POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Ingegneria Elettronica e delle Comunicazioni – XXVII ciclo

Tesi di Dottorato

Vehicular Networks and Outdoor Pedestrian Localization

III Facoltá di Ingegneria
ING-INF/03

Carlo Borgiattino
Matr. 189149

Tutore
Prof. Carla Fabiana CHIASSERINI

Coordinatore del corso di dottorato
Prof. Ivo MONTROSSET

March 2015

Summary

This thesis focuses on vehicular networks and outdoor pedestrian localization. In particular, it targets secure positioning in vehicular networks and pedestrian localization for safety services in outdoor environments.

The former research topic must cope with three major challenges, concerning users' privacy, computational costs of security and the system trust on user correctness. This thesis addresses those issues by proposing a new lightweight privacy-preserving framework for continuous tracking of vehicles. The proposed solution is evaluated in both dense and sparse vehicular settings through simulation and experiments in real-world testbeds. In addition, this thesis explores the benefit given by the use of low frequency bands for the transmission of control messages in vehicular networks.

The latter topic is motivated by a significant number of traffic accidents with pedestrians distracted by their smartphones. This thesis proposes two different localization solutions specifically for pedestrian safety: a GPS-based approach and a shoe-mounted inertial sensor method. The GPS-based solution is more suitable for rural and suburban areas while it is not applicable in dense urban environments, due to large positioning errors. Instead the inertial sensor approach overcomes the limitations of previous technique in urban environments. Indeed, by exploiting accelerometer data, this architecture is able to precisely detect the transitions from safe to potentially unsafe walking locations without the need of any absolute positioning systems.

Acknowledgements

I would like to thank my supervisor Prof. Carla Fabiana Chiasserini together with Prof. Claudio Ettore Casetti for their great guidance and support.

Furthermore, I am thankful to Prof. Marco Gruteser and his research team at Winlab, a wireless research center of Rutgers University, New Jersey. The period spent there was really valuable. My thanks also goes to Shubham Jain and Francesco Bronzino for their support and help.

Finally I am grateful to Francesca, who has always supported me in difficult moments.

Contents

List of Figures

Chapter 1

Introduction

In recent years, vehicular networks are gaining more importance due to a steady increase of interest in safety and entertainment applications for Intelligent Transportation Systems (ITS). Roadways can be made safer by letting vehicles communicate road and traffic conditions, as well as their position and velocity. In such scenario, a continuous knowledge of vehicles position is required by several applications such as real-time traffic monitoring, e-tolling and liability attribution in case of accident. However, constant tracking of vehicles position presents strong implications in terms of security and user privacy. Therefore, public disclosure of identity and position of drivers should be avoided.

Another important aspect that ITS are now facing is pedestrian safety. Traffic accidents with pedestrians still account for a significant number of injuries or fatalities and there is mounting evidence that mobile device distractions of pedestrians are exacerbating this problem. Smartphones, which are part of the problem, can also be part of the solution. Using existing activity recognition techniques, mobile devices can sense when they are being used while walking. If they could also detect potentially dangerous situations, they could generate much more targeted and noticeable alerts. In the context of a vehicle safety communication system, this information could be shared with vehicles over radio links using available standards (e.g IEEE 802.11p).

In this thesis we focus on vehicular networks and outdoor pedestrian localization. First of all we propose a new architecture that identifies and tracks vehicles position. For its evaluation, different scenarios have been studied through simulations and measurements on the fields using real prototypes. We then analyze the advantages given by the use of non conventional low frequency bands in vehicular networks. Finally, we focus on pedestrian localization for safety services. We design and develop different solutions that exploit positioning techniques and inertial sensors for distinguishing safe and unsafe walking locations. We carried out walking trials in environments ranging from suburban to metropolitan areas in order to evaluate their robustness.

In Chapter 2 we introduce a new lightweight privacy-preserving framework for continuous tracking of vehicles. In the proposed architecture, each vehicle broadcasts anonymous position beacons. The nearby cars collect such information and report it to a central authority, which verify the locations announced by vehicles. Furthermore, in presence of unverified location claims, the authority is able to infer the actual position of malfunctioning or misbehaving vehicles. In a nutshell, this system allows a trusted authority to securely collect and verify the positions claimed by vehicles without resorting to computationally expensive asymmetric cryptography.

In Chapter 3 we analyze the benefits given by the use of low frequency bands in vehicular networks. Frequency spectrum regulations have licensed 5.9 GHz band or dedicated short-range communication (DSRC) for ITS. DSRC channels are of limited capacity and recent studies have highlighted their scarcity, in comparison to the broad range of services that are envisioned in vehicular networks. At an identical transmitter power, low-frequency bands offer a significantly larger coverage than 5-GHz DSRC implementations. This translates into the possibility for the vehicles to interact with the ITS in advance. We developed a real vehicular testbed, with infrastructure nodes operating in the UHF band as well as roadside units (RSUs) operating at 5 GHz to evaluate the perfomance of a novel content downloading architecture. The proposed protocol leverages the 700MHz band for control messages and the high-throughput 5-GHz bands for data delivery.

Chapter 4 proposes a novel localization technology for pedestrian safety in outdoor environments. Using the GPS position obtained from a smartphone and a map, we design an algorithm able to detect when a pedestrian is going to cross a street. This technique identifies the street closest in distance from the pedestrian's current location. Then our algorithm achieves crossing detection by predicting a pedestrian's path of motion and checking if it intersects with any of the streets nearby. We analyze the requirements for pedestrian risk detection from rural to urban environments. In addition, we study the limits of our approach through walking trials in different environments. The results obtained show that GPS-based approaches are feasible in rural and suburban areas while they are not applicable in dense urban environments. Indeed, large errors in positioning and delays in detection make these approaches not suitable for critical applications.

In Chapter 5 we focus on pedestrian safety in urban scenario. In order to overcome the limitations of the GPS-based approach described in Chapter 4, we propose a new technique based on shoe-mounted inertial sensors. Using shoe-mounted accelerometers, we can measure the feet inclination and consequently compute the slope of the ground. Our system is able to identify transitions between sidewalks and streets by searching for sloped transitions (ramps) from the inclination of the ground. In addition, by keeping track of the instantaneous variation in the accelerometer magnitude, we are able to recognize the stepping over a curb, event that often occurs when crossing a street. We developed a prototype to evaluate the effectiveness of this approach both in laboratory experiments and in approximately 40 hours of walking. In addition, we designed a slope-based localization algorithm that, given a profiling database, permits to locate the sidewalk a pedestrian is walking on. In summary, our solution is able to precisely detect the transitions from safe to potentially unsafe walking locations without the need of any absolute positioning systems.

Finally, in Chapter 6, we present the conclusions we have drawn from this work.

Chapter 2

Verification and Inference of Vehicular Positions

A number of vehicular networking applications require continuous knowledge of the location of vehicles and tracking of the routes they follow, including, e.g., real-time traffic monitoring, e-tolling, and liability attribution in case of accidents. Locating and tracking vehicles has however strong implications in terms of security and user privacy. On the one hand, there should be a mean for an authority to verify the correctness of positioning information announced by a vehicle, so as to identify potentially misbehaving cars. On the other, public disclosure of identity and position of drivers should be avoided, so as not to jeopardize user privacy. In this chapter, we address such issues by introducing A-VIP, a secure, privacy-preserving framework for continuous tracking of vehicles. A-VIP leverages anonymous position beacons from vehicles, and the cooperation of nearby cars collecting and reporting the beacons they hear. Such information allows a location authority to verify the positions announced by vehicles, or to infer the actual ones if needed, without resorting to computationally expensive asymmetric cryptography. We assess the effectiveness of A-VIP via realistic simulation and experimental testbeds.

The content of this chapter is organized as follows. Sec.2.1 introduces the secure positioning problem while related works are presented in Sec.2.2, We describe the system scenario and communication protocol in Sec.2.3. Sec 2.4 details the location verification and inference procedures, while Sec.2.5 discusses the resilience of A-VIP to attacks by adversarial vehicles. The performance of A-VIP in both simulated and real-road environments is shown in Sec.2.6 Finally, Sec.2.7 concludes the chapter.

2.1 Introduction

Borrowing from a well-established communication pattern in wireless LANs, vehicular networks have adopted the term *beaconing* to indicate the periodic broadcasting of messages to neighboring vehicles or road-side units (RSUs). These messages, defined, e.g., in the SAE J2735 specifications, can be used for safety purposes as well as for cooperative awareness. The information they carry (e.g., vehicle ID, timestamps and location information) may be secured through the use of an on-board tamper-proof Hardware Security Module (HSM) as well as signatures, cryptography and certificates [1].

Secure beacons for vehicle position identification and tracking are needed in a number of scenarios where vehicle position accountability is a requirement in order to provide services to the community or to drivers. Secure reporting of vehicle location can substantiate drivers' claims in case of accidents. At the same time, secure location verification by authorities can provide accountability for those involved.

However, ensuring secure positioning must cope with three major problems, concerning (i) users' privacy, (ii) computational costs of security and (iii) the system trust on user correctness. As for the first aspect, when not strictly required, public disclosure of the vehicle identity to all receiving devices in the proximity of a beaconer is an issue. Vehicles can be tracked, jeopardizing drivers' privacy and requiring complex pseudonym management [2]. Thus, there is a need for separating secure position identification by authorities and the possibility of undesirable user tracking by peers in the vehicular network. As for the second aspect, standard security mechanisms based on, e.g., asymmetric cryptography, induce significant protocol overhead and computational complexity. In fact, their use is recommended to be largely dependent on the applications and circumstances, and avoided whenever possible [3]. Finally, basic solutions cannot guarantee the correctness of the location information provided by a user who owns the required cryptographic material, but has a malfunctioning GPS receiver or can tamper with GPS data before they are input to the HSM.

In this chapter, we address the issues above by proposing A-VIP (Anonymous Verification and Inference of Positions), a framework that, unlike previous work:

(i) allows a trusted authority to securely collect and verify the positions claimed by vehicles without resorting to computationally expensive asymmetric cryptography – as is instead done in the IEEE 1609.2 standard [4];

(ii) in presence of unverified location claims, grants the authority the capability to infer the actual position of malfunctioning or misbehaving vehicles;

(iii) does so by safeguarding drivers' privacy with respect to other vehicles participating in the network, and without any requirement for uninterrupted radio coverage from roadside infrastructure.

To achieve such goals, A-VIP leverages *anonymous position beacons* from vehicles, which prevent overhearing nodes from identifying or tracking their source, but still allow authorized third parties – sharing secret information with the beaconing vehicles – to perform such operations. Then, an authenticated reciprocal beacon reporting scheme grants an authority the possibility to verify the locations claimed by vehicles and infer unverified positions by efficiently solving an optimization problem.

2.2 Related work

When considering the problem of location verification and inference, an extensive literature can be found in the domain of wireless sensor networks, including among others [5–8]. However, it is commonly acknowledged that solutions designed for static nodes do not fit the highly mobile vehicular scenarios we target.

Specific to the vehicular environment, many works have focused on pure ad-hoc vehicular network environments where no infrastructure or central authorities are considered. There, the aim is the verification of reciprocal position information so as to secure cooperative awareness and multihop routing. To that end, a number of different solutions have been proposed that leverage diverse metrics, including the distance among nodes and their relative mobility [9, 10], the Time-of-Flight (ToF) distance bounding [11] and ranging [12], the Received Signal Strength (RSS) within a two-hop neighborhood [13], or the presence of Non Line-of-sight (NLOS) conditions [14]. There have also been proposals to use dedicated hardware, such as multiple directional antennas [15], or original data structures, such as trusted routing tables [16].

With A-VIP, we take a different approach, considering the problem of vehicle position verification from the viewpoint of a trusted authority that collects car-generated location information through a roadside or cellular infrastructure. To the best of our knowledge, ours is the first attempt at designing a position verification system that considers such a perspective – which, although less visionary than the ad-hoc one, is more consistent with that expected to be the most viable architecture for secure vehicular communication systems [17]. Thus, our optimization problem formulation and the resulting centralized solution are hardly comparable with the techniques presented in the papers mentioned before.

We also remark that A-VIP does not only allow for position verification, but also addresses the problem of inferring the location of untrusted cars, and does so by granting user anonymity. Solutions have been proposed that specifically tackle the latter problems individually, such as [18, 19], but ours is the first work to present a comprehensive framework for secure, privacy-preserving localization.

Finally, our work is the first to experimentally evaluate the performance of a position verification and inference system through real-world testbeds.

2.3 Anonymous positioning procedure

We consider a WAVE-based vehicular network composed of vehicles communicating with each other and, occasionally, with RSUs. No assumption is made on the deployment of RSUs, so vehicles may travel along road segments where no RSU coverage is available. Vehicles may have also a 3G/LTE radio interface, through which they can access the cellular network that fully covers the road topology. Both RSUs and cellular base stations allow vehicles to contact a Location Authority (LA), which is in charge of collecting location claims, verifying them and inferring the actual positions of vehicles deemed to announce incorrect locations.

Vehicles are equipped with GPS, thus, unless otherwise specified, they know their own position and share a common time reference. Each vehicle owns cryptographic material, i.e., a certified identity and a long-term secret key, used to establish a secure channel with the LA at any time, through either an RSU or the cellular infrastructure. Solutions already exist for the distribution and management of long-term pairwise keys in vehicular environments (see, e.g., [4, 17]), and their discussion is out of the scope of this chapter.

Vehicles that comply with the A-VIP mechanism are defined as *correct*, while the others may be: (i) *faulty*, that is, they follow the protocol but provide incorrect information due to, e.g., GPS malfunctioning, or (ii) *adversarial*, i.e., their aim is to announce a fake position and have it verified, so as to obtain some advantage, discredit nearby users, or disrupt the A-VIP operation. To that end, adversarial nodes can either deviate from the A-VIP communication protocol procedure, or comply with it but inject false information. In this work, we consider internal adversaries, more challenging than external ones, as they own the cryptographic material to participate in the protocol. We consider however that adversaries are unable to forge messages on behalf of other nodes whose keys they do not have. Adversaries may be further distinguished as independent or colluding: in this chapter we focus on the former. However, in our analysis we also evaluate Sybil attacks, which can be viewed as a worst case of an attack carried out by colluding adversaries.

2.3.1 A-VIP goals

A-VIP aims at verifying the positions announced by correct vehicles while guaranteeing their privacy with respect to the other ones, and at detecting faulty or adversarial nodes while inferring their actual locations. Such goals are to be achieved with low computational complexity.

We stress that, barring the use of complex pseudonym management schemes [2], anonymity *cannot* be implemented simply by letting a vehicle issue beacons where its identity is encrypted with the long-term shared key it shares with the LA. Indeed, some form of plaintext ID, attached to the encrypted beacon, would be needed for the LA to recognize the beacon originator and choose the appropriate key to decrypt the remainder of the message. Clearly, the presence of a plaintext ID would jeopardize the vehicle

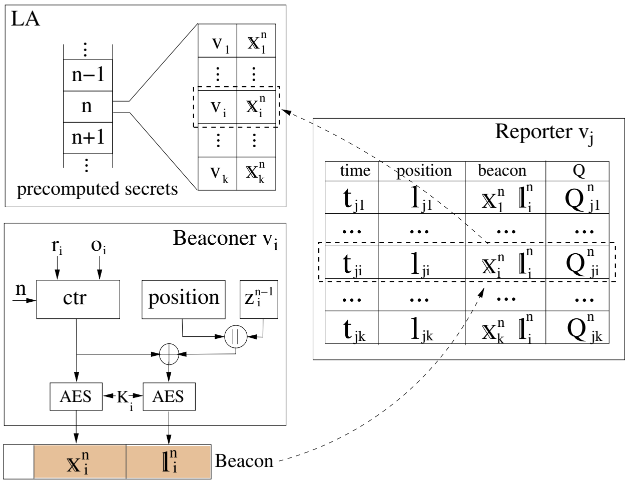


Figure 2.1. A-VIP procedures by beaconer, reporter and LA.

privacy, whose protection is one of our goals, allowing for overhearing and tracking by unauthorized receivers.

Additionally, we cannot just rely on encrypted cellular upload of the positioning information by vehicles. As a matter of fact, A-VIP is designed to be compatible with the cellular access infrastructure, but not to depend on it. More importantly, as discussed in Sec. 2.1, direct cellular upload does not allow for a verification of the location claimed by vehicles, which is instead part of the goals of A-VIP.

2.3.2 Communication procedures

The procedures in the A-VIP protocol are described below, while a schematic overview is shown in Fig. 2.1.

Registration. The registration procedure takes place every time a vehicle is started, and is repeated after a registration validity time has expired. It is performed over the secure channel between the vehicle and the LA, established with the long-term key via the RSU infrastructure, if available, or through 3G/LTE, otherwise.

Let us assume that a generic vehicle v_i sends a registration request at time instant $t_{i,0}$. The LA records such an instant and returns to the vehicle a registration triplet (K_i, r_i, o_i) where K_i is a short-term 128-bit AES symmetric key, and r_i, o_i are random integers. The triplet is used to compute a time-dependent secret $\mathbb{x}_i(t)$, shared between the vehicle and the LA. As detailed later, when sent by v_i to the LA, $\mathbb{x}_i(t)$ allows the LA to verify the freshness of a beacon transmission and the identity of its originator. In order to compute it, the two entities initialize a counter to r_i and increment it by o_i every τ_b seconds, e.g., at every beacon transmission. The updated counter is then encrypted with K_i using AES in counter mode (AES-CTR) [20]. Thus, in general, if $t_{i,0} + n\tau_b \leq t < t_{i,0} + (n+1)\tau_b$, then $\mathbb{x}_i(t) = E_{K_i}\{r_i + no_i\} = \mathbb{x}_i^n$. Note that both r_i and o_i can be picked at random since the chances of collision among $\mathbb{x}^n$ values, related to different vehicles at the same time, are negligible.

The LA is then in a position to precompute all the upcoming values of $\mathbb{x}_i^n$ for a period that depends on the registration validity time. Clearly, the amount of memory needed by the LA to store precomputed values of $\mathbb{x}_i^n$ depends on the average number of vehicles in the area served by the LA and the time interval for which the $\mathbb{x}_i^n$ values are precomputed. As an example, in the scenario described in Sec. 2.6.1 and assuming a validity time equal to 1 hour, the required storage space at the LA for all vehicles amounts to an easily accommodated 52 Mbytes of memory.

Anonymous beaconing. When traveling, all correct vehicles broadcast a beacon every τ_b, as foreseen by current standards. Beacon messages are broadcast by nature, and thus they are subject to undetected collisions over the wireless medium. While this may affect A-VIP operations, we stress that our approach can easily incorporate solutions to dynamically reduce channel contention such as [21]. For the purposes of this chapter, we assume that all beacon transmissions occur at a power level common to all correct vehicles and at the basic data rate. Also, we assume the beacon to be split into two parts: an encrypted one, for the purposes set forth in this chapter, and an unencrypted one, where plaintext content can be broadcast for such purposes as collision avoidance or cooperative awareness. We assume however that the beacon is anonymous, i.e., it does not include the vehicle identifier and it uses a fresh random MAC-layer address [17]. When not transmitting, the vehicle listens to the channel, overhearing beacons from other vehicles and collecting the information therein for later reporting to the LA.

The beacon content is assembled using the triplet assigned to a vehicle during the registration. Specifically, the n-th beacon issued by a vehicle v_i carries two pieces of information, as shown in the “Beaconer” box of Fig. 2.1:

(a) the time-dependent secret $\mathbb{x}_i^n$, which can be computed by v_i and by the LA, independently of each other;

(b) the encrypted current location announced by the vehicle $\mathbb{l}_i^n = E_{K_i}\{(\, l_i^n \,||\, z_i^{n-1}\,) \oplus (r_i + no_i)\}$, computed using the short-term pairwise key K_i from the triplet. The plaintext location l_i^n is concatenated with the one-bit flag z_i^{n-1} used to notify the LA whether the beacon issued at step $n-1$ was affected by a replay attack (as explained in Sec. 2.5). Such a string is then XOR’ed with the plaintext counter value $(r_i + no_i)$, to ensure freshness of the beacon positioning content and thwart partial-replay attacks (as also detailed in Sec. 2.5).

Reporting. When a beacon issued by a vehicle v_i is correctly received by a vehicle v_j, the latter is required to store the following entry in a *report table*, such as the one depicted in the “Reporter” box of Fig. 2.1:

- the time t_{ji} at which the beacon is received;
- its own position l_{ji} at the time the beacon was received;

- the secret $\mathbb{x}_i^n$ carried in the beacon;
- the encrypted position $\mathbb{l}_i^n$ of v_i carried in the beacon;
- an optional field Q_{ji}^n, indicating the received signal quality (e.g., the received signal power computed by the radio interface driver).

Every τ_r seconds (report interval), v_j generates a *report message* including the report table, populated with data collected from all newly overheard beacons. The report is transmitted to the LA, ensuring authentication and integrity through standard procedures. The transmission may occur via the RSU or via the cellular network if RSUs are scarce and real-time positioning is required. Additionally, multihop vehicle-to-vehicle communication can be exploited to reach nearby RSUs and, hence, speed up the report delivery.

We remark that vehicles normally act as both beaconers and reporters, and that the LA needs to receive only report messages, not the beacons broadcast by vehicles. Also, the communication procedures outlined above allow for a fully anonymous information exchange, preventing overhearing and thus ensuring user privacy.

2.4 Position verification and inference

When the LA receives reports from vehicles, it processes them so as to (i) determine the locations announced by cars in the system, (ii) verify such locations and (iii) infer the actual positions of vehicles deemed to have advertised an incorrect location.

Let the LA divide the road topology into discretized spatial *tiles*, whose set is denoted by $\mathcal{S}$. Also, let $\mathcal{V}$ be the set of vehicles that the LA has to verify. Upon receiving a report message from vehicle $v_j \in \mathcal{V}$, the LA processes one report table entry at a time, as follows:

- it extracts the time t_{ji} at which v_j received the beacon;
- for each $v_k \in \mathcal{V}$, it computes n such that $t_{k,0} + n\tau_b \leq t_{ji} < t_{k,0} + (n+1)\tau_b$, i.e., $n = \lfloor (t_{ji} - t_{k,0})/\tau_b \rfloor$, and it looks up the precomputed secret value $\mathbb{x}_k^n$ that matches the $\mathbb{x}_i^n$ in the report table entry (LA box in Fig. 2.1).

When a match is found, the LA identifies v_i as the vehicle that sent the beacon and retrieves the triplet associated to it [1]. Then, the LA performs the following actions:

(1) it decrypts the $\mathbb{l}_i^n$ field entered by v_i in the beacon reported by v_j, extracting the announced position l_i^n and the flag z_i^{n-1};

(2) if the z_i^{n-1} flag is set, it discards the entry;

[1] If no match for $\mathbb{x}_i^n$ is found, the report entry is discarded. This may happen as a result of, e.g., replay attacks or node malfunctioning.

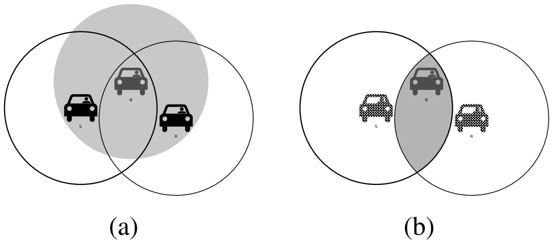


Figure 2.2. Q-unaware approach: v_i's beacon is reported to the LA by v_k and v_j. The shaded circle in (a) represents the transmission range of v_i, while the white circles denote the receiving range of v_k and v_j. The shaded area in (b) represents the possible locations resulting from the combination of the two reports.

(3) otherwise, if z_i^{n-1} is unset, it stores n, the position l_i^n included in the beacon by v_i and the position l_j^n announced by v_j in the report table entry. If present, the LA also stores the signal quality indicator, Q_{ji}^n, that v_j measured on the beacon received from v_i.

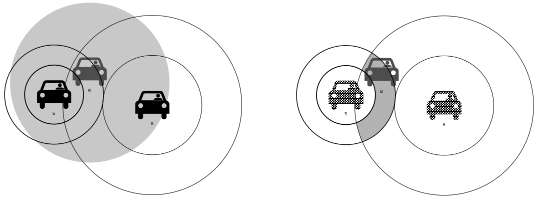
The LA leverages the information extracted from the report table entry to identify the possible tiles corresponding to a vehicle position, thus verifying the location claim and possibly determining the actual vehicle location in case of mismatch. Clearly, the same beacon, characterized by a single $\mathbb{x}_i^n$, may be reported by multiple vehicles traveling in the proximity of v_i when the latter broadcasted it. If so, the LA can obtain a better estimation of the beaconer position by combining the received reports. The steps required by this operation are detailed in the rest of this section, where, for sake of clarity, we drop the time notation and assume that all measures refer to the same beacon broadcast interval n.

2.4.1 Cooperative position identification

Depending on whether the quality indicator Q_{ji} is included in the report or not, the LA can adopt two different approaches to identify the tiles corresponding to the position of the beaconer v_i. The two techniques, named Q-unaware and Q-aware, operate as follows.

The Q-unaware approach. In this case, the LA does not have any information on the quality level with which the beacon signal was received at the reporter. Thus, for each pair of tiles $(s,t) \in \mathcal{S}^2$, it can assume a simple 0-1 propagation model to state whether a beacon sent from tile s can be heard in t or not, i.e., $h(s,t) : \mathcal{S}^2 \to \{0,1\}$. We remark that any methodology could be used to determine the vehicle radio range: from a simple unit disc model, displayed in Fig. 2.2, to a signal map drawn from real-world measurements [22].

