CycleGuard: A Smartphone-based Assistive Tool for Cyclist Safety Using Acoustic Ranging

WENQIANG JIN[*][†], Hunan University, China
SRINIVASAN MURALI[‡], The University of Texas at Arlington, USA
YOUNGTAK CHO, The University of Texas at Arlington, USA
HUADI ZHU, The University of Texas at Arlington, USA
TIANHAO LI, The University of Texas at Arlington, USA
RACHAEL THOMPSON PANIK, Georgia Institute of Technology, USA
ANIKA RIMU, The University of Texas at Arlington, USA
SHUCHISNIGDHA DEB, The University of Texas at Arlington, USA
KARI WATKINS, Georgia Institute of Technology, USA
XU YUAN, University of Louisiana at Lafayette, USA
MING LI, The University of Texas at Arlington, USA

Every year 41,000 cyclists die in road traffic-related incidents worldwide [47]. One of the most startling and infuriating conflicts that cyclists experience is the so-called "right hook". It refers to a vehicle striking a cyclist heading in the same direction by turning right into the cyclist. To prevent such a crash, this work presents CycleGuard, an acoustic-based collision detection system using smartphones. It is composed of a cheap commercial off-the-shelf (COTS) portable speaker that emits imperceptible high-frequency acoustic signals and a smartphone for reflected signal reception and analysis. Since received acoustic signals bear rich information of their reflecting objects, CycleGuard applies advanced acoustic ranging techniques to extract those information for traffic analysis. Cyclists are alerted if any pending right hook crashes are detected. Real-time alerts ensure that cyclists have sufficient time to react, apply brakes, and eventually avoid the hazard. To validate the efficacy of CycleGuard, we implement a proof-of-concept prototype and carry out extensive in-field experiments under a broad spectrum of settings. Results show that CycleGuard achieves up to 95% accuracy in preventing right hook crashes and is robust to various scenarios. It is also energy-friendly to run on battery-powered smartphones.

[*] The work was done when the author was a Ph.D. student at The University of Texas at Arlington.
[†] Equal contribution.
[‡] Equal contribution.

Authors' addresses: Wenqiang Jin, Hunan University, Changsha, Hunan, China, wqjin@hnu.edu.cn; Srinivasan Murali, The University of Texas at Arlington, Arlington, TX, USA, srinivasan.murali@mavs.uta.edu; Youngtak Cho, The University of Texas at Arlington, Arlington, TX, USA, youngtak.cho@mavs.uta.edu; Huadi Zhu, The University of Texas at Arlington, Arlington, TX, USA, huadi.zhu@mavs.uta.edu; Tianhao Li, The University of Texas at Arlington, Arlington, TX, USA, tianhao.li@mavs.uta.edu; Rachael Thompson Panik, Georgia Institute of Technology, Atlanta, GA, USA, rtpanik@gatech.edu; Anika Rimu, The University of Texas at Arlington, Arlington, TX, USA, anika.rimu@mavs.uta.edu; Shuchisnigdha Deb, The University of Texas at Arlington, Arlington, TX, USA, shuchi.deb@uta.edu; Kari Watkins, Georgia Institute of Technology, Atlanta, GA, USA, kari.watkins@ce.gatech.edu; Xu Yuan, University of Louisiana at Lafayette, Lafayette, LA, USA, xu.yuan@louisiana.edu; Ming Li, The University of Texas at Arlington, Arlington, TX, USA, ming.li@uta.edu.

2474-9567/2021/12-ART163 $15.00
https://doi.org/10.1145/3494992

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Cyclist safety, collision avoidance, acoustic ranging

ACM Reference Format:
Wenqiang Jin, Srinivasan Murali, Youngtak Cho, Huadi Zhu, Tianhao Li, Rachael Thompson Panik, Anika Rimu, Shuchisnigdha Deb, Kari Watkins, Xu Yuan, and Ming Li. 2021. CycleGuard: A Smartphone-based Assistive Tool for Cyclist Safety Using Acoustic Ranging. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 163 (December 2021), 30 pages. https://doi.org/10.1145/3494992

1 INTRODUCTION

In the US, over 36,560 bicyclist-related crashes were reported in 2017. More than 800 of them resulted in deaths. This number rose to over 850 in 2018, accounting for a 6.3% increase from the fatalities in 2017 [40]. Bicycles may be the most difficult detection problem that autonomous vehicle systems face [14]. Compared with a car, a bicycle has much less mass and more variation in appearance. Because of this, cyclists are vulnerable on roads in the era of autonomous vehicles. In addition to being difficult to detect, bicycle users are more vulnerable in crashes than vehicular travelers, which makes the detection a compounding issue. According to a study of crash data over 10 years in Minneapolis [7], 41% of bicyclist-motorist crashes happen at intersections, and another 40% occur within 50 feet of intersections. Moreover, the most common pre-crash maneuvers for cyclists are vehicle following roadway (42%) and vehicle making a right turn (16.4%).

Often the focus of cyclist safety research centers on mitigating crashes through infrastructure design, policy changes, and educational programs. Automotive companies have developed a number of collision avoidance systems (CAS). Many of them utilize LiDAR, radar sensors, or a combination of these. However, these sensors are too big and too expensive (typically costing thousands of dollars) for a bicycle. Camera-based CAS such as Mobileye [33] has also been developed for cars. In addition to the concern of cost, battery is another issue. There is a growing interest in using V2X [6] for cyclist safety. The idea is to exchange positioning information between vehicles and bikes for collision avoidance. However, cyclists cannot expect all the vehicles to be instrumented with the above-mentioned advanced techniques for their safety. It is likely to take decades before such systems can achieve adequate penetration among vehicles.

As opposed to automotive research, somewhat limited research efforts are devoted to developing protection technologies for bicycles. To our knowledge, sensor systems for bicycles have been explored by a few research teams [22, 23, 56, 65, 66] and companies [13, 28, 29]. To detect the presence of nearby vehicles, conventional sensing modalities have been employed in prior works [22, 23, 29, 65, 66], in which special-purpose sensory devices are built. The price is typically at least hundreds of dollars. Besides, cyclists need to have experienced skills to install and calibrate the sensors for accurate detection. Another common approach [13, 28, 56] uses cameras to sense traffic around the cyclists. However, its performance is susceptible to lighting conditions. Among the existing technologies, the idea that is closest to ours is from Kawanaka et al. [24]. They leveraged built-in microphones on smartphones to detect nearby vehicles by analyzing wind and tire sound. Nevertheless, the received sound can be easily polluted by environmental noises.

Given that the “right hook” is one of the most prevalent bike-car crash types (more than 16%) [7, 58], we aim to protect cyclists from such a hazard, especially near an intersection where the traffic is more dense and complex. Instead of building a new sensory device, our system, called CycleGuard, is composed of a smartphone and a cheap commercial off-the-shelf (COTS) portable speaker. CycleGuard emits ultrasonic signals continuously and analyzes the reflections to detect potential right-hook collisions. The entire system runs on the smartphone in realtime. Cyclists will be alerted immediately once a potential hazard is detected.

To achieve the above goal, it is essential to figure out the motion status of each vehicle nearby, characterized by, for example, its relative distance to the bicycle, relative direction, and moving trajectory. To this end, we exploit advanced acoustic ranging techniques that take advantage of smartphone's audio hardware to characterize motions status of vehicles nearby. Nowadays, smartphones are typically equipped with more than two microphones to serve various applications, such as voice recognition, stereo recording, and noise cancellation. They provide a high sampling frequency and a wide frequency response range. For example, Galaxy S5, S8+, iPhone 7, and iPhone 8 exhibit good frequency response up to 23 KHz. CycleGuard uses high-frequency acoustic signals inaudible to human ears. Their reflections, bounced off from nearby vehicles, bear rich information regarding their moving pattern. They are then picked up by the smartphone's dual microphones for further analysis.

Although the idea of acoustic sensing is not new, turning it into a tool for collision detection is faced with several unique challenges. First, both the speaker and microphones are omnidirectional. Received signals contain not only reflections from vehicles but also all the surroundings, which include the cyclist's body, bike, trees, and other infrastructures. The interference needs to be removed to enhance the signal quality and bring out the vehicle-related reflections prominently. To achieve this, we model the multi-path propagation between the speaker and reflective objects by using the least-square estimation model. Interference caused by objects close to the cyclist is then reconstructed and subtracted from the received signal. The target vehicle is further identified from cross-correlations between the transmitted signals and received reflections. Its relative position to the bike is characterized by the relative distance and direction. The former is calculated based on the reflection's round trip time. In practice, the speaker and microphones are not perfectly synchronized due to software processing latency. It will result in non-negligible errors in calculating cross-correlations. To address this issue, we first benchmark the latency and then take it into account during the calculation. Relative directions are then estimated from the time-difference-of-arrival (TDoA) between received reflections at two microphones. With the vehicle's time-series relative positions, we are then able to track its trajectory and decide if the target vehicle tends to make a right turn. Our design further deals with the impact from arbitrary handlebar maneuvers. In practice, cyclists adjust the handlebar's orientation from time to time to keep the bicycle balanced while riding. As a result, the steering maneuver changes the phone's orientation and thus impacts the detection accuracy of CycleGuard. Our approach first characterizes the relationship of vehicle's positions before and after the steering maneuver. The relationship is then employed to compensate for the offset in the measures caused by the steering maneuver.

CycleGuard cannot always detect a right-turn vehicle faster than a cyclist, especially when the cyclist is fully focused. We treat CycleGuard as an assistive tool that provides an added layer of protection for cyclists on top of their own visual and auditory sensing system to avoid collisions. CycleGuard would be effective in protecting cyclists in the following scenarios.

Cyclists may fail to quickly detect right turning vehicles under poor lighting conditions by solely relying on their visual observations. In the meantime, they can become hard to spot by car drivers from rearview mirrors because of the darkness. According to a report from National Highway Traffic Safety Administration (NHTSA) [40], cyclist fatalities were 54% during lower visible times of the day in dawn, dusk and night-time, compared to 46% during daylight. Since CycleGuard relies on acoustic signals for detection, it works regardless of lighting conditions. It is not rare that cyclists get distracted when riding. In a study regarding distracted biking in the Boston area, a total of 1,974 bicyclists were observed. Among them, a total of 615 (31.2%) were distracted [64]. Moreover, in the case of Spain, the analysis of the cyclists' crash reports from the period 2008–2013 showed that up to 89.3% of the 25,439 traffic crashes suffered by cyclists involved cycling distractions as one of their concurrent causes [35]. Potential distraction sources include cyclist's headphones, road-side billboards, and behavior of other road users. Our app can protect cyclists in these cases too.

The contributions of our work are summarized as follows.

- We develop CycleGuard, a smartphone-based collision detection system, to enhance cyclist safety at intersections by assisting them in taking appropriate reactions in the presence of frontal right-turn vehicles.

Although vehicular collision avoidance systems have been investigated for a long time, similar protection for bicycles has rarely been studied.

- We address distinct challenges when acoustic ranging is applied to collision detection in the form of two pivotal contributions. Firstly, we deal with signal interference from the cyclist's surrounding objects. With this basis, we further tackle the non-negligible cross-correlation errors caused by software processing latency when deriving the target vehicle's position. We also develop an effective approach to compensate the measurement offsets in the target vehicle's positions caused by arbitrary handlebar maneuvers.
- A novel method is proposed that exploits the position curve of each object to identify the right-turn vehicle with the presence of other static and non-static objects.
- We implement CycleGuard on a COTS speaker and a smartphone without involving any other sophisticated sensors. We conduct a systematic literature review added with testing in an indoor simulation environment to derive suitable parameters for selected components in CycleGuard and validate the feasibility of CycleGuard via extensive in-field testings.

