
SiemensVDO, Interior and Infotainment, Infotainment Solutions, Im Gewerbepark C 27, 93059 Regensburg, Germany

Received: 9 March 2005; Revised: 20 June 2005; Accepted: 29 June 2005

Abstract. The main tasks of car navigation systems are positioning, routing, and guidance. This paper describes a novel, two-step approach to vehicle positioning founded on the appropriate combination of the in-car sensors, GPS signals, and a digital map. The first step is based on the application of a Kalman filter, which optimally updates the model of car movement based on the in-car odometer and gyroscope measurements, and the GPS signal. The second step further improves the position estimate by dynamically comparing the continuous vehicle trajectory obtained in the first step with the candidate trajectories on a digital map. This is in contrast with standard applications of the digital map where the current position estimate is simply projected on the digital map at every sampling instant.

Keywords: GPS signals, vehicle positioning, Kalman filter, pattern matching, map matching, dead reckoning

1. Introduction

The development of the modern car navigation systems was enabled by the ongoing improvement of electronic devices as well as by the availability of new position information sources such as GPS and digital maps. In general, there are two concepts of navigation systems. The first type is given by centralized systems, where there is a continuous two-way communication with the vehicles requesting the navigation service. The information from the on-board vehicle sensors is transmitted to the navigation center, which estimates the car position and transmits the guidance commands back to the driver. On the contrary, autonomous navigation systems process all the information on-board and calculate optimal route and the necessary guidance commands without participation of an external server. Due to the lower costs, the autonomous navigation systems have become the standard in car industry. This paper discusses novel methods for vehicle positioning in autonomous navigation system.

The autonomous car navigation systems have to perform three distinctive tasks [1]: positioning, routing, and navigation. Positioning implies the continuous determination of the vehicle position based on the available sensory and the digital map information. The car build-in sensors used for this purpose are the odometer and the gyroscope. The odometer provides information about the traveled distance while the gyroscope calculates the heading (orientation) change with a given time interval. Both of these sensors are, due to the financial reasons, of limited accuracy and subject to drift. Additional information sources used for car positioning are the GPS signals and the digital map of the roads.

The Global Positioning System (GPS) [2] is a system of 24 satellites with synchronized atomic clocks, which continuously transmit the time and the needed satellite identification information. The GPS receiver that is available within the vehicle detects the number of satellites whose signals are strong enough to be processed. The receiver delivers the vehicle position and the information about its velocity as two separate outputs. Until May 2000, the GPS accuracy was intentionally deteriorated in order to make the civilian use of the system less accurate than its military applications. This property is called Selective Availability and, as stated above, is currently turned off. Nevertheless, the decision whether to reactivate is subject to a yearly review by the US government. In addition, the coverage (the number of visible satellites) of the GPS system is not evenly distributed, and in some areas (urban, mountainous regions) the number of visible satellites is further reduced.

Digital maps contain information about the road network including the road properties (highway, one way street, etc.). The road representation within the digital map is piecewise linear. Nowadays the digital maps used in standard navigation systems provide additional information about the type and location of different services of interest to the car passengers such as hotels, shopping mall locations, etc.

The information from the on-board sensors, the GPS and the digital map has to be combined appropriately in order to maximize the accuracy of the estimated vehicle position and heading. This is a standard sensor fusion problem where different sensors are of different accuracy and whose properties are also changing over time. Typically, the sensor fusion is performed stepwise. In the first step, the odometer and gyroscope information is combined within the process called dead reckoning [1]. The result of the dead reckoning process is the estimated vehicle trajectory. In a further step, the dead reckoning position is projected on the digital map. This step is called map matching. If the dead reckoning position is in between several roads, several projections will be made. Each projection is then seen as a possible alternative whose likelihood is estimated over time based on the GPS information and the trajectory time evolution. If the discrepancy between the matched position and the GPS signal is too high, the so-called "GPS reset" is performed, i.e., the position is assigned to the current GPS estimate. Not only do the sensor fusion algorithms have to deliver an accurate position estimate, but they also have to accomplish this task under very stringent computational and memory constraints. Hence, the final solution is always a tradeoff between the optimality of the mathematical approach and the implementation limitations.

Routing [3] is calculation of the route between the location A and location B that is optimal with respect to the selected criterion. Once the optimal route and the current position are known, the guidance algorithm issues recommendations to the driver so that the vehicle remains on the selected route.

2. Dead Reckoning Improvement by Kalman Filter

If the initial vehicle location is known, the odometer and gyroscope measurements can be used to reconstruct the traveled trajectory. The process of incremental integration of the vehicle trajectory relative to a known location is called dead reckoning [1] and is illustrated in Fig. 1. The standard dead reckoning equations are as follows:

$$
\begin{aligned}
x_i &= x_0 + \sum_{j=1}^{i} l_j \cos\left(\varphi_j\right) \\
y_i &= y_0 + \sum_{j=1}^{i} l_j \sin\left(\varphi_j\right) \\
\varphi_i &= \varphi_0 + \sum_{j=1}^{i} \Delta\varphi_j
\end{aligned}
\tag{1}
$$

where x and y are the planar coordinates, while l and $\Delta\varphi$ represent the traveled distance and the change in orientation. The Eq. (1) do not adequately describe movement on a circle and, hence, this paper introduces corrected dead reckoning estimate [4] as in Eq. (2).

$$
\begin{aligned}
x_i &= x_0 + \sum_{j=1}^{i} \frac{\sin\left(\Delta\varphi_j/2\right)}{\Delta\varphi_j/2} l_j \cos\left(\varphi_{j-1} + \Delta\varphi_j/2\right) \\
y_i &= y_0 + \sum_{j=1}^{i} \frac{\sin\left(\Delta\varphi_j/2\right)}{\Delta\varphi_j/2} l_j \sin\left(\varphi_{j-1} + \Delta\varphi_j/2\right) \\
\varphi_i &= \varphi_0 + \sum_{j=1}^{i} \Delta\varphi_j
\end{aligned}
\tag{2}
$$

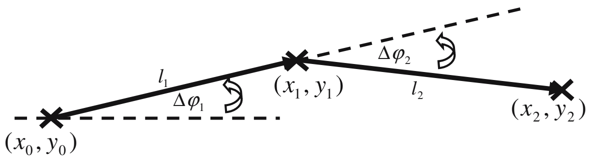

Figure 1. Dead reckoning as vector addition.

