# Requirements on Delay of VRU Context Detection for Cooperative Collision Avoidance

Michel Morold*, Quang-Huy Nguyen†, Marek Bachmann*, Klaus David*, and Falko Dressler‡
*Chair for Communication Technology, University of Kassel, Germany
†Dept. of Computer Science and Heinz Nixdorf Institute, Paderborn University, Germany
‡School of Electrical Engineering and Computer Science, TU Berlin, Germany
{michel.morold,marek.bachmann,klaus.david}@comtec.eecs.uni-kassel.de
{nguyen,dressler}@ccs-labs.org

***Abstract*—Every year, approximately 350 000 vulnerable road users (VRUs) still lose their lives in road traffic accidents worldwide. To reduce this number, cooperative VRU collision avoidance systems aim at complementing current vehicle or infrastructure-based systems for VRU protection. The cooperative approach assumes user equipment (UE) like a smartphone or a smartwatch on the VRU side, which allows to obtain and exchange movement and contextual information with other road users for determining the current risk level for a collision. In recent publications, the usage of additional contextual information, such as the pedestrian's current activity (e.g., standing, walking or crossing a curb), has been shown to improve the collision detection accuracy. However, those approaches focused on the detection accuracy, but did not investigate how time delays for both detection and communication of VRU contexts affect the ability to detect collisions. We fill this gap by investigating the influence of both activity recognition and communication delays on the collision detection performance. As a baseline, we use the standardized Euro NCAP test protocol. For the evaluation, we exemplarily consider the usage of a curb detection module for position correction, which improves the collision detection accuracy at the moment the pedestrian crosses the curb.**

## I. Introduction

Every year, about 350 000 VRUs, mostly pedestrians and cyclists, lose their lives in road traffic accidents [1]. Several approaches aim to reduce the number of these accidents by VRU detection and collision avoidance systems, e.g., using cameras, radar, or LIDAR mounted on vehicles or the road infrastructure. However, those approaches usually depend on a direct line-of-sight (LOS) to the VRU and may struggle in non-line-of-sight (NLOS) scenarios to detect VRUs in a reliable and timely manner. In recent years, cooperative VRU collision avoidance has emerged as a complementary approach for vehicle and infrastructure-based detection systems to mitigate those limitations [2], [3]. The cooperative approach assumes UEs, like smartphones or smartwatches, on the VRU side. Those devices are equipped with Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) sensors and allow tracking and exchange of movement information with nearby vehicles or the infrastructure [2] (see Fig. 1).

However, there are still a lot of challenges to be addressed when integrating UEs into cooperative collision avoidance systems. Most notably, the low accuracy of current smartphone GNSS makes it difficult to accurately track VRU movements.

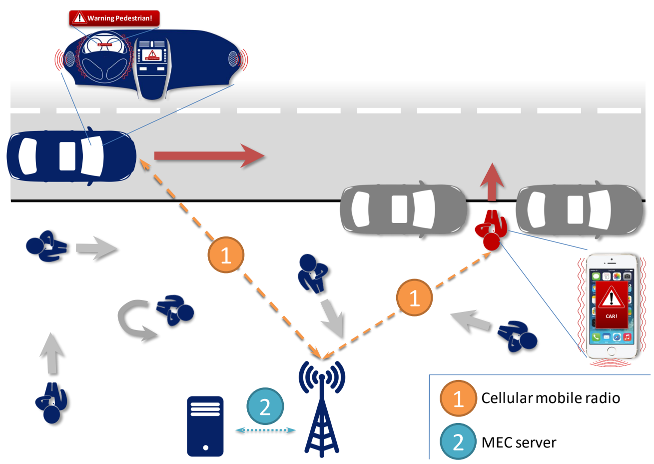


Fig. 1. Overview over our proposed system architecture.

As shown in [4], [5], a position accuracy of at least 0.5 m is required for reliable collision avoidance, which is not achieved by current smartphones.

In order to address these challenges, more recent publications [6]–[10] proposed the usage of VRU-specific contextual information, which have been shown to be beneficial for improving the performance of cooperative collision avoidance systems. For example, the detection that a pedestrian has crossed a curb [11] represents a valuable input for a cooperative system, which can be used to increase both the position accuracy and the collision detection probability [6].

