Car2Pedestrian Positioning: Methods for Improving GPS Positioning in Radio-Based VRU Protection Systems

Sebastian Engel
and Claudia Kratzsch
Department of Vehicle Safety
AUDI AG

Klaus David
Chair for Communication
Technology
University of Kassel

Dominik Warkow
Chair of Transport Systems
Information Technology
TU Dresden

Marco Holzknecht
Chair of Computer Systems
in Engineering
University Magdeburg

***Abstract*—Current developments in the field of Car2X communication deal with the communication between vehicles as well as between vehicles and the infrastructure. Vulnerable road users, like pedestrians or bicyclists, are not yet regarded in these developments. We present an approach for integrating vulnerable road users into Car2X communication by using smartphones as sensor devices and for the communication. Since the position information is of crucial importance in Car2X communication systems, we developed a method to improve the position data that is gathered from the GPS sensor of the smartphones. A motion recognition is combined with walking speed estimation to develop a dead reckoning algorithm for pedestrians that takes their current movement state into account. The results of our algorithm are compared to a stand-alone GPS positioning solution of the smartphone to show the improvements that can be achieved with our solution.**

I. Introduction

According to the "White Paper on transport", the European Commission aims at reducing the number of fatalities in road transport to nearly zero by 2050. Therefore road safety technologies, like driver assistance systems, eCall and cooperative systems shall be deployed [1]. Regarding the cooperative systems, the communication between the road users becomes necessary, as it is the basis for cooperative behaviour in cooperative systems. The technology for exchanging information by communication between vehicles and their environment is called Car2X communication.

Current developments in the field of Car2X technology are engaged in the communication between vehicles as well as the communication between vehicles and the infrastructure. However, VRUs (Vulnerable Road Users) are involved in a quarter of all fatal traffic accidents in Germany [2]. Therefore this group of traffic participants has to be included in the development of cooperative safety systems.

In this paper, we present an approach for integrating VRUs into cooperative safety systems by using smartphones. This approach is named *Car2Pedestrian communication* in the following. This paper is organized as follows: In the next section, the Car2Pedestrian communication system is presented, which uses smartphones as sensor devices and for communicating the gathered information to vehicles. Since the position data of the VRU, that is determined by GPS, is a crucial information in our proposed system, in the third section results for improving the positioning solution for pedestrians by a combination of motion recognition, walking speed estimation and dead reckoning for pedestrians using smartphones are presented. Section four concludes the approaches of this paper.

II. Car2Pedestrian communication with smartphones

In this section our basic system concept for Car2Pedestrian communication is presented. This is necessary to understand the need for further examinations regarding the GPS positioning of smartphones. The concept is shown in Fig. 1. It consists of two modules, which are implemented on the smartphone and the vehicle, respectively.

The smartphone is carried by the pedestrian and is used to determine the position, speed and walking direction by the GPS sensor. The position coordinates are transformed into an Euclidean coordinate system by applying an UTM (Universal Transverse Mercator) projection. Afterwards the position, speed and direction data are transmitted to the vehicle via WLAN. Since the automotive WLAN 802.11p is not yet available in off-the-shelf smartphones, WLAN 802.11 b/g is used to transmit the data to a router which is installed in the vehicle.

In the vehicle, the position and driving direction are determined by a GPS receiver. The position coordinates are also transformed into Euclidean coordinates by an UTM projection. Further movement data like the speed or acceleration of the vehicle are readout from the CAN bus (Controller Area Network). Afterwards the future positions and movements of the pedestrian and the vehicle are predicted using a CV model (Constant Velocity) and a CA model (Constant Acceleration), respectively.

The equation for predicting the position and movement data of the pedestrian from time step $k-1$ to k is shown in Eq. 1, where n and e denote the position of the pedestrian in UTM coordinates, v the walking speed and φ the walking direction. Note that the north-component is predicted by the sinus ratio of the direction angle, the east-component by the

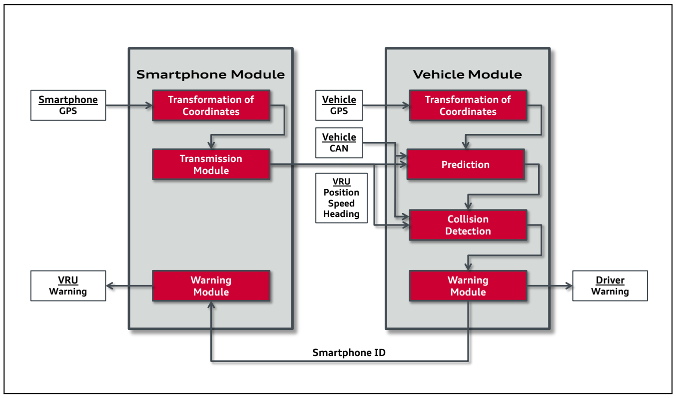

Fig. 1. Basic system concept for Car2Pedestrian communication using smartphones. Based on [3].