According to the fact that dead reckoning is a discrete time process, the orientation at the sampling instants and the orientation between two samples differ: The orientation of the traveling car in between two sample points is given by the mean of the two orientations at the two sample instants. Therefore, the averaged orientation has to be used for dead reckoning. For practical reasons this averaged orientation can be determined by the orientation of a first sample instant plus half of the orientation change within the following time step. The differences between Eqs. (1) and (2) become clearly visible, when a car is driving on a circle as it happens in parking garages and as is illustrated in Fig. 2.

In addition to using an appropriate heading during dead reckoning, the traveled length has also to be adjusted to avoid inaccuracies. The car physically drives on the circle, such that the length of a circle section is measured by the odometer. On the other hand, the discrete dead reckoning uses straight lines, which create a shortcut between two sample instants. Accordingly, the length measured by the odometer has to be corrected by the term $\sin(\Delta\varphi/2)/(\Delta\varphi/2)$ as applied in Eq. (2), where $\lim\limits_{\Delta\varphi\to 0}[(\sin(\Delta\varphi/2))/(\Delta\varphi/2)] = 1$.

As it can be seen, dead reckoning is the simplest way of determining the vehicle position. Unfortunately, the accuracy of this method is very low since the incremental information is noisy and the error accumulates over the time. The standard way of dealing with the inaccuracy of the dead reckoning is to project the so determined vehicle position on the roads depicted in the digital map and use GPS to evaluate the certainty of different projections. The approach described in this paper is different. The novelty of the herein presented approach is that the dead reckoning trajectory is corrected by the GPS measurement every time when the "quality" of the GPS reception is acceptable. Hence, the GPS information is combined with the odometer and gyroscope measurements before the digital map is used. This sensor fusion is achieved by a suitable Kalman filter implementation.

There are several possible ways in setting up the sensor fusion as a Kalman Filter problem [1, 5, 7, 8, 11, 12]. The "complete" Kalman filter representation would use the dead reckoning equations as state equations and the GPS as measurements. In this case the states would be the actual positions. Although this is a common sense approach, it has the following unfavorable characteristics:

- The state equations are non-linear. The non-linearity stems from the presence of the sine and

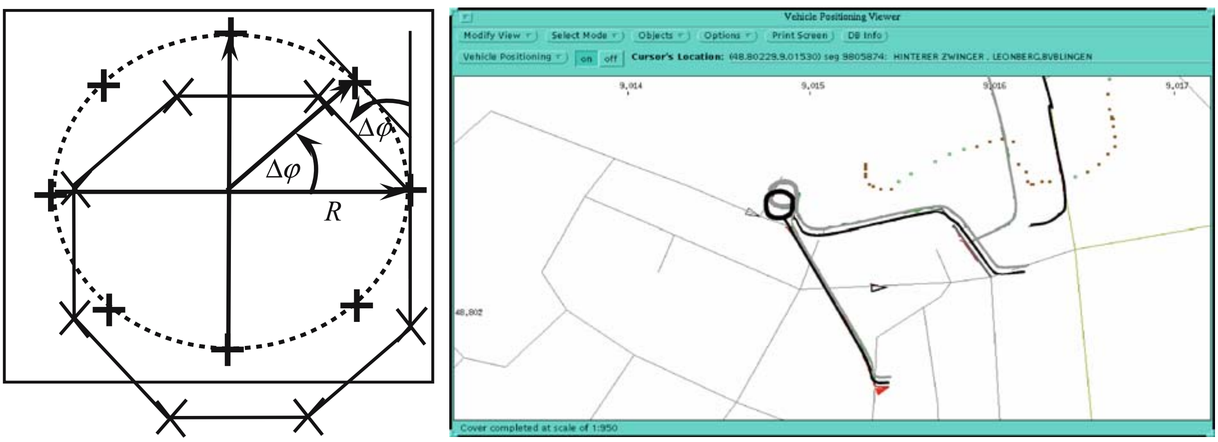

Figure 2. The *plus signs* (+) represent true positions on a circle. *Crosses* (×) represent the dead reckoning positions based on Eq. (1). Using the dead reckoning positions based on Eq. (2), the true and dead reckoning positions coincide. An example of a driving situation in a parking garage is shown in the *right half* (*black*: true position; *grey*: estimated position using old DR).

cosine function as well as from the transformation needed to adjust the plane assumption of the dead reckoning to the curvature of the Earth.

- The additive Gaussian noises in the two state equations are not uncorrelated. The actual independent measurements are the incremental traveled length and the orientation. Since both longitude and latitude positions are functions of the latter two variables, they are correlated.