Given that v_j located in tile $t \in \mathcal{S}$ received the beacon from v_i, then the LA can identify a set of tiles, $\mathcal{S}_i^{(j)} \subseteq \mathcal{S}$, where the beaconer could have been. Due to the simple propagation model, all tiles in $\mathcal{S}_i^{(j)}$ correspond to the right location with equal probability. Thus, the probability that the beaconer was in tile s when v_j heard its beacon is given by:



(a) (b)

Figure 2.3. Q-aware approach: the beacon from v_i is reported by v_k and v_j. The shaded area in (a) represents the transmission range of v_i. The annuluses denote the set of locations from which a beacon could be received by, respectively, v_k and v_j with the quality level indicated in their report. In (b), the intersection of the annuluses represents the possible positions of v_i.

$$p_{i,s}^{(j)} = \begin{cases} \frac{1}{|\mathcal{S}_i^{(j)}|} & \text{if } s \in \mathcal{S}_i^{(j)} \\ 0 & \text{otherwise} \,. \end{cases} \tag{2.1}$$

Note that, if multiple reports from correct nodes are available, the corresponding sets of tiles may intersect, as depicted in Fig. 2.2(b). In this case, the LA can obtain a better estimate of the true position of v_i. Considering the intersection translates into computing the following probability:

$$P_{i,s}^{(\mathcal{R}_i)} = \frac{\prod_{j:v_j \in \mathcal{R}_i} p_{i,s}^{(j)}}{\sum_{u \in \mathcal{S}} \prod_{j:v_j \in \mathcal{R}_i} p_{i,u}^{(j)}} \quad \forall s \in \mathcal{S} \,, \tag{2.2}$$

where $\mathcal{R}_i$ is the set of vehicles that reported v_i's beacon. We stress that $p_{i,s}^{(j)}$ represents the probability that v_i was in s while sending the beacon, computed taking only one report into account. $P_{i,s}^{(\mathcal{R}_i)}$ represents the same probability, yet computed by combining the information received from multiple correct reporters.

The Q-aware approach. When a report includes the Q_{ji} value related to a beacon reception, such information can be exploited to refine the position estimate of beaconer v_i. To do so, the LA needs an accurate model of the propagation conditions in the area where the broadcast transmission took place, including received signal quality information. Again, deterministic (e.g., ray-tracing), stochastic, or measurement-based models can be used: the Q-aware procedure does not change and is performed as follows.

Let the propagation model be a function $h(s, t, Q_{ji}) : \mathcal{S}^2 \times \mathbb{R} \to [0, 1]$ that, for any pair of tiles (s, t) and any signal quality value Q_{ji}, provides the probability $\mathbb{P}(R_t^{(j)} | B_s^{(i)}, Q_{ji})$ that a beacon sent by v_i from tile s can be received by $v_j \in \mathcal{R}_i$ in tile t, with the quality level Q_{ji} reported by v_j.

By applying Bayes' theorem, the LA can use such values to compute the probability $\mathbb{P}(B_s^{(i)} | R_t^{(j)}, Q_{ji})$ that the beaconer was in title s, given that the beacon was heard by v_j in tile t, with a quality level Q_{ji}. Specifically,

$$\begin{aligned} p_{i,s}^{(j)} &= \mathbb{P}(B_s^{(i)} | R_t^{(j)}, Q_{ji}) = \\ &= \frac{\mathbb{P}(R_t^{(j)} | B_s^{(i)}, Q_{ji}) \cdot \mathbb{P}(B_s^{(i)})}{\sum_{u \in \mathcal{S}} \mathbb{P}(R_t^{(j)} | B_u^{(i)}, Q_{ji}) \cdot \mathbb{P}(B_u^{(i)})} \end{aligned} \tag{2.3}$$

where $\mathbb{P}(B_x^{(i)})$, $x = s, u$, is the probability that the broadcasting vehicle v_i is in tile x at the time of transmission. This value may depend on the vehicle density and on the size of the considered area. We assume however a generic scenario where no such knowledge is available, and the probability is equally spread among all tiles, i.e., $\mathbb{P}(B_s^{(i)}) = 1/|\mathcal{S}|$ for any v_i and any tile in $s \in \mathcal{S}$.

Upon receiving multiple reports, the LA can again resort to (2.2) to combine the Q-aware probabilities computed as in (2.3). Then, it can determine the tiles the beaconer could have been at the moment of the broadcast, with the associated probabilities $P_{i,s}^{(\mathcal{R}_i)}$. Note that, unlike the Q-unaware case, such probabilities are now Q-dependent and provide better location estimates, assuming that the underlying signal quality model is accurate enough.

A simple example of the Q-aware approach is portrayed in Fig. 2.3, where two reporters, v_k and v_j, include different quality levels for a beacon received from v_i. For simplicity, in the figure we considered that the area corresponding to the value of Q, indicated by a reporter, maps onto an annulus comprised in its reception range. Then, the set of possible locations of the beaconer is given by the intersection of the two annuluses, i.e., the shaded area in Fig. 2.3(b).

2.4.2 Assessing the trustworthiness of vehicles

The technique used in the previous subsections to combine multiple reports assumes that all reports come from correct nodes. Unfortunately, as we discussed in Sec. 2.1, faulty or adversarial users may report fake position information and invalidate any attempt by the LA to estimate their own and other vehicles' positions. Thus, it is therefore essential to determine the *trustworthiness* of vehicles in order to tell apart faulty or adversarial nodes. A-VIP performs this task by leveraging the information contained in reports sent by vehicles. Hence, the trust attribution process described below is only performed, within each time step, for the subset $\overline{\mathcal{V}} \subseteq \mathcal{V}$ of vehicles that satisfy two conditions: (i) having transmitted a beacon and (ii) having had such beacon reported by others.

The trustworthiness probability of vehicle v_i at a generic time step, which we will refer to as $\gamma_i \in [0, 1]$, is determined by the LA through a three-phase process:

(1) the *location probability* $\Phi_{i,s}^{(\mathcal{R}_i)}$ that v_i is at any tile $s \in \mathcal{S}$ upon beacon transmission is computed by taking into account the (unknown) trustworthiness of vehicles in $\mathcal{R}_i$, i.e., those that reported the beacon sent by v_i;

(2) the location and trustworthiness probabilities of all vehicles in $\overline{\mathcal{V}}$ are combined into a global *consistency function*, χ, corresponding to the average number of vehicles that are correctly estimated to be in their declared location;

(3) the trustworthiness probability γ_i of each vehicle in $\overline{\mathcal{V}}$ is computed so that the consistency function χ is maximized.

Phase one above is achieved by letting the LA combine the information it received in the reports in a similar fashion as that of (2.2). This time, however, the unknowns represented by the trustworthiness of the vehicles participating in the process are integrated in the expression. Specifically, for each $s \in \mathcal{S}$, the LA evaluates the location probability $\Phi_{i,s}^{(\mathcal{R}_i)}$ that v_i was in tile s when sending the beacon, as:

$$\Phi_{i,s}^{(\mathcal{R}_i)} = \sum_{\mathcal{Z} \in \wp(\mathcal{R}_i)} \left(P_{i,s}^{(\mathcal{Z})} \prod_{j:v_j \in \mathcal{Z}} \gamma_j \prod_{k:v_k \in \mathcal{R}_i \setminus \mathcal{Z}} (1 - \gamma_k) \right) . \tag{2.4}$$

In (2.4), $\wp(\mathcal{R}_i)$ is the power set of $\mathcal{R}_i$, i.e., all possible subsets (proper and not) of reporters in $\mathcal{R}_i$. The terms $P_{i,s}^{(\mathcal{Z})}$ are calculated as in (2.2), using the probabilities $p_{i,s}^{(j)}$. Recall that the latter probabilities are computed using either (2.3) or (2.1), depending on whether the information on the Q-value is available or not. Also, we define $P_{i,s}^{(\emptyset)} = \mathbb{P}(B_s^{(i)})$. In words, the expression in (2.4) states that, if only the reporters in the subset $\mathcal{Z}$ are trustworthy, which happens with probability $\prod_{j:v_j \in \mathcal{Z}} \gamma_j \cdot \prod_{k:v_k \in \mathcal{R}_i \setminus \mathcal{Z}} (1 - \gamma_k)$, then the probability that the beaconer v_i was in s is obtained by considering the reports sent by such vehicles ($v_j \in \mathcal{Z}$) and neglecting the others ($v_k \in \mathcal{R}_i \setminus \mathcal{Z}$). Consistently, the term $P_{i,s}^{(\emptyset)}$ corresponds to the case where no trustworthy vehicle exists, hence the probability that v_i was in s is $\mathbb{P}(B_s^{(i)})$, which does not depend on any report. Finally, note that, if $\gamma_j = 1$ $\forall j : v_j \in \mathcal{R}_i$, the expression in (2.4) reduces to $P_{i,s}^{(\mathcal{R}_i)}$, i.e., to the probability associated to the intersection of all the reported beacon receptions, as illustrated in Sec. 2.4.1.

In the second phase, the LA defines the global *consistency function*, χ, as the (expected) number of correctly estimated positions for vehicles in $\overline{\mathcal{V}}$:

$$\chi = \sum_{i:v_i \in \overline{\mathcal{V}}} \left(\gamma_i \Phi_{i,l_i}^{(\mathcal{R}_i)} + (1 - \gamma_i) \sum_{s \in \mathcal{S}} \Phi_{i,s}^{(\mathcal{R}_i)} \right) . \tag{2.5}$$

For each vehicle v_i, the first term of the sum in (2.5) corresponds to the case where v_i is correct (which happens with probability γ_i), and it represents the probability that v_i was in the tile including the position l_i that it announced in its beacon. The second term, instead, corresponds to the case where v_i cannot be trusted (which happens with probability $1-\gamma_i$) and it accounts for the probability that v_i could have been in any of the possible tiles, $s \in \mathcal{S}$. Note that the expression in (2.5) has the following interesting property: when all vehicles are trustworthy, i.e., $\gamma_i = 1$ $\forall i$, it reduces to $\chi = \sum_{i:v_i \in \overline{\mathcal{V}}} P_{i,l_i}^{(\mathcal{R}_i)} \leq |\overline{\mathcal{V}}|$. In this

Algorithm 1 Identifying the set of trustworthy vehicles.

```
Require: $\gamma_i$, $\forall v_i \in \overline{\mathcal{V}}$
$\mathcal{T} \leftarrow \emptyset$
$\mathcal{T}' \leftarrow \emptyset$
repeat
    $\mathcal{T} \leftarrow \mathcal{T}'$
    $v_i \leftarrow \arg\max_{h:v_h \in \overline{\mathcal{V}} \setminus \mathcal{T}} \gamma_h$
    $\mathcal{T}' \leftarrow \mathcal{T} \cup \{v_i\}$
until $\left(\exists v_k\colon v_i \in \mathcal{R}_k \wedge \max_{\mathcal{S}} P_{k,s}^{(\mathcal{T}_k')} = 0\right) \vee \mathcal{T}' = \overline{\mathcal{V}}$
return $\mathcal{T}$
```

case, χ is a measure of the accuracy of the estimation based on the cooperative position identification described in Sec. 2.4.1, which correctly increases as χ approaches $|\overline{\mathcal{V}}|$.

In phase three, the LA determines the trustworthiness γ_i of each vehicle in $\overline{\mathcal{V}}$ by solving following problem:

$$\begin{array}{ll}
\max & \sum_{i:v_i \in \overline{\mathcal{V}}} \left(\gamma_i \Phi_{i,l_i}^{(\mathcal{R}_i)} + \delta_i \sum_{s \in \mathcal{S}} \Phi_{i,s}^{(\mathcal{R}_i)}\right) \\
s.t. & \Phi_{i,s}^{(\mathcal{R}_i)} = \sum_{\mathcal{Z} \in \wp(\mathcal{R}_i)} \left(P_{i,s}^{(\mathcal{Z})} \prod_{j:v_j \in \mathcal{Z}} \gamma_j \prod_{k:v_k \in \mathcal{R}_i \setminus \mathcal{Z}} \delta_i\right) \\
 & 0 \leq \gamma_i \leq 1 \quad 0 \leq \delta_i \leq 1 \quad \gamma_i + \delta_i = 1\,.
\end{array}$$

In the problem above, the objective imposes to maximize the consistency χ. The first constraint enforces the definition of Φ, and is equivalent to (2.4). The second constraint ensures that γ values are between 0 and 1. The last two constraints introduce a set of auxiliary variables $\delta_i = 1 - \gamma_i$. By introducing these variables in the problem, we make the objective *posynomial*[2]. Posynomial problems can be reduced to a convex form and thus maximized in polynomial time [23].

2.4.3 Detecting fake identities

The above mechanism is based on the consistency among the reported positions of all vehicles. However, it may not be sufficient against Sybil attackers, which control multiple identities and use them to consistently report false positions. To this end, we put in place a specific mechanism to uncover fake identities, named *buddy detection*.

The key idea is fairly simple: if two (or more) vehicles *consistently* appear to be colocated, there is something suspicious. More formally, we say that two vehicles v_i and v_j consistently appear to be together if the sets $\mathcal{R}_i^k$ and $\mathcal{R}_j^k$ overlap for more than a

[2] A posynomial is a function of the form $f(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} \cdots x_n^{a_{nk}}$ where all x_i and coefficients c_k are positive real numbers, and the exponents a_{ik} are real.

fraction f, i.e., if

$$\sum_k |\mathcal{R}_i^k \cap \mathcal{R}_j^k| \geq f \cdot \min\left(\sum_k |\mathcal{R}_i^k|, \sum_k |\mathcal{R}_j^k|\right). \tag{2.6}$$

If the above condition is verified, we set $\gamma_i = \gamma_j = 0$, i.e., the LA declares both v_i and v_j as non-trustworthy. Note that the minimum in (2.6) implies that an attacker alternating several fake identities will still be detected.

The parameter f should be set taking into account two factors. First, some beacons and reports may be lost, and the overlap between the sets may be not complete. Second, an attacker controlling more than $f \cdot |\mathcal{R}_i|$ fake identities will not be detected. In Sec. 2.6.3, we show the effectiveness of the buddy detection mechanism in terms of false positives/negatives, already for low values of f.

2.4.4 Deriving the vehicle positions

As a result of the above procedure, the LA obtains the γ_i values for all vehicles in $\overline{\mathcal{V}}$, i.e., those that, during the time step under consideration, have broadcast a beacon that was then reported to the LA. Then, the LA can run Alg. 1 with the goal to determine the set $\mathcal{T} \subseteq \overline{\mathcal{V}}$ of vehicles deemed to be trustworthy.

At the outset, the LA initializes the set of trustworthy vehicles, $\mathcal{T}$, to the empty set (line 1). Then, at each step, it selects the vehicle v_i in $\overline{\mathcal{V}}$, but not in $\mathcal{T}$ yet, for which the probability to be trustworthy is the highest. It adds the vehicle to the set $\mathcal{T}'$, which is thus given by $\mathcal{T} \cup \{v_i\}$ (lines 5–6). If the information provided by v_i is consistent with the one provided by vehicles already in $\mathcal{T}$, then v_i is deemed trustworthy as well and included in $\mathcal{T}$.

More precisely, let us denote by $\mathcal{T}_k'$ the set of vehicles that have reported the beacon sent by v_k and are in set $\mathcal{T}'$. Then, for each vehicle v_k, for which v_i has reported a consistent information with respect to all other trustworthy reporters, there will be at least one tile s with non-zero probability, $P_{k,s}^{(\mathcal{T}_k')}$, associated to it (line 7). That is, the intersection among the location sets corresponding to the reports sent by the trustworthy vehicles and by v_i will not be empty. If this is the case, v_i is added to $\mathcal{T}$ (line 4). Otherwise, v_i, and all the reporters with a value of trustworthiness probability lower than γ_i, are tagged as non-trustworthy, and the procedure ends. Thus, the last computed set $\mathcal{T}$ includes all vehicles in $\overline{\mathcal{V}}$ that are deemed to be trustworthy by the LA.

After running Alg. 1, for each vehicle v_i in $\overline{\mathcal{V}}$, the LA determines the position set $\mathcal{L}_i \subseteq \mathcal{S}$ corresponding to the locations where the vehicle is deemed to be. In particular, if $v_i \in \mathcal{T}$, the LA considers the position $\mathcal{L}_i = \{l_i\}$, where l_i is the location declared by v_i in its beacon. Otherwise, the LA associates to v_i the set of possible locations $\mathcal{L}_i = \{s | P_{i,s}^{(\mathcal{T}_i)} > 0\}$. Note that, if $v_i \notin \mathcal{T}$ and no trustworthy vehicle has reported the beacon from v_i, i.e., $\mathcal{T}_i = \emptyset$, we have $P_{i,s}^{(\emptyset)} = 0 \; \forall s \in \mathcal{S}$, hence $\mathcal{L}_i = \emptyset$ and no position estimation is available at this time instant for v_i.

As a last step, the LA checks for all vehicles in $\overline{\mathcal{V}}$, for which $\mathcal{L}_i \neq \emptyset$, if their location was missing at some of the previous time instants. Let us consider the case where the LA finds missing position information for v_i at all time instants $k \in (n, n+T)$, while $\mathcal{L}_i^n$ and $\mathcal{L}_i^{n+T}$ are not empty. Then, the LA can estimate $\mathcal{L}_i^k$ as follows. For each pair of tiles $s \in \mathcal{L}_i^n$ and $u \in \mathcal{L}_i^{n+T}$, the LA exploits the empirical probability density function of the traveling time from s to u and verifies whether the probability that v_i was in the generic tile $t \in \mathcal{S}$ at time k is greater than 0. If so, t is added to the set $\mathcal{L}_i^k$. By doing so, the LA obtains a set of possible positions for v_i at k, along with their probabilities.

2.5 Attacks against A-VIP

Next, we discuss some possible attacks targeted at disrupting the position verification process described above. Our focus is on attacks orchestrated by single or multiple, albeit independent, adversaries. Colluding adversaries would indeed have the additional burden of continuous platooning to be successful, thus we consider these attacks as unpractical.

Transmit-power attack. The A-VIP position identification technique described in Sec. 2.4.1 relies on the fact that all correct vehicles transmit their beacons at the same power level. An attacker may maliciously increase or decrease its transmit power, thus affecting the Q-unaware and Q-aware approaches to the position verification and pretending to be closer or farther from the reporters than it actually is. However, while fooling a part of its neighbors, the attacker cannot help but appear inconsistent to the rest, since its announced position does not match the expected physical behavior of the transmission. Thus, A-VIP successfully detects transmit-power attacks, as shown in Sec. 2.6.3.

False location attack. It aims at pretending to be at a location different from the actual one, and at the same time at disrupting the operation of the beacon-reporting process. Specifically, the attack consists in a vehicle transmitting a beacon that includes the right time-dependent secret but a false position information. The announced position will not be coherent with the locations advertised by vehicles receiving the beacon in their reports, which may generate problems in the verification process. However, our results in Sec. 2.6.3 demonstrate that the A-VIP verification mechanism described in Sec. 2.4.2 is robust to this kind of attack.

Replay attack. Adversarial users replay beacons from correct vehicles. Although the attacker can retransmit a copy of the beacon, it cannot tamper with its content, as both the secret $\mathbb{x}_i^n$ and the beaconer position information are encrypted. We remark that encrypting the location l_i^n together with the current counter value, as described in Sec. 2.3.2, univocally ties l_i^n to $\mathbb{x}_i^n$. This prevents *partial replay* attacks, where the adversary only replays $\mathbb{x}_i^n$ and modifies the encrypted field $\mathbb{l}_i^n$ that contains the position information .

Still, by performing a full replay at locations other than those of the original broadcast, the attacker could induce the LA to tag correct nodes as faulty. In such cases, the timing of the replay is of the essence:

- in case of a replay attack occurring more than τ_b seconds after the legitimate beacon was broadcast, the LA will no longer be able to match the secret in the beacon with any precomputed secret during that time frame, and the report table entry will be ignored;
- in case of a replay attack occurring less than τ_b seconds after the legitimate beacon was broadcast, the replayed message will be reported multiple times by one or more witnesses. The LA will easily detect the presence of a duplicate, delayed entry in reports and reject it.

In the latter case, the trustworthiness of the original beacon sender would be tarnished. However, this sender can detect the replay of its own beacon and report the misdeed by setting the z_i^{n-1} bit in its following beacon, as introduced in Sec. 2.3.2. Recall that z_i^{n-1} can only be set by the original beacon sender, since it is encrypted along with the vehicle position within $\mathbb{I}_i^n$ and its freshness is ensured by the counter value. The LA will thus know that the beacon is invalid without affecting the vehicle credibility. The only result an attacker can achieve is thus to occasionally invalidate beacons from random vehicles. Jamming could yield the same effect with lower system complexity.

Wormhole attack. The replay attack can be combined with a wormhole attack, so that a full replay occurs less than τ_b seconds after the legitimate beacon was broadcast and in a different region (to avoid detection by the original sender). As a result, the replayed beacon will also be reported by witnesses other than those within the sender's communication range. In this case, the LA can detect the inconsistency by noting that the same beacon is heard by multiple witnesses farther apart than the nominal transmission range. The LA will thus be able to disregard both the original and replayed beacon entries without affecting the trustworthiness of the original sender. Additionally, the information collected at the LA allows for locating the wormhole ends, which have to be placed within the communication range of the reporters receiving the duplicate beacon. Since A-VIP implicitly counteracts wormhole attacks, we do not experimentally assess its robustness to them.

Phantom attack. An adversarial vehicle can run a phantom attack by never broadcasting beacons, nor reporting to the LA: such a vehicle would thus be completely transparent to the system. Its advantages are dubious. If, on the one hand, the attack could be used by a vehicle who is trying to escape liability after causing a car wreck, on the other, a phantom attacker falsely accused of being involved in an accident would be unable to prove it was elsewhere.

Additionally, phantom attacks could pose a threat to commercial applications such as e-toll enforcement. In such cases, the onboard devices are required to be tamper-resistant

HSMs integrating the antenna apparatus, so that no vehicle can successfully disappear from the network. For these reasons, countering this attack is out of the scope of A-VIP.

Teleport attack. An adversarial user could impair local transmissions of its own beacons and have a colluder broadcast those same beacons at a location other than that where it actually is. We refer to this as teleport attack, enabling the adversary to, e.g., deny liability in any accident in which she is involved by having her beacons broadcasted at a distant, safe location. The same discussion as for the phantom attack applies here as well, and an integrated-antenna HSM is required to prevent teleport attacks when the goal is determining liability. Thus, we do not assess the robustness of A-VIP to such attack.

Sybil attack. In a vehicular network, a Sybil attack is run by a single car that owns multiple identities and can thus impersonate several vehicles [24]. In the context of localization, a Sybil attacker can autonomously corroborate the fake position it advertises. More specifically, an adversarial user could avoid broadcasting beacons (i.e., perform a phantom attack), yet have multiple impersonated vehicles reciprocally (though falsely) report each other's beacons. Such attackers could thus claim any possible position.

We stress that Sybil attacks are difficult by nature. In a system of communicating vehicles, identities cannot be fabricated but they must have been legitimately obtained, hence successively stolen by the adversary. Such a hurdle makes the Sybil attack often infeasible, or only feasible for a short time before the identity theft is discovered.

Nonetheless, A-VIP is designed to cope this attack. The *buddy detection* procedure described in Sec. 2.4.3 aims precisely at discovering fake identities. We prove its effectiveness in Sec. 2.6.3.

2.6 Evaluation

Our evaluation of A-VIP is carried out in a simulated, yet realistic, vehicular scenario, as well as in real-world live testbeds. They are presented in Sec 2.6.1, along with the metric adopted to assess the quality of the A-VIP results. Our simulative and experimental study has two main goals. Firstly, in Sec. 2.6.2 we aim at acquiring a better understanding of the accuracy of the position estimation provided by our framework. Secondly, in Sec. 2.6.3, we focus on testing the A-VIP resilience to a range of different attacks.

2.6.1 Scenarios and metrics

Simulation scenario. Simulations are run on a map representing a 1×1.5 km^2 section of the urban area of Ingolstadt, Germany. The scenario models a total of 2792 vehicles over a period of about 1 hour, with a mean trip time of 5 minutes and 24 seconds and a mean road traffic density of 300 vehicles per km^2 [19]. The vehicular mobility is generated using the well-known Simulator of Urban MObility (SUMO), capable of reproducing real-world microscopic and macroscopic road traffic. The RF signal propagation is modeled through

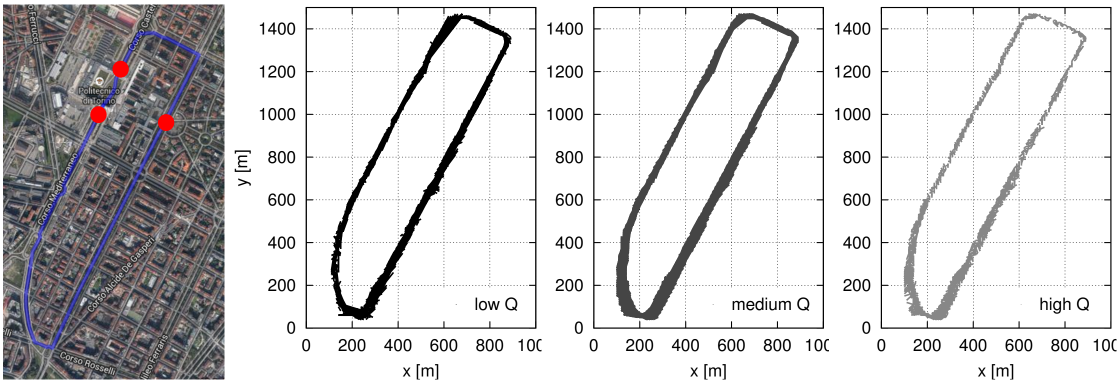


Figure 2.4. Urban testbed: route (left) and RF signal maps.