2 RELATED WORK

2.1 Cyclist Safety Techniques

Cyclist safety techniques are designed to prevent or reduce the severity of collisions between cyclists and other objects on the road. A common approach uses cameras to observe the surrounding traffic and applies computer vision-based algorithms to detect and alert oncoming vehicles [13, 28, 56]. However, their performance is subject to lighting conditions. Some prior works investigate other sensing techniques for cyclist safety, such as LiDAR [66], laser [22, 23], sonar, or their fusion [65]. They rely on sophisticated sensors that are typically expensive and large in size. Hence, these techniques have not been widely adopted so far. Hisaka et al. [19] proposed to install ZigBee receivers on vehicles to capture the received Zigbee signals transmitted from bicycles for collision detection. This approach requires the installment of specialized devices for all vehicles, rendering it impractical to deploy. Kawanaka et al. [24] leveraged the smartphone's built-in microphone to detect approaching vehicles from the wind and tire noises. However, audio recordings can be easily polluted by background noises coming from constructions and other traffic. So far, most existing works focus on the detection of vehicles approaching from behind. Instead, we target a more challenging cyclist safety issue; a careless driver cuts in front of the cyclist to make a right turn at an intersection. Such a scenario is much more complicated to handle. In addition to estimating the vehicle's distance to the bicycle, we need to further analyze and trace the target's trajectory. Hence, none of the prior approaches is applicable here.

An alternative solution is to deploy the collision avoidance system (CAS) in vehicles. Since the demonstration of the first modern version in 1995, it has attracted massive attention in the past decades. Vehicle manufacturers, like Audi [1], Tesla [57], Mercedes Benz [4], and Volvo [11], have been developing generations of CAS to alert the driver of their surroundings on the road, including the cyclists. Nonetheless, CAS is primarily designed to protect vehicle drivers from collisions, while we focus on cyclist's safety. Besides, not all vehicles are equipped with advanced CAS to date.

2.2 Acoustic-Based Collision Avoidance for Pedestrian Safety

There has been some research leveraging acoustic signals to sense the surrounding environment for pedestrian mobility safety, especially pedestrians with vision impairments. For instance, UltraSee [63] uses smartphones to transmit ultrasonic signals to detect sudden changes on the ground and alerts pedestrians about dangerous road conditions such as tripping, stairs, and open manholes. BumpAlert [60] proposes to jointly utilize the smartphone camera, speaker, and microphones for obstacle detection. ObstacleWatch [62] provides more fine-grained information about sidewalk obstacles, such as their sizes and directions, and thus provides more accurate detection. To improve the navigation experience of blind people, researchers have augmented regular white canes with advanced sensors to derive the obstacle's distance [10, 30, 54, 67] and shape [51, 55, 70] to decide if the environment is hazardous to users. However, all these works aim to detect static or slow-moving obstacles, e.g., walls, tables, trees, and other pedestrians nearby, while we target moving vehicles. Auto++ [26] records the traffic sounds and uses machine learning to identify approaching cars. As a note, none of the above studies or products are relevant to right-turn vehicle detection, which highlights this work's unique contribution to the field of cyclist safety.

2.3 Other Applications of Acoustic Sensing

A plethora of works have revolved around the field of acoustic sensing. For example, FingerIO [38], Strata [68], and Vernier [69] provide finger tracking on device surface using built-in speakers and microphones of smartphones. A similar idea is adopted in EarphoneTrack [9] that employs earphones for finger motion tracking. Beyond that, acoustic sensing has been explored in monitoring user's breathing pattern [37], human body activities behind the wall [39], object detection [32], indoor mapping [49], and acoustic ranging [31]. The above systems consider a short ranging distance within a couple of meters. Since we aim to detect motion status of nearby vehicles, the ranging distance is significantly longer, up to 30 meters. Because of the long ranging distance, the received signals in our scenario are composed of interference/reflections from various objects, including static ones (e.g., trees, buildings, and roadside trash-bins) and non-static ones (e.g., other pass-by vehicles). Thus, as an indispensable part of our design, CycleGuard should be capable of extracting useful ingredients from the highly polluted received signals. Further, it needs to reconstruct the target vehicle's trajectory from received signals, and more importantly, to decide whether it would pose any threat to the cyclist. In summary, the existing approaches are not readily applicable to our scenario.

3 PRELIMINARIES

3.1 Design Rationale

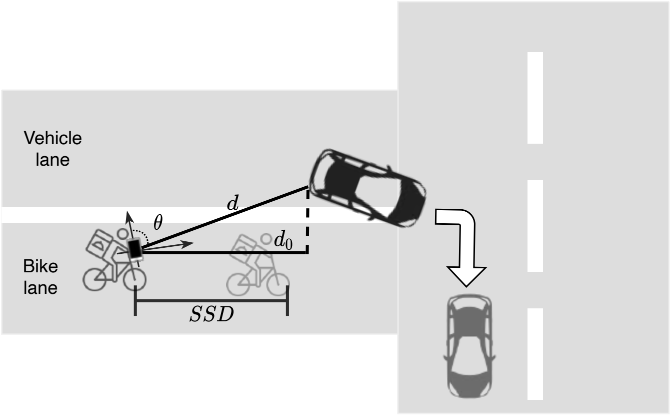


Fig. 1. Design rationale of CycleGuard.

When a bicycle is in the blind spot of a driver, the cyclist could be in danger caused by the driver's right-turn maneuver, as shown in Fig. 1. To avoid this hazardous situation, CycleGuard continuously monitors frontal traffic and alerts the cyclist if any potential collision is forecasted. Denote by d_0 the bicycle-vehicle (b-v) distance on

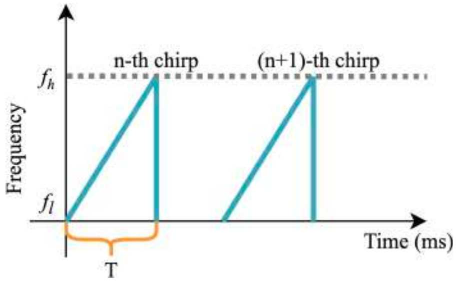


Fig. 2. Time-frequency representation of chirp signals.

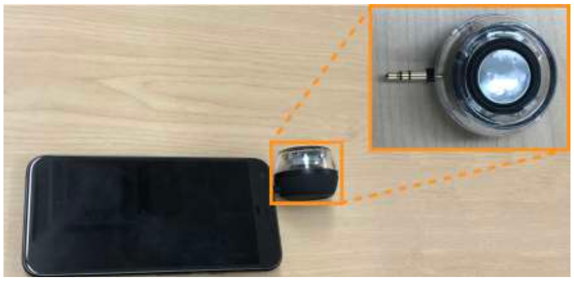


Fig. 3. Connecting COTS speaker with smartphone.

the biking direction, the moment the vehicle starts to make a right turn. Meanwhile, the cyclist reacts to the perceived turning vehicle and applies brakes to stop the bicycle. The distance that the bicycle travels before a complete stop is called stopping sight distance (SSD). It is a near worst-case distance the cyclist needs to be able to see, react, and brake to have room to stop before colliding with the vehicle. If $d_0 > SSD$, the cyclist can stop the bicycle within SSD. This inequality can be used as one of the conditions for cyclist safety. A vehicle that stays in the vehicle lane and moves ahead will not pose any threat to the cyclist. This is the same case for vehicles that travel across the intersection in a vertical direction of biking. Thus, our system should be able to distinguish the frontal vehicle's right-turn trace from the others. (A detailed discussion will be provided in Section 4.3.) In short, the detection becomes positive if 1) $d_0 \leq SSD$ and 2) the vehicle is making a right turn. For expression simplicity, we leave the discussion of how to determine if a vehicle is making a right turn in Section 4.3. As shown in Fig. 1, d_0 is a function of the b-v distance d and direction θ, i.e., $d_0 = d \sin\theta$. Now our task is converted to measure d and θ of the target vehicle and to derive its moving pattern from time-series measures of d's and θ's.

3.2 CycleGuard Signal Design

CycleGuard uses the smartphone to emit acoustic chirp signals periodically. As shown in Fig. 2, a chirp's frequency linearly sweeps from the minimum value f_l to the maximum value f_h over time. In the time domain, one chirp signal is expressed as

$$S_c(t) = A_0 \cos(\pi \frac{B}{T} t^2 + 2\pi f_l t) \tag{1}$$

where A_0 is the amplitude, $B = f_h - f_l$, $t \in (0, T]$, and T is the chirp duration. We set $f_l = 16KHz$ and $f_h = 20KHz$ so that the chirps can be hardly picked up by human ears unless they are transmitted at a high power*. The same frequency range has been adopted by quite a few novel applications, such as biometric sensing [50] and acoustic imaging [32].

One concern of applying acoustic ranging is that the signal may be polluted by background noise in outdoor environments. To examine the potential impact caused by the background noise, we collected traffic noises, including vehicle's engine sound as it passes by, from actual bike rides in different city areas. Fig. 4 shows the frequency constituents of the recorded sound at local roads, roads adjoining large shopping complexes, and city highways during their respective rush hour periods. We find that the components are mainly concentrated on the lower end of the frequency spectrum, i.e., below 5kHz. Recall that the adopted chirp spans from 16kHz to 20kHz. There is a clear gap between the above two. By applying a high-pass filter to the received signal, the background noise can be easily eliminated.

*Although the frequency range of human hearing is generally considered from 20 Hz to 20 KHz, high frequency sounds must be much louder to be noticeable (including children and young adults) [52].

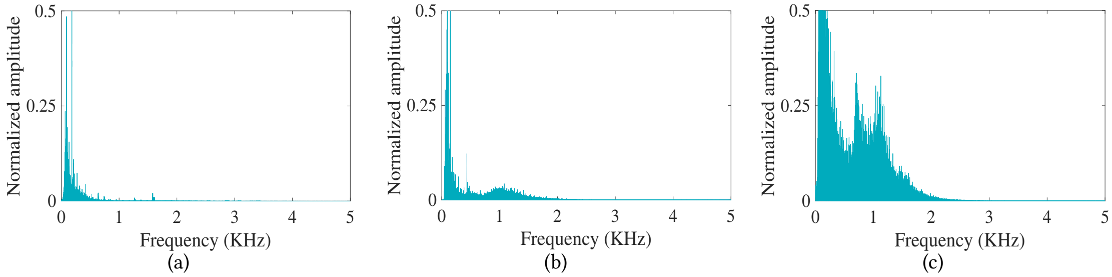


Fig. 4. Frequency constituents of real-world traffic noises in city areas: (a) Residential roads. (b) Shopping complex roads. (c) Highway roads. The accumulated frequency components of noise collected from highway roads are the largest among the three because the loudest noise is observed in highway roads.

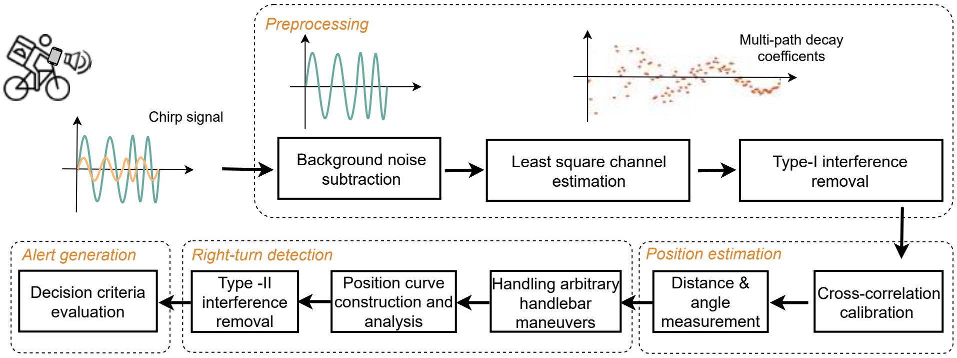

Preprocessing
Multi-path decay coefficents
Chirp signal
Background noise subtraction
Least square channel estimation
Type-I interference removal
Alert generation
Right-turn detection
Position estimation
Decision criteria evaluation
Type -II interference removal
Position curve construction and analysis
Handling arbitrary handlebar maneuvers
Distance & angle measurement
Cross-correlation calibration

Fig. 5. Overall system design.

We use a COTS speaker to emit acoustic chirps. As shown in Fig. 3, the speaker is connected to the smartphone via an aux audio Jack plug-in[†]. CycleGuard does not utilize the smartphone's built-in speaker due to its low working power. As a note, our experiment results show that the ranging distance with the built-in speaker is limited to 8 meters, which is insufficient in our application.