Consequently, the decision was made to use the "decentralized" Kalman filter [1] implementation, where the odometer and gyroscope data will be enhanced with the available GPS measurements. A decentralized Kalman filter, in contrast with the standard Kalman filter implementation, is applied only to the part of the dynamics of the positioning system. The complete system dynamics in this case would have had the actual position coordinates of the vehicle as its states. In the current implementation, the Kalman filter is applied only to the angle and traveled distance measurements, which are used after the estimation to compute the vehicle position. The odometer and gyroscope drift is modeled with extra parameters P^{gyro} and P^{odo}. In the state–space representation of the Kalman filter (3) these two parameters appear as two auxiliary states driven by additive noise. Due to the fact that the gyroscope drift is more severe than the odometer drift, and due to the computation and memory limitation of the system, the following Kalman filter is implemented [9]:

$$\left.\begin{cases} \varphi_{i+1} = \varphi_i + (\Delta\varphi_i)_{gyro} + P_i^{gyro} + \vartheta_i^1 \\ P_{i+1}^{gyro} = P_i^{gyro} + \vartheta_i^2 \\ P_{i+1}^{odo} = P_i^{odo} + \vartheta_i^3 \end{cases}\right\}\text{state equations}$$

$$\left.\begin{cases} y_i^{(1)} = \varphi_i + \eta_i^{(1)} \\ y_i^{(2)} = (l_i)_{odo} + P_i^{odo} + \eta_i^{(2)} \end{cases}\right\}\text{measurement equations} \tag{3}$$

The heading and traveled distance measurements $y^{(1)}$ and $y^{(2)}$ are obtained from the received GPS information. The GPS receiver delivers independently the position estimate as well as the velocity and the heading. The Kalman filter presented herein relies only on the GPS velocity and heading information, while the GPS position estimate is not explicitly used. The limitation of the usage of the GPS information as a "teacher" comes from the fact that the quality of the velocity and heading (as well as the position) estimate vary based on the number of visible satellites and their positions. Consequently, a novel proprietary algorithm of Siemens AG [13] is applied that changes the covariance matrices of the additive measurement noise $\eta^{(1)}$ and $\eta^{(2)}$ based on the GPS signal quality. The worse the GPS reception is, the more noisy are the measurements and, as a result, the smaller is the correction of the odometer and gyroscope information. In the extreme case when the GPS information is completely unreliable, no correction of the gyroscope and odometer information is performed. Due to the changes of the GPS quality, the Kalman filter is implemented with a suitable forgetting factor which forces the statistics to weight the past information less and less over time [11].

The resulting corrected heading and traveled distance estimates are further used to determine the vehicle position according to the following formula:

$$\begin{pmatrix} \text{Long}_{k+1} \\ \text{Lat}_{k+1} \end{pmatrix} = \begin{pmatrix} \text{Long}_k \\ \text{Lat}_k \end{pmatrix} + \\ +\psi\left\{ \frac{\sin(\Delta\varphi_k/2)\left(1+P_k^{odo}\right)(l_k)_{odo}}{\Delta\varphi_k/2} \begin{pmatrix} \cos(\varphi_k + \Delta\varphi_k/2) \\ \sin(\varphi_k + \Delta\varphi_k/2) \end{pmatrix} \right\} \tag{4}$$

where ψ stands for the transformation from the planar to the WGS84 coordinate system [1], which takes into account the geometry of the Earth surface.

A typical dead reckoning improvement after the Kalman filter implementation is depicted in Fig. 3. The dashed line shows the dead reckoning path without the Kalman filter while the dotted line shows the Kalman filter produced dead reckoning. The true driven path is depicted by a solid line. As seen, the Kalman filter corrected dead reckoning improves significantly the accuracy of the original dead reckoning since the GPS information is used.

The computational burden of the Kalman filter had to be optimized such that its complexity was acceptable for use on a 68000 processor from Motorola and using a PSOS real-time operating system. Hereby, the fact that the matrices in the Kalman filter (see Eq. (3)) are sparse and block diagonal has been used in the C-code implementation. Instead of complete matrix calculations only the parts of the equations with non-vanishing

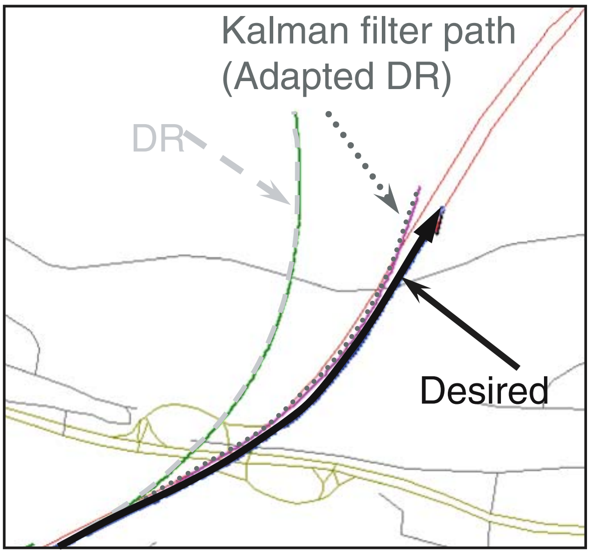

Figure 3. Dead reckoning paths without (*dashed*, *light grey*) and with (*dotted*, *dark grey*) Kalman filter implementation. The *black solid line* depicts the true trajectory.

elements are determined. The required number of elementary operations has been reduced from 489 to merely 188, i.e., a reduction by more than 60% has been achieved. In addition, it was found that most of the equations could be implemented using an integer representation. Due to the large and relevant change in the value range of the Kalman gains, this part has been implemented using floating point operations. As a result, the optimized implementation achieved the same positioning accuracy as the complete implementation using floating point operations, yet with acceptable computational requirements.

Once the Kalman filter adapted dead reckoning is available, it should be further combined with the digital map information. The process of comparing the calculated vehicle trajectory against known roads is called map matching [1, 6]. The next section describes a novel pattern matching based method for map matching.