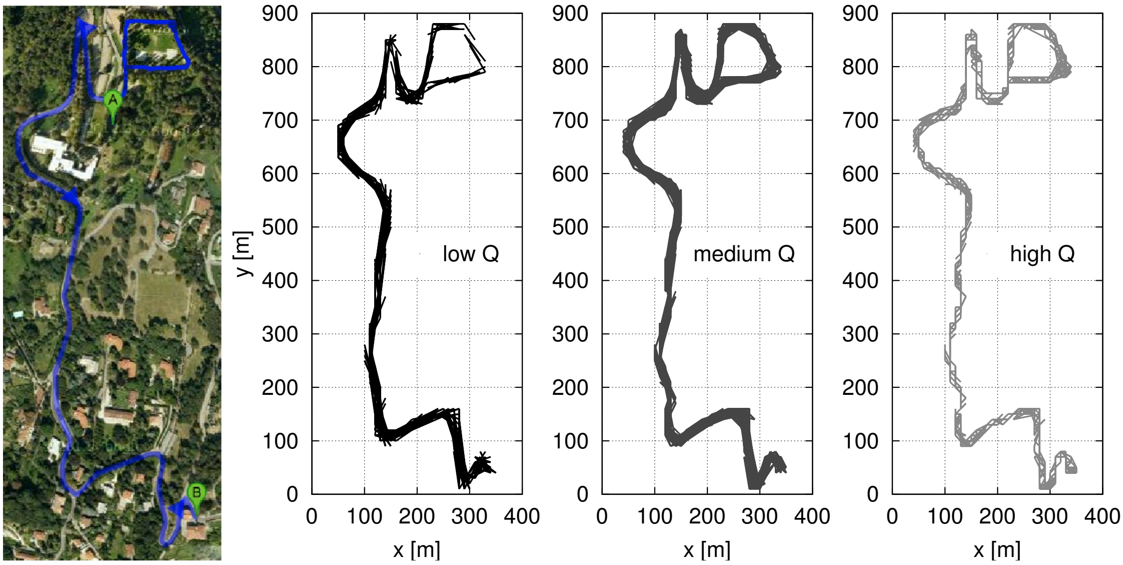


Figure 2.5. Suburban Testbed: route (left) and RF signal maps.

the 802.11p/DSRC radio shadowing technique proposed in [22]. The model accounts for buildings, and has been validated via real-world measurements in urban environments. As a result of the coupling of the vehicular mobility and signal propagation, we record vehicles to have an average of 69.83 neighbors, i.e., potential reporters per beacon in A-VIP. The availability of RF signal propagation information in the evaluation scenario lets us leverage the Q-aware technique presented in Sec. 2.4.1 to compute the probabilities $p_{i,s}^{(j)}$. We also model the 802.11p channel access and collisions that may take place among simultaneous transmissions.

While assessing the impact of malicious behavior, we consider a challenging scenario where 10% of the vehicles are randomly selected as adversaries, unless otherwise specified.

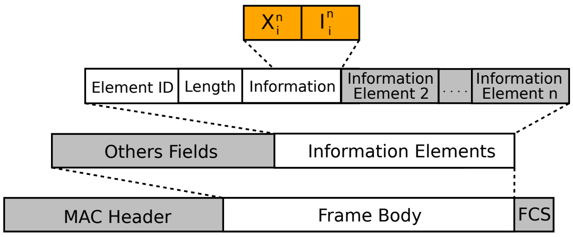


Figure 2.6. Testbed: A-VIP information ($\mathbb{x}_i^n$ and $\mathbb{l}_i^n$, in yellow) integration in the legacy IEEE 802.11 beacon.

Experimental testbed. We implemented the A-VIP protocol on commercial off-the-shelf hardware, in order to assess its position estimation capabilities and robustness to attacks in live testbeds. One of the testbeds is deployed in a urban area in the center of Turin, Italy and it consists of a 5-km road loop. A large portion of the road loop passes around the Politecnico di Torino, where we have installed 3 RSUs (802.11a APs) as shown in the left plot of Fig. 2.4. Using these devices we could cover a large portion of the selected area. All tests were completed during normal working days using 6 vehicles. The route follows arterial roadways, mostly straight and with 2-4 lanes in each direction, and features 12 traffic lights. The testbed vehicles did not follow any predefined formation but they tried to proceed within each other's radio range. The other testbed covers instead a 2-km road loop nearby Turin, Italy, and is composed of portions of a public road and of a private road in a suburban woodland area. The testbed comprises 5 vehicles, which follow the route portrayed in the left image of Fig. 2.5. A single RSU is deployed in the testbed, providing intermittent Internet access to up to five vehicles circulating at a time in the road loop within each other's range for most of the time.

From a technical viewpoint, the RSU and the vehicles are equipped with an Alix PC Engines motherboard, with an AMD Geode 500 MHz processor and one Ubiquiti Networks XtremeRange 5 radio IEEE 802.11a card. Vehicles carry one 5-dBi omnidirectional antenna on their rooftops, and are configured to transmit at an output power of 18 dBm. Finally, GPS receivers provide vehicle localization data.

A-VIP is implemented as a user-level application capable of transmitting and receiving beacons in ad hoc mode between vehicles, and sending reports to the RSU. Beacons are generated and broadcast every τ_b seconds, which is a configurable system parameter. The beaconing application exploits native IEEE 802.11 beacons, by including the information required for A-VIP operation, i.e., the secret $\mathbb{x}_i^n$ and the encrypted location information $\mathbb{l}_i^n$ of the emitting vehicle v_i. Such data is 32-byte long and is injected in the Vendor Specific Information Element (Vendor IE) field of the 802.11 beacon, as depicted in Fig. 2.6, without any need to edit the wireless card drivers. Upon reception of a new beacon from v_i, a vehicle v_j retrieves and stores the A-VIP data in its report table, along with a 4-byte reception time t_{ji}, a 8-byte current position l_{ji} and a 1-byte received signal quality indicator Q_{ji}^n.

In our live testbed, the propagation model used for the Q-aware computation of the probabilities $p_{i,s}^{(j)}$ is derived from experimental measurements. The corresponding propagation maps are depicted in the three right plots of Fig. 2.4 and Fig. 2.5, where, for clarity of presentation, the values of received signal power have been discretized into high (-40 to -60 dBm), medium (-60 to -80 dBm) and low (-80 to -95 dBm) signal quality bins. Consistently with intuition, we remark that shorter distances correspond to better signal quality.

In the case of attacks, we use 5 (6) vehicles in the suburban (urban) testbed and randomly select 2 of them as adversaries.

Location error. In order to express the quality of the LA estimates, we introduce a metric called *location error*. Formally, for the n-th beacon issued by a vehicle v_i whose actual position at the broadcast time is ℓ_i^n, the location error is defined as follows:

$$e_i^n = \sum_{s \in \mathcal{L}_i^n} P_{i,s}^{(\mathcal{T}_i)} d(\ell_i^n, s)\,. \tag{2.7}$$

Note that, in case of a vehicle v_i deemed trustworthy (i.e., $v_i \in \mathcal{T}$), $\mathcal{L}_i^n = \{l_i^n\}$, hence the location error represents the distance between its actual (ℓ_i^n) and declared (l_i^n) positions. Thus, in this case e_i^n = 0 if the vehicle is actually correct and its GPS is precise.

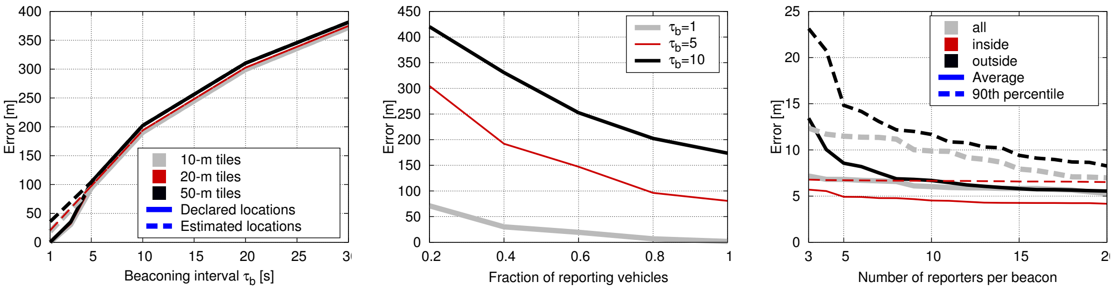
Instead, if v_i is not deemed trustworthy, the location error is the average of the distances between its actual location and the centers of the tiles representing its possible locations. The average is weighted by the probability that v_i is in each of such tiles $s \in \mathcal{L}_i^n$, according to trustworthy vehicles that received the n-th beacon from v_i (i.e., $P_{i,s}^{(\mathcal{T}_i)}$). In this case, e_i^n is the error that the LA incurs when trying to recover the actual position of an untrusted vehicle from the reports of trustworthy cars.

We stress that the second situation occurs also in the case of a vehicle v_i announcing its position with a frequency lower than that used by the LA to verify locations. Indeed, this forces the LA to estimate the location of v_i, as described in Sec. 2.4.4. In the following, we set the reporting periodicity τ_r equal to the beaconing interval τ_b, and consider that the LA verifies the position of all vehicles at every second, thus computing the location errors with a 1-Hz frequency.

2.6.2 A-VIP position estimation quality

We first assess the quality of the position estimation described in Sec. 2.4.1, both via simulation and our tow live testbeds, in absence of faulty or adversarial users. In this case, the uncertainty comes from the RF signal propagation, which is time-varying and may induce errors in the estimation process, possibly up to the point where some vehicles are tagged as untrustworthy. Additionally, beacons and reports may be lost due to channel errors, contributing to impair the verification by the LA.

Simulation results. Fig. 2.7 shows how the location error, averaged over all vehicles, is



(a) (b) (c)

Figure 2.7. Simulation: location error vs. (a) the beaconing interval τ_b, (b) the fraction of reporting vehicles, and (c) the number of reporters per beacon when $\tau_b = 10\,\text{s}$. Grey/red/black colors identify different tile sizes in (a) and reporter positions in (c); solid/dashed lines refer to declared/estimated locations in (a) and to average/90th percentile in (c).

affected by different system parameters in the simulation scenario.

In Fig. 2.7(a), we assume that all vehicles periodically report to the LA according to the procedure described in Sec. 2.3.2, and we evaluate the impact of the per-vehicle beacon transmission periodicity τ_b. Colors denote different spatial granularities (i.e., tile side lengths), ranging from 10 to 50 m. In these tests, A-VIP correctly tags all vehicles in the simulation as trustworthy, thus the framework does not generate any false positive. According to the location error definition in (2.7), the positions considered by the LA are those *declared* by the vehicles in their reports (solid lines in the plot). However, for the sake of completeness, we also report the error measured on positions *estimated* from other cars' reports through the Q-aware approach presented in Sec. 2.4.1 (dashed lines). This allows us to comment on the quality of the cooperative position identification.

We can first observe that τ_b has a dramatic impact on the location error, under all configurations. As the LA computes the location error every second, τ_b values larger than one second result in missing position information and trigger the estimation process of missing intermediate locations presented in Sec. 2.4.4. Clearly, the longer the τ_b, the more distant the position samples, leading to a less accurate estimation of intermediate locations. When this effect is marginal (τ_b is between 1 and 3 seconds), the error ranges in the order of the tile size.

When comparing the errors yielded by declared and estimated positions, they only differ when the absolute error is small, and, even then, the distance between the two is always in the order of the tile size. This allows us to conclude that reports are an efficient source of information to estimate the actual location of vehicles, and that we can trust the Q-aware cooperative position identification in case no positioning information is explicitly provided by a vehicle.

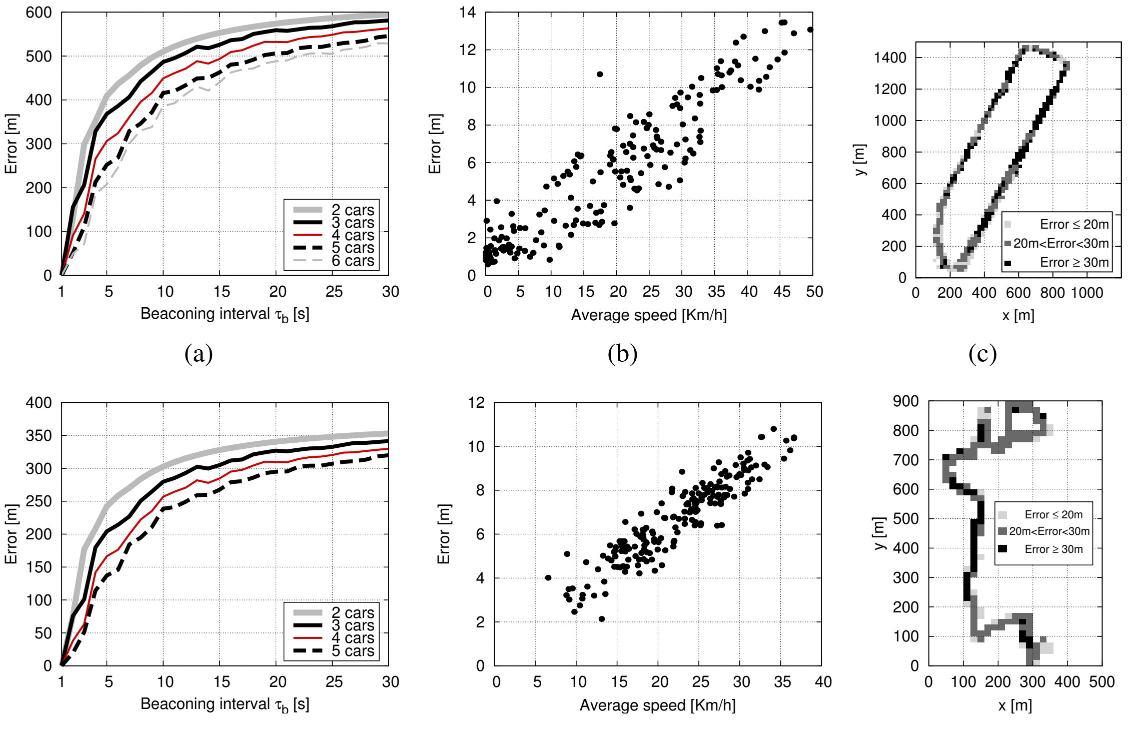
Fig. 2.7(b) shows the impact of the fraction of vehicles participating in the A-VIP reporting, when the tile size is set to 10 m. When all vehicles upload report messages, i.e., the fraction is equal to 1, the error corresponds to that measured in Fig. 2.7(a). However, as participation in reporting dwindles, i.e., for lower values on the x axis, the error tends to grow, slowly at first and faster later on. This effect, consistent through all values of τ_b, is due to the fact that, in presence of smaller sets of reporting vehicles, beacons are less likely to be received by any reporter. Non-reported beacons will never reach the LA. The latter will treat these situations as missing position information cases, thus estimating the location of vehicles whose beacons have not been reported as from Sec. 2.4.4. As discussed before, the estimate accuracy decreases as more beacons remain unreported. However, A-VIP appears robust to the lack of reporting, as errors become significant only if the majority of vehicles do not upload reports.

Fig. 2.7(c) shows a breakdown of the location error depending on the number and position of the reporters, for a tile size of 10 m and τ_b = 10 s. The overall average error (solid grey line) is not affected by the number of vehicles reporting the beacon, while the 90th percentile (dashed grey line) is. This implies that a low number of reporters can generate a few large error situations, namely, when beaconers fall outside the polygon whose vertices are the reporting vehicles (black solid and dashed lines). Conversely, if the beaconer is within such a polygon (red solid and dashed lines), the error remains low even for a small number of reporters. Indeed, in the latter case and when the reporters $v_j \in \mathcal{R}_i$ of a beacon from v_i are farther apart, the intersection of the sets $\mathcal{S}_i^{(j)}$ is smaller, as shown in Fig. 2.2, and the location estimate is much more accurate. We can conclude that large position estimation errors only concern vehicles whose beacons are reported by a few neighbors clustered on one side of the beaconer.

Testbed. A direct comparison of testbed and simulation results is not viable due to the very different settings that characterize the two environments, including the covered area, the number of cars and the propagation conditions. However, the qualitative behavior of the location error versus the beaconing interval τ_b observed in the experimental evaluation, in Fig. 2.8(a)and Fig.2.8(d), matches the simulated one in Fig. 2.7(a). Also in the testbed case, longer time intervals between back-to-back beacon transmissions determine higher location errors. The reason lies again in the difficulty of inferring intermediate locations between distant position samples.

We consider the match above as a positive result implying that real-world RF signal propagation, despite its complex and time-varying nature, does not induce dramatic errors in the A-VIP position estimation process. Similarly, real-world beacon and report message losses, measured to affect around 1% of messages in our experiments, do not impair the verification process at the LA. As a consequence, the experimental curves confirm that location errors in the order of the tile size (set to 10 m in these tests) are achievable in real settings if τ_b = 1 s.

Interestingly, the testbed results obtained with a varying number of vehicles (ranging from 1 to 5 (6) and mapping onto different lines in Fig. 2.8(d) (in Fig.2.8(a)) also validate



(a) (b) (c)

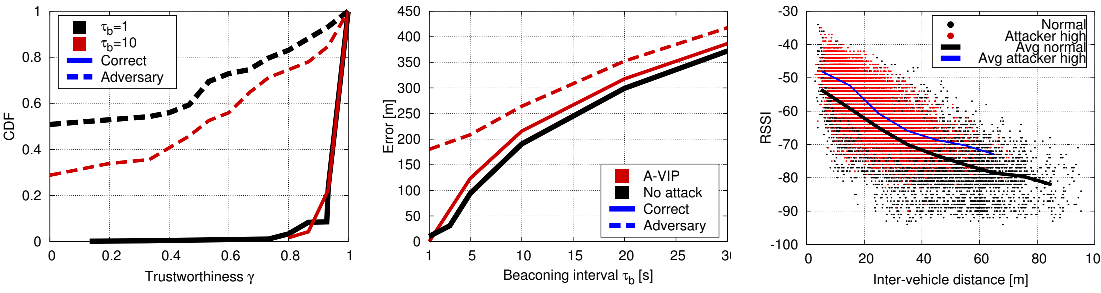
(d) (e) (f)

Figure 2.8. Urban (a,b,c) and Suburban (d,e,f) testbeds: location error versus (a,d) the beaconing interval, (b,e) the vehicle speed (middle), and (c,f) the vehicle geographical position. All plots consider a tile side of 10 m, and the latter two plots refer to the case where $\tau_b = 1$ s and $\tau_b = 10$ s, respectively.

the finding that the error is significantly reduced when the number of vehicles (hence reporters) increases up to the maximum.

From a quantitative viewpoint, however, the urban testbed leads to significantly larger errors, especially for high values of τ_b. This is due to the fact that figuring out intermediate positions in the urban environment is harder than in the suburban one. As a matter of fact, in the urban testbed the speed of vehicles is more heterogeneous due to the presence of traffic lights (forcing zero-speed standstill situations) and fast multi-lane roads (allowing vehicles to reach 50 km/h). Such a variability in the vehicle speed makes it more difficult to continuously track positions from a limited number of known locations.

Finally, the testbed also gave us the possibility to assess the impact that the vehicle speed and geographical position have on the Q-aware estimation accuracy. Fig. 2.8(b) and Fig. 2.8(e) present the relationship between the error on the estimated locations and the vehicle speed averaged over 30 s-intervals, when all vehicles are used and τ_b = 1 s. The vehicle speed is averaged so as to remove outliers due to GPS errors, hence make the plot more readable. The results show a strong correlation between the error on the



(a) (b) (c)

Figure 2.9. Transmit-power attack. (a) Simulation: distribution of trustworthiness probability γ. (b) Simulation: location error vs. beaconing interval τ_b. (c) Testbed: RSSI vs. communication distance for normal and attacker users. In (a) and (b) solid/dashed curves indicate correct/adversary nodes. Black/red colors identify different τ_b in (a), while they differentiate the "A-VIP" and the "No attack" cases in (b).

estimated locations and the traveling speed, explained by the fact that higher speeds introduce a higher variability in vehicle positions and make the estimation less accurate. The geographical analysis of the error, depicted in Fig. 2.8(c) and Fig.2.8(f) for τ_b = 10 s, is consistent with such a conclusion. Indeed, high errors (i.e., dark regions in the plot) are recorded on straight segments of the road when the speed is higher, while the lowest errors (i.e., light dots in the plot) are observed at slow-speed turns.

Results in Fig.2.8(b) and Fig.2.8(c) display the more heterogeneous and overall higher estimation error obtained in the urban environment. We impute such performance to two factors. First, messages are lost more frequently in the urban environment, with a 0.91 beacon reporting probability versus a value of 0.99 in the suburban scenario. This implies that locations have to be estimated using fewer reports. Second, the RF signal propagation tends to vary more significantly over time in the urban case, due to the presence of heavier road traffic and thus of a large number of mobile obstacles that impair the communication. As a result, the average propagation map used by the Q-aware approach is less reliable, leading to larger estimated location errors.

Overall, we conclude that the Q-aware position estimation run by A-VIP yields good performance as confirmed by both simulation and real-world experiments.

2.6.3 Robustness to attacks

Having assessed A-VIP position estimation reliability in presence of correct nodes only, we now evaluate the robustness of our solution (described in Secs. 2.4.2–2.4.4) to attacks led by adversarial nodes. We consider attacks for which tamper-proof HSM is not needed. Consistently with the findings in Sec. 2.6.2 and if not stated otherwise, we will assume τ_b = 1 s, a tile side of 10 m, and all vehicles participating in the reporting process.

Transmit-power attack. We consider the case of transmit-power attacks, described in

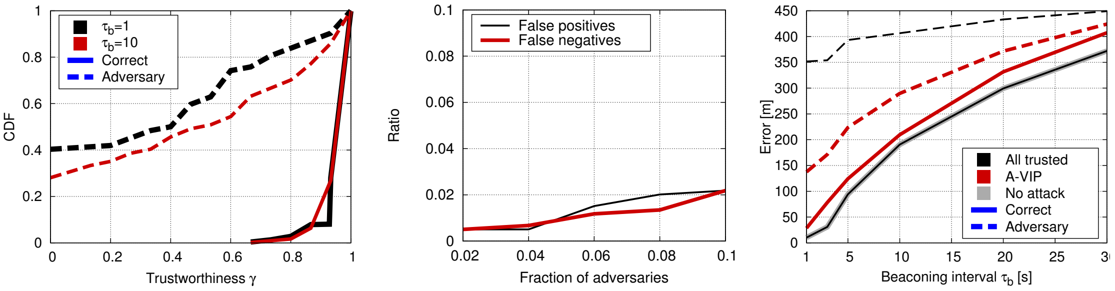
Sec. 2.5, and assess the A-VIP robustness via simulation. Fig. 2.9(a) shows the Cumulative Distribution Function (CDF) of the trustworthiness γ assigned by A-VIP to correct (solid line) and adversarial (dashed line) vehicles. Different colors map onto beaconing intervals τ_b of 1 and 10 seconds, respectively. The distributions clearly show how A-VIP can tell apart correct and misbehaving nodes, assigning high γ values (typically close to one) to the former, and much lower γ values (often near zero) to the latter. Notably, the percentage of adversarial nodes with high trustworthiness is small, a good performance in light of the large percentage of attackers (10% as mentioned in Sec. 2.6.1) and the fact that they are allowed significant freedom, being able to increase their transmit power by up to 20 dB (100 mW).

The value of γ allows the LA to decide which vehicles can be trusted and which cannot, as detailed in Sec. 2.4.4. The impact of such a classification on the accuracy of positions validated by the LA is portrayed in Fig. 2.9(b), in terms of the resulting location error. We can observe that correct vehicles are effectively identified by A-VIP: their errors with (solid red line, “A-VIP correct”) or without (solid black line, “No attack”) adversaries mostly overlap, and they do not suffer from the presence of transmit-power attackers. On the contrary, attackers are tracked down by A-VIP: their actual locations are estimated by the LA (dashed red line, “A-VIP adversary”) with fair accuracy.

We attempted running transmit-power attacks in the experimental testbeds as well. However, the wireless interface cards we employed only allow for very limited transmission power variations, of 5 dB at most. As shown in Fig. 2.9(c), such a small power offset is lost in the Received Signal Strength Indicator (RSSI) variability due to normal RF signal propagation phenomena. Therefore, the interface card limitations did not allow us to implement adversarial nodes that were substantially different from correct vehicles in terms of transmitted power.

False location attack. As described in Sec. 2.5, false location attacks are performed in our evaluation by announcing outdated positions along with consistent cryptographic material. We first study their effect in simulation, assuming that adversaries run false location attacks by including in their beacons the position they were at 10 s before.

Fig. 2.10(a) portrays the CDF of the trustworthiness probability γ for correct and adversarial vehicles, when τ_b is set to 1 s and 10 s. Also in this case, A-VIP reliably separates the two classes of nodes, assigning high γ values to the former and low γ values to the latter. Only 5% to 10% of the attackers are assigned a high trustworthiness, and this mainly occurs for adversarial nodes that did not move significantly during the 10-second delay of the attack. The proper classification of correct and adversarial behaviors leads to extremely low false positives (i.e., attackers tagged as trustworthy) and false negatives (i.e., correct nodes tagged as adversarial). Fig. 2.10(b) depicts false positives and false negatives as the fraction of adversaries varies between 2% and 10%. The results are obtained for $\tau_b = 10\,\text{s}$ and shows that both types of incorrect tagging are limited to less than 2% of vehicles in all cases.

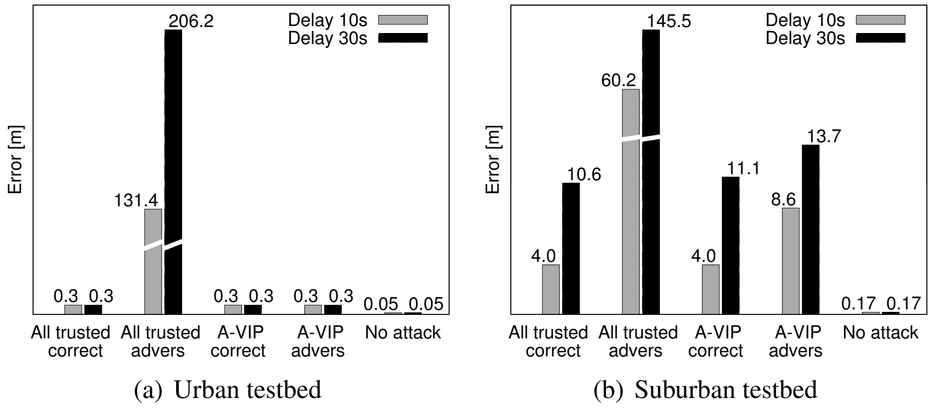


(a) (b) (c)

Figure 2.10. False location attack with 10 s-old positions. Simulation: (a) distribution of the trustworthiness probability γ, for a 10% of adversaries; (b) false positives and negatives vs. fraction of attackers when $\tau_b = 10$ s; (c) location error vs. beaconing interval τ_b, for a 10% of adversaries. In (a) and (c), solid/dashed curves indicate correct/adversary nodes. In (a) black/red colors identify different τ_b, while in (c) black/red/grey colors identify the "All trusted", "A-VIP" and "No attack" cases. In (c), in the "No attack" case, all vehicles are correct, thus only the "Correct" curve appears.