However, the existing approaches which use VRU context to improve VRU safety only focus on detection accuracy, without considering the detection time needed and how it affects the performance of detecting an impending collision, especially in time-critical scenarios. In this paper, we close this gap and investigate how delays for both detection and communication influence the ability to detect impending collisions between VRUs and vehicles. This investigation is conducted for a collision scenario that is representative for 80 % of all traffic accidents involving pedestrians.

Our contributions can be summarized as follows:

- We analyze the delays for different stages of the *Activity Recognition Chain* for the detection of *crossing a curb*;

- we use the Veins LTE simulator to estimate the communication delay for the collision context derived from a standardized European New Car Assessment Programme (Euro NCAP) scenario; and
- we investigate the collision detection accuracy using a curb detection module for position correction. We particularly consider the influence of delays for both detection and communication on the collision detection probability.

## II. Related Work

In recent years, the integration of contextual or VRU-dependent information into cooperative VRU collision avoidance system has drawn considerable attention among several research groups. In [8], the authors used smartphone sensors to detect the VRU's current activity, i.e., *stop*, *walk*, *run*, or *bike*, which is used to choose the appropriate process model of a Pedestrian Dead Reckoning algorithm. The authors showed that their approach was able to provide a higher sampling rate and a higher position accuracy than only using smartphone GNSS. A similar approach was pursued in [7], which first detects the current movement activity of a cyclist, i.e., *waiting* or *starting*, and then chooses an appropriate model based on the detected activity to forecast the cyclist's future trajectory. For the detection of cyclist movements, the authors used a cooperative approach based on sensor data from the cyclist smartphone and a wide angle stereo camera. In conclusion, no accident scenario was considered so far, so the influence of detection and communication delay of VRU context on the collision detection performance remains unclear.

Besides approaches that capture VRU motion sequences, the non-periodic activity *crossing a curb* and stepping onto the road represents a valuable input for a cooperative system as well. It can be detected via dedicated, shoe-mounted inertial sensors [12] or smartphones [11] and is particularly useful for assessing the current degree of collision risk if the exact position of a pedestrian is not known, for example due to a high GNSS inaccuracy of the pedestrian's smartphone. In [6], it has been shown that the usage of a curb detection module for position correction increased the probability of detecting an impending collision from 29.4% to a probability of 75.9%, for an assumed position inaccuracy of 4 m.

However, the delay for both detecting VRU context as well as communication in time-critical scenarios and their influence on the collision detection accuracy has not been investigated so far. In this paper, we extend our prior work [6] and analyze the influence of both detection delay of *crossing a curb* as well as the communication delays on the probability of detecting an impending collision.

## III. Delay Analysis of Context Detection in Cooperative VRU Collision Avoidance Systems

For our system architecture, we assume a Multi-access Edge Computing (MEC) server, which receives Cooperative Awareness Messages (CAMs) via cellular connections (e.g., Long-Term Evolution (LTE)) (see Fig. 1). CAMs sent by UEs of pedestrians contain their current position, direction, and speed as well as the currently detected pedestrian context, i.e., *walking*, *running*, or *crossing a curb*. The collision risk estimation between all vehicles and VRUs is continuously calculated on the server side. In case an impending collision between two road users is detected, the MEC server sends a Decentralized Environmental Notification Message (DENM), to initiate collision avoidance measures. The server is assumed to have unlimited computing resources so that the remote execution time can be neglected. In case a *crossing a curb* event is detected, the longitudinal position accuracy for collision detection is corrected. The exact position of the curb is assumed to be known.

The detection of a pedestrian's current activity usually comprises different stages within the *Activity Recognition Chain* for (1) data acquisition (2) preprocessing and segmentation, (3) feature extraction, and (4) classification [13]. Each stage takes a certain amount of time, which contributes to the overall detection delay. The first stage collects and stores sensor data (e.g., acceleration) at a given sampling rate, but introduces a small, possibly negligible delay. The next stage, preprocessing and segmentation, usually involves cleaning up the data (e.g., removing outliers) or applying filters and then segmenting the data. Commonly, data is segmented by applying a sliding window approach with fixed-length window sizes, while some approaches allow consecutive windows to overlap. The window size impacts the detection delay of an activity directly. It determines the time, all subsequent stages have to wait until they can continue processing the data. In time critical scenarios like VRU safety, it is mandatory to find the best possible trade-off between window size (i.e., delay) and accuracy to ensure both a timely and robust detection. However, while some activities (like standing or sitting) can still be detected reliably when reducing the windows size significantly (e.g., to less than 100 ms), some other activities may require larger window sizes to capture temporal connections in sensor data. Subsequently, the third stage extracts representative features (e.g., mean or variance) from preprocessed data based on segments, and feeds those features into a pre-trained machine learning model for classification (e.g., decision tree). Both steps are computed locally on the smartphone and the time needed depends on the available hardware resources, e.g., available processor and storage speed.