4 SYSTEM DESIGN

Fig. 5 shows the overall system design of CycleGuard. Upon receiving reflected signals, CycleGuard applies a high-pass filter to remove low-frequency components. As discussed in the previous section, this step is to remove background noises. Since the received signals are mixed with reflections from objects nearby, such as the bike, cyclist's body, and ground, we apply the least-square-estimation method to get rid of them. The above three steps belong to the *preprocessing module*. The *position estimation module* aims to locate the target vehicle, i.e., measuring the b-v distance d and direction θ. In particular, the former is derived by calculating the cross-correlation between transmitted and received signals. The latter is obtained by exploring the time difference of arrival (TDoA) of received signals at dual-mics. For the *right-turn detection module*, it continuously monitors the target vehicle's trajectory. Its inputs are time-series measures of d's and θ's. As discussed later, they exhibit unique patterns for

[†] CycleGuard works with speakers in other connection modes, such as Bluetooth.

right-turn vehicles. This module further eliminates background reflections from static objects and other moving vehicles. Further, the effect of arbitrary handlebar maneuvers is also taken into account. The *alert generation module* creates alarms, indicating the detection of any hazardous right-turn vehicle, if the two decision criteria (discussed in Section 3.1) are met.

4.1 Preprocessing

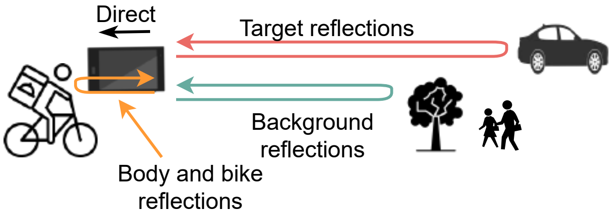


Fig. 6. Illustration of two types of interference.

In addition to the background noise, CycleGuard needs to get rid of other two types of interference. As shown in Fig. 6, the *type-I interference* denotes reflections from objects near the smartphone, including the cyclist's body, bike, and ground. *Type-II interference* stands for reflections from objects further away, such as trees, roadside trash-bins, buildings, and other pass-by vehicles. Due to a short distance away from the smartphone, type-I interference typically has a much higher magnitude than type-II interference, and even higher than the useful signals, i.e., reflections from the target vehicle. We categorize interference into two types because their corresponding removal techniques are distinct in our design. We first discuss how to combat type-I interference here while the approach of dealing with type-II interference will be covered in Section 4.3.

Once a chirp is emitted, it propagates through multiple paths, reaches different objects, and then bounces back. Denote by τ the round-trip duration a chirp travels through path i. We express the received signal following the classic multi-path model [59]

$$R(t) = \sum_{i \in U_1} h_i S_c(t - \tau_i) + \sum_{i \in U_2} h_i S_c(t - \tau_i), \tag{2}$$

where h_i stands for the decay coefficient of the propagation path i. U_1 represents the entire set of type-I interference. In our setting, their sources are the objects close to the smartphone, say, within 150 cm. U_2 are the components propagating through other paths. They include reflections from the target vehicle and type-II interference. Given the above expression, if we are aware of decay coefficients $h_i, i \in U_1$, we can remove the type-I interference by subtracting the first summation from (2). For this purpose, we first apply the least-square-estimation method to estimate h,

$$\underset{h_i}{\text{Minimize}} : \sum_t \left[R(t) - \sum_{i \in U_1 \bigcup U_2} h_i S_c(t - \tau_i) \right]^2 . \tag{3}$$

Since both $R(t)$ and $S_c(t)$ are known, the optimization problem (3) searches for a group of h_i's that produce the minimum objective function. This is an unconstrained least-square optimization problem, which can be efficiently solved via classic methods, such as *gradient descent* [8] and *Newton's method* [12]. In our implementation, (3) is solved using the Apache math tools [3]. Fig. 7(a) shows the derived decay coefficients when the target vehicle is 5 m away from the bicycle.

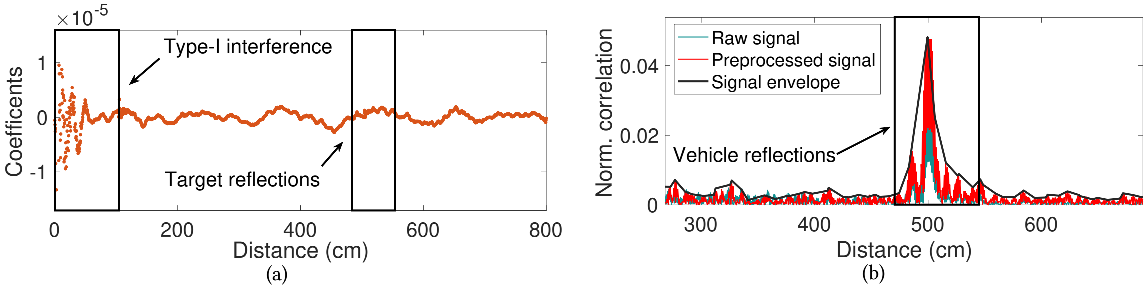


Fig. 7. Preprocessing of received signals. (a) Decay coefficients of received signals. (b) Received signals before/after type-I interference removal.

4.2 Estimation of B-V Distance and Direction

4.2.1 Estimation of B-V Distance d. The b-v distance d is derived by estimating the round-trip propagation delay of the chirp signal reflected by the target vehicle. Specifically, d is calculated as $v_c \times \text{delay}/2$, where $v_c = 340m/s$ is the sound speed. Then the remaining task is to figure out the round-trip propagation delay.

Given the received signal $R(t)$ (after pre-processing), we calculate its cross-correlation with the transmitted signal $S_c(t)$,

$$C(t) = \int_{\tau=-\infty}^{+\infty} S_c(t)R(t-\tau)d\tau. \tag{4}$$

Fig. 7(b) shows a normalized $C(t)$. In the envelop, strong peaks appear around the distance of 5 m, which is exactly where the vehicle exists in the testing. Hence, b-v distance d is obtained by identifying peaks in the envelop of $C(t)$. The classic peak detection algorithm [5] is applied. We also find in Fig. 7(b) that the crest starts from 4.8 m and ends at 5.6 m. The signals within this duration are reflections from different parts of the vehicle. In our design, d is deemed as the distance that associates with the strongest peak.

While type-I interference is eliminated via the pre-processing module, type-II interference still remains. Therefore, multiple peaks may exist in $C(t)$ above; each represents reflections from one prominent object, i.e., the target vehicle and sources of type-II interference. Yet, which one is produced by the target vehicle is unknown. The approach of locating the target vehicle among other objects will be discussed in Section 4.3.

The calculation above assumes that $S_c(t)$ and $R(t)$ are well aligned. In practice, timelines at the speaker and mic may not be perfectly synchronous due to the processing delay at mics. As a result, the calculation of d would be erroneous. Denote by τ_0 the misalignment in mic's and speaker's timelines.

To address the issue, we propose to adjust the received signal $R(t)$ as $R(t+\tau_0)$. Now the question is how to estimate τ_0. In particular, we first calculate the cross-correlation $C(t)$ without the presence of any vehicle. The first peak in $C(t)$ should associate with the direct speaker-mic signal transmission. Let t_0 be the corresponding measured transmission delay. Due to the misalignment, t_0 appears larger than the ground truth. Let the speaker-mic distance be L, which is a known value given a specific smartphone. Then we have $L = v_c \times (t_0 - \tau_0)$. Thus, τ_0 is calculated as $\tau_0 = t_0 - \frac{L}{v_c}$. As a note, the rest computations adopt the adjusted received signal $R(t+\tau_0)$ to eliminate the impact of signal misalignment.

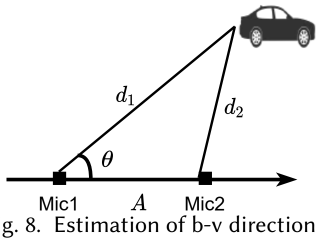


Fig. 8. Estimation of b-v direction θ.

4.2.2 Estimation of B-V Direction θ. The basic idea is to exploit the time-difference-of-arrival (TDoA) of signals received at two mics. As shown in Fig. 8, given the b-v distance measured at mic1 and mic2 as d_1 and d_2 respectively,

θ can be calculated based on their geometrical relationship as

$$\theta = \arccos(\frac{d_1^2 + A^2 - d_2^2}{2 \times d_1 \times A}) \tag{5}$$

where A is the inter-mic distance. It is a constant for smartphones of the same type but may vary for different ones. These values are available from smartphone's specification databases, such as [17, 18, 53]. In real-world implementations, such information can be pre-loaded to CycleGuard, which automatically adjusts parameter A in accordance with the smartphone's specs when the app is activated for the first time. As a note, the similar idea has been widely adopted in prior works that utilize dual-mic on smartphones for acoustic sensing, such as indoor localization, activity classification, and gesture recognition [27, 38, 49, 61, 68].

4.3 Right-Turn Detection

The goal of this component is to analyze the target vehicle's trajectory and decide if it is making a right turn. This task is non-trivial. While the type-I interference has been removed, the received signals are still mixed with the type-II interference. Recall that type-II interference consists of background reflections from static objects (e.g., trees and buildings) and non-static ones (e.g., other passing-by vehicles). In the following, we start from a simple case where only the target vehicle is present and then extend it to more general scenarios.

4.3.1 A Simple Scenario. The idea of the basic scheme is to analyze the vehicle's moving trace and decide if it matches the pattern of a right-turn vehicle. To facilitate the discussion, we set up a 2-D coordinate system. The top mic, denoted as mic1, serves as its origin, whereas the direction of mic1 (top mic)$\longrightarrow$ mic2 (bottom mic) is the x-axis, as shown in Fig. 10. Given the b-v direction θ, target vehicle's position is expressed as $(d_1 \cos\theta, d_1 \sin\theta)$. Fig. 11 shows the vehicle's time-series positions, called *position curve*, in the smartphone's coordinate system. The following observations are made toward a right-turn vehicle. First, θ increases, whereas d decreases over time. Second, as the detection is made the moment the vehicle starts turning, the vehicle exists at the left-hand side of the bicycle. Hence, the observed positions should all belong to the first quadrant in the coordinate system, i.e., $\theta \in [0, \pi/2]$. We will discuss later vehicles in other moving traces. Our basic scheme thus monitors the above three aspects in an evolving time window. A positive decision is made once the three criteria are met.

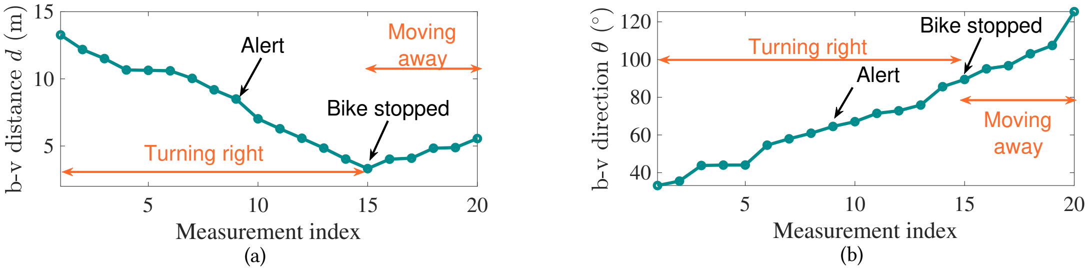


Fig. 9. Measured b-v distances and directions of a right-turn vehicle.

To better illustrate our idea, we plot b-v distance (d) versus time and b-v direction (θ) versus time as a vehicle is making a right-turn in front of the bike in Fig. 9(a) and Fig. 9(b), respectively. Once the app is turned on, it continuously estimates d and θ. Fig. 9 shows the sequential measures of these two values. It is observed that d decreases while θ increases over time before the bike stops. These motion features indicate that the vehicle is making a right-turn. An alert is generated once $d_0 \leq SSD$, i.e., the detected vehicle is too close to cause a potential crash.