Similarly, the good classification performance of A-VIP leads to limited location errors, in Fig. 2.10(c). Once more, longer beaconing intervals result in higher location errors, for all cases. However, differences emerge when we focus on different curves. When all nodes are correct and we do not have any attack (solid grey line, "No attack"), we have the standard position estimation error already discussed in Sec. 2.6.2. By introducing a 10% of attackers (red lines), we observe that the location error of correct nodes (solid red line, "A-VIP correct") – properly identified by A-VIP as previously shown – does not change significantly. Positions announced by adversarial nodes are discarded: their actual locations, estimated through the cooperative Q-aware technique (dashed red line, "A-VIP adversary"), show again a fair accuracy. For completeness, the plot also shows the error values in the case where all nodes are trusted (black lines). We note that correct nodes (solid black line, "All trusted correct") exhibit the minimum error, since the position they advertise is trusted by the LA and matches their actual location. However, the LA believes also adversarial vehicles (dashed black line, "All trusted adversary"), which leads to a very high error in their position.

Fig. 2.11 shows the resilience of A-VIP to delay attacks through the measured location error for $\tau_b = 1$ s and a delay of 10 s or 30 s. Each pair of bars refers to one of the cases plotted in Fig. 2.10(c) and, for sake of readability, the numerical value of the error (expressed in meters) is reported on top of each bar. In Fig.2.11(b) we note that a delay of 10 s allows attackers to correctly announce positions that are up to 60 m away if the trustworthiness mechanism of A-VIP is not used (2^{nd} grey bar from the left, "All trusted adversary"). When such a mechanism is employed, the error (4^{th} grey bar from the



(a) Urban testbed

(b) Suburban testbed

Figure 2.11. False location attack. Testbed (2 attackers) : location error for $\tau_b = 1$ s and when the announced position is 10 s or 30 s old.

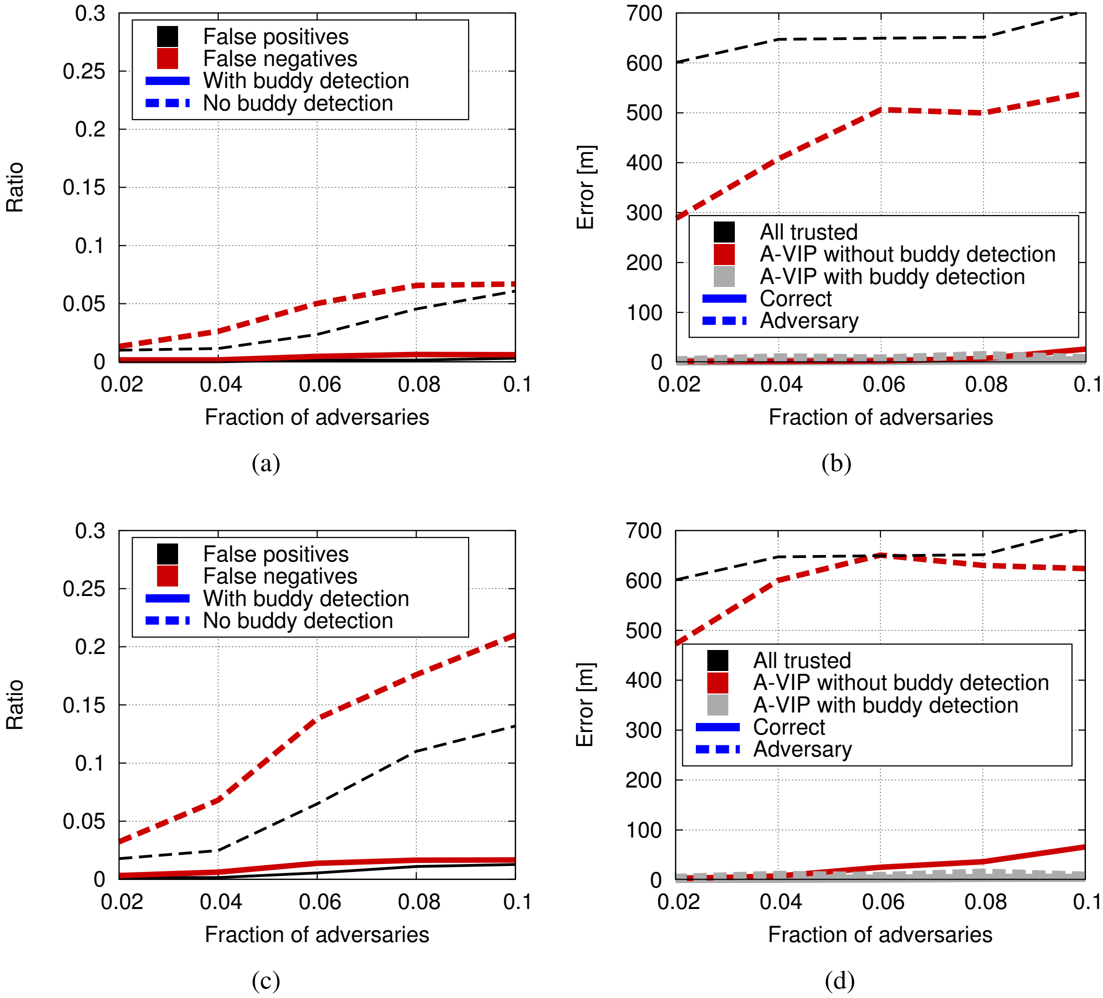
left, “A-VIP adversary”) becomes negligible, meaning that adversaries are correctly identified and their actual locations are estimated within the accuracy limits of the Q-aware technique.

Increasing the attack delay to 30 s leave more room for misbehavior. We observe in Fig.2.11(b) that without verification, adversarial vehicles can announce positions almost 150 m away from their actual location (2^{nd} black bar from the left, “All trusted adversary”). Instead, the trustworthiness mechanism of A-VIP still allows us to reliably tell apart incorrect nodes. Their true position is estimated with an error in the order of meters (4^{th} black bar from the left, “A-VIP adversary”) with respect to the case where attackers are absent (5^{th} black bar from the left, “No attack”).

The qualitative behavior of Fig.2.11(a) is the same as that observed in Fig.2.11(b). However, the attack space of adversarial vehicles seems larger in the urban environment, which leads to higher location errors when the positions they announce are always trusted by the LA (“All trusted adversary” bars). This is mainly due to the larger size of the road loop, which lets adversaries declare positions that are more distant from their actual ones.

As a final remark, when comparing Fig.2.11(a) to Fig.2.11(b), we note that the error achieved by A-VIP is lower in the urban environment. This is due to the different adversarial/correct vehicle ratios: 4 correct vehicles (out of the total 6) are present in the urban testbed, while only 3 correct vehicles (out of 5) participate in the suburban case.

Sybil attack. The last type of threat considered in our performance evaluation is the one brought about by the Sybil attack, detailed in Sec. 2.5. The limited number of vehicles available in our testbed does not allow us to experimentally evaluate A-VIP resilience to Sybil attacks. Therefore, in the following we resort to simulation. More precisely, we consider that 2% to 10% of the vehicles run Sybil attacks, and set the f parameter of the A-VIP buddy detection mechanism presented in Sec. 2.4.3 to 0.05. As shown by the results below, this low fraction is already sufficient to detect Sybil attacks in most cases.



(a) (b) (c) (d)

Figure 2.12. Sybil attack. Simulation: adversaries own one additional identity ((a), (b)) and three additional identities ((c), (d)). (a) and (c): False positives and negatives vs. fraction of attackers. (b) and (d): Location error vs. fraction of attackers. Curves are differentiated by the combination of color and line pattern. In (a) and (c), black/red colors indicate false positives/negatives, and solid/dashed lines the presence/absence of buddy detection. In (b) and (d), black/red/grey colors identify the cases where all nodes are trusted or A-VIP is used without/with buddy detection, while solid/dashed curves map onto correct/adversary nodes.

We first consider the case where adversarial nodes own cryptographic material granting them one additional identity. Indeed, this is barely sufficient for attackers to pass A-VIP verification even when no buddy detection is used. As shown in Fig. 2.12(a), false positives and negatives remain below 7% in all cases (dashed lines), as self-reporting via one additional identity is not sufficient to overcome the honest reporting by correct nodes. At any rate, the buddy detection mechanism (solid lines) greatly impairs the success probability of Sybil attacks as well as the chances of misclassifying correct vehicles. Fig. 2.12(b) shows that adversaries can modify their location by several hundred meters, when no verification is run (dashed black line) as well as when no buddy detection is used (dashed red line). By employing A-VIP with buddy detection, no maneuvering room is left to Sybil attackers: they are identified and their actual location is accurately estimated (dashed grey line).

Adversaries capable of impersonating three vehicles in addition to their actual identity make a great case for a solution such as A-VIP. Fig. 2.12(c) proves that three illicitly owned identities are sufficient to grant a Sybil attacker significant probability of success. Specifically, more than 20% of the attacks (dashed red line) are successful if no buddy detection is employed. As a side effect, up to 14% of correct vehicles (dashed black line) are at risk of being tagged as untrustworthy. The adoption of the buddy detection mechanism however limits both false negatives and positives to values below 2% in the worst case (solid lines).

The positive impact of the buddy detection naturally translates into much lower location errors, shown in Fig. 2.12(d). With three additional identities, Sybil attackers can modify their location by more than 600 m without being detected by the LA, if no A-VIP trustworthiness mechanism (dashed black line) or buddy detection (dashed red line) are employed. Conversely, A-VIP with buddy detection bounds the error to nearly zero values.

2.7 Conclusions

We presented A-VIP [25, 26], a lightweight privacy-preserving framework for verification and inference of vehicle positions by a Location Authority. A-VIP leverages computationally-inexpensive symmetric cryptography and reciprocal reporting of anonymized beacons by vehicles. Simulation and experiments in real-world testbeds have shown A-VIP capable of achieving its goals in both dense and sparse vehicular settings. Our results also show that A-VIP can effectively cope with several feasible attacks on a position verification system, with a small percentage of false positive/negatives.

Chapter 3

Content Downloading using UHF band

In this chapter, we examine the capacity of the frequency bands used in vehicular networks. This kind of networks are expected to support both safety and non-safety applications, through Dedicated Short Range Communications (DSRC) in 5-GHz bands. These channels, however, are of limited capacity and recent studies have highlighted their scarcity, in comparison to the broad range of services that are envisioned in vehicular networks. We therefore explore the benefit of using UHF bands for the transmission of control messages, so as to acquire more capacity. Specifically, we focus on content downloading, and design a protocol that leverages the UHF band for control messages and the high-throughput, 5-GHz bands for data delivery. We developed a testbed to quantify the performance of our approach, and show a 3x throughput gain in content delivery with respect to the case where only 5-GHz bands are used.

The rest of the chapter is organized as follows. Sec.3.1 introduces the Content Downloading problem and reviews previous work, while Sec. 3.2 describes the network scenario that has been implemented in our vehicular testbed. The protocol message exchange for content downloading, on both the UHF and the 5-GHz bands, is introduced in Sec. 3.3. Sec. 3.4 details the testbed set up, while the results derived from our measurement campaign are presented in Sec. 3.5. Sec. 3.6 concludes the chapter highlighting directions of future research.

3.1 Introduction

In the last few years, the interest in Intelligent Transportation Systems (ITS) has been steadily increasing, fueled by the need for safety and entertainment applications. Roadways can be made safer by letting vehicular users communicate road and traffic conditions, as well as position and velocity. Also, since a person often spends in the car between one and two hours per day, most newly-manufactured vehicles boast multimedia capabilities, which beg for advanced infotainment services (email/social network access,

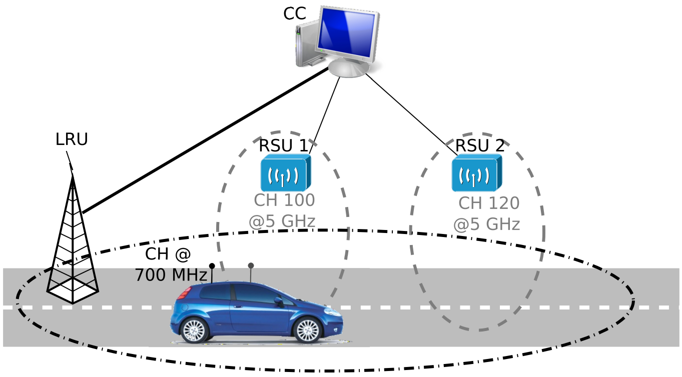


Figure 3.1. Abstract representation of the network scenario in the testbed.

newscasts, or local touristic clips).

In order to support such applications, frequency spectrum regulations have licensed 5.9 GHz band or dedicated short-range communication (DSRC) for ITS, while the IEEE 802.11p specifications have standardized vehicular communications over the allocated spectrum. In particular, 802.11p foresees a time division technique to let a vehicle equipped with one radio operate on the control and service channels. Also, it allocates one frequency channel for control message exchange and safety applications, and six channels for other services, all of them in the 5.9 GHz band.

Several research studies [27–31] suggest that, under high vehicle density or emergency situations, this bandwidth will likely be insufficient for either safety or non-safety services. To alleviate the spectrum demand, a number of solutions have been proposed. The work in [30] considers vehicles equipped with a DSRC and a UHF radio, and analytically derives the performance gain yielded by a cognitive radio system that allows the use of additional bands. Vehicles equipped with two radios are also considered in [31]. There, Kim et al. introduce a cognitive ad hoc network architecture to allow vehicle opportunistic access to WiFi channels, and present a cognitive routing protocol leveraging geographical location and sensed channel information. A simulation-based study is described in [29], where vehicles sense the UHF spectrum licensed to TV broadcasters and report their measurements to roadside processing units. The latter are in charge of identifying the frequencies available for widening the 802.11p control channel spectrum.

Motivated by the aforementioned observations and studies, in this chapter we focus on the use of low-frequency channels, namely, the UHF band at 700 MHz, in support to the channels at 5 GHz commonly used in ITS. Note that the use of UHF frequencies at 700 MHz for vehicular communications have been already attracting a great deal of interest, initially by the Japanese Transportation Institute and, more recently, by the FCC [32]. Indeed, low-frequency bands offer a significantly larger coverage than 5-GHz DSRC implementations. At an identical transmitter power, a low-frequency signal will have greater range than a high-frequency one, due to decreased free space attenuation and lower absorption by buildings and obstacles. The advantage of using the UHF bandwidth is that control information can be exchanged between vehicles and network infrastructure independently of the coverage provided by roadside radio devices. This translates into the possibility for the vehicles to interact with the ITS in advance, and get ready for the (high-throughput) connectivity with an upcoming roadside device. As a result, the time under coverage of the latter can be fully exploited for data transfers, thus reducing the experienced delay.

We develop a real vehicular testbed, with infrastructure nodes operating in the UHF band as well as roadside units (RSUs) operating at 5 GHz, while vehicles are equipped with both a UHF radio and a 5-GHz radio. An abstract representation of the network scenario in the testbed is depicted in Figure 3.1. We focus on content downloading applications, and design a message protocol that leverages the UHF channel for control information and 5-GHz service channels for data delivery. We then investigate, through our testbed, the benefits of such an approach.

To our knowledge, ours is one of the very few existing vehicular testbeds that exploit white spaces or UHF bands [33]. Furthermore, although in this work we limit our attention to the 700-MHz band and to content downloading, our study could be extended to the case of other low-frequency channels, like those used by Digital Mobile Radio (DMR), as well as to include control messages for the support of safety and other non-safety applications.

3.2 System scenario

As already hinted, our objective is to devise a fast reservation and scheduling mechanism that can support the transfer of content from a server to moving vehicles, exploiting (i) the longer transmission range of UHF communication to prefetch and schedule the delivery in suitable advance and (ii) the high transmission rates and extensive spacial reuse that communication in ISM bands can afford.

We focus on a roadside network consisting of the following actors, which are supposed to be deployed in an area supporting downloading services for vehicular users.

- *Central Controller (CC)* acting as coordinator between content requests from vehicles and scheduled downloads on the vehicular network.
- *RoadSide Units (RSUs)* providing short-range coverage to send downloaded content to passing vehicles; RSUs are supposed to be connected to the CC either through a wireline or through a wireless multihop connection (hereinafter referred to as “CC-RSU link”).
- *Long Range Units (LRUs)* base stations operating on UHF bands, used to collect movement updates and content requests from vehicles.

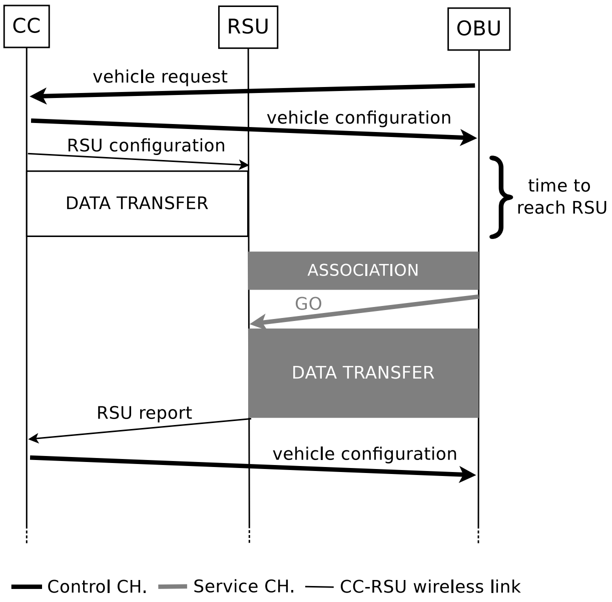


Figure 3.2. Protocol exchange among CC, RSU and OBU.

- *On-Board Units (OBUs)* used by vehicles to request content from the CC through the LRU and to download it from RSUs.

Additionally, we assume that each vehicle has a location device (e.g., a GPS) attached to its OBU and that the CC knows the locations of all RSUs under its control. The appropriate UHF channel is automatically supplied to the OBU by a radio map lookup service available on the OBU itself, possibly integrated with sensing channel information [29,31].

3.3 Protocol description

In the following, we describe the protocol interactions between the four actors. We will refer to this protocol as Locate-Fetch-Transfer (LFT), which summarizes the three tenets of its design.

Locate A *Vehicle Beat* message (similar to the CAM specified by ETSI) is broadcast by each OBU every second in the UHF band. This type of message carries geolocation data (latitude, longitude, direction and speed), along with additional (e.g., safety-related) information of a specific vehicle identified by its MAC address. The LRU receives the Vehicle beats and forwards the data to the CC, which then updates each vehicle information and its average speed (computed over the last ten seconds).

Fetch Content is requested by vehicle users through a URL (either provided by the user application, or manually inserted) pointing to an Internet resource or to data locally cached at the CC. The request, along with the MAC address of the requesting vehicle, is received by the LRU on the UHF band and forwarded to the CC. If no LRU is available (the request is not acknowledged), the request is periodically reissued until successful.

When the CC receives the request, it downloads a local copy of the requested content, if not already available. Then, it selects the closest (or the most suitable) RSU to the vehicle from its database and determines if, based on the vehicle position, its predicted movement, and the expected download rate, the content needs to be split across multiple RSUs along the vehicle path. After identifying the first RSU that the vehicle is likely to come across, the CC sends a *Vehicle Configuration* message toward the vehicle (through the LRU), detailing the network information, such as IP address, netmask, channel and BSSID, needed by the vehicle to connect to the selected RSU. Additionally, the CC partitions the content in one or more macroblocks (depending on the expected number of RSUs involved and on their coverages), and it sends an *RSU Caching* to the RSU nearest to the vehicles. Such a message includes retrieval information for the first macroblock, along with the vehicle ID (e.g., its MAC address). The selected RSU downloads the macroblock through the CC-RSU link, further partitions it into chunks, each of which can fit in a MAC frame, and waits for the vehicle arrival.

Transfer After the OBU of the vehicle has associated to the RSU using the information provided by the Vehicle Configuration message, it starts sending short *Go* messages to the RSU until the first chunk is received from the RSU. The chunks, sent over UDP and with the help of an application-level window protocol, are transferred until either the macroblock is complete, or the vehicle leaves the RSU coverage. When the transfer thus ends, the RSU returns an *RSU report* message to the CC, informing it of the final status of the transfer. The CC can then schedule the next RSU, possibly repartitioning the remaining data of the requested content among one or more macroblocks.

Figure 3.2 summarizes the LFT exchanges upon the issuing of a vehicle request, among three of the four actors: for the sake of simplicity, the communication between CC and OBU is always assumed to go through the LRU.

3.4 Testbed setup

To validate the framework in a real scenario, we have relied on our TV White Spaces (TVWS) testbed, in the Viù Valley, a mountain area in north-western Piedmont (Italy). There, we have selected a TV frequency that is allocated to a broadcaster, but that is not currently used. We have installed a bidirectional communication system based on the IEEE 802.11 specifications, as described below.

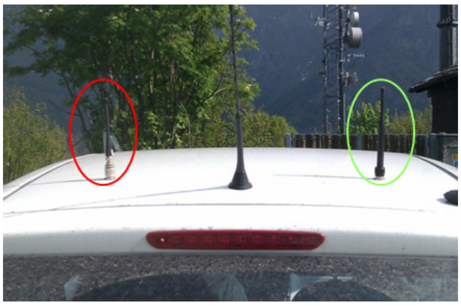


Figure 3.3. Antenna configuration on the testbed car: 700 MHz (red circle) and 5 GHz (green circle) antennas.

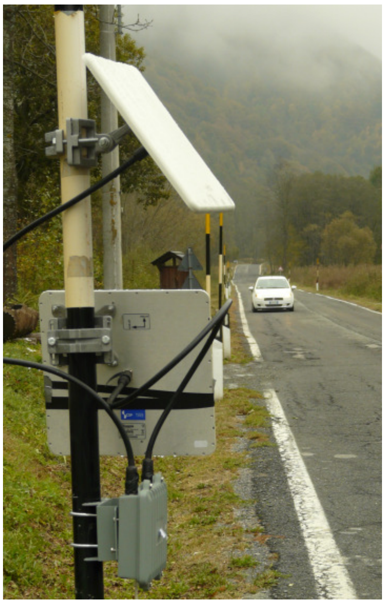


Figure 3.4. Antenna configuration at the RSU: antenna link with CC at the top, and with the OBU at the bottom.

3.4.1 Hardware configuration

The coverage of the valley is guaranteed through an LRU with the following characteristics:

- central frequency: 763 MHz, channel bandwidth: 5 MHz;
- antenna: 70-degree span, 9 dBi gain;
- transmission power: 18 dBm.

Vehicles are equipped with two omnidirectional antennas, at 5 GHz and 700 MHz, respectively; the former has 5 dBi gain and uses a transmit power of 22 dBm, the latter has 6 dBi gain and uses a transmit power of 18 dBm. On the vehicle, we have installed a device with two miniPCI cards, one for the 5-GHz network and the other for the channel at 700 MHz.

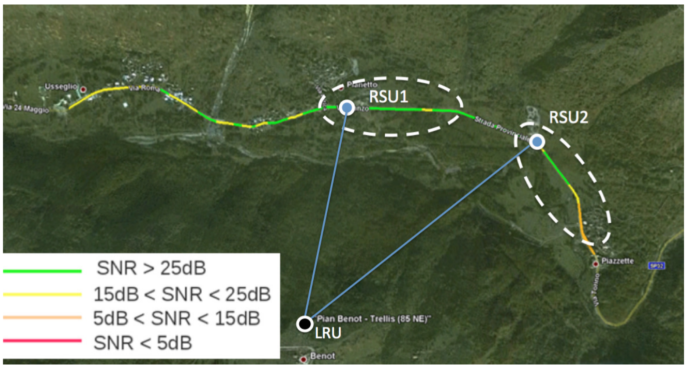


Figure 3.5. RSSI values at 700 MHz in the testbed road.

RSUs have been installed as APs operating at 5 GHz. As shown in Figure 3.4, RSUs are equipped with two directional antennas (30-degree span, 23 dBi gain) at 5 GHz, one of which is used to handle data exchange towards vehicles, while the other is used for the CC-RSU link.

As described in [34], we compared the performance of the UHF system along the road to that of a device operating at 5 GHz. The wireless cards used the MadWiFi driver, with the Minstrel rate adaptation algorithm activated. We evaluated the received signal strength index (RSSI) and throughput in both bands; the values of RSSI measured at 700 MHz, shown in Figure 3.5, are such that a good data rate is always guaranteed between vehicles and LRU.

Finally, the OBU aboard the vehicle has two IP addresses. The first one is used for exchange of signaling messages with the LRU and the CC, the second address is dynamically configured, as described above, and is used during the data downloading from the RSUs.

3.4.2 LFT parameters

We have implemented the LFT protocol described in Section 3.3, and tested it with one vehicle travelling on the stretch of road in Figure 3.5. We installed two RSUs, namely, RSU 1 and RSU 2, operating in the 5 GHz bands using channel 100 and 120, respectively. In order to represent the passage under several RSUs along the road, the vehicle proceeds as follows. It starts outside the coverage of RSU 1, then enters it and associates to the RSU. Next, the vehicle leaves the coverage of RSU 1 and, a little later, enters the coverage of RSU 2 and associates to it. Finally, it leaves RSU 2, turns around and drives back, repeating the procedure in reverse order. The vehicle is driven back and forth until the transfer is complete.