In our previous work [14], we already investigated local computation times for feature extraction and classification using different machine learning algorithms with varying window lengths and sampling frequencies of sensor data. The results show that the average local computation time does not exceed 17 ms for all cases. The experiments were performed on a Nexus 6 smartphone, which was released in 2014. Current smartphones provide a much faster computation time. Therefore, we neglect these delays here.

In case of the activity *crossing a curb*, Jahn et al. [11] obtained an overall recall of 86 % and a precision of 90.6 %, with a window size of 250 ms with an overlap of 0.8, using accelerometer data at 32Hz. The mean duration of a curb step was found to be 1 s. This configuration performs classification every 50 ms, resulting in 20 instances on average per *crossing a curb* activity. When segmenting the training data, all segments are labeled by means of a majority vote, i.e., a segment is labeled as *crossing a curb* if the number raw data, labeled as *crossing a curb*, exceeds 50 % (i.e., 125 ms or 5 values) of the window. Considering a sampling rate of 32Hz, a segment with 250 ms is labelled as *crossing a curb* if the number of values is at least 5, which results in a minimum detection time of $\Delta t_{\text{curb,min}} = 156.25$ ms. Given a mean duration of a curb step of 1 s, and a recall of 86 % for *crossing a curb*, on average, 3 of 20 instances are not recognized correctly. In the worst case, these erroneously detected instances occur directly at the beginning of *crossing a curb*, which results in a maximum expected detection time of $\Delta t_{\text{curb,exp}} = 306.25$ ms. Nevertheless, it should be noted that this estimation is based on averages, so there might be cases in which *crossing a curb* is detected later than 306.25 ms or not detected at all.

## IV. Evaluation Setup

In this section, we describe our assumptions and configurations for both the Veins LTE simulation as well as the simulator used to calculate the collision detection probability. We also introduce the models used to specify the pedestrian mobility and the collision detection.

### A. Scenario

For our simulation, we chose the CPNC-50 (Car-To-Pedestrian Nearside Child 50 %) scenario from the Euro NCAP test protocol for VRU safety systems [15]. In this scenario, a pedestrian crosses the street while a car approaches from his left-hand side, while the line of sight is visually obstructed by parked cars (see Fig. 2). This scenario covers about 80 % of all road traffic accidents involving pedestrians [5]. In comparison to CPNC-50, we virtually added a curb in our simulation, which is reached by the pedestrian after 1.8 m. The remaining parameters are kept identically and are shown in Fig. 2. We model the scenario in a 2-dimensional Cartesian coordinate system in which the geometry of the car (c) is represented as a rectangle with a length of $L_c = 4$ m and a width of $W_c = 2$ m. The geometric center of the car's rectangle represents its current position, while the pedestrian (p) is represented as a point.

### B. Movement Modeling and Collision Detection

We assume a linear movement for the car and the pedestrian. Thus, the position of the car and the pedestrian depending on the time $t$ is given by the linear movement equation for $i \in \{\text{p}, \text{c}\}$ as

$$\mathbf{r}_i(t) = \begin{pmatrix} x_i \\ y_i \end{pmatrix}(t) = v_i \cdot t \cdot \begin{pmatrix} \sin(\phi_i) \\ \cos(\phi_i) \end{pmatrix} + \mathbf{r}_i, \quad (1)$$

where $\mathbf{r}_i$ is the current position, $v_i$ is the speed, and $\phi_i$ is the direction. $\mathbf{r}_i(t)$ represents the geometric center of the road user's geometry at time $t$. We call $\mathbf{m}_i = ((x_i, y_i)^T, v_i, \phi_i)$, i.e. the parameters of (1), the movement vector of road user $i$. An impending collision is detected based on linear extrapolation