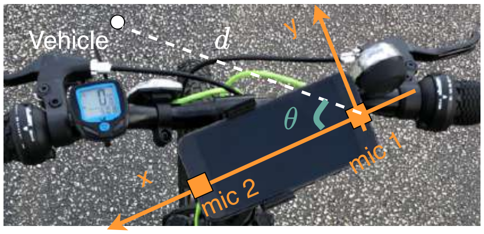


Fig. 10. Illustration of smartphone's coordinate system and target vehicle's position in the coordinate system.

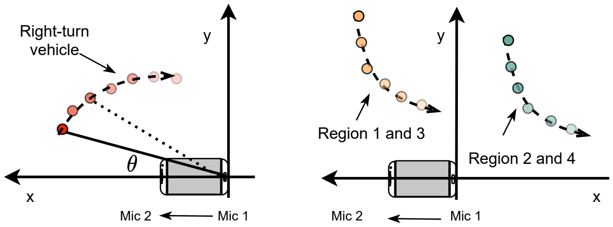


Fig. 11. Position curves of a right-turn vehicle and static objects in different regions around an intersection.

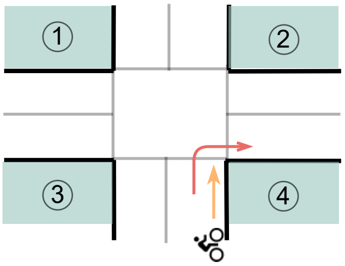


Fig. 12. The area around an intersection is divided into four regions.

Table 1. Properties of d and θ for static objects at different regions.

	θ	d	Range of θ
Region 1	↓	↓	$[0, \pi/2]$
Region 2	↑	↓	$[\pi/2, \pi]$
Region 3	↓	↓	$[0, \pi/2]$
Region 4	↑	↓	$[\pi/2, \pi]$
Right-turn vehicle	↑	↓	$[0, \pi/2]$

4.3.2 Handling Static Objects Nearby. In practice, there are various static roadside objects that can reflect acoustic signals. Hence, these objects, especially the large ones, such as trash-bins, trees, and buildings, can be detected by our system. Their reflections constitute part of the type-II interference. It is thus critical to extract the target vehicle from them. Under the framework of the basic scheme, we propose to analyze their position curves. To facilitate the discussion, we divide the area near the intersection into four regions as shown in Fig. 12. We first look at static objects from region 1. As the cyclist heads to the intersection, both its distance d and direction θ with respect to the object decreases. Besides, the position curve falls into the first quadrant i.e., $\theta \in [0, \pi/2]$. The properties of time-series d's and θ's of objects from all four regions are summarized in Table 1. It is observed that right-turn trajectories exhibit distinctive properties compared with those of static objects. We thus exploit this phenomenon to identify the right-turn vehicle from received signals mixed with interference from static objects.

4.3.3 Presence of Multiple Vehicles. Type-II interference also includes reflections from other vehicles passing by. This part discusses how to tackle them. The idea is similar as above. As shown in Fig. 13, all traffic trajectories at the intersection are categorized into twelve types. In the following, we pick several representative ones and analyze their position curve patterns. A comprehensive analysis is given in Table 2. A vehicle in trace 9 moves straight forward. Both its distance d and direction θ to the bicycle increase over time. Besides, its position curve belongs to the first quadrant, i.e., $\theta \in [0, \pi/2]$. For a vehicle in trace 6, it also takes a right turn but in a different lane. Thus, it would not pose any danger to the cyclist. Its direction θ to the bicycle first increases and then decreases, whereas d keeps on decreasing. The corresponding position curve falls into the first quadrant, i.e., $\theta \in [0, \pi/2]$. Vehicles at intersections are regulated by traffic lights and signs. Vehicles in trace 5 should be stopped by the red light, if the bicycle can pass through (with the green light). That explains why both its d and θ

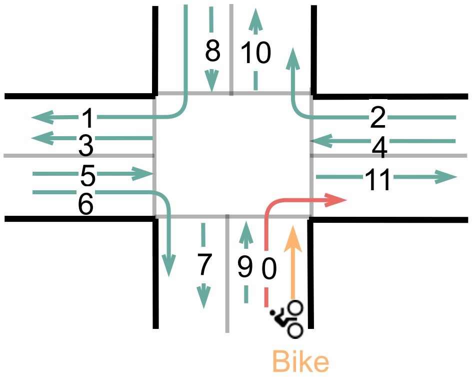


Fig. 13. Illustration of vehicle moving traces of different types.

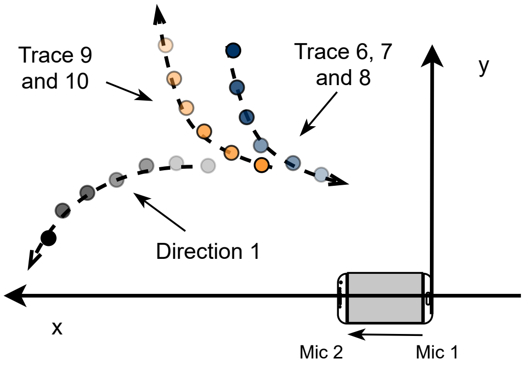


Fig. 14. Position curves of vehicles in different traces.

drop with $\theta \in [0, \pi/2]$. Table 2 and the analysis above show that the right-turn vehicle produces a unique pattern in its position trace compared with vehicles in other traces. CycleGuard thus monitors those patterns in received signals to identify any right-turn vehicles.

Table 2. Properties of d and θ for vehicles in different moving traces.

Trace	1	2	3	4	5	6	7	8	9	10	11	Right-turn vehicle (0)
θ	↓	↓	↓	↓	↓	↑↓	↓	↓	↑	↑	↑	↑
d	↓↑	↓↑	↑	↓	↓	↓	↓	↓	↑	↑	↑	↓
θ range	Q1	Q2	Q1	Q2	Q1	Q1	Q1	Q1	Q1	Q1	Q2	Q1

Q1: $\theta \in [0, \pi/2]$, Q2: $\theta \in [\pi/2, \pi]$

4.4 Adjustment of Phone Orientation

The phone orientation also affects detection performance. Basically, we need to avoid pointing mics toward the cyclist body, as received signals would be blocked. This part explains how to adjust the phone to a suitable orientation.

As demonstrated in Fig. 15, the angle between the biking direction and the y-axis is treated as the phone orientation, denoted by $\hat{\theta}$. In Section 6.3 we evaluate the performance of CycleGuard with regard to $\hat{\theta}$ and find out that each type of smartphone achieves the best performance with slightly different $\hat{\theta}$. Take Google Pixel XL as an example, its best $\hat{\theta}$ exists around 30°. $\hat{\theta}$ is determined by the relative position of two mics.

To allow a cyclist to mount the phone properly with an ideal $\hat{\theta}$, our app displays a reference line on the screen, as shown in Fig. 16. By aligning this line parallel to the bike's handlebar, the smartphone's orientation becomes exactly $\hat{\theta}$. Hence, the cyclist only needs to adjust the phone's orientation before proceeding to the road. Since the direction of the reference line may not be the same across various types of smartphones, we propose to derive them offline. Once the app is installed on the smartphone, the suitable reference line is automatically selected according to the phone's specifications. The entire process is transparent to cyclists.

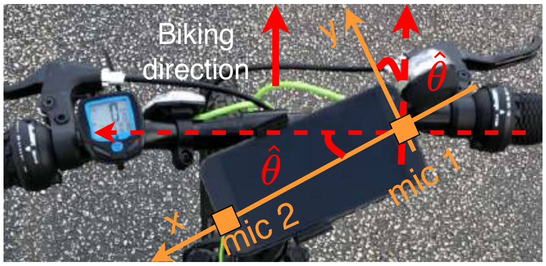
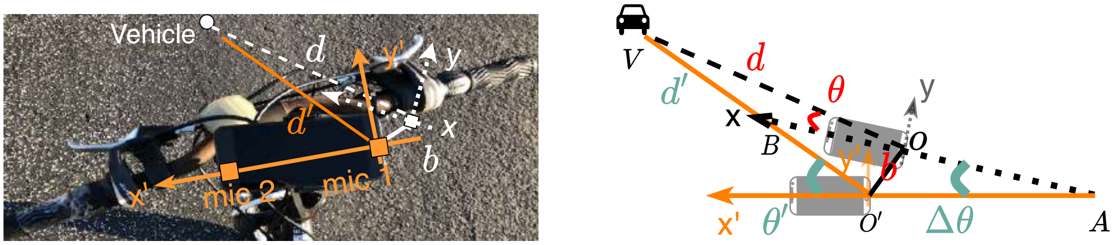


Fig. 15. Illustration of phone orientation $\hat{\theta}$.

Fig. 16. The displayed reference line helps cyclists to adjust phones to a suitable orientation $\hat{\theta}$.



(a)

(b)

Fig. 17. Effect of arbitrary handlebar maneuvers. (a) The handlebar maneuver changes the coordinate system, from $x - y$ to $x' - y'$. (b) Illustration of the relationship between θ and θ', between d and d'.

It is noteworthy that perfect alignment is desired but not mandatory. It is quite often that cyclist's manual adjustment bears some error. We further evaluate the impact of alignment errors on the performance of CycleGuard in Section 6.3. It shows that the performance mildly degrades about 2% when the error is 12°.

4.5 Effect of Arbitrary Handlebar Maneuvers

Cyclists adjust the handlebar's orientation from time to time to keep the bicycle balanced while riding. Consequently, the steering maneuver changes the phone's orientation and thus impacts the detection accuracy of CycleGuard. As the measurement of b-v distance d and direction θ serves as the cornerstone of our scheme, we would like to examine the impact of arbitrary handlebar maneuvers on them, and more importantly, how to remedy the effect.

Denote d/d' and θ/θ' the b-v distance and direction before/after the handlebar maneuver, respectively. Let b be the displacement of the coordinate origin, i.e., from O to O'. As shown in Fig. 17, since $d, d' \gg b$, then we have $d \approx d'$. Hence, the impact of arbitrary handlebar maneuvers on the measurement of d is negligible. Now we turn our attention to θ/θ'. The goal is to derive the relationship between θ and θ', such that the former can be derived from the latter. Let A be the intersection between the x-axis and x'-axis. Denote by $\Delta\theta$ the angle $\angle OAO'$. Then $\Delta\theta$ is deemed as the phone's orientation change caused by steering maneuver. Because $d, d' \gg b$, d and d' are approximately parallel to each other. Then we have $\angle OBO' = \theta$. In triangle $\Delta O'AB$, we have $\theta' = \Delta\theta + \angle OBO' = \Delta\theta + \theta$. Thus, θ can be expressed as $\theta = \theta' - \Delta\theta$. Now the remaining task is to find out $\Delta\theta$, i.e., the phone's orientation offset caused by steering maneuver. We propose to estimate it by accessing the phone's gyroscope readings. We first record the initial reading when the handlebar is placed at the original orientation. Then, as the cyclist adjusts the handlebar, the real-time yaw velocity of the handlebar maneuver can be recorded by the gyroscope and converted to $\Delta\theta$.

4.6 Piecing All Together

The design rationale of CycleGuard is to estimate whether a cyclist has sufficient time to stop his/her bicycle before a potential right-hook collision. As discussed, it is equivalent to monitor two conditions. First, $d_0 > SSD$; per the discussion in Section 3.1, the calculation of d_0 relies on the values of b-v distance d (Section 4.2.1) and b-v direction θ (Section 4.2.2), i.e., $d_0 = d \sin\theta$. By further taking into account the adjustment of phone orientation $\hat{\theta}$ (Section 4.4) and the phone's orientation offset $\Delta\theta$ caused by steering maneuver (Section 4.5), d_0 is adjusted as $d_0 = d\sin(\theta' - \Delta\theta - \hat{\theta})$. Recall that θ' denotes the measured b-v direction under handlebar maneuver. The second condition is to decide the target vehicle's right-turn maneuver from its moving trajectory (Section 4.3). Our scheme is capable of handling interference from both static objects and other vehicles passing by.