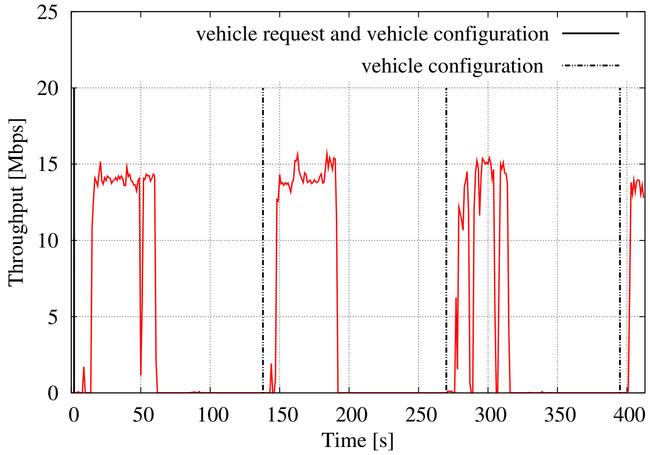


Figure 3.6. Instantaneous throughput vs. time, worst case; speed: 20 km/h.

We have run standalone tests where the vehicle uses LFT to request and download a 200-Mbyte file in each experiment. We then compared the attained performance to a case where a 2-Mb/s dummy download was activated from each RSU toward an additional OBU in a parked vehicle.

Each file was split into 168,810 chunks of 1296 byte each. In each test, the vehicle travels either at a steady speed of 20 km/h or 40 km/h. Given the coverage attained with the directional antennas at the RSUs, such vehicle speeds result in the scheduling at RSUs of 61,728 and 45,000 chunks, respectively. If no chunks were lost, 3 and 4 contacts with RSUs, respectively, would have been enough to complete the whole file transfer.

Table 3.1. Transfer summary at 20 km/h: worst case, 4 contacts (top), and best case, 3 contacts (bottom)

	RSU1	**RSU2**	**RSU2**	**RSU1**
From CC [chunks]	61728	61728	53053	13163
To OBU [chunks]	57869	57888	39890	13163
Coverage time [s]	71	69	63	26
Throughput [Mb/s]	8.45	8.70	6.56	5.25

	RSU1	**RSU2**	**RSU2**
From CC [chunks]	61728	61728	54334
To OBU [chunks]	54939	59537	54334
Coverage time [s]	85	78	63
Throughput [Mb/s]	6.70	7.91	8.94

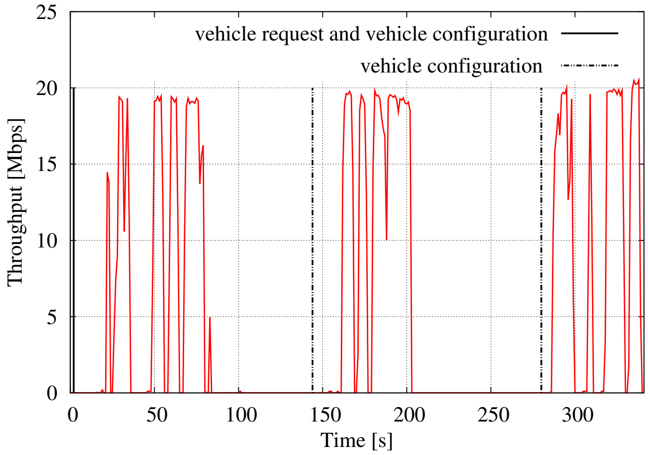


Figure 3.7. Instantaneous throughput vs. time, best case; speed: 20 km/h.

3.5 Experimental results

We now present and discuss the performance recorded on the previously described testbed. It is worth pointing out that, due to the duration of each test, not many of them could be run in the same environmental conditions (namely, over a few hours' span, meteorological conditions in a mountain valley are bound to change dramatically). Therefore, we could not provide a solid statistical averaging of metrics and we resorted to showing the worst-case results recorded across each type of experiment. Occasionally, we will also provide a set of best-case results for the sake of comparison.

The first set of results, showed in Figure 3.6, illustrates the instantaneous application-layer throughput recorded during each of the four contacts with the RSUs (each contact being separated by a vertical line), in the worst recorded case, driving at 20 km/h. Since the signal quality did not allow to exceed 15 Mb/s of throughput, one contact more than necessary had to occur for the entire file to be transferred. It is however to be noted that, in the best case (Figure 3.7), three contacts, as predicted, are enough to complete the transfer, thanks to a sustained throughput of almost 20 Mb/s.

Table 3.1 further details the transfers showing, respectively in each row, the macroblock size scheduled by the CC in anticipation of the upcoming contact[1]; the number of chunks actually downloaded by the OBU of the passing vehicle; the time under coverage[2] of the RSU and, finally, the average application-layer throughput while under coverage.

In the second set of results in Figure 3.8, the test was repeated while driving at a steady speed of 40 km/h. As expected, coverage under each RSU lasts for a shorter time, hence the greater number of contacts needed. Specifically, 9 and 7 contacts were needed in the

[1] For clarity, this quantity is expressed in number of chunks; recall, however, that the macroblock is divided into chunks only at the RSU.

[2] Here and in the following tables, for the last passage, this value represents the time under coverage till download is complete.

Table 3.2. Transfer summary: worst case, 40 km/h

	RSU1 Avg	**RSU2 Avg**	**Per-contact Avg**
From CC [chunks]	34681	44456	39025
To OBU [chunks]	16964	20997	18757
Coverage time [s]	36	36	36
Throughput [Mb/s]	4.94	5.96	5.40

Table 3.3. Transfer summary: 5 GHz only, worst case, 40 km/h

	RSU1 Avg	**RSU2 Avg**	**Per-contact Avg**
From CC [chunks]	31916	33117	32492
To OBU [chunks]	8487	4874	6752
Coverage time [s]	48	39	44
Throughput [Mb/s]	1.85	1.29	1.61

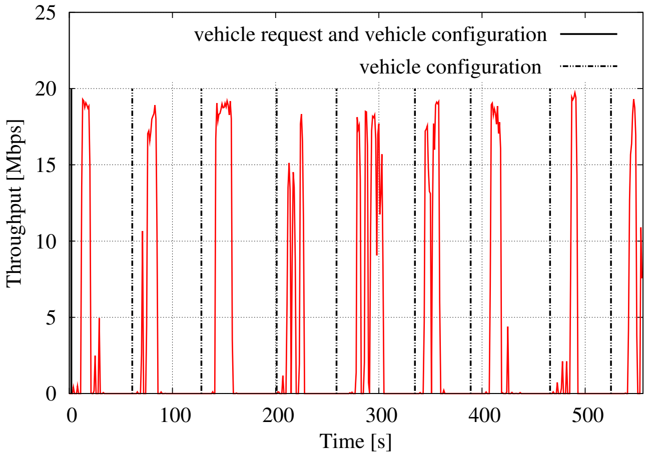


Figure 3.8. Instantaneous throughput vs. time, worst case; speed: 40 km/h.

worst and in the best case, respectively (only the former is shown in the figure). Table 3.2 refers to the worst case and reports the average per-contact number of chunks, scheduled and transferred, as well as the average per-contact coverage time and throughput.

The comparison of the previous case with the scenario including background traffic shows a performance degradation, which is mainly due to the additional flow carried on the channel at 5 GHz (results are omitted for brevity).

Finally, we asked ourselves what the impact of the locate-and-fetch components of the LFT protocol is, by comparing it to the case where only 5-GHz bands are used for both control and data messages. Note that, in these tests, the OBU sends the request through the RSU and the downloading of the remaining chunks is negotiated at the time of every contact on the 5 GHz channel. This results in a plain vanilla file transfer lacking the benefits of preemptive feeding of content to the upcoming RSU, as the vehicle has no means to send its updated position to the CC while travelling outside the RSU coverage. As shown in Table 3.3, the average number of chunks transferred to the OBU significantly decreases with respect to the case where the UHF channel is used, yielding a much lower throughput. Indeed, precious time under RSU coverage ends up being wasted in negotiating the download of the file leftover. Overall, the average throughput during content downloading resulted to be 1.61 Mb/s, implying a 3.4x gain yielded by the usage of the UHF band for the transmission of control messages.

3.6 Conclusion and future work

We defined a protocol for content downloading services, which leverages 5-GHz bands for data delivery and UHF bands for the transmission of control messages (aimed at locating vehicles and collecting requests) [35]. We assessed the benefits of exploiting UHF bands, providing much larger coverage than the 5-GHz frequencies, through a vehicular testbed. Our experimental results show that a 3x throughput gain in content delivery can be achieved with respect to the case where only 5-GHz bands are used. Such a gain is due to preemptive data feeding to the upcoming RSU and to the fact that RSU coverage time is fully exploited for high-throughput data transfers.

Future work will expand along the following directions: (i) experimental tests on more complex road topologies, (ii) implementation of a mechanism for dimensioning the content resource to be transferred to the RSU, based on the expected RSSI, (iii) implementation of fast authentication procedures as the vehicle moves in and out the coverage of different RSUs.

Chapter 4

GPS-Based Pedestrian Positioning

This chapter and the following one focus on outdoor pedestrian localization. We aim at designing and developing a framework able to detect potentially risky situations for pedestrian in outdoor environments. In this chapter we study the use of positioning techniques for sensing when pedestrians are at an increased risk of a traffic accident. Such sensing techniques could support augmented reality applications that increase pedestrian safety. We discuss requirements for pedestrian risk detection from rural to urban environments and consider algorithms relying on inertial and positioning sensors for distinguishing safe and unsafe walking locations. In conclusion, we study the limits of this approach through walking trials in different environments.

The rest of the chapter is organized as follows. Sec.4.1 introduces the pedestrian safety problem and related works. Sec.4.2 describes the scenarios we are dealing with and their challenges. The system overview is reported in Sec.4.3. Experimental activites in outdoor environments are explained in Sec.4.4. Sec.4.5 evaluates the perfomance of the positioning based approach while Sec.4.6 concludes the chapter.

4.1 Introduction

As mobile and wearable computing technologies pose increasing distractions, an important role of augmented reality technology can be to steer our attention back to the real world. This role can be particularly valuable in a pedestrian safety context. Traffic accidents with pedestrians still account for a significant number of injuries or fatalities and there is mounting evidence that mobile device distractions of pedestrians are exacerbating this problem. From 2000 to 2009, the United States saw more than 47,700 pedestrians deaths in traffic accidents and 688,000 pedestrians were injured [64]. Pedestrian deaths account for nearly 14% of all traffic fatalities [74]. According to a study, 26% of pedestrians text or email, 51% talk on the phone and 36% listen to music while crossing the street [72]. Distracted walking has also attracted significant media attention [39,46]. Mobile augmented reality technology could warn pedestrians when they are about to step into the street, using a variety of techniques. For example, they could underlay the composed text message of the user with a camera view of the street and a "Look Up!" notification. With a phone-to-vehicle communication system, it would also be possible to warn oncoming drivers.

Related Work. To support such applications and trigger notifications at the right time, mobile devices must sense and evaluate accident risks. Prior work in this area has focused on detection of oncoming cars using cell phone cameras as special sensing hardware. Gandhi et. al. [86] provide an overview of video, radar and laser distance measurement approaches for active pedestrian safety. RFID-based approach is discussed by Fackelmeier et. al. [81]. This approach doesn't need line of sight, but has a limited communication range and needs additional device to be carried by the pedestrian and the car. Another approach that needs no line of sight and is based on 3G and WLAN is presented by Sugimoto et. al. [92]. David et. al. [80, 82, 93] present a radio approach that assumes that the GPS location is precise up to 10 to 80 cm. Their solution also relies on an external server, with a considerable connection establishment time, to coordinate communication between pedestrian and driver. They also add movement recognition to the radio-based solution [83]. Another pedestrian safety app by Wang et. al. [94] uses the smartphone's rear camera to detect vehicles approaching the pedestrian when he is talking and walking. This approach works only when the pedestrian is on a call and can easily drain the smartphone's computational resources and battery.

Approach. In this chapter, we ask whether mobile devices can use their in-built sensing capabilities for pedestrian risk assessment. With the plethora of user interaction designs that augmented reality technology provides, we believe that the problem can be relaxed from detecting imminent collision to sensing increased pedestrian risk. Augmented reality interfaces offer many choices for subtler awareness cues rather than only raising startling alarms. Leaving aside the specific user interaction design, we focus on defining relevant pedestrian risk scenarios and studying whether mobile positioning and inertial sensing techniques can sense scenarios that pose increased risk. In summary, the contributions of this chapter are:

- identifying pedestrian risk scenarios that are amenable to detection with in-built sensors
- defining requirements for a positioning and inertial sensing approach
- evaluate the limits of positioning techniques across these scenarios for pedestrian in-street detection

4.2 Scenarios and challenges

With a vehicle safety communication system such as the DSRC [65,71] system supported by the US Department of Transportation, it is feasible for a pedestrian's device to notify oncoming vehicles when the pedestrian is in-street. The vehicle could alert the driver, perhaps using augmented reality displays on the windshield. The requirements and challenges for smartphone-based in-street detection techniques differ significantly across scenarios. In particular, in-street detection can assume different meanings depending on the environment. We discuss some of these scenarios here, and the corresponding application assumptions.

- **Rural Out-of-Town Environments.** In out-of-town environments such as rural roads or highways, pedestrians are rare and not expected by most drivers. Thus walking along such roads, which typically do not have sidewalks, can be particularly hazardous. Since pedestrians are sparse and tend to walk along roads for extended periods, the accuracy requirements on a positioning system are relatively low, in the order of tens of meters. It thus suffices to let an approaching driver know if a pedestrian is walking along a street.

- **Suburban Environments.** In suburban or residential areas, there may or may not be sidewalks, and one may expect occasional pedestrians, walking in the street (when no sidewalks present) or on the sidewalk. In the absence of sidewalk, any approaching car can be warned. For a pedestrian walking on the sidewalk, solely detecting that there is a pedestrian walking along the street would result in uselessly warning an approaching driver. It is thus beneficial to identify the events when this person may be stepping into the street, putting himself at a higher risk of being hit by car.

- **Urban Environments.** Most urban downtowns and cities are well developed and have sidewalks, with a large number of pedestrians walking along the street. Just knowing that a pedestrian is walking along a street would create far too many warnings for drivers and cause warning fatigue. In such a scenario, more fine-grained differentiation of pedestrians at risk is not only beneficial but necessary. Since a pedestrian is usually safe when walking on the sidewalk, our approach is to identify pedestrians that are in the roadway. Accurate detection of street crossing is therefore a key concern.

Considering the application scenarios listed here, it is evident that outdoor walking activity detection, determining that they occur near streets (street matching) and crossing detection are key building blocks. Those rely on localization, which is the biggest challenge for such a detection venture. A careful analysis of these scenarios also reveals that

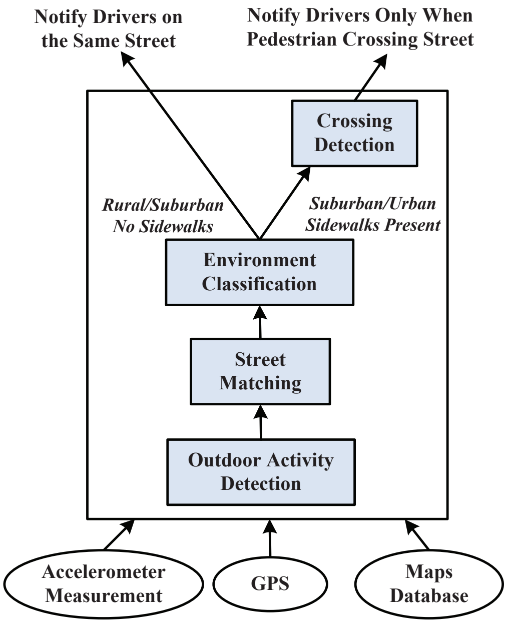

Notify Drivers on the Same Street
Notify Drivers Only When Pedestrian Crossing Street
Crossing Detection
Rural/Suburban No Sidewalks
Suburban/Urban Sidewalks Present
Environment Classification
Street Matching
Outdoor Activity Detection
Accelerometer Measurement
GPS
Maps Database

Figure 4.1. System overview.

accurate positioning is not a stringent requirement in rural areas, but is increasingly valuable in suburban, and an absolute necessity in urban environments. While GPS localization might work well in some rural environments with no or very few buildings around, using GPS for street level localization in suburban and urban environments might face more significant difficulties. To better understand these challenges, we therefore study the limits of such a positioning-based approach in the latter two environments based on sample algorithms we describe next.

4.3 Design and methodology

Our primary focus is to test the limits of position-based detection for the scenarios discussed in Section 4.2. Here we outline how position-based sensing could be part of a pedestrian risk assessment system. Given that a large number of people use smartphones, we explore smartphone-based techniques to localize pedestrians and determine their position with respect to the street-sidewalk frame of reference.

Fig 5.3 provides a system overview. The bottom layer depicts the raw inputs available from the smartphone. This includes the GPS, accelerometer and maps data. The components in the box characterize the system. The accelerometer and GPS data help determine when a person is walking outdoor. Using the maps data and the GPS location, we can identify the street the pedestrian is walking along. Using more detailed information available from a maps database, we can classify the environment as rural, suburban or urban.

We also identify whether or not the street has sidewalks. In case of a pedestrian walking along a street in a no sidewalks rural/suburban environment, the system can notify all the vehicles approaching the pedestrian. In the presence of sidewalks, the system runs an additional crossing detection algorithm, that identifies when a pedestrian is in-street (not on the sidewalk) and notifies only when such crossing events occur. The system components are discussed here in more detail.

Outdoor Activity Detection. Prior to running a detection on the pedestrian's location, we want to verify that this is done only when the pedestrian is outdoors. There exist multiple works on distinguishing whether the user is indoor or outdoor, based on GPS information, or accelerometer data tracking [48, 51, 55]. We can use these techniques to determine when a person is walking outdoor.

Street Matching. Once we have established that a person is walking outdoors, we use the location coordinates to determine whether the pedestrian is near a street by matching the location against a map. In the context of a vehicle safety communication system, this information could be shared with vehicles over radio links such as 802.11p (which have been demonstrated in smartphones) with vehicles. When approaching intersection, two different streets appear to be at the same distance from the pedestrian. This is indeed useful, because at an intersection, cars traveling on both streets must be alerted if the pedestrian is in-street.

Environment Classification. The GPS location and a relevant maps database allows us to classify the area the pedestrian is walking in, as discussed in Section 4.2. It provides us an estimate of the detection requirement for the area. For a pedestrian in a rural environment where there are no sidewalks, we deduce that the pedestrian is walking in-street and that it is adequate to identify the street, and notify the cars driving on that street. In an urban location, it becomes imperative that the algorithm identifies potentially unsafe transitions into the street and alerts the cars only in relevant occurrences of such events. We can obtain the underlying street network information and check for the presence of sidewalks using OpenStreetMap [62].

OpenStreetmap is an open-source maps database that uses a topological data structure composed of nodes, ways, relations and tags. Tags are used to store metadata about the map objects [38]. Of the various road tags in the dataset, the *sidewalk* tag, *location* tag and *highway* tag, are used in our heuristics to determine if a street has sidewalk. These tags are able to show roads' information on sidewalk, location and road type, respectively. Using the *sidewalk* tag it is possible to indicate if a sidewalk is present on either both sides, the left side, or the right side of a street.

For validating our heuristics, we randomly select 30 streets from 5 regions in New Jersey. These regions are shown in Fig 4.2. For 6 streets in each region, we wanted to determine if it has a sidewalk or not. We acquire the information from OpenStreetMap and find the corresponding street on Google maps to compare our identification with the ground truth. If it is consistent, we say that the sidewalk is accurately detected. We calculate the sidewalk detection rate, which is the percentage of sidewalks that are accurately

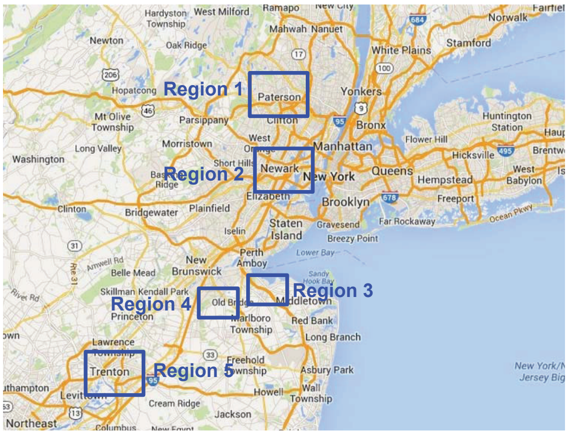


Figure 4.2. Regions for sidewalk presence heuristics validation.

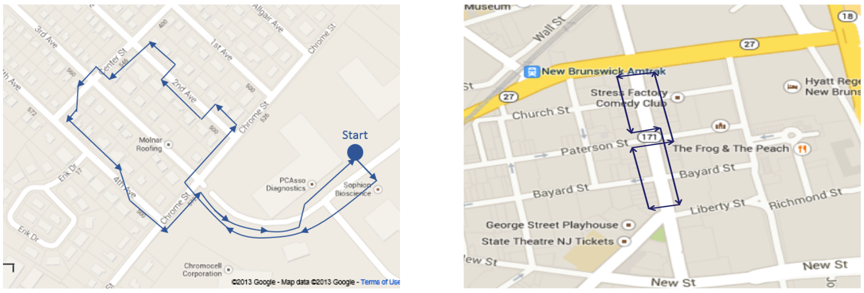
identified. We observe that we can achieve 94% sidewalk detection rate. This observation indicates that our heuristics can identify streets with sidewalks accurately. The coverage for the *sidewalk* tag is variable around the U.S., being best in Washington DC [36], Toronto and a number of other places.

Crossing Detection. As a final step, we need to detect when a pedestrian is in-street. It is fairly straightforward when there is no sidewalk present. We identify this as walking in street, potentially risky and warn any oncoming cars. In dense urban environments, with sidewalks and pedestrians abound, we do not want to send too frequent unnecessary alerts to pedestrians and approaching cars. Hence, more fine-grained information about pedestrians that are actually in the street is useful. This occurs usually when the person is crossing a street. Our algorithm achieves crossing detection by predicting a pedestrian's path of motion and checking if it intersects with any of the streets nearby. Such an intersection of the pedestrian's predicted path and the street centerline indicates when the user might be purposed to cross the street. To predict the path of motion, we extrapolate a user's heading by a distance *d*. In our scenario, the streets are two-way streets, single lane in each direction. The typical lane width in New Jersey is about 3.5 meters [49]. We chose *d* accordingly.

4.4 Experiment description

To test the proposition developed in the earlier section, we use the GPS information from the pedestrian's smartphone and analyze the extent to which positioning technologies can support in-street detection. The GPS on the smartphone provides us with useful information, such as the latitude and longitude for a pedestrians' location, their heading, speed and accuracy of the location provided.

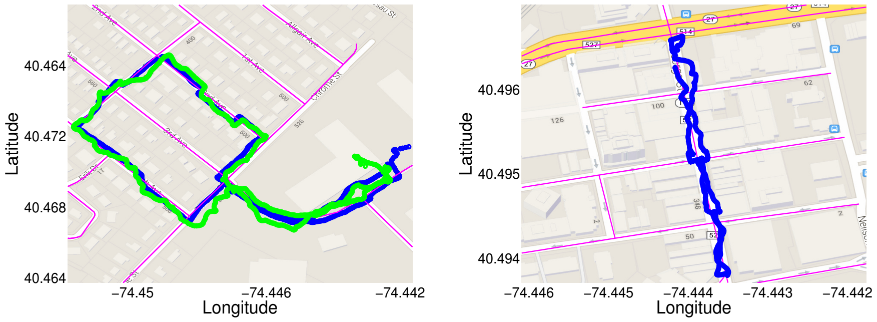
We chalked out two separate test paths. 15 volunteers from our lab walked along these



(a) Suburban test path.

(b) New Brunswick test path.

Figure 4.3. Experiment test paths.



(a) GPS trace in suburban test bed with phone in hand (blue trace) and phone in pocket (green trace).

(b) GPS trace in urban test bed.

Figure 4.4. GPS traces plotted on underlying street network.

two test paths for the purposes of data collection. An Android application continuously collected sensor data. The test paths are shown in Fig 5.6. The path in Fig 5.6(a) is a suburban test path that includes a residential area, where most buildings have two floors. This area had a sidewalk only along one street, while for the rest of the path the pedestrian had to walk in the street, along the edge. The street was crossed for a total of 14 times on this path. The exact same path was traversed 20 times. This path incorporated various turning and street crossing scenarios.

Our second test path, as shown in Fig 5.6(b) was the downtown area of a small city, New Brunswick. Fig 5.6(b) shows only one of the many paths we covered. The street was crossed about 10-12 times in each trial. 12 different people walked a total of 28 similar loops for data collection. This test bed is a well developed urban area and had sidewalks throughout.

For both the test paths, the pedestrian carried the phone in hand or in pocket. The Android application in context recorded raw timestamped data from GPS, Accelerometer, Gyroscope, Magnetometer, Rotation Vector, Linear Acceleration and Gravity Acceleration. The information logged from the GPS was the latitude, longitude, bearing, speed, time of fix, accuracy and number of satellites used for that fix. We set the GPS to log data at the maximum available rate, which allows us to obtain one GPS location per second, on an average. The application also allows us to record the ground truth, i.e. the exact moment when the pedestrian enters a street, by touching a button on the screen. We use these timestamps to compare and evaluate the detection performance of our algorithm.

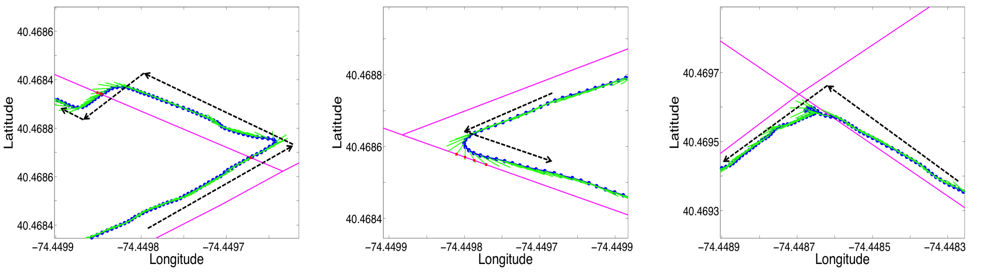
4.5 Evaluation

Keeping in mind the gravity of pedestrian risk detection applications, we evaluate the performance of GPS based algorithms for such risk awareness. We also want to explore how this limit varies with environments. We selected suburban and urban scenarios because they are more challenging environments with a complex infrastructure. Rural out-of-town environments can be handled by existing localization techniques as discussed in Section 4.3. We also analyze detection delay, rather than just detection accuracy, since timeliness is important for the effectiveness of pedestrian safety warnings.