cosine ratio of the direction angle, since the geographical, not the mathematical angle is used.

$$\begin{pmatrix} n_k \\ e_k \\ v_k \\ \varphi_k \end{pmatrix} = \begin{pmatrix} n_{k-1} \\ e_{k-1} \\ v_{k-1} \\ \varphi_{k-1} \end{pmatrix} + \begin{pmatrix} v_{k-1} \cdot \cos(\varphi_{k-1}) \\ v_{k-1} \cdot \sin(\varphi_{k-1}) \\ 0 \\ 0 \end{pmatrix} \cdot dt \tag{1}$$

The equation for predicting the position and movement data of the vehicle from time step $k-1$ to k is shown in Eq. 2, where n and e denote the position of the vehicle in UTM coordinates, v_n the driving speed in northern direction and v_e the driving speed in eastern direction. The accelerations in the northern and eastern direction are denoted by a_n and a_e.

$$\begin{pmatrix} n_k \\ e_k \\ v_{n,k} \\ v_{e,k} \end{pmatrix} = \begin{pmatrix} n_{k-1} \\ e_{k-1} \\ v_{n,k-1} \\ v_{e,k-1} \end{pmatrix} + \begin{pmatrix} v_{n,k-1} \\ v_{e,k-1} \\ a_{n,k-1} \\ a_{e,k-1} \end{pmatrix} \cdot dt + \begin{pmatrix} \frac{a_{n,k-1}}{2} \\ \frac{a_{e,k-1}}{2} \\ 0 \\ 0 \end{pmatrix} \cdot dt^2 \tag{2}$$

If the predictions of Eq. 1 and 2 indicate that the positions of the pedestrian and the vehicle intersect in the near future and little time for braking is left in the vehicle, a warning is given to the driver. Furthermore a warning is sent to the smartphone. Therefore also the pedestrian is warned, who can react to the critical situation by raising his attention or changing his movement.

As it can be seen, the GPS data are the only information source for the movement prediction of the pedestrian. Therefore it is of major importance that this information is reliable. However, with off-the-shelf smartphones the position accuracy might have an error of approx. 10 m [4]. In bad environmental situations like urban canyons, where multipath and shadowing influence the accuracy of the positioning, the error might increase. In the following section, our approach for improving the GPS data by a combination of motion recognition, walking speed estimation and dead reckoning for pedestrians is presented.

III. Methods for Improving the GPS Positioning with Smartphones

Our approach for improving the GPS positioning of smartphones for pedestrians is split into three parts. At first, the current motion status of the pedestrian is estimated and grouped into the categories *stop*, *walk*, *run* or *bike*[1]. After the current motion state has been determined, the walking speed of the pedestrian is estimated in the case the state equals *walk* or *run*. In the last step, the pedestrian dead reckoning (PDR) is executed based on the different motion states and the walking speed. The results of the PDR algorithm are improved data regarding the position, speed and walking direction of the pedestrian. The different steps are explained in the next subsections.

A. Motion recognition with smartphones

The motion recognition is used to distinguish between different motion states that influence the choice of movement models in the PDR algorithm. Motion recognition by the evaluation of accelerometer data has been first described in [5], where multiple sensors were placed on the test subjects body to gather the information. However, the approach presented in this paper is more realistic because only a smartphone is used. Hence the pedestrian is not restrained in the mobility. Using smartphones for motion recognition is also known from [6], [7] and [8]. In contrast to those works, we use the smartphone not only for recording the data, but we also implement the motion recognition on the device. Therefore a real-time evaluation of the current activity is possible.

For the motion recognition, the data of the acceleration sensor and the gyroscope in the x-, y- and z-axis are captured with a sample rate of 40 Hz. A windowing is applied to the data to calculate different features of each data window. The windows contain 128 data samples, resulting in a window length of 3.2 seconds. The motion recognition is triggered every 800 ms, which results in a window overleap rate of 75 %. The following features are extracted from these windows for the classification:

- standard deviation and spectral entropy of acceleration (x-, y- and z-axis)
- standard deviation of angular speed (x-axis)
- spectral entropy of angular speed (x-, y- and z-axis)

This results in a feature vector containing 10 elements for each data window. As classification algorithms, a k-nearest-neighbour (kNN) and a decision tree (DT) were used. For the training phase, 12 test persons (six females, six males) performed each of the four activities for two minutes. The test persons differed in height and weight and wore different

[1] In this paper, only the algorithms for pedestrians are explained.