3. Pattern Recognition for Improving the Map Matching Process

The Kalman filter adaptation of the odometer and gyroscope information not only provides their optimal estimates but also the standard deviation of their errors. This error information is propagated in the positioning Eq. (1) leading to the error distribution in the determined position, i.e., the region where the vehicle true position might be. This position error distribution is then used in the standard navigation systems for determining the candidate positions on the road map. If there are more than one road intersecting the region with possible vehicle positions obtained from the Kalman filter updated dead reckoning, several candidate on-road positions will be determined. In the standard positioning approach each candidate position is then separately evaluated based on different criteria such as the heading, connectivity with respect to the candidates in the previous time step, closeness to the dead reckoning position estimate, etc. The candidate position with the highest likelihood is adopted as the true position at the current time instant. The problem with this approach is that the evaluation of the candidates is based only on the instantaneous information that is prone to error.

The novelty of the herein presented approach to map matching is that the historical information over several time integration steps in dead reckoning is compared in a suitable way in order to determine the optimal map matching position [10]. The underlying idea is to extract features from both dead reckoning trajectory and digital map that could be compared. The problem is that the resolutions of the two representations are different. The road representation in the digital map is piecewise linear while the dead reckoning trajectory is smooth due to the small time step in odometer and gyroscope time sampling. If the goal is to represent each of these trajectories as a sequence of curve and straight-line patterns, a method is needed to initially extract these features and to compare them in reliable manner.

4. State Machines for Feature Generation

State machines are used herein to process the available information in order to determine the apparent features “straight” and “turn.” The features themselves are derived by summation of values over a defined interval. The feature “straight” corresponds to the distance between the estimated vertices of two successive turns. The feature “turn” gives the change in heading in a curve.

During the development of the rules for state transitions, it has been found that intermediate states have to be defined in addition to the original states “straight” and “turn” (see Figs. 4 and 5). The intermediate states take care of the inaccuracies in the sensor and map data and, therefore, they enhance the reliability of the pattern matching process. There are two critical aspects during the development of state machines, the selection of states and the definition of the rules for the state transitions. The rules are typically defined as a comparison of a variable of interest with the appropriate threshold (see Figs. 4 and 5). Therefore, the thresholds have a major impact on the instance when a transition takes place and they simultaneously define if a feature is considered as significant or not. The evaluation criteria for feature generation applied in the state machines are: the current heading change, the accumulated heading change since start of the turn, the curvature, and the distance traveled while the state machines remains in the current state. Instead of the radius, the curvature, i.e., the inverse radius of the turn is considered in order to avoid singular behaviour for straight roads. Since both types of paths (sensor and map) are available with a different resolution, separate state machines are chosen for each of them. One of the main differences between the analysis of the sensor and map data is that the sensor data are continuous (smooth) while the digital map data are piece wise linear. Hence, two different state machines were derived, one for each path type. Figure 4 depicts the state machine for sensor data with four possible states and the rules for the transitions. $\Delta\varphi$ is the heading change in the current time step, $\Sigma\Delta\varphi$ is the accumulated heading change up to the current time step, $C=\Delta\varphi/\Delta L$ is the curvature where $\Sigma\Delta L$ describes distance traveled while the state machine remains in the current state. A curvature of 1,000 means a heading change of 1,000°/km=1°/m, where the corresponding turn has a radius of about 60 m.

In Fig. 5 the state machine for map data is depicted—with the three possible states and the rules for the transitions. The state machine, which analyzes the candidate trajectories on the digital map, has one state less (turn start) than the state machine for sensor data. The curve begin is not needed due to the piecewise linear road representation. The curve end state is still needed since a curve can be represented by a sequence of incremental angle changes, which should be ultimately summed up together.

In general, the state machine for map data uses different rules and different thresholds compared with the rules and thresholds of the state machine for sensor data (e.g., 4° heading change compared to 2°). This difference is caused by the difference in resolution of sensor and map data. Similar to sensor data the tuning of thresholds for the map data has been achieved based on test data—and this has been performed such that sequences of features of an equal length for sensor and map data are achieved.

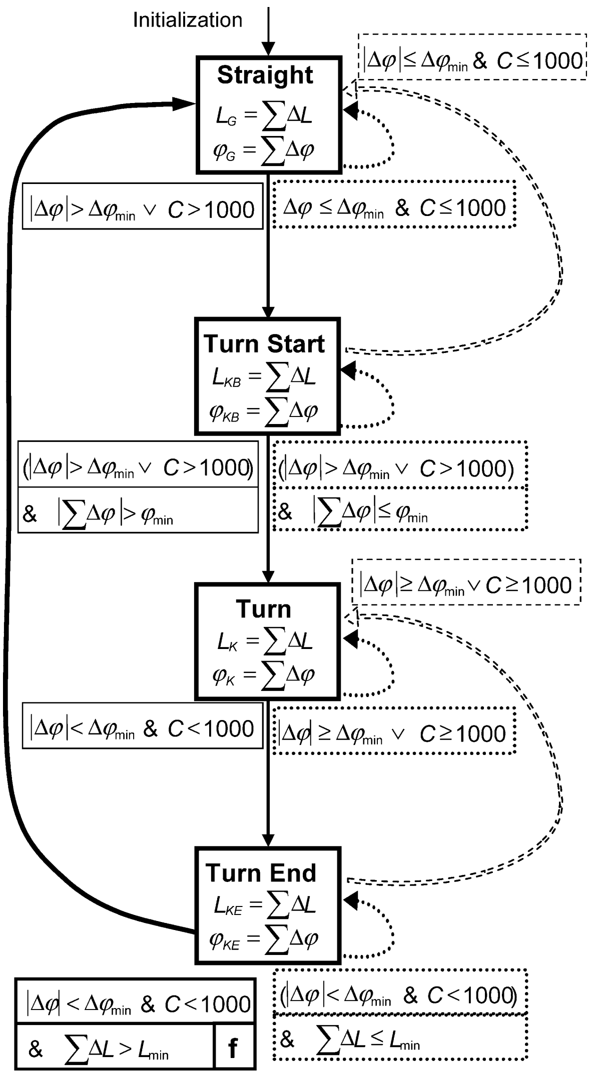

Figure 4. State machine for sensor data, where $\Delta\varphi_{min}$=2°, φ_{min}=30°, and L_{min}=10 m. Here, C=1,000 means C=1,000°/1 km=1°/m.