We first show here how the collected GPS traces look as compared to the actual path walked. The GPS trace from one of the trials for the suburban test bed is plotted on an underlying street map in Fig 4.4(a). The magenta lines are the center of the streets obtained from OpenStreetMap, the blue path is the GPS trace when the phone is in the user's hand, and the green path is the GPS trace when the phone is in user's pocket. It is evident from these traces that the GPS positioning accuracy deteriorates when the phone is in the pedestrian's pocket, compared to when in hand. Fig 4.4(b) shows the GPS trace of the urban downtown, plotted on the underlying street network. This trace substantiates that the GPS positioning quality declines rapidly in urban areas.

In-street detection for rural and suburban scenarios can be performed using the location provided by GPS and the sidewalk tag from OpenStreetMap. The heuristic used for this approach was discussed in the earlier section. For urban settings with sidewalks, we want to identify the events when a pedestrian enters the street. Therefore, we implement and evaluate a crossing detection algorithm based on GPS-alone.

Crossing Detection Performance Evaluation. Fig 4.5 shows GPS traces from segments of one of the trial walks. The magenta lines mark the center of the streets obtained from OpenStreetMap. The black dashed arrows indicate the actual path walked by the pedestrian and the blue path is the GPS trace obtained from the smartphone. The green lines are the predicted path of motion obtained by extending the pedestrian's GPS bearing by a distance *d* (4 meters in this case), at each location update. Each time an intersection of the predicted path with the street is detected, a crossing prediction is said to be made. A red marker marks this crossing prediction. Based on whether or not the pedestrian crossed the street, this prediction can be a true detection or a false alarm.



(a) Examples for correct crossing detections.

(b) An example false detection caused by change in bearing.

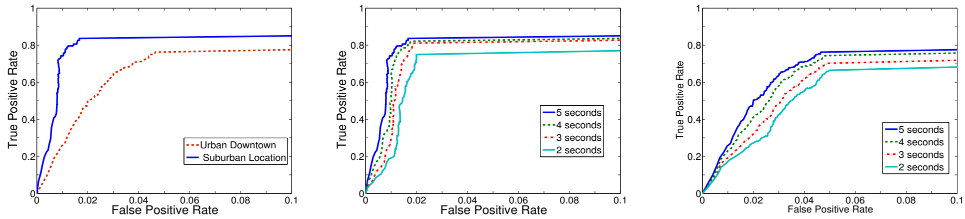
(c) An example missed detection caused by GPS inaccuracy.

Figure 4.5. Examples of various crossing detection scenarios. The magenta lines mark the center of the street. The black arrows indicate the actual path walked. The blue trace is the person's GPS trajectory. The green lines are the predicted path of motion. The red markers are the points where a predicted path intersects with the line marking the center of the street.

Fig 4.5(a) shows two accurately detected crossings, identified by the intersection of the predicted path and the street. The red markers are the points where a predicted path intersects with the line marking the center of the street. An example false positive is shown in Fig 4.5(b). In this example, the pedestrian turned around the corner at an intersection (shown by the blue GPS trace) to walk along the perpendicular street, rather than crossing that street. This false alarm was caused by the change in the pedestrian's path of motion. GPS positioning suffers from delays in location updates and hence fails to capture the unanticipated changes in the pedestrian's path and bearing. A false negative is a missed detection, mostly caused due to GPS inaccuracy. Fig 4.5(c) is an example of a missed detection. In this case, the GPS is unsuccessful in accurately identifying which side of the street the pedestrian is on, resulting in missing the crossing.

From the discussion of the extrapolated path length d in the previous section, we can see that the longer this extended distance, the earlier the crossing detection would be. On the other hand, a longer extended path would also lead to more false positives. To analyze how this path length affects our detection, we varied d over a range of 0.1 m to 10 m, in steps of 0.1 m.

Fig 4.6(a) shows the Receiver Operating Characteristic (ROC) curves for the suburban and urban test beds. This curve plots the true positive rate against the false positive rate. We define a true positive as a detection made at most 8 seconds before and k seconds after the exact instant the pedestrian entered the street. As discussed earlier, the entrance time was recorded by means of an Android application. In Fig 4.6(a), the value of k used is 5 seconds, varied over d. This implies that for any detection to be counted as a true detection, the warning can occur as late as 5 seconds from the time of the exact entrance into the street.



(a) In-street detection performance for urban and suburban testbeds at ground truth window = 5 seconds.

(b) In-street detection performance for the suburban test bed at varying ground truth windows.

(c) In-street detection performance for the urban test bed at varying ground truth windows.

Figure 4.6. Crossing street detection performance analysis.

We refer to this window as the *ground truth window*. For practically useful applications, the allowed time *k*, after the exact entrance must be small. This implies that the event must be detected as soon as the person enters the street, or within the allowable time window *k*. We vary *k* to assess the algorithm for delay performance. Fig 4.6(b) shows the ROC curves for the suburban area for varying ground truth windows. We see that the algorithm provides a very good detection rate of 85% with as few as 1.8% false positives. This performance degrades as the allowed delay (*k*) after the actual entrance is decreased.

Fig 4.6(c) shows the performance for the crossing detection algorithm in an urban environment, such as the downtown of a small city. We can observe a visible drop in the performance compared to that in a suburban environment, with a detection rate of 78% for 4.5% false positives. The maximum rate at which we can sample GPS is approximately once per second. It is evident from the performance curves that GPS does not work well for applications with a stringent timing and fine grained localization requirement, such as the pedestrian safety applications.

4.6 Discussion and conclusion

We analyzed the performance of positioning and inertial sensing techniques for pedestrian in-street detection [56]. We identified in-street detection in rural, out-of-town areas as a scenario that is likely to be feasible and our experiments show promising results even for sidewalk-street differentiation in suburban environments, but it still presents a challenge in urban environments. It is evident that GPS does not serve well the fine-grained positioning needs of pedestrian safety applications in dense urban environments. It is impaired with large errors in positioning and delays in detection, rendering it unfit for such critical applications.

One approach for further work that might overcome the many false positives in the GPS-only prediction algorithm, that occur due to GPS location and bearing inaccuracy, is

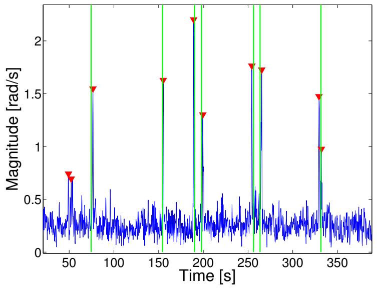


Figure 4.7. Turns detected by gyroscope.

the use of inertial sensors to detect key movement features by pedestrians. For example, we saw that GPS fails to capture the sudden changes in the pedestrian's path of motion close to an intersection. Such sudden turns can be detected with the in-built gyroscope in a smartphone. Fig 4.7 shows a plot of time vs gyroscope magnitude from a walking trial. Here, the vertical green lines depict the ground truth and mark the exact time instance when a user turned and the red markers mark the turn detected by our algorithm, based on gyroscope magnitude. We can see that most turns are detected.

Together such risk assessments can enable new applications that warn distracted pedestrians as well as drivers, when a safety communication system is available. As our electronic devices are increasingly drawing our attention away from real world hazards, we believe that this will be an important feature of future mobile augmented reality systems.

Chapter 5

Inertial Sensor-Based Pedestrian Positioning

This chapter presents an improved solution for pedestrian localization in an urban environment compared to the one described in the previous chapter. It seeks to detect the transitions between sidewalk locations and in-street locations using inertial sensors, to enable applications such as alerting texting pedestrians when they step into the street. In this work, we use shoe-mounted accelerometers for location classification based on surface gradient profile and step patterns or pedestrian-to-car communication. The shoe sensors relay inertial sensor measurements to a smartphone, which extracts the step pattern and the inclination of the ground a pedestrian is walking on. This allows detecting transitions such as stepping over a curb or ramps that lead into the street. We carried out walking trials in environments ranging from suburban to metropolitan area (Manhattan) and show that we can accurately determine transitions between sidewalk and street locations. We also show how, in certain scenarios, surface profile matching techniques can provide more accurate location estimates than existing positioning technologies.

The content of this chapter is organized as follows. Sec.5.1 introduces the pedestrian safety problem and summarizes the main contribution of this work. Sec.5.2 describes the possible applications which can take advantage from our solution and the related challenges. Sec.5.3 provides the system overview while Sec.5.4 analyzes in detail its building blocks. The system implementation is reported in Sec.5.5 while the experimental scenarios are described in Sec.5.6. The robustness of the algorithm is evaluated in Sec.5.7. The related work are describe in Sec.5.8 while Sec.5.9 concludes the chapter.

5.1 Introduction

Evidence is mounting that technology distractions have a negative impact on pedestrian traffic safety. In the last decade, from 2000 to 2009, American streets have witnessed more than 688,000 pedestrians injured in traffic accidents, 47,700 of them fatally [64]. Pedestrians account for nearly 14% of all traffic fatalities in the US [74] and about 22% worldwide [63]. While motorist fatalities have declined, pedestrian fatalities have been rising at an annual rate of 4.9% from 2009-2012 [61] (the newest 2013 estimates look more positive, fortunately). Econometric analysis has shown that at high densities of cell phone use, the life-taking effect due to distractions outweighs the life-saving effect of improved emergency medical response times [89]. According to a study, 26% of pedestrians text or email, 51% talk on the phone and 36% listen to music while crossing the street [72]. While the evidence is not yet fully conclusive, it is significant enough that municipalities have started stenciling “LOOK" signs at crosswalks in an attempt to alert pedestrians who text while crossing the street (see Fig. 5.1).

We believe that a technology solution would be more effective for the problem at hand. Smartphones, which are part of the problem, can also be part of the solution. Smartphones can sense when they are being used while walking using existing activity recognition techniques [90]. If they could also sense potentially dangerous situations, they could generate much more targeted and noticeable alerts. Smartphones could also be integrated with the emerging wireless traffic management and safety infrastructure (e.g., DSRC [71], [65]). In prior work, the Walksafe project has presented preliminary results on using smartphone cameras for detecting oncoming vehicles [94]. Even if significant improvements in accuracy will make this approach feasible, the energy consumption of continuous camera operation is likely to remain a challenge.

Pedestrian safety improvements, however, do not necessarily require such near-collision detection. Since the majority of pedestrian fatalities occur during road crossing at intersections or midblock locations [76], safety improvements could also be achieved by supporting good crossing habits or by alerting distracted pedestrians who enter the roadway.[1] Phones can easily monitor when they are used for texting and other potentially distracting purposes. The challenge, however, is to determine when pedestrians are crossing and when they are in relatively safe areas on a sidewalk. Existing localization technologies such as cellular, WiFi, and satellite positioning do not consistently achieve the accuracy necessary to discriminate sidewalks and roadways. The Global Positioning System may come close under ideal open-sky conditions but cannot achieve the same accuracy in many urban centers, where it is most needed.

In this work, we address this challenge through a shoe-based step and terrain gradient profiling technique that can detect transitions between sidewalks and streets, as well as assist in localization. It exploits the trends towards wearable sensing in shoes for health and fitness [57,70]. We show that similar inertial sensors are not just useful for exercise tracking and posture analysis, but can also monitor step and ground profiles that are important

[1] Indeed, after a sharp 2012 uptick in traffic fatalities in New York City, the Transportation Commissioner blamed it in part on distracted walking and lamented: “I don’t think that the iPhone has invented an app yet that will ping you when you hit a crosswalk." [69].

LOOK UP.
Drivers aren't always looking out for you.
WALK SMART. ARRIVE ALIVE DE
LOOK!

(a) (b)

Figure 5.1. Signage at crosswalks in Delaware and New York City [67,68].

in a pedestrian safety context. We note that urban environments (at least the more developed regions) follow relatively consistent design guidelines [77] with curbs separating roadways from sidewalks and increasingly ramps leading into dedicated crossings. We therefore develop algorithms that can detect the stepping over a curb, which often occurs when crossing a street. We also demonstrate that it can track the inclination of the ground and detect the sloped transitions (ramps) that are installed at many dedicated crossing to improve accessibility. The ground gradient measurements could further be matched to a known terrain profile to assist in localization. We have developed a prototype sensing system based on a device affixed to a shoe to evaluate the effectiveness of this approach both in laboratory experiments and in approximately 40 hours of walking spread over the downtown area of a small city and the midtown area of Manhattan.

In summary, the major contributions of this work are as follows:

- Proposing the use of shoe-mounted inertial sensors not just for step counting but for step and ground profiling to support fine-grained pedestrian location classification
- Designing algorithms for detecting transitions from the sidewalk to street, by stepping off the curb or by walking over ramps leading into designated pedestrian crossing
- Extending the algorithm to match ground gradient profiles for pedestrian localization
- Experimentally analyzing the performance of these algorithms in a controlled laboratory environment and based on approximately 40 hours of walking traces in a small city and in Manhattan

5.2 Applications and challenges

There are multiple applications that could benefit from a sensing technique to classify pedestrian walking locations and in particular to detect the transitions from sidewalks to

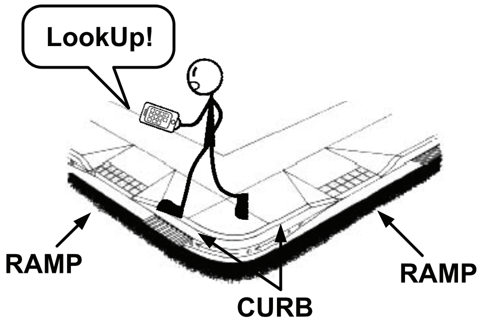


Figure 5.2. Illustration of sidewalk to street transition points.

streets. Examples include:

Virtual Look Up! Rather than relying on signs painted on the roadway, which may lead to warning fatigue, smartphones could issue a much more targeted electronic alert when an apparently distracted pedestrian is about to step into the street (illustrated in Fig. 5.2).The exact mode of alert or notification could range from visual (for texting users) to auditory, and might differ based on the robustness of detection. More intrusive alerts might be desirable when the system can attain high confidence that the user is at risk; more peripheral warnings may be suitable when there is lower confidence. We leave the exact design of such alert mechanisms to application designers and focus on robustness of the underlying sensing techniques.

Driver Pedestrian Awareness. Government transportation departments are working with car makers towards a vehicle communication network that can be used to provide vehicles with better situational awareness [75]. To support safety applications and increasingly automated driving, over time all vehicles are expected to announce their current position and other vehicle dynamics information to vehicles in the vicinity. Prototype smartphones exist that can participate in such an 802.11p based network [71]. This would not just enable smartphones to learn about oncoming cars, but also to potentially announce the presence of the pedestrian to nearby vehicles. In urban areas, however, such announcements would have to be more selective, so that the system can distinguish pedestrians that are safely walking on the sidewalk from pedestrians that may be at risk.

Feedback on Crossing Habits. Not all applications require distinguishing pedestrian walking locations in near real-time. Offline analysis of historical walking traces can also be useful to monitor and provide feedback on good crossing-habits. For example, a sensing system could allow parents to monitor whether children that walk to school are stopping (to look) before an intersection or whether they dash into the street.

5.2.1 Challenges

Consider a straw-man approach that attempts to localize the pedestrian using existing methods and distinguish sidewalks and roadways by comparing the position coordinates with the location of known roadways on a map. While such an approach may be suitable in detecting a pedestrian walking along a rural road (where no sidewalks exist), it can be expected to face significant challenges in more urban environments where the majority of pedestrian accidents occur. Particular challenges are:

Accurate positioning. In many suburban and urban environments the most widely available, precise localization methods are those based on global navigation satellite systems. Typical accuracies are in the order of a few meters under good conditions and the accuracy can quickly degrade to tens of meters in more challenging urban canyons. Cellular and WiFi positioning rarely improves accuracy outdoor. Since sidewalks are often only a few meters wide, this level of accuracy makes it extremely challenging to distinguish a pedestrian on the sidewalk from one in the street let alone determine the event when a person transitions from the sidewalk to street.

Sidewalk map. Even with accurate positioning, the system would have to compare the position coordinates with a map to distinguish sidewalks from streets. Unfortunately, road maps such as those available from OpenStreetMap [62] record only road centerlines. Whether a sidewalk is present along the roadway is not stored. Even if it were, the boundary between streets and sidewalks would need to be estimated based on assumptions of typical road width. This further limits accuracy or requires additional effort in constructing such a map.

Robustness to environment. There exist considerable variations across urban environments. High-rise buildings in certain environments might lead to localization challenges. Street widths can vary and features such as pedestrian overpasses or center medians may need to be handled. Pedestrians' paths are not as well-constrained as those of vehicles. This makes it challenging to achieve robust and consistent performance across such varied environments.

To overcome these challenges, we explore a different sensing approach that detects significant events in a local frame of reference for the pedestrian, instead of relying on absolute positioning.

5.3 Inertial profiling approach

Our primary idea is to distinguish the pedestrian's location through inertial sensing of ground features, and in particular through the design features that demarcate roadways and sidewalks. In more developed regions, engineers are required to follow increasingly precise and consistent sidewalk design guidelines [60, 77] to improve accessibility. In particular sidewalk-roadway transitions should remain easily detectable by the visually impaired yet do not pose barriers for wheelchair users. In the United States, sidewalks and roadways are separated by curbs, which are lowered at designated crossings through a ramp. The ramp design often includes changes in surface texture to make detection easier.

We explore an inertial sensing system to detect such unique and consistent features. The proposed system acquires acceleration data from a shoe-mounted inertial sensor, to detect changes in step pattern, and to estimate the inclination of the ground at each step. Based on the obtained profiles, the algorithm classifies the pedestrian's current location as street or sidewalk. The most important feature of our approach is to detect the events when a pedestrian transitions from the sidewalk to the street or vice versa. The system therefore focuses on detecting steps over curbs as well as ground slope changes that indicate walking over ramps. We further explore the use of slope profiles for more fine-grained localization and surface roughness as an additional indicator to distinguish sidewalks and streets. This approach addresses the aforementioned challenges because it does not rely on absolute localization or maps and achieves robustness by relying on consistent sidewalk design features.

5.3.1 System overview

A shoe-mounted sensor has considerable advantages over other approaches (e.g., using phone's GPS). It moves along with the foot and allows us to trace its exact movement. Thus, the shoe-mounted sensor can measure the feet inclination at any given time point and further helps to compute the slope of the ground. Inertial sensing directly on the phone would not allow determining the surface gradient. Placing a sensor on each foot gives us the added benefit of detecting *step off* and *step on* events, irrespective of the foot the pedestrian uses for the action. Toward this end, our approach utilizes inertial sensors to capture step and ground profiles to guard pedestrian safety. It consists of three main tasks: *curb detection*, *ramp detection*, and *slope-based localization*. Fig 5.3 displays the flow of our system.

The lowest layer takes the input from the inertial sensors on both feet. As an example, the figure shows the raw data collected from the shoe-mounted three-axis accelerometers, by connecting it to a smartphone over Bluetooth. The data is obtained from each of the axis: x, y and z. The center layer has been split into three main components of our system. The curb detection, the ramp detection and localization based on slope profiling. Each of these layers processes the data from left and right sensor in the same way and provides a combined detection output.

To perform curb detection, our approach identifies significant changes in accelerometer readings when the user steps off the curb. It utilizes raw accelerometer data from the lower layer. In particular, we consider the acceleration magnitude of the y-axis and z-axis. After removing the high frequency noise components, we further smoothen the data. We apply a threshold to the resultant signal, to segregate the peaks. These peaks reflect the transitions made by stepping off the curb.

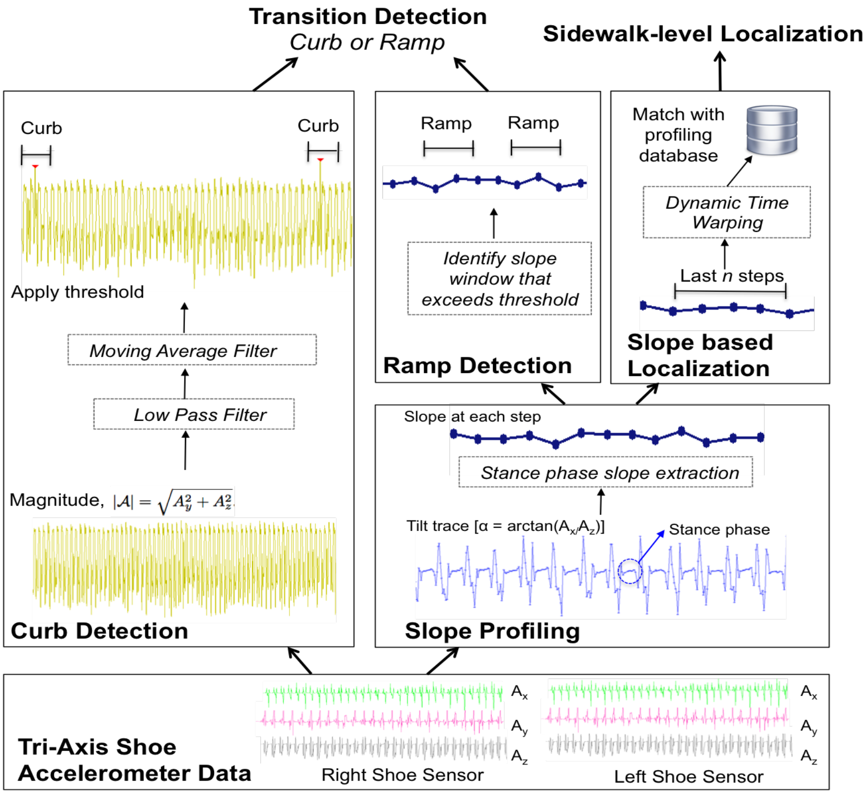


Figure 5.3. System overview of transition detection and localization.

Both ramp detection and localization rely on slope profiling, which includes two steps, *Tilt Tracking* and *Stance Phase Detection*. The x and z-axis accelerometer data is used for ground surface gradient profiling. Fig 5.4 shows the orientation of the accelerometer axis with respect to the shoe. *Tilt tracking* refers to generating the tilt trace. The tilt trace represents the inclination of the shoe-sensor system in time. This inclination changes as a person walks. It is also observed that the different phases of walking are recognizable from this trace (Fig 5.4). *Stance phase detection* refers to detecting the duration when the foot is flat on the ground. From each of the stance phases, our system extracts the mean inclination of the foot every time it is in contact with the ground. The intuition behind this measurement is that this value represents the inclination of the ground at that step.

The ramp detection layer accomplishes the ramp detection by applying a threshold to the slope change along the step trace. Based on this threshold, it can identify when a person is walking on the ramp. The localization layer uses the step trace to identify the exact sidewalk a person is walking on. It uses last n steps from the step trace to match it against a profiling database, which contains previous traces from all sidewalks in that region. We use dynamic time warping to align the traces and match step patterns.

The curb and ramp detection, together, form the transition detection system. The slope based localization allows us to localize a pedestrian to the sidewalk he is walking on. An application that alerts pedestrians in potentially dangerous situations uses these transitions for generating warnings because they mark the instants when a pedestrian enters or exits a sidewalk. The next section discusses the detection algorithms and implementation in detail.

5.4 Step and slope profiling

Our approach is based on classifying the pedestrian's location derived from the gradient profile of the ground surface. The gradient profile of the surface refers to a sequence of slope measurements of the ground. The sensor was oriented such that the x axis pointed forward in the direction of motion, the z-axis pointed in the direction of gravity and the y-axis pointed sideways, as shown in Fig 5.4. Our algorithm accepts raw accelerometer readings as input to measure the inclination of the ground and provides fine-grained localization based on the gradient profile.

5.4.1 Curb detection

At midblock locations, a jaywalker usually has to step off a curb to enter the street, followed by a step on to mark the transition back from street to sidewalk. We noticed that during step off and step on events, the accelerometer records a peak in acceleration along the axis on which the gravity is acting. Therefore we developed an algorithm that keeps track of the instantaneous variation in the accelerometer magnitude along the z-axis and the y-axis. We do not consider the x-axis because it points to the direction the user is moving in, and the peaks on that axis represent lateral movement and not vertical movement. The y-axis accounts for any sideways movements and hence we consider this axis in our evaluation. The resultant magnitude of acceleration $\mathcal{A}$ is given by $|\mathcal{A}| = \sqrt{A_y^2 + A_z^2}$.

In order to remove the noisy components from the raw accelerometer data, we filter this data with a Butterworth low pass filter at a cut-off frequency of 10Hz. Then we smooth the data using a simple moving average filter (SMA) with a span of 3 values. After the filtering operations, our algorithm searches for all peaks that exceed a defined threshold in a small time window (e.g., 2 seconds). In each window we find the highest peak and compare it to the event that occurred in the previous window. In case the current event has a higher magnitude than the previous one, we select it as the detected event of stepping off or stepping on the curb. The pseudo code for the curb detection algorithm is given in Algorithm 1

5.4.2 Slope profiling algorithm for ramp detection

The foot mounted accelerometers are used to obtain the slope of the ground at each step the walker takes and facilitates the measurement of the ramp inclination. This algorithm

Algorithm 1: Curb Detection Algorithm

```
Data: $A_y$ - Accelerometer value along y axis
      $A_z$ - Accelerometer value along z axis
      LastEvent - Event detected in previous window
Result: DetectedEvent - Time instant at which the event took place
begin
    for j = 1 $\rightarrow$ SamplePerWindow do
        $mag(j) = \sqrt{A_y(j)^2 + A_z(j)^2}$;
    maglpf $\leftarrow$ lowpass(mag);
    magSmooth $\leftarrow$ SMA(maglpf,3);
    DetectedEvent = 0;
    forall the $x \in SamplePerWindow$ do
        if magSmooth(x) > threshold and magSmooth(x) > DetectedEvent then
            DetectedEvent = x;
    if DetectedEvent > LastEvent then
        return DetectedEvent;
```

proceeds in three steps. First, it tracks the tilt angle of the accelerometer. It then uses these tilt traces to extract the stance periods of a step, the period when the foot is flat on the ground. The mean tilt of the accelerometer during this period allows us to estimate the absolute slope of the ground at each step. The detailed algorithm is described below.