One last missing piece in the jigsaw puzzle is to figure out SSD. Following roadway design guidelines [25, 45, 46], SSD is estimated by

$$SSD = v \times t_{pr} + \frac{v^2}{2\alpha}, \tag{6}$$

in which v, t_{pr}, and α stand for the cyclist's instant biking speed, perception reaction time, and deceleration rate, respectively. SSD is the sum of two parts: 1) brake reaction distance (i.e., the distance traversed by the bicycle from the instant the cyclist sights an object necessitating a stop to the instant the brake is applied); and 2) braking distance (i.e., the distance needed to stop the bicycle from the instant brake application begins). We will leave the instantiation of t_{pr} and α in Section 5. To measure v, we first access smartphone's GPS readings to derive the bicycle's moving distance within a short time interval. Then, v is approximated by dividing the distance by the time interval. We show in Section 6.2 that this approach causes mild estimation errors smaller than 0.31 m/s.

5 PARAMETER SELECTION FOR SSD

To select the appropriate reaction time t_{pr} and deceleration rate α, we carry out a series of indoor simulations. The instantiation of t_{pr} and α is a joint consideration of results from both resources. For the simulation, a bike simulator from Realtime Technologies Inc. [21] is used. As shown in Fig. 1, the bike simulator consists of a full-size bike mounted with motion sensors, three screens subtending horizontally with a Field of View of 135°, and a simulation server that connects with the screens and the bike. The simulator can be used to create various traffic scenarios, such as highways, city roads, traffic controls, car crashes, lane changes, and monitor the changes and reactions of different traffic participants under the created scenarios. In the simulation, we create the scenario depicted in Fig. 1. The scene is projected on screens via a networked environment of three simulation servers. Ambient vehicle noises, wind, and other on-road noises are also provided in stereo directions by a Dolby 5.1 audio surround system. The cyclist rides as in real roads, as shown in Fig.18. Participants are left sufficient time to get themselves familiar with the simulator. To measure the reaction time t_{pr}, we ask the cyclists to observe, react, and brake, whenever the vehicle starts to make a right turn in the simulated traffic scenario. The moment that the vehicle turns right is recorded as t_1 by the system, while the time that a cyclist applies the brake is recorded as t_2. Then, $t_{pr} = t_2 - t_1$. The deceleration rate α is automatically measured by its mounted sensors. Six volunteers participate; each repeats the experiment 50 times. Our statistic results are thus derived from a total of 300 experiment instances.

Fig. 19 shows the cumulative distribution function (CDF) of the measured reaction time. We observe that the value ranges between 0.61 and 2.21s; more than 90% of the measures are less than 1.996 s. The CDF of measured deceleration rates is shown in Fig. 20. The value is between 1.10 and 5.82 m/s^2. The average equals 2.26 m/s^2 whereas 90% of the measures are larger than 1.95 m/s^2. On the other hand, per the Design Manual for

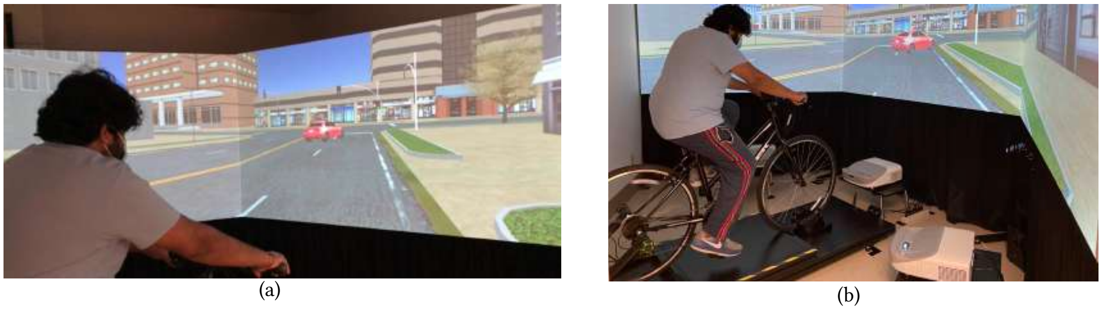


Fig. 18. Simulator setup. (a) Adopted scenario in the simulation. (b) Bike simulator.

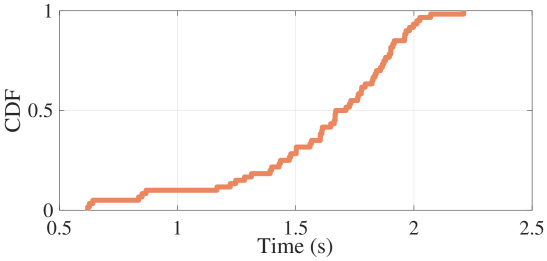


Fig. 19. Reaction time.

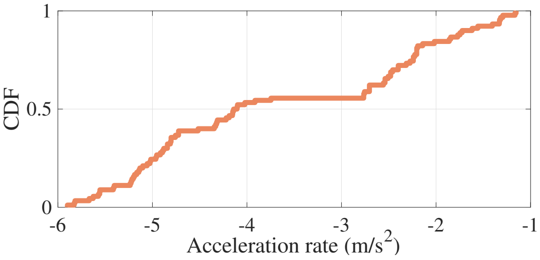


Fig. 20. Acceleration rate.

Bicycle Traffic [48] produced by CROW[‡], the reaction time is estimated as 2 s, whereas the low/high estimate of deceleration is considered as 1.5 m/s^2/2.6 m/s^2. Combining all the results above, we adopt $t_{pr} = 2\ s$ and $\alpha = 1.8\ m/s^2$ in CycleGuard as a conservative setting to accommodate most of the scenarios on roads. By a conservative setting, we mean longer reaction time and milder deceleration adopted by cyclists; that is, cyclists need a longer distance to stop the bike completely.

6 EVALUATIONS

6.1 Experimental Setup

6.1.1 Implementation. As a proof-of-concept implementation, we deploy CycleGuard's prototype on two Android smartphones Google Pixel XL and Galaxy S8. In particular, Pixel XL has Quad-Core Snapdragon 821 CPU (2 × 2.15 GHz and 2 × 1.6 GHz) with 4GB RAM, while Samsung S8 has Octa-Core Snapdragon 835 CPU (4 × 2.35 GHz and 4 × 1.9 GHz) with 4GB RAM. Two vehicles, Volkswagen Tiguan 2016 and Toyota Corolla 2017, are used for testing. The smartphone is placed in the holder that is mounted on the handlebar. A portable COTS speaker, costing around $16, is connected to the smartphone via a 3.5mm audio jack to play chirp signals. The smartphone's two mics are used to capture the reflected signals. An android app is developed to implement all the modules described in Section 4.

[‡] CROW is the technology platform for transport, infrastructure, and public space in the Netherlands. CROW is an organization in which the government and businesses work together in pursuit of their common interests through the design, construction and management of roads and other traffic and transport facilities.

6.1.2 In-field Testing Setup. All tests are conducted at the campus parking lot during weekends when the space is relatively empty. We set up an intersection-alike scenario as shown in Fig. 21. A cyclist rides heading towards the intersection, whereas a vehicle makes a right turn in front. A traffic cone is placed at an SSD distance away from the intersection. Note that SSD is derived in Section 4.6. Hence, if the cyclist applies braking at the traffic cone, the bicycle would stop completely before the intersection. The test is deemed successful if an alert is generated before the bicycle reaches the traffic cone, and failed otherwise. With such design, no danger would be caused to the cyclist, as the bicycle will be stopped before entering the intersection anyways.

6.1.3 Evaluation Metrics. The accuracy performance of our system is mainly evaluated via true positive rate (TPR) and true negative rate (TNR). Specifically, TPR is the proportion of potential collisions that are correctly detected and rise alerts, whereas TNR is the proportion of non-collisions without alert.

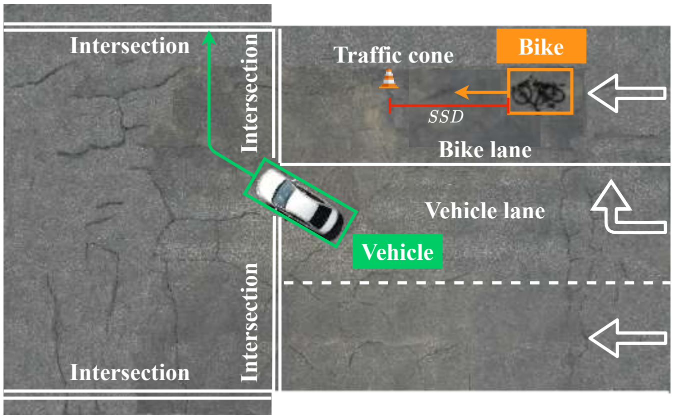


Fig. 21. Testing scenario.

6.2 Micro Benchmarks

6.2.1 Distance Measurements. Fig. 22 evaluates how accurate the measured b-v distances match the ground truth. The vehicle is at a distance of 3-30 m away from the bicycle. In total, 500 times of trials are conducted. It is observed that the measurement error is less than 11.7 cm by average at 30 m. We also find that the error is positively correlated with b-v distance. This is because the vehicle's acoustic reflections decay more significantly via a longer transmission, which results in distortions of the calculated cross-correlations and thus the estimation of b-v distance.

6.2.2 Vehicle Detection. Fig. 23 shows the success rate of vehicle detection by varying the distance from 3 to 30 m. CycleGuard can successfully detect the vehicle in all trials within 9 m. The success rate experiences a slight decrease as the distance increases. The lowest value of 92.3% exists when the vehicle is 30 m away from the bicycle. This is because vehicle's reflections are weak when faraway. Therefore, their peaks in cross-correlations become harder to locate. Still, the overall detection performance is promising.

6.2.3 Speed Measurements. To evaluate the accuracy of speed measurement, we use a speedometer to obtain the ground truth. The speedometer is mounted on a spoke of the bicycle frontal wheel as shown in Fig. 24. The error is calculated as the difference between the ground truth and the measured value by our system. During the testing, participants ride at different speeds ranging from 2 m/s to 6 m/s. As suggested by State Laws [41–44],

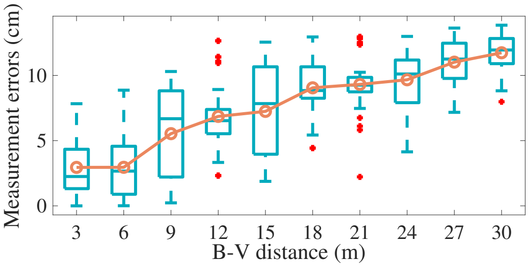


Fig. 22. Measurement accuracy of b-v distance.

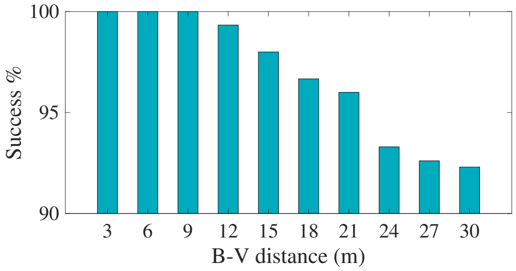


Fig. 23. Performance of vehicle detection.

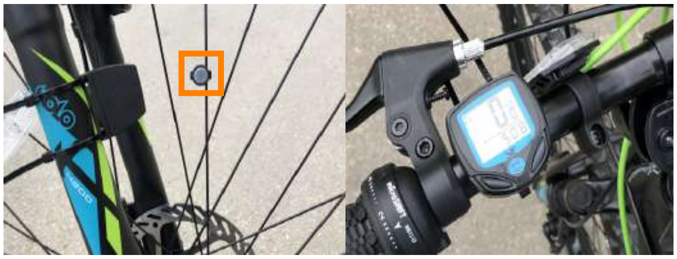


Fig. 24. Speed measurement setup.

cyclists should maintain lower than normal riding speeds when approaching an intersection. Therefore, we limit the biking speed to 6 m/s in the testing. As illustrated in Table 3, the result is promising; measurement errors are within 0.31 m/s under all riding speeds. Given speed of 5 m/s, the error is about 6.2%. The result validates the efficacy of our approach.