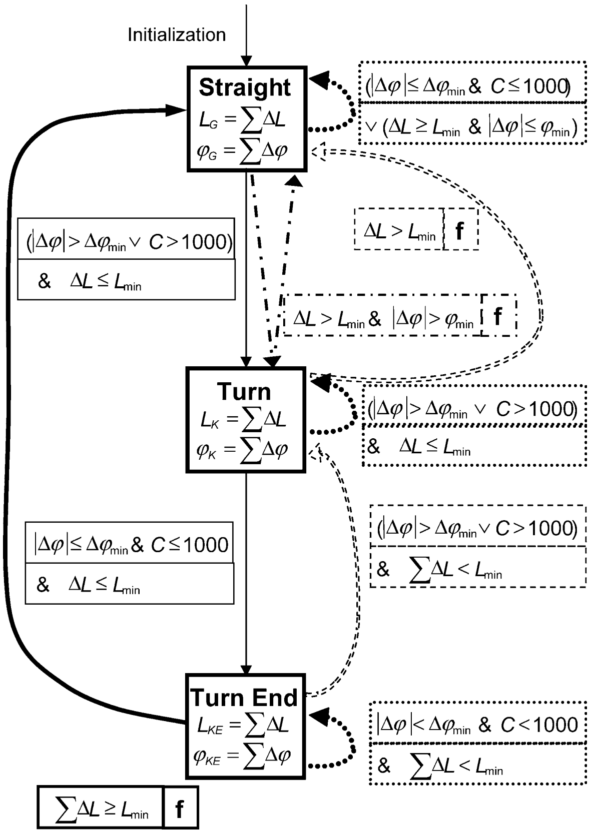

Figure 5. State machine for map data, where $\Delta\varphi_{min}$=4°, φ_{min}=30°, and L_{min}=35 m. Here, C=1,000 means C=1,000°/1 km=1°/m.

The features 'straight' with the length L and 'turn' with the heading change ϕ are simultaneously provided pairwise with the transition from 'turn end' to 'straight.' This transition is indicated by 'f' in Figs. 4 and 5. Features are provided only if the accumulated heading change is larger than φ_{min} (e.g., 30°). With the indices TS for turn start, T for turn, TE for turn end, and S for straight, the following relations hold:

$$L = L_{TE} + L_S + (L_{TS} + L_T)/2, \qquad (5)$$

$$\varphi = \varphi_{TS} + \varphi_T, \qquad (6)$$

The given thresholds have been defined based on the analysis of test drives. For the adjustment of the thresholds the following aspects should be considered:

- $\Delta\varphi$ or C too large: slowly driven turns are not recognized.
- $\Delta\varphi$ or C too small: the state "turn" is never left again.
- φ_{min} determines the heading change for a turn considered as significant, i.e., the heading change which is considered as relevant feature.
- L_{min} determines the minimum distance between two turns to be considered as separate turns and not just a single joined turn.

5. Evaluation of Certainties of Road Alternatives Based on Feature Vector Comparison

The features determined by state machines are stored in two-dimensional vectors. For the sensor signals there is one vector, for the map there are as many vectors as there are possible map alternatives. In the following the term path refers to the fact that feature vectors represent a course, i.e., the pattern.

The likelihood (certainty) of each possible map trajectory is calculated by comparing its feature vector with the features extracted from the dead reckoning trajectory. The comparison is based on the suitable norm of the distance between feature vectors with different weighting on the differences between the straight-line lengths and the curve angles. The error is additive, i.e., each new pair (curve, straight-line) is compared and the corresponding error is added to the already existing evaluation. The comparison starts only when four features have been already identified.

The feature pair (length, heading change) from the dead reckoning (L_{sens}, $\Delta\varphi_{sens}$) and the pair from the road map (L_{map}, $\Delta\varphi_{map}$) are compared according to the Eqs. (7–9). The certainty change due to the heading error ΔPC_w is evaluated based on a Gaussian function (distribution) with the standard deviation corresponding to 30°. This variable is then multiplied with the certainty change based on the length error α_1. The differences are $\Delta\varphi = \Delta\varphi_{sens} - \Delta\varphi_{map}$ and $\Delta L = L_{sens} - L_{map}$ while $L_{err,abs}$ is the allowed error (20 m in this case).

$$\Delta PC_w = 250\left(2e^{-\left(\frac{\Delta\varphi}{\sigma_w}\right)^2} - 1\right) \qquad (7)$$

$$\alpha_l = \begin{cases} e^{-\left(\frac{|\Delta L| - L_{\text{err,abs}}}{L_{\text{err,abs}}}\right)^2} & \text{if } |\Delta L| > L_{\text{err,abs}} \ \& \ \Delta \text{PC}_w > 0 \\ 1 & \text{if } |\Delta L| \leq L_{\text{err,abs}} \vee \Delta \text{PC}_w \leq 0 \end{cases} \tag{8}$$

$$\text{PC}_{\text{new}} = \text{PC}_{\text{old}} + \Delta PC_w \alpha_l \tag{9}$$

Figure 6 illustrates a comparison between a dead reckoning (DR) trajectory and the two possible trajectories on the digital map. The extracted curves with angle changes φ_i and φ_j and the corresponding straight-line length between them are enough to determine that the road 1 is a much more likely candidate than the road 2. The same figure illustrates also the dynamic nature of the pattern recognition approach. Between the start position and the curve φ_{i}, there was only one candidate. At that point, the digital map shows an intersection with two possible successive roads. Both of these roads became candidates with the same initial evaluation inherited from the original "parent" candidate. Both of these candidates are further evaluated based on their own features.