Tilt Tracking. To obtain the tilt from raw accelerometer readings, we measure the change in the static acceleration force. When the device is tilted with respect to earth, the force of gravity distributes over two axis, which in our case are the y and z-axis. This rotated gravitational field vector is used to determine the accelerometer pitch angle. In the case of a foot mounted sensor, the pitch angle, α is calculated around the y-axis, $\alpha = \arctan(A_z/A_x)$, where A_z and A_x are the raw accelerometer readings along the z and x axis respectively while α represents the inclination of the foot with respect to earth.

Stance Phase Detection. After obtaining the corresponding tilt value from every (x, z) pair of accelerometer readings, we use the dynamic acceleration (caused by moving), to identify the different phases of a person's walk. Fig 5.4 shows a tilt trace and the corresponding phases of walking. We require the *stance* phase of the walk, which is when the foot is flat on the ground. The inclination measurement obtained in this phase results from the slope of the ground at that spot, in addition to the initial tilt of the accelerometer.

As a person walks, the inclination of the foot changes continuously, varying rapidly during the swing phase but maintaining a small amplitude during the stance phase. Thus, the stance phase detection algorithm identifies the large negative peaks in the trace and isolates the samples between consecutive peaks. This is one walking cycle. By tracking

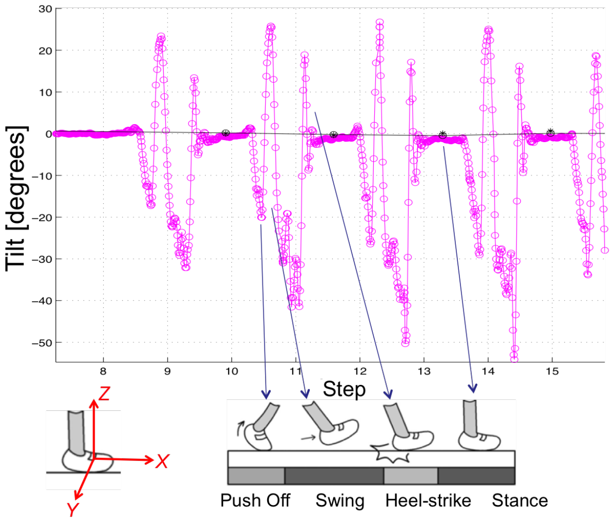


Figure 5.4. An accelerometer slope trace identifying the different phases of a walking cycle.

the variation of the tilt during each cycle, the samples making up each stance phase are determined as the collection of at least *count* number of tilt values within a few degrees of each other. Here *count* depends on the sampling rate of the sensor and the duration for which the foot is in contact with the ground. For our analysis we assume that the foot remains in contact with the ground for at least one-third of a second, during normal walking. At a sampling rate of 50 Hz, this implies that the stance phase should have approximately 15 samples. The mean of the inclination values during the stance phase provides us with the slope of the ground for that step. Thus, each step has a corresponding slope measurement of the ground.

By repeating this calculation for each stance phase, we acquire a sequence of slope values, the slope trace, where each slope corresponds to a step. From this slope trace, we arrive at an estimate for the slope of the ground by subtracting the initial tilt of the accelerometer, caused by the mounting.

5.4.3 Ramp detection algorithm

At designated crossings, the entry and exit is made via a small ramp that leads from the sidewalk to street. With the slope acquired at each step of the pedestrian from slope profiling, we can determine when a person is walking on a ramp. The sidewalk ramp begins a few steps before a person actually enters the street. The slope starts changing from the time a person enters the ramp and till he takes a few steps in the street. This is the window where most of the slope change happens.

The tilt tracking and stance phase detection discussed above are performed independently for each foot. After these procedures, we obtain the slope of the ground at each step from both feet. Each time a walking cycle is completed, we compare the slope of the current step to the slope acquired in the past steps. We look for a continuous gradient change, increase or decrease, higher than a threshold λ. If we discover a sequence of steps with slope change, called the event window, higher than λ in one foot, we look for an overlapping event window in the other foot. If an overlapping window is discovered, we find the step with maximum variation and consider that step S for further analysis.

We know that no two sidewalk ramps exist within 2 steps of each other. To avoid detecting the same ramp more than once and to get rid of false positives, we define a *guard band* around a detection. This guard band ensures that if an event has been detected, another event detected in its guard band will be discarded. We define this guard band as 2 steps. We use this strategy to make sure that we don't miss the events when a pedestrian exits a street and enters the parallel street right away (a common way of getting to the diagonally opposite corner at an intersection). In that case two ramps will exist very close to each other, and hence we use the value of 2 guard band steps. If a detected event S is within the guard band of a previous event, we discard this detection and if it's not, we consider this detection S as a true ramp detection. Algorithm 2 discusses the ramp detection process.

5.4.4 Slope-based localization

The proposed transition detection approach allows us to know when a pedestrian enters the sidewalk and when he exits it. Such knowledge enables our system to localize pedestrians in a set of sidewalks, leveraging slope traces and a profiling database. The smartphone's GPS provides us only a rough estimate of the user's position, to a radius of approximately 20 meters. With this information, we can narrow down the approximate location to a set of 4 possible sidewalks, which are the sidewalks on either side of the current block and those on either side of the next block, towards which the user is heading. The compass and GPS provide us with the heading information.

Each pedestrian has a unique step pattern, which refers to their slope trace. The profiling database contains, for each sidewalk, traces from different walkers. It also takes into account the direction of the pedestrian's walk for each sidewalk. The profiling database for each pedestrian is constructed based on the rationale that a person may repeatedly walks the same path, e.g., a path to the office or a usual lunch place. To localize a pedestrian, our algorithm extracts the slope values from the last *n* steps from sensors on both feet at run-time. We correlate this test trace to all the traces in the profiling database belonging to the same pedestrian to find the best match.

Because this correlation is performed between two traces that vary in time, we employ Dynamic Time Warping (DTW) [78] to measure similarity between two temporal traces. The DTW algorithm is essentially an alignment algorithm that helps us match step patterns, even if the users were walking at different speeds or the speed changed during the walk. We correlate the test trace to all the traces present in the database for the target

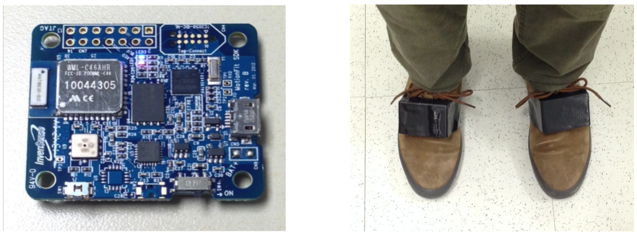
Algorithm 2: Ramp Detection Algorithm

```
Data: $A_x$ - Accelerometer value along x axis
      $A_z$ - Accelerometer value along z axis
      LastEvent - Step at which last event was detected
      guardBand - Minimum number of steps allowed
      between consecutive event detections
Result: $S$ - Step at which the event occurred
begin
    forall the $A_z, A_x$ do
        $\alpha \leftarrow \arctan(A_z/A_x)$;
        tilt ← Savitzky-GolayFilter($\alpha$);
        if walkCycleComplete(tilt) == TRUE then
            (startIndex, stopIndex) = findStancePhase(tilt);
            stepSlope = getMeanSlope(tilt,startIndex, stopIndex);
            gradientChange=getGradientChange (stepSlope);
            if gradientChange > threshold then
                overlap=checkEventOverlapWithOtherFoot() if overlap == TRUE
                then
                    $\mathcal{S} \leftarrow$ step with maximum variation;
                    if $\mathcal{S} >$ LastEvent + guardBand then
                        return $\mathcal{S}$
                    Discard $\mathcal{S}$;
```

area, to obtain a set of possible matches. Each match has an associated sidewalk output, correlation value and distance. All the matches with a correlation value higher than a threshold are considered eligible for decision making. Of these, the current sidewalk is determined as the one that matches with more than 25% of the traces for that sidewalk. In case more than one sidewalks match with the current trace, we choose the one with minimum *distance*.

5.5 Prototype implementation

We implemented an Android application to communicate with the sensors on the shoe via Bluetooth. The application logged the accelerometer data from the two sensors independently. Each accelerometer reading was timestamped and contained raw data from the three axis. Another application on the same smartphone was implemented to collect the ground truth for offline evaluation of the proposed system. This application had buttons



(a) MPU 9150 sensor.

(b) Sensor mounting on shoes.

Figure 5.5. Set up for shoe-mounted sensor.

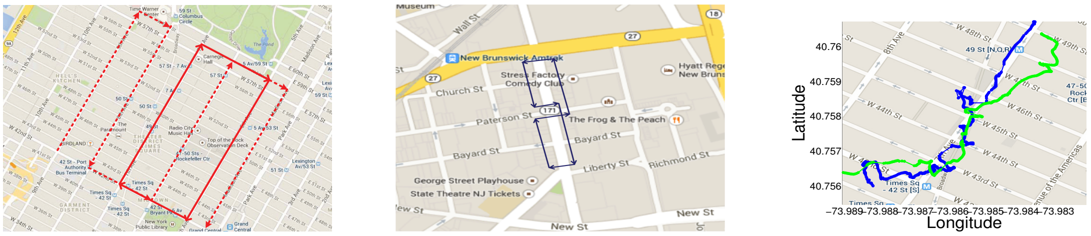
to mark the exact time when the walker transitions between a sidewalk and a street, and whether this transition occurred via a ramp or a curb. For safety reasons the ground truth was entered by a second person near the pedestrian. This ensured that the pedestrian is unbiased and undistracted while crossing the street.

The sensor mounted on the shoe is an Invensense MPU-9150 [87] 9-axis motion sensor, shown in Fig 5.5(a). It is a self-calibrating device, set to collect data at a sampling rate of 50 Hz. This board comprises inertial sensors including a tri-axis accelerometer, a Bluetooth module and a battery. We chose this platform because it offers relatively high precision inertial sensing module (the same chip that is used in latest smartphone models). In order to capture all foot movements, we strapped a sensor on each of the walker's shoes, as shown in Fig 5.5(b). We chose a sensor mount that is similar to current sport activity tracking products. We mount a sensor on each foot to ensure that stepping off/on the curb is detected, irrespective of what foot was used. It must be noted that only the accelerometer data was used for all the analysis.

5.6 Experimental scenarios

To evaluate the performance of our system, 16 volunteers from our laboratories walked along predefined routes in two locations with a sensor attached to each of their shoes. The system collected the ground slope data as well acceleration events for curb detection. All walkers were asked to walk 'normally' and an experimenter accompanied the walker to record ground truth information (i.e., the time when transitions occurred and whether they occurred via a ramp or a curb). We used a smartphone app for ground-truth recording and estimate that the timing accuracy is about 1s. The two locations are:

Midtown area of Manhattan. We chose this location in order to evaluate our system in a more challenging environment: with different kinds of ramps, crowded sidewalks and longer path. Five volunteers walked in their personal styles at different times including the rush hours and weekends. After a brief explanation in our laboratory regarding the apps and mounting the sensors, they autonomously collected the data in Manhattan. We



(a) Manhattan Test Paths.

(b) New Brunswick (city downtown) test path.

(c) Sample GPS traces from Manhattan experiments.

Figure 5.6. Test paths from two cities and sample GPS traces from Manhattan expreiments.

highlight that during their walk, they were free to use either a ramp or a curb to cross the streets. We selected two test paths. A shorter one that we were able to cover more frequently to analyze the variance of our results over the same path. A second longer path helped us understand variations occurring over different locations. The first test path begins from the Times Square (42nd Street, 7th Avenue) and then goes up to Central Park (58th Street, 5th Avenue) before returning to Times Square. This route is shown in Fig 5.6(a) and marked by solid red lines. The entire route is about 2.1 miles. The duration for each loop is approximately 60 mins, and it includes crossing the street about 30-35 times. The dashed path in Fig 5.6(a) shows the 2nd test path (about 4 miles), which is twice as long as the original test path. We walked the shorter path 20 times and the longer path twice, providing a total of nearly 25 hours of sensor data.

City Downtown. We conducted walking trials in New Brunswick's downtown area. Like a typical downtown environment, this testbed comprised mainly sidewalk ramps. Each loop was a 20 minute-long walk. It measured about 800 meters. These trials were conducted by 11 people. The path for each walker was decided beforehand, but the walkers were not aware of the path. They were provided directions as they walked. We did this to ensure covering most sidewalks and ramps in that area. Walking over the same sidewalk more than once allows us to use their step pattern for slope-based localization. Fig 5.6(b) provides a map of our test bed in the downtown area and shows only one of the many paths we covered. This path has 8 crossings, which equals walking over a ramp 16 times. We performed as many as 12 crossings in a loop, amounting to 24 ramps.

5.7 Evaluation

We evaluate the robustness of the detection of transition events and of sidewalk-level localization in the described scenarios.

5.7.1 Transition detection

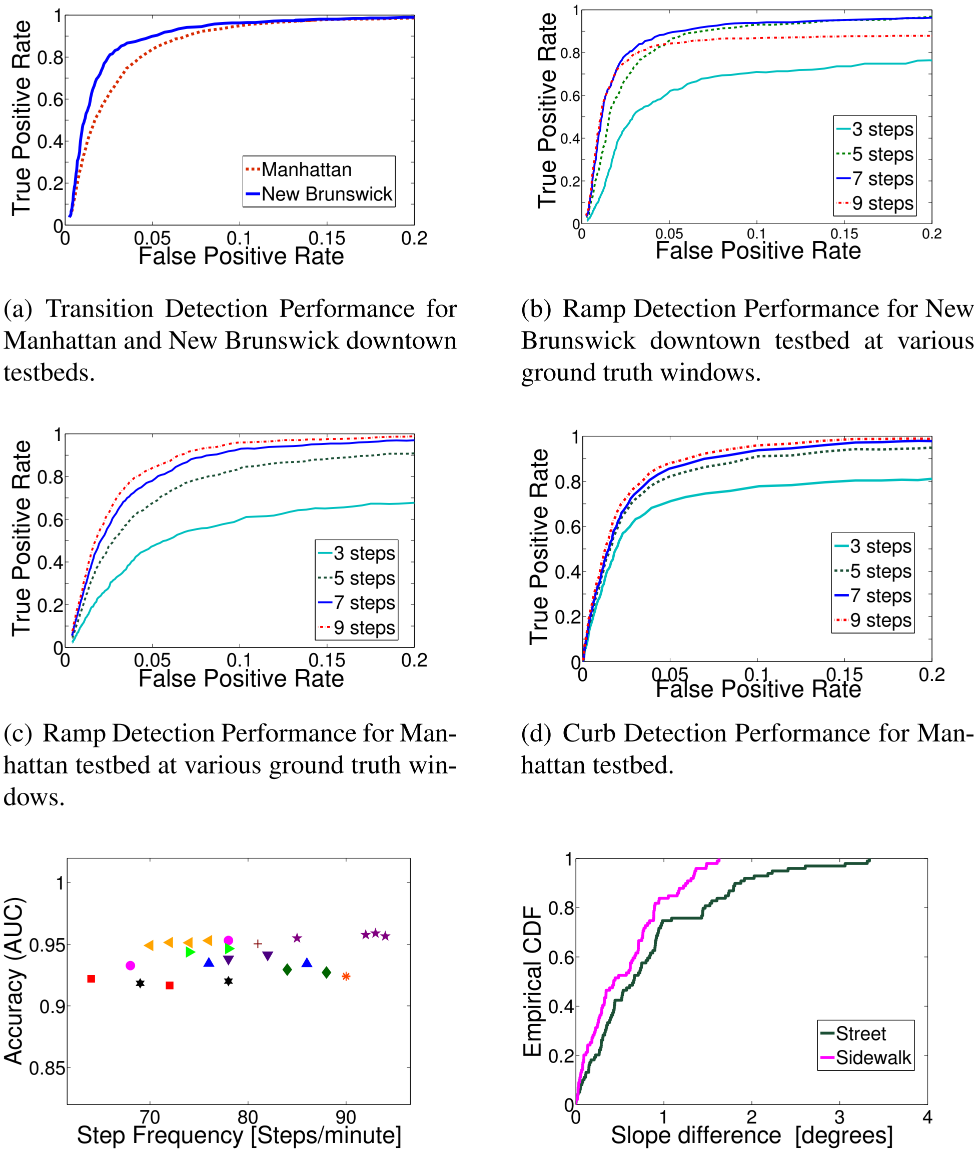
We first evaluate the performance of transition detection for curbs and ramps and analyze whether it is affected by where the pedestrian is walking and the individual's walking style. We use the Receiver Operating Characteristic (ROC) curve to illustrate our transition detection results. We define a true positive as a detection event that falls in a window around the actual transition (as marked during data collection). We refer to this window as the *ground truth window* and define it as a sequence of *k* steps (*k-2* steps on the sidewalk and 2 on the street) around a ground truth transition. We define it in terms of steps instead of time, since there is considerable variation in time when one has to wait for traffic to pass before crossing. By including 2 steps from the street in the ground truth window, we ensure that the point of transition is covered. Any detection event that falls outside such a window is counted as a false positive.

Location based evaluation

Fig 5.7(a) shows our transition detection results for the Manhattan and city downtown areas. It is evident from the figure that our system achieves over 90% true detection rate with just 5% false positives in the city downtown area. For the larger New York city area, our algorithm provides a true detection rate of above 90% with only about 7% false positives. The proximity of the two curves demonstrates that our algorithm performance is not affected by the location. For Manhattan, the performance of the longer loop was exactly similar to the performance of the original loop. This demonstrates that the algorithm performance is not affected by the length or duration of the loop.

City Downtown evaluation. We now break down the results in terms of ramp and curb detection. Our city has a typical US downtown environment with a lot of sidewalk ramps. The streets being busy, people mostly use these ramps for crossing the street, and hence their detection formed the core of our tests in that area. We evaluate our ramp detection algorithm's performance for 11 people who walked to collect sensor performance data. They traversed different paths in that area, and hence the number of ramps varied for everyone. The overall performance obtained by the 11 different walkers is shown in Fig 5.7(b).

We vary the slope detection threshold used in the ramp detection algorithm from 0.1 to 15 degrees, in steps of 0.1. We vary our assumption for the number of steps, *n*, making up the transition window. This accounts for people with different stride lengths, where the number of steps placed on a ramp alters from one person to another. We tried 3, 5, 7 and 9 as the possible values for ground truth window on the ramp. For each of these, we assumed two steps into the street and the rest on the sidewalk. It is visible from the graph that we get similar performance for different ground truth windows, but the window of 7 gives the best performance. We can derive from this graph that to detect the 90% of ramps we encounter 3% of false positives. Whereas 95% of true positive rate corresponds to 6%



(a) Transition Detection Performance for Manhattan and New Brunswick downtown testbeds.

(b) Ramp Detection Performance for New Brunswick downtown testbed at various ground truth windows.

(c) Ramp Detection Performance for Manhattan testbed at various ground truth windows.

(d) Curb Detection Performance for Manhattan testbed.

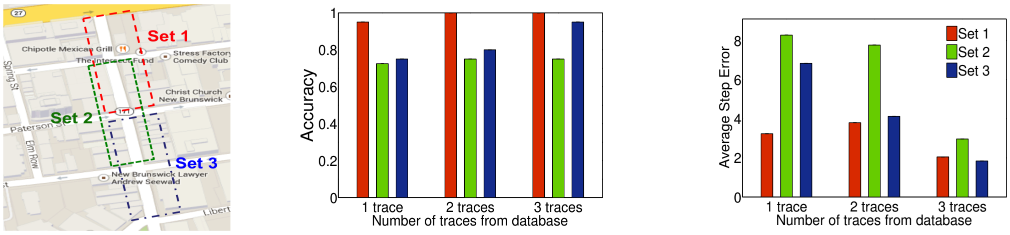
(e) Comparison of performance at different walking rates. Each symbol represents one loop. Similar symbols represent the same person.

(f) Empirical CDF of slope variation, to distinguish between rough and smooth surfaces.

Figure 5.7. System Performance Analysis.

false positive rate. It is noticeable that the performance is affected a lot when the window size is 3. This testbed did not include any curb step off and step on.

Manhattan evaluation. This testbed involved crossing the roadway at the ramp and



(a) Three sets of 4 sidewalks from New Brunswick downtown testbed.

(b) Accuracy of identifying the correct sidewalk, for different database dimension.

(c) Error, in number of steps, in localizing a person on the correct sidewalk.

Figure 5.8. Sidewalk-level localization grounded on pattern matching.

curb. We analyze the performance for each algorithm separately. Fig 5.7(c) shows the ROC curve of the ramp detection algorithm in Manhattan. These curves include the 20 original loops and the 2 longer loops.

Similar to the ramp detection for small downtown area, we varied the ground truth window for the ramp detection in New York over the same range of 3, 5, 7 and 9 steps. It is discernible from the graph that the optimum performance is attained at a ground truth window of 9, but is similar to that at 7. As discussed in the small downtown results, the performance is affected by the size of the ground truth window. With a window of 9, the algorithm can achieve a true detection rate of 90% for just 6% error rate, while for 7, this error rate becomes 8%. So the algorithm will have lower detection rate for pedestrians with longer than average stride length, due to a shorter ground truth window.

Fig 5.7(d) illustrates the performance of the curb detection algorithm. It is apparent from the plot that again, 9 steps provides the optimum performance, while a ground truth window of 3 affects the detection ratio highly. The curb detection also considers ground truth window similar to the ramp detection algorithm. It is worth noticing that the performance is not affected much for a window of 5, 7 and 9. At a window size of 9, we can achieve a true positive rate of 90% at the cost of 6% false positives.

GPS-only approach. To begin with, we wanted to analyze how reliable GPS can be, for our particular problem. We implemented the straw-man approach as discussed in the Applications section. Using the location coordinates of the pedestrian and the heading, both obtained from the GPS, we extrapolate a person's path of motion to approximate his position in the next d meters. We check this extended path for intersection with a nearby street. Such an intersection of the pedestrian's predicted path and the street would indicate that the pedestrian might be purposed to cross the street. The underlying street network data can be accessed from Open Street Map, which is an open-source maps database. Using this technique, we tried to examine our best GPS trace from Manhattan, shown in Fig 5.6(c). This trace was collected using a Nexus 5, that was held in the walker's hand for recording the ground truth data. Even for different values of d, we could not detect any actual crossings. It is also evident from the trace that GPS alone cannot be used for fine grained transition detections between street and sidewalk.

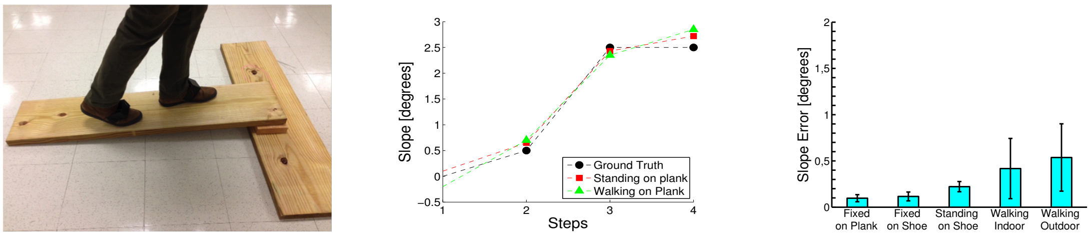
People based evaluation

Walking styles and speeds vary across pedestrians. This leads to difference in duration of the stance phase. To evaluate the effect of variations in walking styles, we conducted our experiments with 11 different people, across our city. Walking speed and stride length are important metrics that distinguish walking styles. We use step frequency as a combined metric for evaluation because it accounts for the changes in speed as well as stride length. Fig 5.7(e) shows the accuracy of the algorithm for different step frequencies. We measure the accuracy in terms of the area under the ROC curve, called AUC, a metric used in machine learning to describe the performance of a classifier. It is the probability that our algorithm will rank a randomly chosen positive instance higher than a randomly chosen negative instance. During the experiments the average walking speed varied between 65 and 95 steps per minutes. To obtain a fair comparison in the frequency calculation, we discarded the time during which a walker was waiting to cross an intersection. This value depends on the traffic condition and not on the walking style. The figure shows the Area Under the Curve (AUC) values for different walkers. Each symbol in the figure represents a loop completed by a walker. Each color and marker style correspond to a different walker. All AUC values range between 0.9 and 0.98. From this graph we can observe that the system has a high accuracy irrespective of the walking speed.

5.7.2 Slope based surface distinction

Apart from transition detections, the slope profiles can help us distinguish the surface a person is walking on, i.e. when a person is walking on the sidewalk and when in the street. This is particularly significant for pedestrian safety applications in situations where a sidewalk is missing and the pedestrian is walking in the street. In such a scenario, a transition detection would not work. Commonly, sidewalks are made of large concrete slabs, while streets are made of asphalt, which results in sidewalks having a more uniform surface than streets. Generally speaking, streets are rough compared to sidewalks, and the slope of the ground changes more often from step to step.

We compare the slope profiles of the traces collected from sidewalk and street. Fig 5.7(f) shows the Empirical CDF plot for the step to step slope variation. We can see that as the variation in slope increases, it becomes more and more probable that the surface is rough, and is more likely to be a street. On an average, the step to step slope variation of a person walking in the street is higher than that of someone walking on the sidewalk. Hence gradient profiles can also be used to distinguish the uniformity of a surface that the pedestrian is walking on.



(a) Indoor ramp set up.

(b) Step vs. Slope at indoor ramp.

(c) Slope measurement error at different error placements.

Figure 5.9. Sensor accuracy analysis.

5.7.3 Pattern matching for sidewalk-level localization

Our localization algorithm allows us to determine the sidewalk a pedestrian is walking on, based on the individual's step pattern. We demonstrate this technique on the data from the city downtown. As shown in Fig 5.8(a), we divide the test bed in three sets of four sidewalks each. Using the approximate location and heading information from GPS, we can discover the appropriate set the user is in. When the pedestrian reaches the end of the block, we update our prediction using the second set of sidewalks and so on.