Table 3. Bike speed measurement errors.

Speed (m/s)	2	3	4	5	6
Errors (m/s)	0.15	0.26	0.16	0.31	0.26

Table 4. Right-turn detection performance.

Distance (m)	10	15	20	25	30
Success rate	98%	97%	95%	92.3%	92%

6.2.4 Right-turn Detection. This part examines if CycleGuard can detect right-turn vehicles correctly. Table 4 shows that the success rate is 95% when the distance is at 20 m. As expected, it decreases with a longer distance. The reason is similar as above, i.e., a longer distance leads to a weaker received signal at mics and thus a lower detection accuracy. Comparing the results of Table 4 and Fig. 23, the accuracy of right-turn detection is only slightly lower than the vehicle detection at the same given distance. For example, when the distance is 30 m, the success rate of the former is 92.3% whereas the latter is 92%. It means the performance degradation of right-turn detection is mainly caused by the long distance. Our proposed scheme (Section 4.3) for right-turn detection performs pretty well given relatively strong received signals.

6.2.5 Computation Time. The computation time of each system module is depicted in Fig. 25(a). The pre-processing module consumes the most time, 220.5 ms on average. This is because the least-square estimation requires solving a complex optimization problem (3). The time for position estimation and right-turn detection is relatively low, with their average values as 65.62 ms and 63.72 ms , respectively. The CDF of total computation

Table 5. Energy consumption of CycleGuard with 30-min usage.

Phone	Battery capacity	Energy consumption	Percentage
Pixel XL	3450 mAh	242 mAh	7%
Galaxy S8	3000 mAh	270 mAh	9%

Table 6. Energy consumption of different applications.

Application	CycleGuard	Video calling	Video streaming	Web browsing
Energy consumption	242 mAh	563 mAh	422 mAh	242 mAh
Percentage	7%	16%	12%	7%

time is presented in Fig. 25(b) of which the average value is 365.2 ms, with 90% of values below 414 ms. Given a biking speed of 5 m/s, a cyclist would cover 2.1 m within 414 ms. It means a cyclist is in fact 2.1 m closer to the vehicle than the SSD when an alert is generated. To address this issue, we set the SSD 414 ms $\times v$ longer than the result derived from (6) in our implementation.

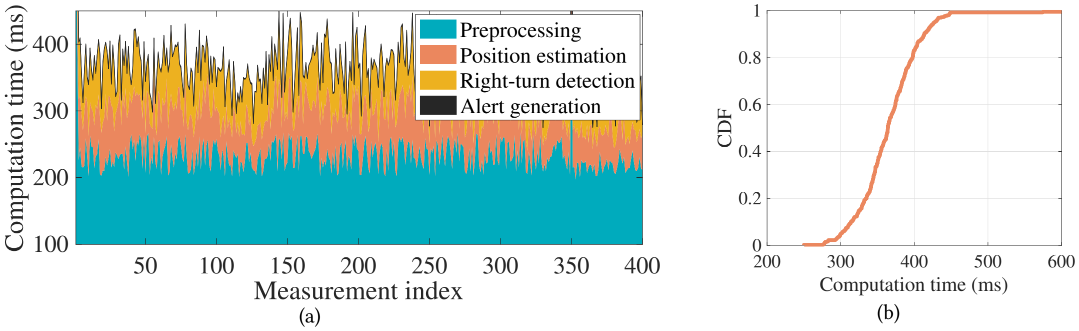
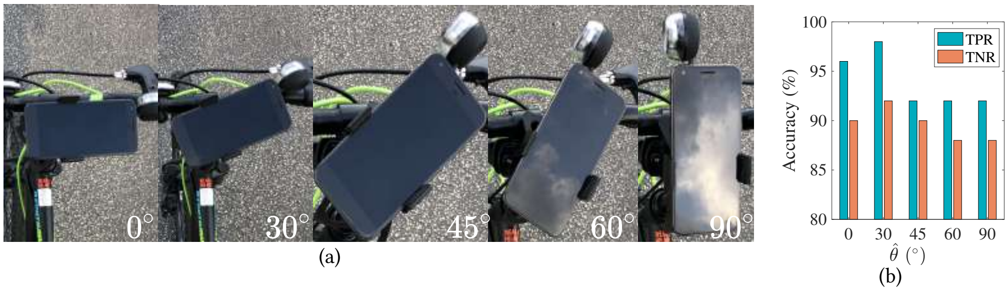


Fig. 25. Evaluation of computation time. (a) Stacked computation time. (b) CDF of total computation time.

6.2.6 Energy Consumption. We assess the energy consumption of our system by running it on two different phones, Google Pixel XL and Samsung S8, for 30 minutes. As illustrated in Table 5, for Pixel XL, the energy cost is 246 mAh, i.e., 7% of its battery capacity. S8 utilizes 270 mAh, which equals 9% of its total battery capacity. We also compare in Table 6 CycleGuard's energy usage with daily phone usage applications, including video calling, video streaming, and web browsing. The most energy-consuming application is video calling, i.e., 563 mAh (16%). CycleGuard has a relatively low energy cost, which is similar to that of web browsing.

6.3 System Benchmark

6.3.1 Impact of Riding Speeds. In the experiments, participants are asked to ride at different speeds varying from 2 m/s to 6 m/s. As shown in Table 7, both TPR and TNR decrease as the speed grows. This is because our system detects vehicle location four times per second; high-frequency detection is desirable to watch for potential collisions. CycleGuard can successfully detect collisions at high accuracy, $TPR = 94\%$ and $TNR = 88\%$, even when



(a)

(b)

Fig. 26. Impact of phone orientation. (a) Illustration of different phone orientations under testing. (b) Accuracy performance.

the riding speed is 6 m/s. Besides, we notice that TNR is lower than the TPR. In general, a lower TNR produces more false alarms. It is more important to have a good TPR to miss as few potential crashes as possible.

Table 7. Impact of riding speeds, b-v distances, and vehicle speeds.

Metric	Bike riding speed (m/s)					Relative distances (m)					Vehicle speed (mph)				
	2	3	4	5	6	10	15	20	25	30	5	7	10	12	15
TPR	98%	98%	96%	94%	94%	98%	96%	96%	94%	92%	98%	98%	96%	92%	92%
TNR	91%	90%	89%	88.7%	88%	92%	90%	89.7%	89.2%	88%	90%	90%	90%	88%	87.5%

6.3.2 Impact of B-V Distances. We also evaluate if different b-v distances impact the accuracy performance. The riding speed is set to 4 m/s by default. As shown in Table 7, the detection accuracy is negatively correlated with b-v distances. This result complies with that in Section 6.2. The performance degradation is due to the weaker received signals traveling through a longer distance. Still, the performance remains satisfactory with at least $TPR = 92\%$ and $TNR = 88\%$.

6.3.3 Impact of Vehicle Speeds. We test speeds from 5 mph to 15 mph. As a note, the speed limit of turning at intersections is 15 mph per many State Laws [41–44]. As shown in Table 7, the detection accuracy drops with the increase of vehicle's speeds. As mentioned above, a vehicle in a higher speed requires more frequent detection to capture its moving pattern. Due to the hardware restriction, the detection frequency of CycleGuard is upper-limited by four times/sec. This restriction is due to the computation time of our algorithms. As discussed in Section 6.2.5, the minimum total computation time is about 250 ms to run all the modules.

6.3.4 Impact of Phone Orientations. In this part, we evaluate the impact of phone rotation $\hat{\theta}$. Fig. 26(a) demonstrates the phone's placement at different $\hat{\theta}$. Fig. 26(b) shows that the best overall performance exists when $\hat{\theta}$ is around 30°. The worst performance exists when $\hat{\theta} = 90°$, as one of the mics points toward the cyclist. The cyclist's body blocks a portion of received signals. The overall TPRs and TNRs are higher than 93% and 87%, respectively, under all settings.

6.3.5 Impact of Phone Orientation Misalignment. As mentioned in Section 4.4, to facilitate phone orientation adjustment, a reference line will be displayed on the screen. In practice, the process of manually adjusting orientation may be imperfect due to human errors. We thus further evaluate the impact of this factor on the detection accuracy. Table 8 shows TPR and TNR with respect to different alignment errors, defined as the

Table 8. Impact of phone orientation misalignment.

Alignment error (°)	-12	-9	-6	-3	0	+3	+6	+9	+12
TPR	96%	98%	98%	98%	98%	98%	98%	98%	96%
TNR	90%	90%	90%	92%	92%	92%	92%	90%	90%

difference between the ground truth and the ideal value of θ'. The impact is negligible. For example, TPR is 98% under perfect alignment while it is still as high as 96% with the error of +12°. Therefore, CycleGuard is robust to human errors during phone orientation adjustment.

Table 9. Performances of CycleGuard under maneuver changes.

Metric	Bike riding speed (m/s)					Relative b-v distances (m)				
	2	3	4	5	6	10	15	20	25	30
TPR	98%	96.8%	95.83%	93.75%	92.3%	96.6%	96%	94.73%	93.9%	93.1%
TNR	90%	89.2%	88.57%	87.5%	87.5%	90.9%	88.4%	88%	86.9%	86.2%

6.3.6 Impact of Maneuver Changes. Cyclists adjust the handlebar's orientation from time to time to keep the bicycle balanced while riding. To alleviate its impact on detection accuracy, in the design we propose to offset the phone's orientation change caused by steering maneuver with the assistance of real-time gyroscope readings. To validate its effectiveness, we carry out a set of tests in which cyclists are asked to proceed on a road and make maneuver changes at a specific point. As shown in Fig. 27, we set up a traffic cone on the road to stand for a pothole. Hence, cyclists need to steer the handlebar to avoid the "pothole". We record the experimental results under this scenario and compare its performance with those when cyclists proceed on regular roads (Table 7 in Section 6.3). The results are summarized in Table 9. We vary both riding speeds and b-v distances to evaluate the system performance in different settings. We find that its performances are comparable to the benchmarks in Table 7 in Section 6.3. To be specific, when the cyclist proceeds straight at speed of 4m/s, $TPR = 96\%$ and $TNR = 89\%$, while it is 95.83% and 88.57% with cyclist's maneuver changes. It implies that maneuver changes cause negligible impact on our system.

Table 10. Impact of nearby object reflections.

Metric	Impact of trees					Impact of buildings				
	10m	15m	20m	25m	30m	10m	15m	20m	25m	30m
TPR	98%	95%	94.11%	92.3%	92%	98%	96%	96%	93.33%	92.68%
TNR	90.9%	90%	88.8%	88%	87.5%	92%	90%	88.23%	87.5%	87%

6.3.7 Impact of Nearby Object Reflections. As received signals include reflections from objects nearby, this part examines their impact on the detection performance. We conduct experiments in the presence of trees and buildings since these objects are commonly found in roadways. Table 10 exhibits a similar trend to that of Table 7. The latter is derived without any main reflection object nearby. The detection performance is downgraded as the b-v distance becomes larger. The presence of roadside objects impacts TNR slightly. For example, when the b-v distance is 20 m, TNR = 89.7% without any nearby objects, while TNR is 88.8% and 88.23% with trees and buildings, respectively. This is caused by false detection triggered by roadside objects. On the other hand, TPR is relatively consistent with or without nearby objects. It indicates that right-turn vehicles will not be missed because of interference from nearby objects, which is a desirable property.

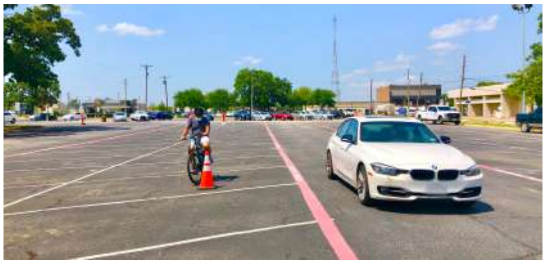


Fig. 27. Testing scenarios for maneuver changes on the road.