A special care had to be taken in order to guarantee robustness of the pattern generation and their comparison. Every time when there is a feature in one domain and not in the other, e.g., in the dead reckoning path but not on the candidate road, the comparison is delayed for some time (or traveled distance) in order to see if the corresponding feature can be detected. If, in spite of everything, the corresponding feature could not be detected this is interpreted as a mismatch between the dead reckoning and the candidate trajectory. As a consequence, the certainty of this candidate trajectory is decreased (i.e., its cumulative error is increased).

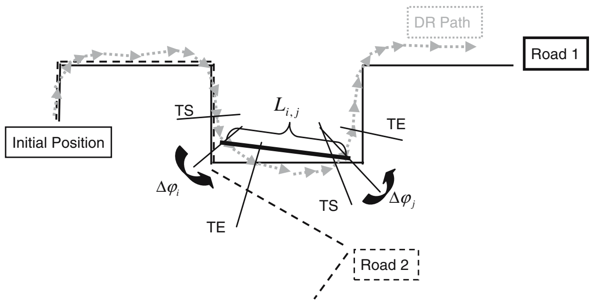

Figure 6. Example of the pattern recognition for evaluation of two possible road trajectories (*solid* and *dashed lines*) based on the digital map. The dead reckoning path is given by a *dotted line of arrows*. States and features corresponding to the dead reckoning path are indicated. In this example, the map alternative 1 is the correct one.

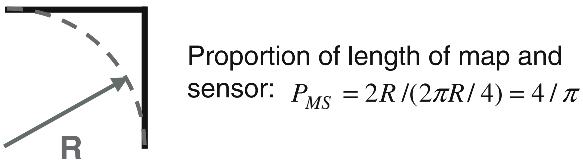

Figure 7. Correction for a turn represented in the map by a singular event.

6. Position Correction Following the Recognition of the Pattern TURN

In general, the differences with respect to the feature length of sensor and map could be used to adapt the odometer model. This step would require that the map is better than the currently used calibration. In most cases this cannot be expected—the currently applied calibration is rather accurate. Accordingly, the information provided by the pattern matching is used to check the position after a turn and to correct it, if appropriate. This can be realized by comparison of the distances that are determined by the state machines after the last turn, respectively. This comparison is unique, if there is only one possible map alternative. And, only in this case a length correction ($\Delta\text{Pos}_{\text{final}}$) is determined and used to correct the position.

The position correction is determined based on three aspects (see Eqs. 10–13). $\Delta\text{Pos}_{\text{init}}$ is always applied and this term is defined by the difference of the length after the last turn given by the length values of the two state machines $\left(\widehat{L}_{\text{sens}}, \widehat{L}_{\text{map}}\right)$. The estimated middle point of the turn, i.e., the mean between start and end of turn, is used for this step. $\Delta\text{Pos}_{\text{sing}}$ compensates inaccuracies in the map, if the last turn was represented in the map by a singular event (see Fig. 7). $\Delta\text{Pos}_{\text{lane}}$ takes into consideration that in the map only the middle of the street is stored, but not each lane (see Fig. 8). The final correction value $\Delta\text{Pos}_{\text{final}}$ is eventually determined by the sum of the used correction factors and considering a tolerance of the width of a lane (LW). Figure 9 shows an example revealing the position improvement achieved.

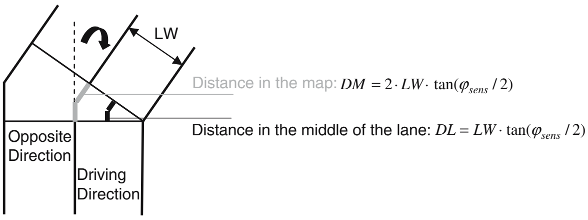

Figure 8. Correction due to consideration of the lane width: DL−DM.

$$\Delta\mathrm{Pos}_{\mathrm{init}} = \widehat{L}_{\mathrm{sens}} - \widehat{L}_{\mathrm{map}} \tag{10}$$

$$\Delta\mathrm{Pos}_{\mathrm{sing}} = \widehat{L}_{\mathrm{sens}} \tfrac{4-\pi}{\pi} \text{ if } 45^\circ < \varphi_{\mathrm{sens}} < 135^\circ \tag{11}$$

$$\Delta\mathrm{Pos}_{\mathrm{lane}} = -\mathrm{LW}\,\tan\left(\varphi_{\mathrm{sens}}/2\right) \text{ if } \varphi_{\mathrm{sens}} < 135^\circ \tag{12}$$

$$\Delta\mathrm{Pos}_{\mathrm{final}} = \begin{cases} 0 & \text{if } \sum \Delta\mathrm{Pos}_{\mathrm{used}} \le \mathrm{LW} \\ \sum \Delta\mathrm{Pos}_{\mathrm{used}} - \mathrm{LW} & \text{if } \sum \Delta\mathrm{Pos}_{\mathrm{used}} > \mathrm{LW} \end{cases} \tag{13}$$

To illustrate the results obtained by the described approach, two test drives have been performed. We refer to these test drives as A and B. Figure 10 shows a typical situation with and without the pattern matching algorithm (test drive A). As visible in the left half of Fig. 10, positioning without pattern recognition is not stable. It jumps from one road alternative to another and even shows an "off-road" candidate as the most probable one. The correct position is identified at the end of the shown trajectory. On the other hand, the positioning system with the pattern matching (right half of Fig. 10) follows smoothly the exact trajectory in a stable fashion.

Another typical situation is depicted by the test drive B (see Fig. 11). The driving direction is from top to bottom. Again the positioning without pattern recognition is not stable—it jumps from one road alternative to another and it takes the complete test drive B to identify the correct position. At the start of the shown drive, the pattern matching approach enables the navigation system to keep the correct map alternative and to identify this map alternative as the true position despite of an initializing error. Accordingly, the precise position becomes clear after a few meters (see top of right half of Fig. 11), when the pattern matching is applied. During test drive B the positioning system with the pattern matching follows smoothly the exact trajectory in a stable fashion.