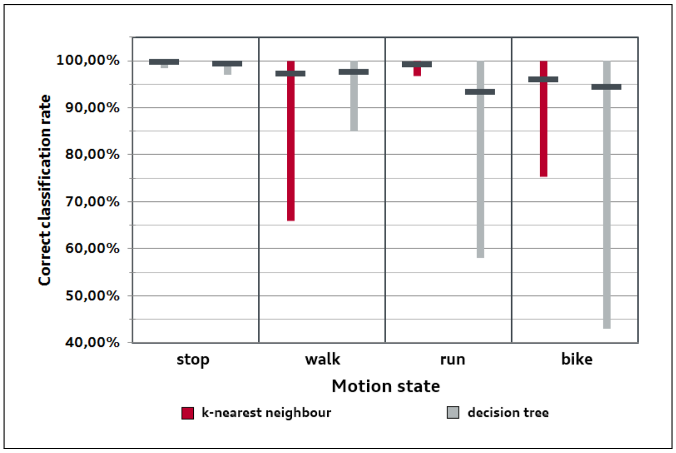

Fig. 2. Correct classification rate for different activities using a k-nearest neighbour algorithm and a decision tree.

Motion	Classified as stop	walk	run	bike	Classification rate true	false
stop	837	0	0	1	99,88 %	0,12 %
walk	0	898	20	5	97,29 %	2,71 %
run	0	5	794	0	99,37 %	0,63 %
bike	20	9	1	756	96,18 %	3,82 %

TABLE I
CONFUSION MATRIX OF THE K-NEAREST NEIGHBOUR ALGORITHM.

Motion	Classified as stop	walk	run	bike	Classification rate true	false
stop	834	0	0	4	99,52 %	0,48 %
walk	0	902	21	0	97,72 %	2,28 %
run	0	35	747	17	93,49 %	6,51 %
bike	15	17	11	743	94,53 %	5,47 %

TABLE II
CONFUSION MATRIX OF THE DECISION TREE.

clothing and shoes. Each record was labeled and used to train both classifiers. In all test runs, the position of the smartphone had to be restricted to achieve unambiguous sensor signals. Therefore, the smartphone was placed in the right trouser pocket with the display showing forwards.

To evaluate the classification accuracy, the leave-one-out strategy was used for all 12 test persons. The classifiers were trained with data of 11 test persons, while the data of the last person were used as the test set. All 12 combinations of training data vs. test data were evaluated to estimate the overall accuracy. The results are shown in Fig. 2, where the average correct classification rate for each motion is illustrated for both classifiers. The vertical bars indicate the range from lowest to the highest classification rate of all 12 test combinations.

The confusion matrices of both classification algorithms are shown in Tab. I and II. With both classifiers, misclassifications between *walk* and *run* occurred. This is due to the similar motion patterns of both activities. The misclassifications of the motion status *bike* can be attributed to the results of one test person, whose motion differed from the other test persons during this activity due to his clothing.

Both classifiers show similar results with a correct classification rate greater than 90 % for each motion state. In overall, with a correct classification rate of 98 %, the kNN algorithm outperforms the DT with a classification rate of 96 %. Even with a small training data set of 11 test persons, a very good accuracy could be achieved. However, one has to point out that these results could only be achieved because the position of the smartphone was restricted and only four activities had to be distinguished.

Based on the results, we decided to implement the DT algorithm on the smartphone. Therefore another DT was trained using all 12 test persons. Even if the classification rate is slightly worse, the computational effort is reduced if a DT is used. The if-else-structure of the DT can easily be implemented on the smartphone after it has been created in the training phase. On the contrary, a more complex data structure has to be implemented on the smartphone if a kNN algorithm is used, to accelerate the search process of the k nearest neighbours. The initialization of this data structure needs some time every time the smartphone module is started. To prevent this delay, the DT was chosen.

B. *Estimation of walking speed*

In this section the approach for estimating the walking speed of the pedestrian presented. This information is used in the dead reckoning algorithm to predict the further movement of the pedestrian in the fusion process.

For the estimation of the walking speed, the step frequency f of the pedestrian is determined at first. Therefore the frequency of the gyroscope data of the smartphone is analyzed. Since the frequency of the gyroscope data was already calculated for the motion recognition, no additional calculations are necessary. Afterwards the frequency value f is applied to a cubic function:

$$v(f) = a \cdot f^3 + b \cdot f^2 + c \cdot f + d \tag{3}$$

To determine the parameters a to d, several test runs were carried through where four test persons walked beside a vehicle that dictated the speed the test persons had to walk. The frequencies of each test person at a given vehicle reference speed was analyzed and the median of each frequency cluster was calculated. The result is shown in Fig. 3. Finally, the medians have been approximated by a cubic function to determine the parameters a to d.