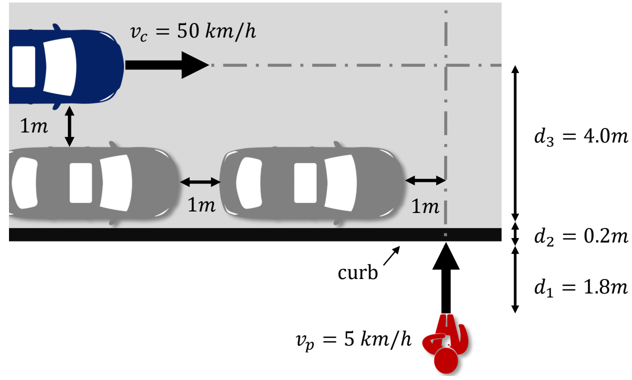


Fig. 2. Scenario with curb (derived from Euro NCAP).

of (1). The function $f(\mathbf{m}_\text{p}, \mathbf{m}_\text{c})$ evaluates if the geometries of two road users intersect at some $t$, the time to collision (TTC).

### C. Modeling Sensor Accuracies

The ability to detect an impending collision is measured by the "probability of collision detection" $P_\text{C}$. This metric was introduced in [5] and evaluates the probability of detecting an impending collision depending on the movement recognition accuracy of a VRU. The calculation of $P_\text{C}$ is based on evaluating $f(\mathbf{m}_\text{p}, \mathbf{m}_\text{c})$ for all possible VRU movement vectors in a set $M_\text{p}$. The set $M_\text{p}$ is composed by adding all values within the coverage interval $\pm 3 \cdot \sigma$ of the error models to the ground truth (gt) movement vector $\mathbf{m}_{\text{p,gt}} = (\mathbf{r}_{\text{p,gt}}, \phi_{\text{p,gt}}, v_{\text{p,gt}})$. We use the error models given in Table I.

Table I
Error Models for the Cooperative System.

| | |
|---|---|
| Position error longitudinal | $X \sim \mathcal{N}(0, \sigma_X)$ |
| Position error lateral | $Y \sim \mathcal{N}(0, \sigma_Y)$ |
| Direction error | $\Phi \sim \mathcal{N}(0, 15°)$ |
| Speed error | $V \sim \mathcal{N}(0, 0.3\,\text{m/s})$ |

### D. Modeling Delays for Curb Detection

At the time at which the pedestrian crosses the curb, its longitudinal position accuracy is set to $d_2 = \sigma_Y = 0.2$ m. Since we assume that the collision detection is performed on the server side, the sum of the detection and communication delay causes an offset between the ground truth position ($\mathbf{r}_{\text{p,gt}}$) of the VRU and the position which is received (rx) by the server ($\mathbf{r}_{\text{p,rx}}$). Since the pedestrian moves along the y-axis, it allows us to express $\mathbf{r}_{\text{p,rx}}$ as $\mathbf{r}_{\text{p,rx}} = (0, y_{\text{gt}} - \Delta t \cdot v)^T$. Thus, to consider the delay in the $P_\text{C}$ calculation, the assumed position $\mathbf{r}_{\text{p,gt}}$ is replaced by $\mathbf{r}_{\text{p,rx}}$ for all $m_\text{p} \in M_\text{p}$. Afterward, $P_\text{C}$ is calculated as described in [5]. After the position correction, the longitudinal position inaccuracy $\sigma_Y$ increases by 0.042m per second until the initial value for $\sigma_Y$ is reached.

### E. Simulation Setup of the Veins LTE simulator

We evaluate the timing performance for messages communication between cars and pedestrian's smartphones in the

Table II
SIMULATION PARAMETERS

| Simulation Parameter | Value |
|---|---|
| Simulated Area | 1 km × 1 km |
| Layout | Intersection (equivalent to Fig. 2) |
| Simulation time | 30 s |
| Repetitions | 10 |
| LTE scheduler | MAXCI |
| Bandwidth | 5 MHz (25 RBs) |
| UE transmission power | 23 dBm |
| eNodeB transmission power | 45 dBm |
| Number of vehicles | 50 |
| Vehicle speed limit | 50 km/h |
| Number of pedestrians | 50, 100, 150, and 200 |
| Pedestrian speed limit | 5 km/h |
| Beaconing interval (pedestrian) | 100 ms and 500 ms |
| Number of background UEs | 50, 100, 150, and 200 |
| Background traffic | 4 kB + uniform(−2 kB, 2 kB) |
| Background traffic interval | 1000 ms + uniform(−500 ms, 500 ms) |
| CAM length | 300 B |

scenario presented in Section III by means of network simulations. Given the popularity of LTE-based communication for safety applications, we focus on LTE-based communication in this study. We only consider the Uu interface since it is already available for smartphones. The LTE-V2X (PC5) interface, which is currently not compatible with LTE smartphones, is left for future work. We used the Veins LTE simulator [16], which supports LTE communication within a vehicular simulation framework. We evaluate the average latency of communication from pedestrian's UEs to cars.