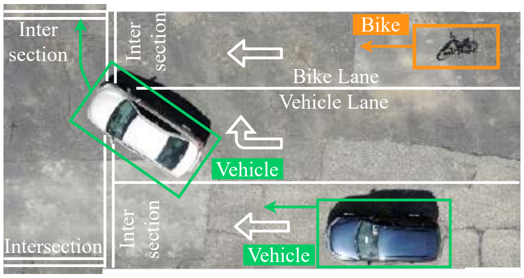


Fig. 28. Testing scenario of multiple vehicles.

Table 11. Impact of multiple co-present vehicles.

Metric	Bike riding speed (m/s)					Vehicle speed (mph)				
	2	3	4	5	6	5	7	10	12	15
TPR	98%	98%	96%	94%	93%	98%	98%	94%	92%	92%
TNR	90%	88.8%	88%	87.5%	86.95%	90%	88.8%	87%	86.6%	85%

6.3.8 Impact of Multiple Vehicles. It is quite often that multiple vehicles are co-present nearby the intersection. We are thus interested in finding out the impact from them. Fig. 28 demonstrates our testing scenario where one vehicle turns right at the intersection while the other proceeds straight. We evaluate the performance by varying the bike speed and vehicle speed. According to Table 11, CycleGuard exhibits better performance at a lower bike/vehicle speed. To illustrate the impact of multiple vehicles, we compare the result with Table 7 which is derived without other co-present vehicles. The TNR is better in the latter. For example, when the vehicle speed is 15 mph, TNR=87.5% whereas it becomes 85% with the presence of a second vehicle. This is because the second vehicle sometimes triggers a false alarm and lowers the TNR. On the other hand, TPR is unchanged in general. We argue that the metric of TPR plays a more vital role than TNR, as a high TPR indicates that hazardous vehicles are less likely to be missed and is thus more crucial to cyclist safety. By combining the results above, we find that CycleGuard is robust to impact from nearby objects, including both the static ones, such as trees and buildings, and the dynamic ones, such as other moving vehicles nearby.

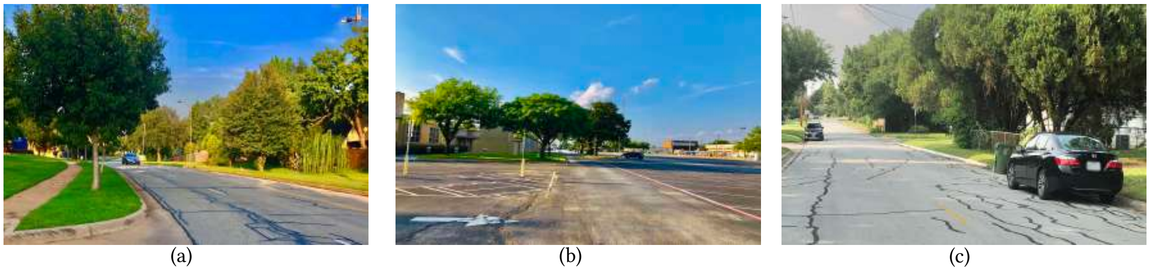

(a)
(b)
(c)

Fig. 29. Testing sites for (a) campus areas, (b) parking lot, (c) residential areas.

6.3.9 Impact of Testing Sites. To evaluate the system in realistic situations, we test CycleGuard on roads that are located at three sites, i.e., residential areas, campus areas, and parking lots. As shown in Fig. 29, highly reflective

Table 12. System performances in residential areas, campus areas and parking lots.

Metric/Setting	Parking lot Relative b-v distances (m)			Campus areas Relative b-v distances (m)			Residential areas Relative b-v distances (m)		
	10	20	30	10	20	30	10	20	30
TPR	98%	96%	92%	96.15%	93.9%	92.3%	95.45%	93.54%	91.83%
TNR	92%	89.7%	88%	91.66%	85.71%	80.55%	90.32%	84.31%	78.84%

objects are around these roads, such as vehicles, trees, signs, and buildings. The results are summarized in Table 12. The overall performance is satisfactory especially when b-v distances are within 20 m. CycleGuard experiences slight performance degradation as the testing site becomes more complex (residential areas > campus areas > parking lots). When the b-v distance equals 30 m, it achieves a TNR of 88% in the parking lot, while it drops to 78.84% in the residential areas. This is because various objects, such as trees, buildings, and vehicles passing by, coexist in complex testing sites. The received signal becomes polluted by their reflections. As a result, it leads to a higher chance of false alarms. Still, CycleGuard maintains good performance for TPR with its values all above 90% across all the testing sites. For the cyclist's safety consideration, a high TPR is more crucial than TNR, since it would miss few potential right-hook crashes.

Table 13. Impact of usage time.

Usage time	Morning	Afternoon	Evening	Night
TPR	96%	98%	95%	96 %
TNR	86%	90%	90%	88%

6.3.10 Impact of Time of Usage. Table 13 shows the performance when CycleGuard is tested at different times of a day, i.e., morning (9 AM), afternoon (2 PM), evening (5 PM), and night (7 PM), of different lighting conditions. It is observed that the performance is rather consistent under different usage time. Thus, the impact of lighting conditions is limited. It is worth mentioning some prior works that leverage computer vision techniques to detect oncoming vehicles [13, 28, 29, 56]. Their performance is subject to lighting conditions.

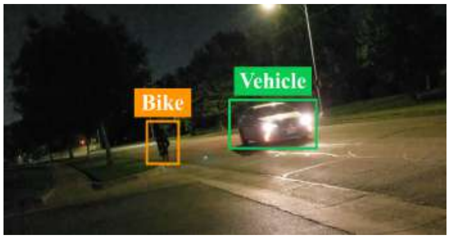


Fig. 30. Nighttime testing scenario.

Table 14. System performances at nighttime.

Metric/Setting	Parking lot Relative b-v distances (m)			Campus roads Relative b-v distances (m)		
	10	20	30	10	20	30
TPR	98%	96.67%	92.3%	96.55%	94.1%	92.85%
TNR	93.9%	91.66%	90.9%	91.89%	86.36%	81.48%

We further test CycleGuard at different sites during the nighttime as shown in Fig. 30. Table 14 summarizes TPR and TNR. It is observed that the performances are comparable to those in the daytime (as shown in Table 7 in Section 6.3). For example, in the parking lot, when b-v distance equals 10 m, $TPR = 98\%$ and $TNR = 92\%$ in the daytime tests; they are about 98% and 93.9% in the nighttime tests. Some prior works, such as Farsight [36] and [15], experience significant performance degradation at night because they use cameras to detect oncoming vehicles.

6.3.11 Impact of Different Devices. Fig. 31 shows the frequency responses of different smartphones' microphones. We observe that both maintain high responses within CycleGuard's operational frequency range, i.e., 16 KHz - 20 KHz. Table 15 further compares the system performance. The result matches our evaluation in Section 6.2; an increased riding speed leads to a drop in detection accuracy. In addition, the detection performance of S8 is slightly better than Pixel XL. For example, when the cyclist's riding speed is 6 m/s, its TPR equals 94% and 92% for S8 and Pixel XL, respectively. This is because S8 has a better microphone response in the high-frequency range, as shown in Fig. 31(b).

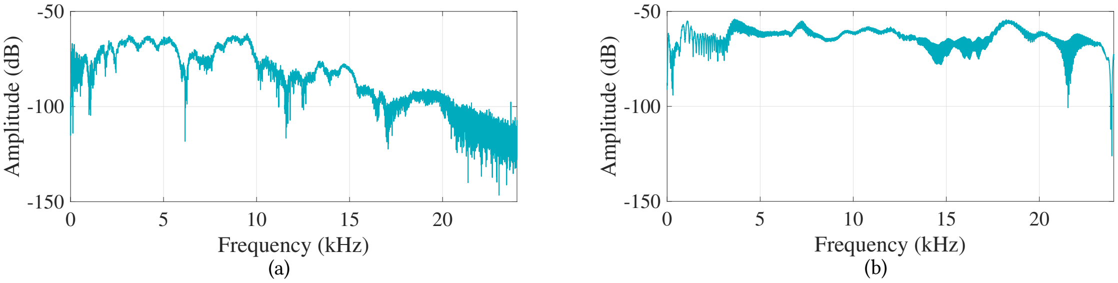


Fig. 31. Frequency response of different smartphone's microphones. (a) Pixel XL. (b) Samsung S8.

Table 15. Impact of different phones.

Phone	Metrics	Riding speed (m/s)				
		2	3	4	5	6
Google Pixel XL	TPR	98%	98%	96%	94%	94%
	TNR	91%	90%	88%	86%	86%
Samsung S8	TPR	98%	97%	95%	93%	92%
	TNR	95%	95%	93%	90%	90%

Table 16. CycleGuard performances across ten volunteers.

ID	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
TPR	98%	96%	98%	93%	93%	92.3%	94.7%	92.8%	96%	95.2%
TNR	86%	88%	86%	88%	90%	85%	84.2%	85.7%	83.3%	85.7%

6.3.12 Impact of Different Users. Ten volunteers are recruited to test our app. Their demographic information is provided in Table 17. During the experiments, volunteers are allowed to use the smartphone and bike following their own preference with no restrictions on phone rotation angles, biking speeds, and relative distances, to emulate a real-world scenario. Table 16 shows the results. The performance is relatively stable across the ten participants. Hence, CycleGuard works irrelevant to whom the user is. This nice property is due to the fact that CycleGuard involves rather limited user operations, simply mounting the smartphone on the handlebar and turning on the app during the whole process.

Table 17. Demographic information of participants.

ID	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Age	25	26	35	36	23	29	21	24	20	29
Bike usage	Medium	Medium	Low	High	Low	Medium	Low	Medium	Medium	Low
Biking expertise	High	High	Medium	High	Medium	Medium	High	High	High	Low

7 USER STUDY

7.1 Procedures

To evaluate the effectiveness and usability of CycleGuard in real-world scenarios, we perform a user study with the demographic group shown in Table 17. Volunteers are recruited from the university's computer science department by advertising our research study through departmental email lists. Ten volunteers participate. At the end of experiments, they are interviewed with open questions and required to fill the questionnaires, which serve as the basis of our user study. Before the actual experiments, a short training session of 15-30 minutes is arranged wherein volunteers are introduced to CycleGuard and its usage instructions. Volunteers are left with sufficient time to use and get familiar with the app. To examine which alert modality is better perceived by cyclists, we have implemented both flashing screens and beeping sounds as alarms.

The field testing is conducted on different sites as shown in Fig. 29. We did not try crowded urban areas with complex traffic conditions due to the following reasons. First, the goal of this work is to investigate the feasibility of utilizing smartphones to detect and avoid right-hook collisions for cyclists; the above-mentioned real-world testing sites should be sufficient to serve this purpose. The second reason is for the consideration of cyclist's safety; we have to admit that there is still room to enhance CycleGuard before handling every situation, among which the crowded urban area is deemed the toughest.

Table 18. Survey results (1: strongly disagree, 3: neutral, 5: strongly agree).

No.	Questions	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Mean	Variance
Q1	Human sight and hearing are insufficient to avoid right hook collisions.	3	5	5	5	4	4	3	5	3	4	4.1	0.69
Q2	Implication of right hook collisions when biking on road.	4	5	5	5	5	4	4	4	3	5	4.4	0.44
Q3	CycleGuard provides additional information to make judgements needed in avoiding right hook collisions.	4	4	4	4	5	5	5	3	3	5	4.2	0.56
Q4	CycleGuard enhances my judgement in right hook scenarios.	5	5	4	3	5	4	5	3	5	5	4.4	0.64
Q5	False alarms produced by CycleGuard discourages me to use it.	1	1	2	3	3	1	1	3	2	3	2	0.8
Q6	I would prefer audio to visual alert modalities.	5	5	5	5	3	3	5	5	4	4	4.4	0.64
Q7	CycleGuard is easy to use.	4	5	4	5	3	5	5	3	5	3	4.2	0.76
Q8	I am willing to adopt CycleGuard in daily commutes.	5	5	3	5	4	5	3	5	5	3	4.3	0.81

7.2 Survey Results.

Table 18 shows the questionnaire survey reported by the volunteers. Questions Q1-Q2 are designed to analyze the cause and impact of right-hook collisions. Q3-Q5 underline how effectively CycleGuard assists the cyclist in collision detection. Finally, Q6-Q8 highlight the usability of CycleGuard. We ask volunteers to provide ratings according to their preference, from the above mentioned perspectives using a 5-point Likert scale (1 = strongly disagree, 3 = neutral, 5 = strongly agree).