The test persons had a similar height and age, therefore the results could be applied well to our data. For a more general approximation, different characteristics of the test persons, e.g. the age and gender, should be varied to obtain different

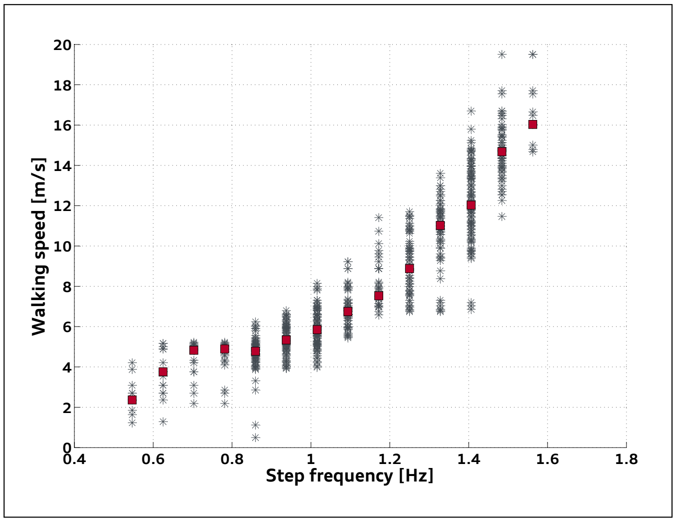

Fig. 3. Step frequencies at given walking speeds of the test persons. The red markers indicate the median value of each frequency cluster.

approximation curves. In the application phase, the best-suiting approximation function for the individual pedestrian would have to be chosen.

A simple step detection was not used in the walking speed estimation, because this would imply the necessity of a given step length. As a consequence, a linear dependency between the step frequency and the walking speed would occur. However, as one can see in Fig. 3, this dependency is non-linear. This also has been shown in [9].

In the next subsection, we explain how the motion recognition and walking speed estimation are used to implement a dead reckoning algorithm for pedestrians, in order to improve the GPS positioning accuracy of smartphones.

C. Dead reckoning for pedestrians

Pedestrian dead reckoning systems are well-known from the literature. In [10], [11], [12] and [13] for example, shoe-mounted sensors are used to detect steps of the test subjects by an accelerometer and their walking direction by a compass or a gyroscope. However, the use of dedicated sensors that are mounted at the shoe increase the hardware effort. In [14] smartphones are used to implement a dead reckoning by step length estimation, heading estimation and map matching. In contrast to our approach, a correction by GPS measurements is not incorporated. A Kalman filter approach for PDR is also described in [15]. Similar to the method described in this paper, step detection is used to determine the travelled distance [16] and the combination of a gyroscope and a magnetic compass delivers the walking direction [17]. Our approach differs from [15], since we have chosen a more realistic approach by just using smartphones, whereas several sensors were placed over the test subjects body in [15]. Beyond that, the extension with motion recognition to enhance the data fusion process has not been examined in detail yet in the authors opinion. Like the motion recognition, our PDR module is implemented on the smartphone to enable a real-time positioning solution without offline post-processing.