In most Car-to-Pedestrian (Car2P) systems, CAMs are supposed to be transmitted, if channel capacity permits, every 100 ms. However, due to limited resources, pedestrian's smartphones may reduce the frequency of sending messages in some special circumstances. Therefore, in our simulations, we configure pedestrian objects to send CAMs to an MEC server with the period of 100 ms and 500 ms. The server then forwards the CAM to all vehicles within communication range. Following modern edge computing concepts, we place the server at the base station. We also take into account the influence of traffic load on the communication delay. To do that, we deployed a number of LTE users at random positions in the simulated area exchanging data with the eNodeB. The most relevant simulation parameters are summarized in Table II.

## V. Results and Discussion

We first determine the delays for communication in our considered scenario. Based on these delays, we simulate and evaluate the influence of the activity detection delay on $P_C$.

Fig. 3 shows the average communication delay from pedestrians to vehicles. Generally, Pedestrian-to-Car (P2Car) delay is less than 500 ms in most cases. Besides, it is obvious that in scenarios where the density of pedestrians and LTE users is low, smaller latencies can be achieved. In our simulation, the average latency for cases where the density is less than 50 people is less than 100 ms. Similarly, the delay increases when

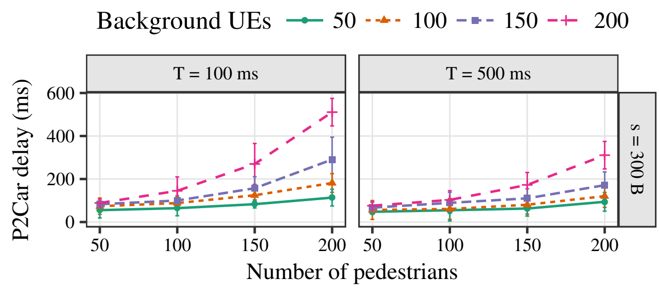


Fig. 3. Average P2Car delay.

the sending frequency of CAMs is higher. These observations can be explained by the limitation of radio resources that makes a timely schedule of the transmission not guaranteed as the load increases.

Similar to the investigation in [6], we performed our evaluation for the assumed pedestrian position inaccuracies ($\sigma_X = \sigma_Y = \sigma_{XY}$) 0.5 m, 1.0 m, 2.0 m, and 4.0 m. We also kept a direction error of $\sigma_\phi = 15°$ and a speed error of $\sigma_V = 0.3$ m/s. As discussed in Section III, the delay for *crossing a curb* varies between $\Delta t_{\text{curb,min}} = 156.25$ ms and $\Delta t_{\text{curb,exp}} = 306.25$ ms. For the communication via LTE between pedestrian and car, we assume delays for $\Delta t_{\text{LTE}}$ of 50 ms, 100 ms, 300 ms, and 500 ms. The results for $P_C$ are shown in Fig. 4. Within every scenario, it is important to note that $P_C$ rises over time, as the pedestrian approaches the collision point, since more movement trajectories inevitably lead to a collision. Moreover, $P_C$ decreases with rising communication delay (especially for a communication delay of 500 ms), which is caused by the permanent offset of the pedestrian. This effect is explicitly noticeable for smaller $\sigma_{XY}$, i.e., 0.5 m and 1.0 m and to a lesser degree for larger $\sigma_{XY}$. For $\sigma_{XY} = 0.5$ m, $P_C$ improves by ≈0.05 at the moment the pedestrian crosses the curb when considering no delay at all. With rising communication delay, improvements of $P_C$ first get smaller and for further increasing delay, i.e. $> 300$ ms, $P_C$ even deteriorates, which is caused by the position offset due to delay. For the maximum expected delay for crossing a curb $\Delta t_{\text{curb,exp}}$, $t_{\text{LTE}} = 50$ ms already causes a decrease of $P_C$ at the moment the curb step is detected. Considering $\Delta t_{\text{curb,exp}}$ and $\Delta t_{\text{LTE}} = 500$ ms, $P_C$ is reduced by 0.32.