The overall advantages of the pattern matching approach are as follows:

- Usage of the historical information in the form of a pattern sequence (curve, straight line) contrary to the standard use of instantaneous information

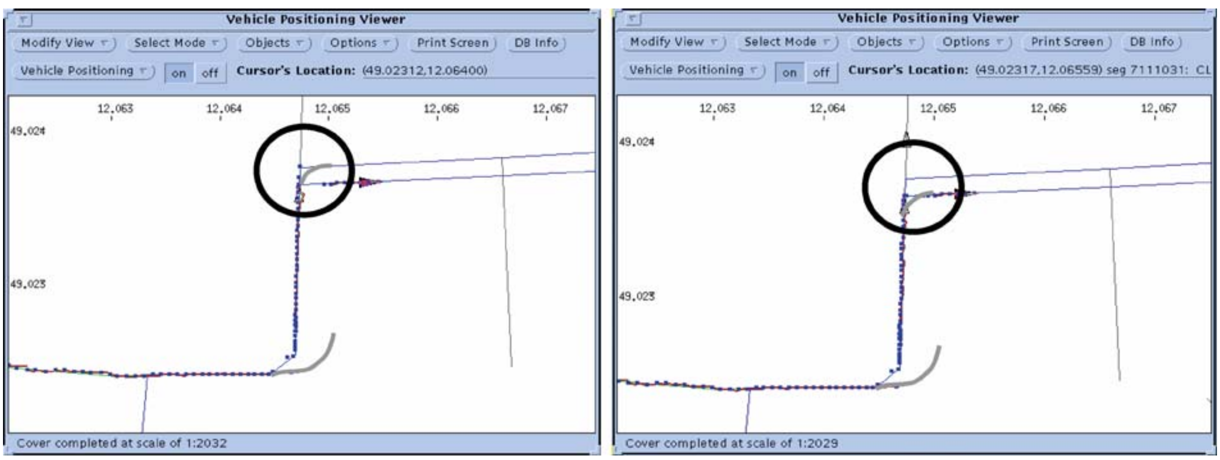

Figure 9. Example for position correction following the recognition of the pattern TURN. After the second turn a significant improvement has been achieved as indicated by the *grey lines*.

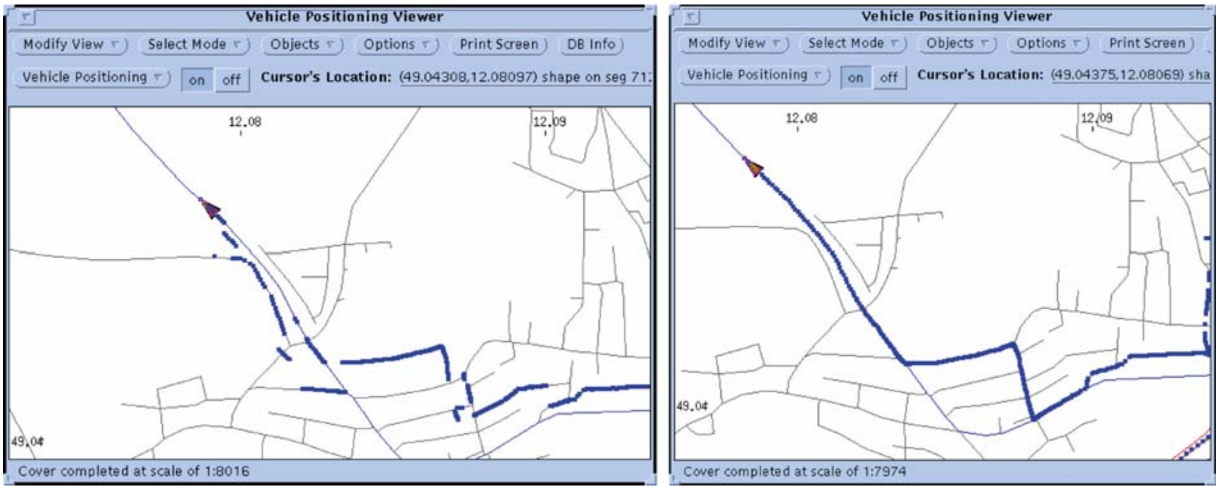

Figure 10. Test drive A without (*left*) and with (*right*) pattern recognition.

(current dead reckoning position, projections on the digital map, etc.). The historical information is herein combined with the classical projection algorithms and it can override the projection based solution.

- Robustness: the feature extraction is performed by integration of the distance and heading change, which leads to noise averaging. In addition, the feature vector matching is performed iteratively leading to the gradual change of the certainty (mismatch).
- Feature vectors generation is independent from the projection on the digital map performed by the standard positioning algorithm.

Other approaches to utilizing digital map data in the car positioning process can be found in [14–18].

7. Conclusions

Two novel sensor and information source fusion methods for vehicle positioning in car navigation systems are presented in this paper. The first sensor fusion is implemented via Kalman filter, which updates the odometer and gyroscope based dead reckoning model by using the appropriate GPS measurements. The so calculated dead reckoning trajectory was extensively tested in Siemens car navigation systems in a set of test drives where it

Figure 11. Test drive B without (*left*) and with (*right*) pattern recognition.

was shown that the resulting improvement in accuracy exceeds 20% in average.

The second novelty presented in this paper is the evaluation of different position candidates on the road map based on the historical information. The historical information is presented in the form of feature vectors consisting of the curve and straight-line elements. Appropriately designed state machines perform the feature extraction from both the dead reckoning and digital map. The certainty (mismatch) of the different position alternatives is iteratively updated every time a new feature is generated.