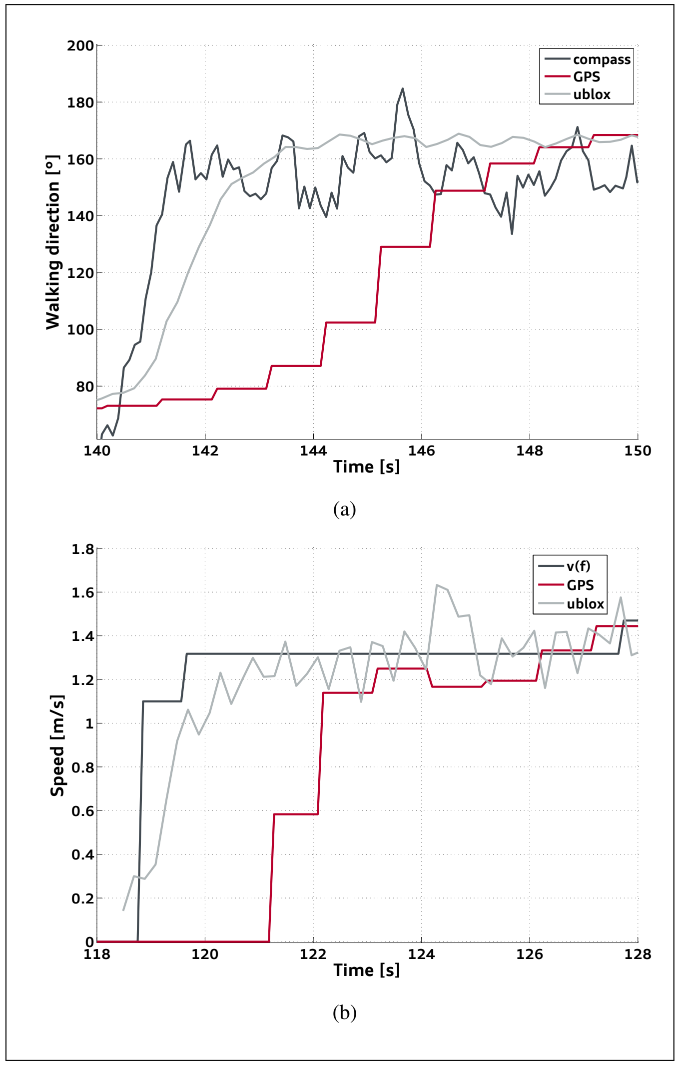

Fig. 4. Comparison of the heading information gathered from the GPS sensor of the smartphone, the *ublox* GPS sensor and the compass of the smartphone (a). Comparison of the speed information gathered from the GPS sensor of the smartphone, the *ublox* GPS sensor and the walking speed estimation (b).

A data fusion algorithm was developed that fuses the sensor data of the acceleration sensor, the gyroscope and the digital magnetic compass with the position and movement data that are delivered by the GPS sensor of the smartphone. The first three smartphone sensors are named *inertial sensors* in the following. Two requirements have been in the focus for the improvement:

1) Increase the sample rate of the GPS position data.
2) Enable a fast detection of position changes which are caused by abrupt movement changes of the pedestrian.

The first requirement is due to the fact that the default sample rate of modern smartphone GPS sensors is at 1 Hz. For an automotive pedestrian safety application this data rate is too low.

The second requirement is closely linked to the first one, because also sudden changes in the movement of the pedestrians must be detected. The data of the GPS sensors of different smartphones in multiple scenarios were analyzed to motivate the second requirement. Fig. 4(a) shows the walking direction of a pedestrian for an exemplary scenario. The test person carried a *Samsung Galaxy S II* smartphone and a *ublox 6T* GPS receiver to compare the results of both sensors. The pedestrian made an abrupt change in the walking direction of 90° in this scenario. We compared the GPS heading information of the smartphone with the heading information of a *ublox 6T* GPS receiver and the data of the compass of the smartphone. Compared to the other sensors, the GPS receiver of the smartphone reacts delayed to the direction change. Fig. 4(b) shows similar results for the walking speed while the pedestrian changes from standing to walking in another scenario. Compared to the speed that is delivered by the *ublox* sensor or our walking speed estimation, the speed delivered by the smartphone GPS sensor reacts delayed at the beginning.

The developed fusion architecture is shown in Fig. 5. The original system concept of Fig. 1 was enhanced by the integration of the inertial sensors. At first, the current motion state of the pedestrian is determined by the recognition algorithm mentioned in section III-A. Afterwards this information is used in the PDR algorithm to choose the process model for the fusion process. Furthermore the estimated walking speed of the pedestrian is used for the prediction process in our fusion algorithm.

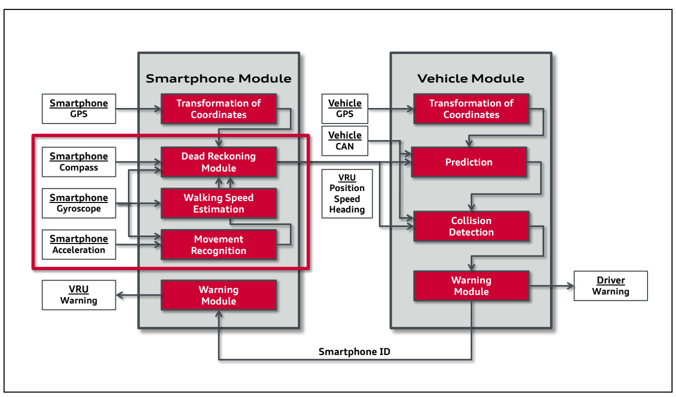

Fig. 5. System concept for Car2Pedestrian communication using dead reckoning for improved GPS positioning.

We use a Kalman filter to fuse the inertial sensors as well as the GPS sensor and to implement the PDR algorithm. The state vector equals to

$$\mathbf{x} = \begin{pmatrix} n \\ e \\ v \\ \varphi \end{pmatrix}, \tag{4}$$

where n and e denote the position of the pedestrian, v the walking speed and φ the walking direction. The observation matrix $\mathbf{H}$ is an identity matrix. The covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ of the process noise as well as the measurement noise are diagonal matrices whose values have been estimated heuristically.