In case of $\sigma_{XY} = 1.0$ m, the general communication delay affects $P_C$ to a lesser extent. Improvements through position correction are still noticeable for delays up to 500 ms for $\Delta t_{\text{curb,min}}$ and 300 ms for $\Delta t_{\text{curb,exp}}$. In contrast, if $\sigma_{XY}$ is 2.0 m and 4.0 m, which represent more realistic position accuracies for current smartphones, we still notice an increase of $P_C$ due to curb correction, even for the highest combined delay (806.25 ms). Although having less time available before the collision (e.g., for braking or evading) after *crossing a curb* was detected and sent to the car, the ability to detect an impending collision is still improved.

In conclusion, when considering the maximum expected delay for the detection of *crossing a curb*, the position

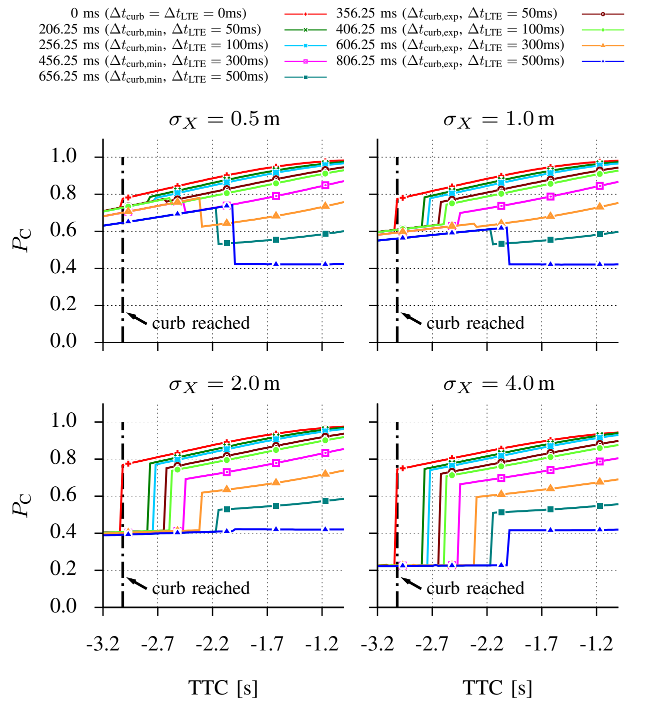


Fig. 4. $P_C$ over time for different position inaccuracies ($\sigma_x$) and different delays, starting at -3.2s before the collision.

correction improves $P_C$ as long as $\sigma_{XY}$ is higher than 2.0 m, or the overall delay for $\sigma_{XY} = 0.5$ m and $\sigma_{XY} = 1.0$ m does not exceed 100 ms and 300 ms, respectively. Thus, a cooperative collision avoidance system must be able to assess all these parameters (i.e., position inaccuracy, delay for detection and communication) at any given time to decide whether to use or discard the received contextual information about the pedestrian. This step is mandatory in order to avoid a deterioration of the collision detection performance caused by delayed information.

## VI. Conclusion

In this paper, we investigated how different delays for activity detection and LTE communication impact the ability to detect impending collisions between vulnerable road users (VRUs) and vehicles. For the evaluation, we focused on a curb detection module running on the VRU smartphone. Therefore, we first estimated the maximum expected delay for crossing a curb to be 306.25 ms. Based on this assumption, we found that if the VRU's position inaccuracy is 0.5 m and 1.0 m, the overall delay for detection and communication must not exceed 100 ms and 300 ms, to still improve the collision detection probability. For higher position inaccuracies over 2 m, the curb detection module is still able to increase the collision detection probability, even considering an overall delay of up to 806.25 ms. Our results clearly show that cooperative collision avoidance systems can substantially benefit from additional contextual information, like the detection of *crossing a curb*. However, delays for both detection and communication have to be taken into account when integrating contextual information in order to ensure a significant improvement of the collision detection accuracy.

## Acknowledgement


We acknowledge partly funding of this research by the FastAR project (HA-Nr. 851/20-11), funded by the State of Hessen and HOLM (House of Logistics and Mobility) within the program "Innovationen im Bereich Logistik und Mobilität" of the Ministry of Economics, Energy, Transport and Housing in Hessen, Germany.