The navigation system including both the Kalman filter improved dead reckoning and the pattern recognition algorithm is a standard commercial product of Siemens AG and since 2000 has been implemented in Opel, Porsche, Alfa Romeo and Lancia models. In January 2002 the German car magazine Auto Bild has evaluated the performance of different navigation systems implemented in ten different cars. An important focus was the evaluation of the positioning accuracy. The Siemens navigation system implemented in Opel cars and in Alfa Romeo cars was graded with eight out of ten possible points. The same evaluation was given for the navigation systems in these cars: Jaguar S-Type, Mercedes ML 55 AMG, Toyota Yaris. However, the other five systems got grades below such that the average evaluation of the ten systems on the positioning accuracy was merely 5.7 points. Considering other aspects like time of route calculation and routing accuracy, the Siemens navigation system implemented in Opel cars was awarded the first place (test winner).

References

1. Y. Zao, "Vehicle Location and Navigation Systems," Artech House, USA, 1997.
2. E. Kaplan, "Understanding GPS, Principles and Applications," Artech House, USA, 1996.
3. P. Toth and D. Vigo (Eds.), "The Vehicle Routing Problem," Discrete Math, Society for Industrial and Applied Mathematics, USA, 2001.
4. H. Lenz et al., "Determination of Sensor Path by Dead Reckoning and Using the Time Discretization," Siemens Internal Report, 2000.
5. C. A. Scott, "Improved Positioning of Motor Vehicles Through Secondary Information Sources," Ph.D. Thesis, University of Technology, Sydney, 1996.
6. R. L. French, "Map Matching Origins, Approaches and Applications," 2nd International Symposium on Land Vehicle Navigation, 1989, pp. 93–116.
7. M. S. Grewal, V. D. Hendersom, and R. S. Miyasko, "Application of Kalman Filtering to the Calibration and Alignment of Inertial Navigation Systems," *IEEE Tr. on Automatic Control*, vol. 36, no. 1, 1991, pp. 4–13 (January).
8. B. Barshan and H. F. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots," *IEEE Trans. Robot. Autom.,* vol. 11, no. 3, 1995, pp. 328–342 (June).
9. D. Obradovic, H. Lenz et al., "Patent Application: Online Memory-based Correction of Dead-Reckoning Position Estimate Using GPS Signals," 2000.
10. D. Obradovic, H. Lenz et al., "Patent Application: Pattern Matching for Car Navigation Systems," 2000.
11. A. H. Jazwinski, "Stochastic Processes and Filtering Theory," Academic, New York, 1970.
12. S. Sukkarieh, E. M. Nebot, and H. Durrant-Whyte, "A High Integrity IMU/GPS Navigation Loop for Autonomous Land Vehicle Applications," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, 1999 (June).
13. Siemens AG proprietary algorithm. Siemens VDO internal report.
14. W. Kim et al., "Efficient Use of Digital Road Map in Various Positioning for ITS," in *IEEE Positioning and Navigation Symposium*, San Diego, USA, 2000.
15. R. R. Joshi, "A New Approach to Map Matching for In-vehicle Navigation Systems: The Rotational Variation Metric," in *ITSC 2001 Conference*, Oakland, USA, 2001.
16. C. Zewang et al., "Development of An Algorithm for Car Navigation System Based on Dempster–Shafer Evidence Reasoning," in *IEEE 5th Conference on Intelligent Transportation Systems*, Singapore, 2002.
17. G. Blewitt and G. Taylor, "Mapping Dilution of Precision (MDOP) and map-matched GPS," *Int. Journal on Geog. Information Science*, vol. 16, no. 1, 2002, pp. 55–67.
18. S. Edelkamp and S. Schroedl, "Route Planning and Map Inference with Global Positioning Trees," *Comp. Sci. in Perspective*, in R. Klein et al. (Eds.), Springer, Berlin Heidelberg New York, 2003.

Dragan Obradovic received the B.E. and M.E. degrees in Mechanical Engineering in 1980 and 1985, respectively, from the University of Belgrade, Yugoslavia. In 1990 he received the Ph.D. degree in mechanical engineering from the Massachusetts

Institute of Technology, Cambridge, USA. Since 1992, Dr. Obradovic has been with the Corporate Technology of Siemens in Munich, Germany where he currently holds a position of Principal Research Scientist in the Information and Communications department. His research involves, among others, the development and application of machine learning methods for system control and signal processing. He has published several dozens of journal and international conference papers and the book "An Information-Theoretic Approach to Neural Computing" (New York, Springer, 1996) and is author of several granted and pending patents. Dr. Obradovic also holds the Adjunct Professor position at Munich University of Technology, where he has been teaching since 2001 in the international Masters of Science in Communication Engineering program within the EE Department.

Henning Lenz received the M. Sc. degree in Electrical Engineering and Computer Science in 1993 from the Portland State University, Oregon, USA. In 1995 he received the Diploma in Electrical Engineering from the Technical University of Karlsruhe, Germany. In 1999 he received the Ph.D. degree in Electrical Engineering from the Technical University of Munich, Germany. Since 1999, Dr. Lenz has been with the Corporate Technology of Siemens in Munich, Germany where he currently holds a position of Principal Engineer in the Information and Communications department. His research involves the development of machine learning methods for traffic control, positioning systems, and diagnostics. He has published several journal and international conference papers and is author of pending patents.

Markus Schupfner received a Diploma of Mathematics in 1995 from the Fachhochschule Regensburg, Germany. He started working for the Siemens AG in 1995, spending most of his time with navigation system development in Sunnyvale, California, USA and Regensburg, Germany. Since 2001, Markus Schupfner has been working for Siemens VDO where he currently holds the position director of navigation department. His development and research involves the development of mathematical algorithms for positioning and route calculation. He has published several patents in the area of navigation systems.