The state vector $\mathbf{x}$ is predicted every 100 ms to obtain new position and movement values at 10 Hz. The update step of the Kalman filter is executed every time new GPS data arrive, hence every 1000 ms. The results of the motion recognition and the walking speed estimation are used to influence the prediction and the update step of the Kalman algorithm: If a motion change of the pedestrian is detected, the GPS data are not incorporated in the update step for a certain time. This is due to the results that can be seen in Fig. 4(b), where the data of the smartphone GPS sensor react delayed to the motion change. A turning detection based on the data of the gyroscope was also developed: If the data of the gyroscope exceed a certain threshold, the data of the GPS sensor are also not incorporated in the update step. This is because of the results that can be seen in Fig. 4(a).

In the default case, if the pedestrian is walking and no change in the motion or turning is detected, the following process model is used in the prediction steps:

$$\mathbf{x_k} = \mathbf{A} \cdot \mathbf{x_{k-1}}, \tag{5}$$

where

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \cos(\varphi_{k-1}) \cdot dt & 0 \\ 0 & 1 & \sin(\varphi_{k-1}) \cdot dt & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

During the prediction steps, the walking direction φ is assumed to be constant. The data of the compass or the gyroscope are not incorporated during these steps. Fig. 4(a) shows an oscillation of the compass data that is caused by the movement of the leg during walking. This oscillation would led to a zigzag course of the pedestrian during the prediction. The walking speed v is also assumed to be constant. However, in the update step, the speeds delivered by GPS and by the walking speed estimation are fused. Likewise, the positions delivered by GPS and by the prediction as well as the walking direction delivered by GPS and by the compass are fused in the update step.

Since the process model of Eq. 5 is very restrictive regarding the movement of the pedestrian, the model is changed if a turning of the pedestrian is detected by the gyroscope data:

$$\mathbf{x_k} = \mathbf{A} \cdot \mathbf{x_{k-1}} + \mathbf{B_1} \cdot \mathbf{w_{1,k-1}}, \tag{7}$$

where

$$\mathbf{B_1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 180/(\pi \cdot \mathrm{dt}) \end{pmatrix} \tag{8}$$

and

$$\mathbf{w_{1,k-1}} = \omega_{k-1} \tag{9}$$

In this case, during the prediction steps the walking direction φ is influenced by the yaw rate ω_{k-1} that is delivered by the gyroscope. In the update step, the data of the GPS sensor are not incorporated. This is due to the delayed behaviour of the GPS sensor in turning scenario. This means, if the pedestrian is turning, the position and movement data are solely calculated based on the inertial sensors of the smartphone for a certain time span. Afterwards the constant movement of Eq. 5 is assumed again, if no further turning or motion change is detected.

A similar approach is chosen if a change in the motion is detected. If the pedestrian changes in between the states *stop*, *walk* or *run*, the following process model is used during the prediction steps:

$$\mathbf{x_k} = \mathbf{A} \cdot \mathbf{x_{k-1}} + \mathbf{B_2} \cdot \mathbf{w_{2,k-1}}, \tag{10}$$

where

$$\mathbf{B_2} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{11}$$

and

$$\mathbf{w_{2,k-1}} = v(f)_{k-1} \tag{12}$$

If the pedestrian changes its motion in between the three categories, an abrupt change of the walking speed v will occur. Therefore, the result of the walking speed estimation $v(f)_{k-1}$ is incorporated in each prediction step. The speed information delivered by GPS is not incorporated in the update step for a certain time, because it reacts delayed to the change. Hence the change in the walking speed during accelerating or stopping can be represented well. If the motion changes from standing to walking or running, also the direction information by GPS is not incorporated in the update step for a certain time. This is due to the fact that no direction information can be delivered by GPS without a movement. If the direction by the compass would be fused with the GPS direction, the result would be erroneous. Therefore, the GPS direction is disregarded in this case.

In the case the pedestrian is standing the following process model is used:

$$\mathbf{x_k} = \mathbf{A} \cdot \mathbf{x_{k-1}} + \mathbf{B_3} \cdot \mathbf{w_{3,k-1}}, \tag{13}$$

where

$$\mathbf{B_3} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{14}$$

and

$$\mathbf{w_{3,k-1}} = \Delta\varphi_{\text{compass},k-1} \tag{15}$$

The walking speed v is set to 0, therefore the pedestrian is not changing its position during the prediction steps. However, the direction changes $\Delta\varphi_{\text{compass},k-1}$ of the compass influence the direction information φ. Therefore the correct direction is known if the pedestrian starts to walk again. The compass instead of the gyro is used, because longer standing times and the necessary integration of the angular speed could lead to erroneous direction information if the gyroscope suffers from an offset. In the update step, the position data of the GPS are incorporated, to allow a convergence to the GPS position, which is assumed to be dependable in longer static measurements. The walking speed v is set to 0, the direction information φ is directly taken from the compass signal.