## References


[1] World Health Organization. (2018) Global status report on road safety 2018. [Online]. https://apps.who.int/iris/bitstream/handle/10665/276462/9789241565684-eng.pdf, accessed: 21 June 2020.

[2] A. Tahmasbi-Sarvestani, H. N. Mahjoub, Y. P. Fallah, E. Moradi-Pari, and O. Abuchaar, "Implementation and Evaluation of a Cooperative Vehicle-to-Pedestrian Safety Application," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 4, pp. 62–75, 2017.

[3] X. Wu, R. Miucic, S. Yang, S. Al-Stouhi, J. Misener, S. Bai, and W.-h. Chan, "Cars Talk to Phones: A DSRC Based Vehicle-Pedestrian Safety System," in *IEEE VTC 2014-Fall*, Vancouver, Canada, Sep. 2014.

[4] P. Themann, J. Kotte, D. Raudszus, and L. Eckstein, "Impact of Positioning Uncertainty of Vulnerable Road Users on Risk Minimization in Collision Avoidance Systems," in *IEEE IV 2015*, Seoul, South Korea, Jun. 2015, pp. 1201–1206.

[5] M. Bachmann, M. Morold, and K. David, "On the Required Movement Recognition Accuracy in Cooperative VRU Collision Avoidance Systems," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, pp. 1–10, Mar. 2020.

[6] M. Bachmann, M. Morold, and K. David, "Improving smartphone based collision avoidance by using pedestrian context information," in *IEEE PerCom 2017, Work in Progress Papers*, Kona, HI, Mar. 2017, pp. 2–5.

[7] M. Bieshaar, S. Zernetsch, M. Depping, B. Sick, and K. Doll, "Cooperative starting intention detection of cyclists based on smart devices and infrastructure," in *IEEE ITSC 2017*, Yokohama, Japan, Oct. 2017, pp. 1–8.

[8] S. Engel, C. Kratzsch, K. David, D. Warkow, and M. Holzknecht, "Car2Pedestrian Positioning: Methods for Improving GPS Positioning in Radio-Based VRU Protection Systems," in *6. Tagung Fahrerassistenzsysteme*, Munich, Germany, Nov. 2013.

[9] R. B. Zadeh, M. Ghatee, and H. R. Eftekhari, "Three-Phases Smartphone-Based Warning System to Protect Vulnerable Road Users Under Fuzzy Conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2086–2098, Jul. 2018.

[10] A. Rostami, B. Cheng, H. Lu, M. Gruteser, and J. B. Kenney, "Reducing unnecessary pedestrian-to-vehicle transmissions using a contextual policy," in *ACM CarSys 2017*. Snowbird, UT: ACM, Oct. 2017, pp. 3–10.

[11] A. Jahn, M. Morold, and K. David, "5G Based Collision Avoidance-Benefit from Unobtrusive Activities," in *EuCNC 2018*, Ljubljana, Slovenia, Jun. 2018, pp. 352–356.

[12] S. Jain, C. Borgiattino, Y. Ren, M. Gruteser, Y. Chen, and C. F. Chiasserini, "LookUp: Enabling Pedestrian Safety Services via Shoe Sensing," in *ACM MobiSys 2015*, Florence, Italy, May 2015, pp. 257–271.

[13] A. Bulling, U. Blanke, and B. Schiele, "A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors," *ACM Computing Surveys*, vol. 46, no. 3, pp. 1–33, 2014.

[14] Q.-H. Nguyen, M. Morold, K. David, and F. Dressler, "Car-to-Pedestrian Communication with MEC-Support for Adaptive Safety of Vulnerable Road Users," *Elsevier Computer Communications*, vol. 150, pp. 83–93, Jan. 2020.

[15] European New Car Assessment Programme (Euro NCAP). (2019, Feb.) Test Protocol - AEB VRU Systems (Version 2.0.4). [Online]. https://cdn.euroncap.com/media/43381/euro-ncap-aeb-vru-test-protocol-v204.pdf, accessed: 21 June 2020.

[16] F. Hagenauer, F. Dressler, and C. Sommer, "A Simulator for Heterogeneous Vehicular Networks," in *IEEE VNC 2014, Poster Session*, F. Dressler, O. Altintas, S. Banerjee, B. Scheuermann, and D. Eckhoff, Eds., Paderborn, Germany, Dec. 2014, pp. 185–186.