We match the current trace with different number of traces from the profiling database. Fig 5.8(b) shows the accuracy obtained in detecting a pedestrian's location in each set. We can see that the performance improves as the profiling database grows. Each color represents a different set (Set 1, Set 2 and Set 3 are colored in red, green and blue respectively). An important observation is that the performance is not similar in all the sets, i.e. set 1 and set 3 have always a better accuracy than set 2. This difference is due to the fact that sidewalks located close may have similar features such as average inclination of the ground and same material. For various walking speed we obtain a different number of steps for the same sidewalk and consequently a different pattern.

Once we know which sidewalk the pedestrian is walking on, we try to evaluate the average user location error. We use the difference in number of steps, between the last step of current window from the test trace and the last step of each best matching window from the profiling database. Both last steps represent the number of steps since the pedestrian entered that particular sidewalk. We are aware of the intrinsic limitation of a step-based metric, but our intention is to demonstrate the algorithm effectiveness in matching a test trace to the correct section of a sidewalk more than knowing the absolute pedestrian location. Fig 5.8 (c) shows the average user location error for different database dimensions. The error reduces as the number of walks included in the database increase. When the database contains three walks, we are able to locate the user position with an absolute error less than four steps.

5.7.4 Ground slope accuracy

To understand the accuracy of the underlying ground slope sensing technique and isolate different sources of error, we conducted the following controlled experiments. These experiments compare the sensed slope against ground truth established with a 0.2 degree accurate inclinometer.

To remove errors from foot movement and walking, we first placed the unmounted sensor on a tilted plank and measured its slope. We then repeated this with a shoe-mounted sensor. The process was further repeated for different plank inclinations. Fig 5.9(c) shows the mean error and standard deviation for these experiments, marked as fixed on plank and fixed on shoe, respectively. As can be seen, the errors are below the specified accuracy of our ground-truth inclinometer. We then continue the experiment with shoe that is actually worn, first by having an experimenter stand on the larger wooden plank shown in Fig 5.9(a), which had roughly the same length and inclination as a sidewalk ramp. In a second experiment, the experimenter walked over the plank. The results in the same figure are labeled standing on shoe and walking indoor. They show that standing results in slightly larger errors than the unworn shoe, likely due to some involuntary movements and that walking adds more noticeable but still small errors of about $0.4\,^{\circ}$. We further compare this with walking on an actual sidewalk ramp, where we measured the ground truth inclination on each footstep. The results show an additional slight increase in errors, presumably because the ground is not perfectly uniform, even within the area of a footstep. Overall, the errors are far below the expected slope variations at sidewalk ramps.

5.8 Related work

Pedestrian tracking using inertial sensors has been of interest to the research community for some time. Most inertial sensors based application use some form of dead reckoning for localization [84]. They estimate the distance traveled from a known initial location, by using the stride length and implementing step count, as done by Cho et. al. [79]

Robertson et. al. [91] explore indoor localization for pedestrians using foot-mounted inertial sensors. Jimenez et. al. [88] use ramp detection in indoor environments to provide drift correction in indoor locations. Woodman et. al. [95] developed a tracking system that uses a foot-mounted inertial sensor, a model of a building, and a particle filter to track a pedestrian in an indoor environments.

Gandhi et. al. [86] provide an overview of video, radar and laser distance measurement based approaches for active pedestrian safety. RFID based approach is discussed by Fackelmeier et. al. [81]. Another approach that needs no line of sight and is based on 3G and WLAN is presented by Sugimoto et. al. [92]. David et. al. [80,82,93] present a radio based approach that assumes that the GPS location is precise up to 10 to 80 cm. They also add movement recognition to the radio-based solution [83]. Another pedestrian safety app by Wang et. al. [94] uses the smartphone's camera to detect vehicles approaching the pedestrian when he is talking and walking. This approach works only when the pedestrian is on a call.

Gallo et. al. [85] use TOF camera for curb and ramp detection in the context of safe parking. This approach requires infrastructure that may be hard to equip a pedestrian with. Pedestrian safety is now a high priority for car manufacturers. In AKTIV [58], a German road safety project launched in 2006, cameras and radar sensors installed on the vehicle are used to monitor its surrounding. Many car producers [59, 66, 73] are now integrating night vision, active breaking and automatic steering solutions in their new models to reduce the pedestrian accidents. Honda is developing a Vehicle-to-Pedestrian technology able to detect a pedestrian with a Dedicated Short Range Communication (DSRC) enabled smartphone [71].

5.9 Conclusions and discussion

Motivated by pedestrian safety applications, we explored how effectively shoe-mounted inertial sensors can profile ground gradients and step patterns to detect street-sidewalk transitions [96]. Our results show promise achieving detection rates of 90% at 5-7% false positives, even in the complex midtown Manhattan pedestrian environment. We also showed through first experiments that slope profiles can be matched to specific sidewalk locations, if a map of such profiles is available. While perhaps not yet robust enough on its own, the information could be combined with other positioning information to achieve more precise urban localization. We observed that performance can be sensitive to the mounting method and therefore believe that performance could be further improved with more robust mounting designs.

The approach does, of course, rely on instrumentation of and power in shoes but it could potentially be an additional feature of existing shoe-mounted exercise tracking devices. It might also be suitable for special applications for some of the most vulnerable pedestrians (e.g., children, the elderly, people with disabilities). While our current prototype is not optimized for energy consumption, we expect that future careful circuit design with wake-up when walking and optimized low energy communications could achieve battery lifetimes of months to years, depending on the amount of walking activity.

The transition detection approach also relies on consistent sidewalk designs and it is therefore limited to countries that have implemented such guidelines. At least in the United States and the European Union, we are aware of conscious effort to follow consistent guidelines for improved accessibility, although regional customization of the algorithms may be necessary. In our experiments, we also still encountered some locations with different designs where transitions were hard to detect. We believe that over time consistency in sidewalk design will improve and such efforts will eventually spread around the world.

Overall, we hope that this work demonstrates the broader applicability of shoe-based sensing and inspires developments that go far beyond current exercise tracking applications.

Chapter 6

Conclusions and Future Work

In this thesis, relevant aspects of vehicular systems considering security, efficient content downloading and safety services for pedestrians have been investigated. All these aspects are paramount to the efficient deployment of vehicular networks and to the use of vehicular communications. We proposed easy, original and effective solutions for them. The most relevant results are summarized in the following.

The first problem we investigated was how to ensure secure positioning in vehicular networks. Secure beacons for vehicle position identification and tracking are needed in a number of scenarios where vehicle position accountability is a requirement in order to provide services to the community or to drivers. Secure reporting of vehicle location can substantiate drivers' claims in case of accidents. At the same time, secure location verification by authorities can provide accountability for those involved. However, ensuring secure positioning must cope with three major problems, concerning (i) users' privacy, (ii) computational costs of security and (iii) the system trust on user correctness.

We addressed this issue by proposing a framework that leverages anonymous position beacons from vehicles, which prevent overhearing nodes from identifying or tracking their source, but still allow authorized third parties, sharing secret information with the beaconing vehicles, to perform such operations. Then, an authenticated reciprocal beacon reporting scheme grants an authority the possibility to verify the locations claimed by vehicles and infer unverified positions by efficiently solving an optimization problem. We proved that our solution is capable of achieving its goals in both dense and sparse vehicular settings through simulation and experiments in real-world testbeds.

Studying the first problem, we noticed the limited channel capacity offered by the 5-GHz bands. We therefore explored the benefit given by the use of low frequency bands for the transmission of control messages. Specifically, we focused on content downloading, and design a protocol that leverages the UHF band for control messages (vehicles position and contents requests) and the high-throughput, 5-GHz bands for data delivery. We assessed the benefits of exploiting UHF bands, providing much larger coverage than the 5-GHz frequencies, through a vehicular testbed. Our experimental results show that a 3x throughput gain in content delivery can be achieved with respect to the case where only 5-GHz bands are used.

Then, motivated by the significant number of traffic accidents with pedestrian, we proposed two different localization solutions for pedestrian safety services in outdoor environments. The first framework we designed uses the GPS position and a map to detect when a pedestrian is going to cross a street. The results obtained show that this approach is feasible in rural and suburban areas while it is not applicable in dense urban environments. Indeed, large errors in positioning and delays in detection make the GPS-based approaches not suitable for critical applications in urban areas. The second solution we designed was specifically targeting urban scenarios. In order to overcome the limitations of our GPS-based approach, we proposed a new technique based on shoe-mounted inertial sensors. We designed an algorithm able to detect transitions from the sidewalk to street, such as stepping off the curb or walking over ramps, leading into designated pedestrian crossings. Our results achieves detection rates of 90% at 5-7% false positives, even in the complex midtown Manhattan pedestrian environment. The transition detection approach also relies on consistent sidewalk designs and it is therefore limited to countries that have implemented such guidelines. At least in the United States and the European Union, we are aware of conscious effort to follow consistent guidelines for improved accessibility, although regional customization of the algorithms may be necessary.

Future work will focus on improving previous vehicular studies with experimental tests on more complex road topologies. Regarding the pedestrian localization, energy consumption and pedestrian-to-vehicle communication solutions will be developed. Indeed, our current prototype is not optimized for energy consumption. We expect that future careful circuit design with wake-up mechanism when walking and optimized low energy communications could achieve battery lifetimes of months to years, depending on the amount of walking activity. Moreover, additional effort will be invested in integrating our framework within vehicular networks. In the following years, all vehicles are expected to announce their current position and other vehicle dynamics information to vehicles in their vicinity. Recently, prototype smartphones that can participate in such an 802.11p based network have been released. Such smartphones can learn about oncoming cars and announce their position to nearby vehicles. Exploiting the information provided by our system, vehicles can distinguish pedestrians that are safely walking on the sidewalk from pedestrians that may be at risk.

Bibliography

[1] Thales ISS, Thales, Jan. 2011 [Accessed on July 2012].

[2] B. Wiedersheim *et al.*, “Privacy in Inter-Vehicular Networks: Why Simple Pseudonym Change Is not Enough,” *IEEE WONS*, 2010.

[3] E. Schoch, F. Kargl, “On the Efficiency of Secure Beaconing in VANETs,” *ACM WiSec*, 2010.

[4] IEEE 1609.2 Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages, 2013.

[5] J. Hwang, T. He, Y. Kim, “Detecting Phantom Nodes in Wireless Sensor Networks,” *IEEE Infocom*, Anchorage, AK, May 2007.

[6] E. Ekici, S. Vural, J. McNair, D. Al-Abri, “Secure Probabilistic Location Verification in Randomly Deployed Wireless Sensor Networks,” *Elsevier Ad Hoc Networks*, vol. 6, no. 2, pp. 195-209, 2008.

[7] S. Čapkun, K. Rasmussen, M. Cagalj, M. Srivastava, “Secure Location Verification with Hidden and Mobile Base Stations,” *IEEE Trans. on Mobile Computing*, vol. 7, no. 4, pp. 470–483, 2008.

[8] J. Chiang, J. Haas, Y. Hu, “Secure and Precise Location Verification Using Distance Bounding and Simultaneous Multilateration,” *ACM WiSec*, Zurich, Switzerland, Mar. 2009.

[9] T. Leinmüller, C. Maihöfer, E. Schoch, F. Kargl, “Improved security in geographic ad hoc routing through autonomous position verification,” *ACM VANET*, Los Angeles, CA, Sept. 2006.

[10] T. Leinmüller, E. Schoch, F. Kargl, “Position Verification Approaches for Vehicular Ad Hoc Networks,” *IEEE Wireless Communications*, pp. 15–21, Oct. 2006.

[11] J.-H. Song, V. Wong, V. Leung, “Secure Location Verification for Vehicular Ad-Hoc Networks,” *IEEE Globecom*, New Orleans, LO, Dec. 2008.

[12] M. Fiore, C. Casetti, C.-F. Chiasserini, P. Papadimitratos, “Discovery and Verification of Neighbor Positions in Mobile Ad Hoc Networks,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, 2013.

[13] M. Abu-Elkheir, S.A. Hamid, H.S. Hassanein, I.M. Elhenawy, S. Elmougy, “Position verification for vehicular networks via analyzing two-hop neighbors information,” *IEEE LCN On-Move*, Bonn, Germany, Oct. 2011.

[14] O. Abumansoor, A. Boukerche, "A secure Cooperative Approach for Non line-of-Sight Location Verification in VANET," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 275–285, 2012.

[15] Z. Ren, W. Li, Q. Yang, "Location Verification for VANETs Routing," *IEEE WiMob*, Marrakech, Morocco, Oct. 2009.

[16] X. Xue, N. Lin, J. Ding, Y. Ji, "A trusted neighbor table based location verification for VANET routing," *IET ICWMMN*, Beijing, China, Jan. 2010.

[17] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure Vehicular Communication Systems: Design and Architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[18] K. Sampigethaya, M. Li, L. Huang, R. Poovendran, "AMOEBA: Robust Location Privacy Scheme for VANET," *IEEE JSAC*, vol.25, no.8, pp.1569–1589, Oct. 2007

[19] D. Eckhoff, C. Sommer, T. Ganseny, R. German, F. Dressler, "Strong and Affordable Location Privacy in VANETs: Identity Diffusion Using Time-Slots and Swapping," *IEEE VNC*, 2010.

[20] W. Diffie, M. Hellman, "Privacy and Authentication: An Introduction to Cryptography," *Proceedings of the IEEE*, vol. 67, no. 3, pp. 397–427, Mar. 1979.

[21] F. Klingler, F. Dressler, J. Cao, C. Sommer, "Use Both Lanes: Multi-channel Beaconing for Message Dissemination in Vehicular Networks," *WONS,* 2013.

[22] C. Sommer, D. Eckhoff, R. German, F. Dressler, "A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments," *IEEE WONS*, 2011.

[23] S.P. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[24] S. Chang, Y. Qi, Ho. Zhu, J. Zhao, X. Shen, "Footprint: Detecting Sybil Attacks in Urban Vehicular Networks," *IEEE Trans. on Paralled and Distributed Systems*, vol. 23, no. 6, pp. 1103–1114, Jun. 2012.

[25] F. Malandrino, C. Borgiattino, C. Casetti, C.-F. Chiasserini, M. Fiore, R. Sadao Yokoyama, *"Verification and Inference of Positions in Vehicular Networks through Anonymous Beaconing"*, in Transactions on Mobile Computing (TMC), Vol. 13, n. 10, pp. 2415-2428, ISSN 1536-1233

[26] F. Malandrino, C. Casetti, C.-F. Chiasserini, M. Fiore, R.S. Yokoyama, C. Borgiattino, *"A-VIP: Anonymous Verification and Inference of Positions in Vehicular Networks"*, in IEEE INFOCOM MiniConference, Torino (Italy), April 2013, pp. 105-109

[27] G. Staple, K. Werbach, "The End of Spectrum Scarcity [Spectrum Allocation and Utilization]," *IEEE Spectrum,* vol. 41, no. 3, pp. 48–52, 2004.

[28] D. Niyato, E. Hossain, P. Wang, "Optimal Channel Access Management with QoS Support for Cognitive Vehicular Networks," *IEEE Transactions on Mobile Computing,* vol. 10, no. 5, pp. 573–591, 2011.

[29] K. Fawaz, A. Ghandour, M. Olleik, H. Artail, "Improving Reliability of Safety Applications in Vehicle Ad hoc Networks through the Implementation of a Cognitive Network," *IEEE International Conference on Telecommunications,* Doha, Qatar, Apr. 2010.

[30] N. Kirsch, B.M. O'Connor, "Improving the Performance of Vehicular Networks in High Traffic Density Conditions with Cognitive Radios," *IEEE Intelligent Vehicles Symposium,* Baden-Baden, Germany, June 2011.

[31] W. Kim, S.Y. Oh, M. Gerla, K.C. Lee, "CoRoute: A New Cognitive Anypath Vehicular Routing Protocol," *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC),* Istanbul, Turkey, July 2011.

[32] R. Sevlian, C. Chun, I. Tan, A. Bahai, K. Laberteaux, "Channel Characterization for 700 MHz DSRC Vehicular Communication," *Journal of Electrical and Computer Engineering,* Jan. 2010.

[33] G. Marfia, M. Roccetti, A. Amoroso, M. Gerla, G. Pau, J.-H. Lim, "Cognitive Cars: Constructing a Cognitive Playground for VANET Research Testbeds," *ACM Conference on Cognitive Radio and Advanced Spectrum Management (CogART 2011),* Barcelona, Spain, Oct. 2011.

[34] A. Masini, G. Mazzini, A. Ghittino, M. Maglioli, N. Di Maio, G. Riva, "WhiteFi: the Usage of UHF Frequencies for Bi-directional Services in Mountainous Scenarios," *International Conference on Electromagnetics in Advanced Applications,* Turin, Italy, Sep. 2011.

[35] C. Borgiattino, C. Casetti, C.-F. Chiasserini, N. Di Maio, A. Ghittino, M. Reineri, *"Experiences with UHF bands for content downloading in vehicular networks"*, in IEEE Wireless Communication and Networking Conference (WCNC) Workshop, Parigi, France, April 2012, pp. 333-337

[36] ito world. `http://goo.gl/sWMVHT`.

[37] Openstreetmap. `http://www.openstreetmap.org/`.

[38] Openstreetmap wiki. `http://goo.gl/66Vlki`.

[39] UK Daily Mail. Texting while walking blamed in the nationwide increase of pedestrian deaths. `http://goo.gl/kaKQgc`.

[40] K. David and A. Flach, "Car-2-x and pedestrian safety", *Vehicular Technology Magazine*, 2010.

[41] Andreas Fackelmeier, Christian Morhart, and Erwin Biebl, "Dual frequency methods for identifying hidden targets in road traffic", In *Advanced Microsystems for Automotive Applications*. 2008.

[42] A. Flach and K. David, "Combining radio transmission with filters for pedestrian safety: Experiments and simulations", In *Vehicular Technology Conference Fall*, 2010.

[43] A. Flach, A.Q. Memon, Sian Lun Lau, and K. David, "Pedestrian movement recognition for radio based collision avoidance: A performance analysis", In *Vehicular Technology Conference (VTC Spring)*, 2011.

[44] Transportation for America. Dangerous by design, 2011.

[45] T. Gandhi and M.M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges", *Intelligent Transportation Systems, IEEE Transactions on*, 2007.

[46] Boston Globe. Put that phone down and just walk. `http://goo.gl/EwgrMg`.

[47] Liberty Mutual Insurance. Study shows three out of five pedestrians prioritize smartphones over safety when crossing street, June 2013.

[48] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. "Sensloc: sensing everyday places and paths using less energy". In *Embedded Networked Sensor Systems (SenSys), 2010.*

[49] New Jersey Department of Transportation. Roadway design manual.

[50] U.S Department of Transportation. Traffic safety facts, August 2013.

[51] C. Qin, X. Bao, R. Roy Choudhury, and S. Nelakuditi. "Tagsense: a smartphone-based approach to automatic image tagging". In *International Conference on Mobile Systems, Applications, and Services (MobiSys), 2011.*

[52] C. Sugimoto, Y. Nakamura, and T. Hashimoto. "Prototype of pedestrian-to-vehicle communication system for the prevention of pedestrian accidents using both 3g wireless and wlan communication". In *3rd International Symposium on Wireless Pervasive Computing*, 2008.

[53] C. Voigtmann, Sian Lun Lau, and K. David. Evaluation of a collaborative-based filter technique to proactively detect pedestrians at risk. In *Vehicular Technology Conference (VTC Fall)*, 2012.

[54] Tianyu Wang, Giuseppe Cardone, Antonio Corradi, Lorenzo Torresani, and Andrew T. Campbell. "Walksafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads", In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, HotMobile '12.

[55] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen. "Iodetector: A generic service for indoor outdoor detection", In *Embedded Networked Sensor Systems (SenSys), 2012.*

[56] S. Jain, C. Borgiattino, Y. Ren, M. Gruteser, Y. Chen, *"On the Limits of Positioning-based Pedestrian Risk Awareness"*, in ACM MobiSys Workshop (MARS), Bretton Woods, USA, June 2014, pp. 23-28

[57] Adidas Speed Cell. `http://goo.gl/Xs9CXK`.

[58] Aktiv project. `http://www.aktiv-online.org/english/aktiv-as.html#KAS`.

[59] Audi: Night vision assistant with highlighting of detected pedestrians. `http://goo.gl/21GD2g`.

[60] European Union Pedestrian Planning. `http://goo.gl/r2QNsT`.

[61] International Transport Forum, Road Safety Annual Report 2013.

[62] Openstreetmap. `http://www.openstreetmap.org/`,.

[63] Reversal in Three-Year Uptick in Pedestrian Fatalities. `http://goo.gl/I389KR`.

[64] Transportation for America, Dangerous by Design 2011. `http://t4america.org/docs/dbd2011/Dangerous-by-Design-2011.pdf`.

[65] U.S. Department of Transportation, Dedicated Short Range Communication. `http://www.its.dot.gov/factsheets/dsrc_factsheet.htm`.

[66] Volvo car corporation's emergency brake assist. `http://www.volvocars.com/intl/top/corporate/volvo-sustainability/pages/sustainability-news.aspx?itemid=280`, 2011.

[67] The Big Story, NYC Crosswalks urge pedestrians to look! `http://bigstory.ap.org/article/nyc-crosswalks-urge-pedestrians-look`, September 2012.

[68] Delaware Online, if you text and walk, remember to look up. `http://goo.gl/ww5pRc`, May 2012.

[69] New York Times, Deaths Rise for Drivers, Bikers and Walkers on City Streets. `http://goo.gl/DFhqP`, 2012.

[70] Nike+ Sensor. `http://www.apple.com/ipod/nike/`, 2012.

[71] Honda Demonstrates Advanced Vehicle-to-Pedestrian and Vehicle-to-Motorcycle Safety Technologies. `http://www.honda.com/newsandviews/article.aspx?id=7352-en`, August 2013.

[72] Liberty Mutual Insurance, Study shows three out of five pedestrians prioritize smartphones over safety when crossing street. `http://goo.gl/5z2DzQ`, June 2013.

[73] Toyota develops new pedestrian safety technology. `http://www.toyota.com/esq/safety/active-safety/toyota-develops-new-pedestrian-safety-technology.html`, 2013.

[74] U.S. Department of Transportation, Traffic Safety Facts, August 2013.

[75] U.S. Department of Transportation announces decision to move forward with vehicle-to-vehicle communication technology for light vehicles. `http://goo.gl/UlrBzx`, February 2014.

[76] Federal Highway Administration, Crash-Type manual for pedestrians, April 1997.

[77] Federal Highway Administration, Sidewalk Design Guidelines, July 1999.

[78] Berndt, D. J., and Clifford, J. Using Dynamic Time Warping to Find Patterns in Time Series. In KDD Workshop (1994), pp. 359-370.

[79] Seong Yun Cho and Chan Gook Park. "Mems based pedestrian navigation system", *Journal of Navigation*, 59(01), pp. 135-153, 2006.

[80] K. David and A. Flach. " Car-2-x and pedestrian safety", *Vehicular Technology Magazine, IEEE*, 5(1), 2010.

[81] Andreas Fackelmeier, Christian Morhart, and Erwin Biebl. "Dual frequency methods for identifying hidden targets in road traffic", In *Advanced Microsystems for Automotive Applications 2008*, pp. 11-20. Springer, 2008.

[82] A. Flach and K. David. "Combining radio transmission with filters for pedestrian safety: Experiments and simulations", In *Vehicular Technology Conference Fall*

(VTC 2010-Fall), 2010 IEEE 72nd, 2010.

[83] A. Flach, A.Q. Memon, Sian Lun Lau, and K. David. "Pedestrian movement recognition for radio based collision avoidance: A performance analysis", In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, 2011.

[84] E. Foxlin. "Pedestrian tracking with shoe-mounted inertial sensors", *Computer Graphics and Applications, IEEE*, 25(6), pp. 38-46, Nov 2005.

[85] Orazio Gallo, Roberto M, and Abbas Rafii. "Robust curb and ramp detection for safe parking using the canesta tof camera", In *Computer Vision and Pattern Recognition Workshop*, 2008.

[86] T. Gandhi and M.M. Trivedi. "Pedestrian protection systems: Issues, survey, and challenges", *Intelligent Transportation Systems, IEEE Transactions on*, 8(3), 2007.

[87] Invensense. Invensense MPU-9150 product specification. `http://www.invensense.com/mems/gyro/documents/PS-MPU-9150A-00v4_3.pdf`.

[88] Zampella F. Prieto J.C. Guevara J. Jimenez A.R., Seco F. "Pdr with a foot-mounted imu and ramp detection", *Sensors*, 11(10),9393–9410, Oct 2011.

[89] Peter D. Loeb and William A. Clarke. "The cell phone effect on pedestrian fatalities", *Transportation Research Part E: Logistics and Transportation Review*, 45(1), pp. 284-290, 2009.

[90] Kazushige Ouchi and Miwako Doi. "Indoor-outdoor activity recognition by a smartphone", In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pp. 537-537, New York, NY, USA, 2012. ACM.

[91] Patrick Robertson, Michael Angermann, and Bernhard Krach. "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors", In *Proceedings of the 11th International Conference on Ubiquitous Computing*, Ubicomp '09, pp. 93-96, New York, NY, USA, 2009. ACM.

[92] C. Sugimoto, Y. Nakamura, and T. Hashimoto. "Prototype of pedestrian-to-vehicle communication system for the prevention of pedestrian accidents using both 3g wireless and wlan communication", In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, 2008.

[93] C. Voigtmann, Sian Lun Lau, and K. David. "Evaluation of a collaborative-based filter technique to proactively detect pedestrians at risk", In *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, 2012.

[94] Tianyu Wang, Giuseppe Cardone, Antonio Corradi, Lorenzo Torresani, and Andrew T. Campbell. "Walksafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads", In *Proceedings of the Twelfth Workshop on Mobile Computing Systems; Applications*, HotMobile '12, 2012.

[95] Oliver Woodman and Robert Harle. "Pedestrian localisation for indoor environments", In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, pp. 114-123, New York, NY, USA, 2008. ACM.

[96] S. Jain, C. Borgiattino, Y. Ren, M. Gruteser, Y. Chen, C.-F. Chiasserini *“LookUp: Enabling Pedestrian Safety Services via Shoe Sensing”*, in ACM MobiSys, Florence, Italy, May 2015.