Fig. 6 shows the results of the PDR algorithm in a scenario where the pedestrian starts to walk and executes a right turn after a while. The red triangles indicate the position information by the smartphone GPS, the gray circles the dead reckoning positions and the light gray line shows the comparable *ublox* trace. The black connection lines illustrate the corresponding positions of the smartphone GPS and the dead reckoning algorithm during the update step.

In the beginning phase, the positions of the PDR algorithm indicate a faster change since the speed of the pedestrian is directly based on the walking speed estimation. On the contrary, the smartphone positions react delayed during this phase. A similar behaviour can be seen during the turning phase. It takes some steps, until the smartphone positions converge to the dead reckoning and *ublox* positions.

All in all, the requirements regarding the increased sample rate and the detection of abrupt movement changes can be fulfilled with the PDR algorithm. However, if the absolute positions of the smartphone GPS sensor are erroneous over a longer time period but similar to the calculated position

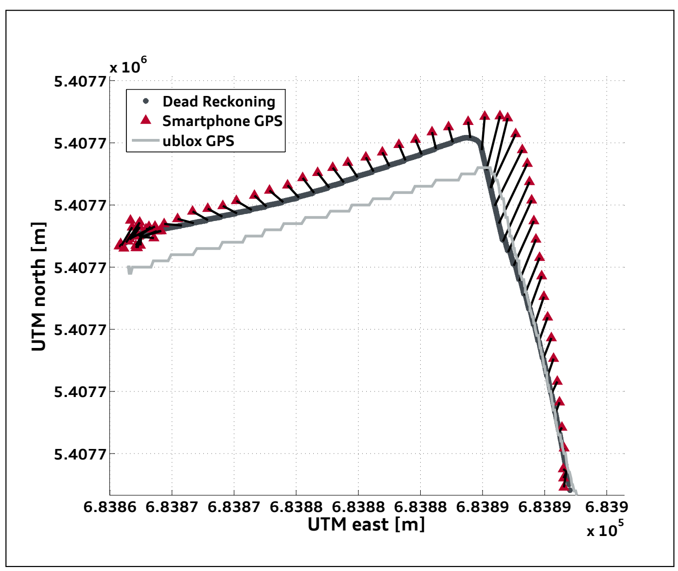

Fig. 6. Comparison of GPS position coordinates delivered by the smartphone GPS sensor, the *ublox* GPS sensor and the dead reckoning algorithm.

based on the inertial sensors, no improvement by the algorithm can be achieved. This can be seen in the phase between the beginning and the turning. Both position traces are parallel with the dead reckoning trace slowly converging to the GPS trace. This is due to the fact, that another reference is missing in this case. Therefore, short-term corrections are possible with the algorithm, but no long-term corrections.

D. Computing time on smartphones

In this subsection the necessary execution times of our algorithm are analyzed. Further test runs under the same conditions as in section III-C were executed, where the PDR algorithm was executed on the smartphone while the test person was walking outside and the data were transmitted. A *Samsung Galaxy S II* and a *HTC Sensation* were used in our tests. In the test runs, the GPS was captured with a sample rate of 1 Hz. The algorithm was executed with a frequency of 10 Hz, hence delivering position data every 100 ms. Fig. 7 shows the execution times for the PDR algorithm in an exemplary test run with both smartphones. For both smartphones the execution times are in the range of 2 ms or less with some outliers due to internal processes of the smartphones. Evaluating all of our test runs, we could ascertain that 95 % of all executed dead reckoning tasks were completed within 2 ms. With a position update every 100 ms the realtime requirement is fulfilled, allowing even higher position update rates.

IV. Conclusion

We presented an approach for improving the GPS data that are delivered by off-the-shelve smartphones by applying a dead reckoning algorithm for pedestrians. The results show that an improved position accuracy is possible in special scenarios like turnings or sudden motion changes. However, the approach suffers from some restrictions at this point of time. To unambiguously evaluate the sensor signals of the smartphone, its position had to be restricted. Further research is necessary to determine the motion state of the pedestrian if the smartphone is placed in another way, e.g. inside of a handbag. Moreover, a more differentiated walking speed estimation is necessary which takes the pedestrians individual walking behaviour into account. However, we could show that all necessary calculations for the PDR can be executed on the smartphone in realtime. This shows that the approach of using smartphones is promising for the development of Car2Pedestrian communication systems for vulnerable road users.

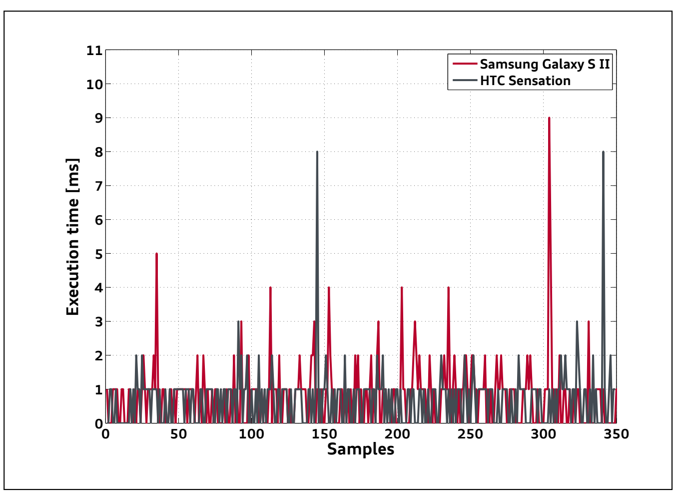

Fig. 7. The necessary execution times for the algorithm implemented on two smartphones show that the realtime requirement is fulfilled.

References

[1] European Commission, "White Paper on transport," 2011, http://ec.europa.eu/transport/themes/strategies/doc/2011_white_paper/white-paper-illustrated-brochure_en.pdf, last access: 23.09.2013.

[2] Statistisches Bundesamt, "Verkehrsunfälle - Zeitreihen 2012," 2013, https://www.destatis.de/DE/Publikationen/Thematisch/TransportVerkehr/Content75/ListeVerkehrsunfaelle.html, last access: 04.10.2013.

[3] S. Engel, C. Kratzsch, and K. David, "Car2Pedestrian-Communication: Protection of vulnerable road users using smartphones," in *Advanced Microsystems for Automotive Applications*, J. Fischer-Wolfarth and G. Meyer, Eds. Springer, 2013, pp. 31 – 41.

[4] D. Thielen *et al.*, "A Feasibility Study on a Cooperative Safety Application for Cyclists crossing Intersections," in *ITSC*, 2012.

[5] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data," in *Pervasive 2004*, A. Ferscha and F. Mattern, Eds. Springer, 2004, pp. 1 – 17.

[6] S. L. Lau and K. David, "Movement Recognition using the Accelerometer in Smartphones," in *Future Network and Mobile Summit*, 2010, pp. 1 – 9.

[7] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity Recognition from Accelerometer Data on a Mobile Phone," in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, ser. Lecture Notes in Computer Science, S. Omatu, M. Rocha, J. Bravo, F. Fernández, E. Corchado, A. Bustillo, and J. Corchado, Eds. Springer, 2009, vol. 5518, pp. 796 – 799.

[8] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explorations Newsletter*, vol. 12, pp. 74 – 82, 2011.

[9] J. E. Bertram and A. Ruina, "Multiple walking speed-frequency relations are predicted by constrained optimization," *Journal of Theoretical Biology*, vol. 209, no. 4, pp. 445 – 453, 2001.

[10] R. Stirling *et al.*, "An Innovative Shoe-Mounted Pedestrian Navigation System," in *Global Navigation Satellite Systems Conference*, 2003.

[11] L. Ojeda and J. Borenstein, "Personal Dead-reckoning System for GPS-denied Environments," in *IEEE International Workshop on Safety, Security and Rescue Robotics*, 2007.

[12] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Computer Graphics and Applications*, vol. 25, no. 6, pp. 38 – 46, 2005.

[13] R. Feliz, E. Zalama, and J. G. García-Bermejo, "Pedestrian tracking using inertial sensors," *Journal of Physical Agents*, vol. 3, no. 1, pp. 35 – 42, 2009.

[14] B. Shin *et al.*, "Indoor 3D pedestrian tracking algorithm based on PDR using smarthphone," in *12th International Conference on Control, Automation and Systems*, 2012.

[15] V. Gabaglio, Q. Ladetto, and B. Merminod, "Kalman Filter Approach for Augmented GPS Pedestrian Navigation," in *Global Navigation Satellite Systems Conference*, 2001.

[16] Q. Ladetto, "On foot navigation: continuous step calibration using both complementary recursive prediction and adaptive Kalman filtering," in *ION GPS*, 2000.

[17] Q. Ladetto, V. Gabaglio, and B. Merminod, "Combining Gyroscopes, Magnetic Compass and GPS for Pedestrian Navigation," in *International Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation*, 2001, pp. 205 – 212.