In response to Q1, 7 out of 10 users highly agree that human sight and hearing are insufficient to make appropriate judgments in avoiding right-hook collisions. For example, P5 and P8 stated that they could not clearly know if the front vehicle will make right turns by hearing its brake sounds. P3 mentioned that the chances of experiencing right hook collisions could be relatively high when the cyclist is distracted from the road, e.g., listening to music. The responses of Q2 have low variance and high mean values. It implies that almost everyone agrees that right-hook collisions have considerable impacts on the cyclist's safety. The survey results of Q3 highlight that CycleGuard helps cyclists to make better decisions to avoid right-hook collisions. P1 and P2 specifically mentioned that it is hard to react and apply brakes when the driver makes a sudden right turn. The alert from CycleGuard allows them to react timely. In response to Q4, most of them agreed that CycleGuard is helpful in enhancing their judgement, as described by the high mean score of 4.4. In response to Q5, 6 out of 10 agreed that false alarms did not discourage them from using it, while 4 of them held a neutral opinion. Some of them expressed that it is unlikely that they would entirely depend on CycleGuard to take reactions during riding. They treat the app as a supportive tool to enhance their judgments. Q6-Q8 highlight the evaluation of CycleGuard's usability. In the response to Q6, 8 out of 10 volunteers prefer an auditory alert over visual alert modality. P4 mentioned that beep sounds draw his attention more quickly than the visual alerts. In Q7, almost everyone agrees that CycleGuard is easy to use. In response to Q8, most volunteers expressed their willingness to adopt CycleGuard in their daily commutes.

7.3 Open Questions.

After the in-field testing, volunteers were encouraged to share their experience of using CycleGuard and discuss with each other to exchange their thoughts. The following questions were asked:

"**Q1:** *What is the potential role of CycleGuard in your daily commute?*"

"**Q2:** *What are possible aspects to improve CycleGuard?*"

In response to Q1, most participants stated that CycleGuard would not override their own judgment. Instead, it serves as a great assistive tool that provides timely collision alerts. Some of them also expressed that they would double confirm the road condition if CycleGuard alerts.

In response to Q2, all ten participants perceived CycleGuard as an assistive tool that enhances their outdoor biking experiences. The following themes emerged. **iOS app:** P7 and P9 suggested providing an alternate version of the app on iOS. This could allow iPhone users to adopt CycleGuard in their daily commutes. **Sensor fusion:** P1 and P3 suggested to enhance the system by incorporating other sensing modalities, such as camera and motion sensors. **Notify the driver:** Some participants mentioned that vehicle drivers should also be alerted if their maneuvers pose threats to nearby cyclists. This would help to avoid potential collision too by having the driver play a role. We plan to investigate the feasibility of these suggestions in our future work.

8 DISCUSSIONS

8.1 Scenarios CycleGuard Cannot Handle.

CyclceGuard cannot prevent every collision, for example, when the instant b-v distance is too short. Recall that SSD is a near worst-case distance the cyclist needs to be able to see, react, and brake to have room to stop before colliding with the frontal vehicle. If the instant b-v distance is shorter than SSD and the frontal vehicle suddenly makes a right turn, then a crash might happen. At this moment, even if CycleGuard generates an alert, it cannot prevent the crash.

Besides, CycleGuard cannot detect correctly if the cyclist rides too fast. The ranging distance of our system is about 35 m, beyond which the detection accuracy deteriorates significantly. For example, if a cyclist rides at 30 km/h (pretty fast for normal commuters), then its corresponding SSD, i.e., near worst-case distance the cyclist needs to be able to react and stop before colliding with the frontal vehicle, is about 36 m, which is even larger than the ranging distance 35 m. In this case, CycleGuard does not work. In fact, the maximum cycling speed CycleGuard can handle is around 26 km/h which covers most normal commuters' riding speeds [20]. To further extend the applicable scenarios of CycleGuard, a direct solution is to employ a speaker with higher operational power. Since the speaker works as plug-and-go in our system, the user is able to select one that best fits her riding speed.

8.2 Increasing Detection Frequency.

Under the current design, the computation time of the entire detection process is about 250 ms at least. It implies that the detection frequency is upper-bounded by 4 times per second. In general, the higher the frequency at which the detection is performed, the more fine-grained the detection can be in the time domain. This is because a position curve will thus be composed of more measurement points that allow accurate characterization of an object's trajectory. Correspondingly, the detection accuracy is enhanced. As our future work, we plan to optimize the design of each module so as to shorten the processing latency. Computationally efficient algorithms will be applied and tested.

8.3 Handling Other Collision Scenarios.

This work focuses on the detection of pending right hook crashes. In the real world, there are other bicycle-vehicle crash scenarios, such as "left hook". It occurs when a vehicle turns left without yielding to a cyclist in the oncoming lane. Then the vehicle turning left may strike the cyclist or cut the cyclist off and cause a collision. To recognize other collision scenarios, we plan to investigate multi-target tracking models that can assign measurement points to individual vehicles. Position measurement refinement will be achieved by employing other advanced techniques from control theory, such as extended Kalman filtering. Moreover, since smartphones are equipped with various on-board sensors, we plan to integrate other sensory modalities, such as cameras, into collision detection. It is expected that multi-modality sensing will provide a more comprehensive interpretation of the traffic in cyclist's surroundings. Sensor fusion techniques (such as [65], [34],[2]) will be applied.

8.4 Positioning of CycleGuard.

CycleGuard does not intend to overwrite a cyclist's judgment; instead, it should be treated as an assistive tool that provides an added layer of protection that alerts cyclists with potential crashes. The primary goal of this work is to explore the feasibility of turning smartphones into a "mini-sonar" to sense cyclist's surroundings and protect them from bicycle-vehicle crashes. As discussed, prior approaches either build some sophisticated sensory devices that cost hundreds of dollars or are restricted to detecting approaching vehicles from behind.

According to our experimental results, TPR is above 90% in almost all the testing scenarios. It indicates that CycleGuard can detect right-turn vehicles with high precision. In the meantime, we have to admit that there is still room for improvement before it can handle every situation. As an assistive tool, if a conflict occurs between a cyclist's judgement and the detection result, the cyclist still holds the responsibility of decision making. We provide a set of suggestive choices in the following three cases: 1) When either source, human visual/auditory system or CycleGuard, indicates a potential collision, cyclists need to apply the brake for safety considerations; 2) when neither source produces any alarm, cyclists can continue to proceed with care; 3) when both sources forecast a potential collision, cyclists should take actions immediately.

8.5 Enhance the Detection Accuracy.

The performance of CycleGuard is far from perfect. There is still room for improvement, e.g., further reducing the false alarm. As our future work, we plan to exploit other sensory modules on smartphones, such as LiDAR and RF antennas, to capture more information on road conditions and enhance detection accuracy. Nowadays, more and more smartphones, such as iPhone 12 Pro, Lenovo Phab 2 Pro, and Asus ZenPhone, are equipped with LiDAR. It can capture object's contours, which is useful to differentiate the vehicle from its backgrounds. It could be a desired addition that helps to reduce the false alarms triggered by roadside objects.

8.6 Impact of Ultrasonic Signals to Animals.

CycleGuard operates in the high-frequency range, i.e., 16kHz to 20kHz. Though the emitted chirp signal can hardly be noticeable by humans, it might fall into the hearing range of some animals, such as dogs and bats. CycleGuard's speaker has a sound level of 65.5 dB by using its maximal power. In the meantime, acoustic sounds quickly decay as they propagate in an open air. We observe that the signal strength drops to 35 dB at a distance of 5.05 m. As a reference, a human whisper is typically around 30 dB, whereas conversations are at 60 dB [16] and traffic noises are at the level of 70-80 dB at a distance of 15 m. Therefore, chirps may be perceptible by some animals when they are very nearby but will not bother them louder than the average traffic sounds. Moreover, various applications have been recently developed based on acoustic sensing, such as acoustic imaging [32], pedestrian safety [26, 62] and biometric sensing [50]. Our work can be treated as another application of acoustic sensing. Further, the authors do plan to work with researchers from the University's Biology Department to conduct a systematic study regarding the possible effect of our app on common pet animals, such as dogs and cats.

9 CONCLUSION

This paper presents CycleGuard, an assistive tool to protect cyclists from right hook crashes. Its technical contribution lies in addressing unique challenges of applying acoustic ranging into b-v collision detection. To estimate the target vehicle's position, cross-correlations between emitted and received signals are utilized. Due to software processing latency, the derived cross-correlation is drifted from the ground truth. To address this issue, a practical method is proposed to benchmark and correct the error. Realizing that cyclist's handlebar maneuvers would cause offset in b-v direction measurement, we formulate the relationship between the vehicle's positions before and after handlebar maneuvers. With this basis, we are able to compensate such a measurement offset. Additionally, a novel method is proposed to recognize a right-turn vehicle's moving trace from those of other static and non-static objects. CycleGuard is implemented using a COTS portable speaker and a smartphone. Extensive in-field experiments show that its detection accuracy can reach 95%.

ACKNOWLEDGMENTS

This work was supported by a grant from the Center for Transportation Equity, Decisions and Dollars (CTEDD) funded by U.S. Department of Transportation Research and Innovative Technology Administration (OST-R) and housed at The University of Texas at Arlington.

REFERENCES

[1] Audi AG. 2021. Audi-MediaCenter. https://www.audi-mediacenter.com/en/technology-lexicon-7180/driver-assistance-systems-7184. (2021).

[2] Furqan Alam, Rashid Mehmood, Iyad Katib, Nasser N. Albogami, and Aiiad Albeshri. 2017. Data fusion and IoT for smart ubiquitous environments: A survey. *IEEE Access* 5 (2017), 9533–9554.

[3] Apache. 2021. Apache Math Tool. (2021). http://commons.apache.org/proper/commons-math/

[4] Mercedes Benz. 2021. Mercedes Benz. (2021). https://www.la.mercedes-benz.com/en/passengercars/mercedes-benz-cars/models/eqc/safety.pi.html/mercedes-benz-cars/models/eqc/safety/parking-gallery/parktronic

[5] Eli Billauer. 2021. Peak detection. (2021). http://www.billauer.co.il/peakdet.html

[6] Sebastian Blanco. 2021. CES 2021: Bicycles want to get into the connected vehicle game. (2021). https://www.forbes.com/sites/sebastianblanco/2021/01/15/ces-2021-bicycles-want-to-get-into-the-connected-vehicle-game/?sh=7992e0ee1bd6

[7] S Blenski. 2013. Understanding bicyclist-motorist crashes in Minneapolis, Minnesota. *City of Minneapolis Public Works Department, Tech.Report* (2013).

[8] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization.* Cambridge university press.

[9] Gaoshuai Cao, Kuang Yuan, Jie Xiong, Panlong Yang, Yubo Yan, Hao Zhou, and Xiang-Yang Li. 2020. EarphoneTrack: Involving earphones into the ecosystem of acoustic motion tracking. In *Proceedings of the Conference on Embedded Networked Sensor Systems.*

[10] J. Castillo Guerrero, C. Quezada-V, and D. Chacon-Troya. 2018. Design and implementation of an intelligent cane, with proximity sensors, GPS localization and GSM feedback. In *Proceedings of the IEEE Canadian Conference on Electrical Computer Engineering.*

[11] Volvo Car Corporation. 2021. Volvo. (2021). https://www.media.volvocars.com/global/en-gb/media/photos/158390/autonomous-drive-technology-ultrasonic-sensors

[12] John E Dennis Jr and Robert B Schnabel. 1996. *Numerical methods for unconstrained optimization and nonlinear equations.* SIAM.

[13] Cycling Designs. 2021. CycleSight. (2021). https://https://cyclingdesigns.com/store/

[14] Peter Fairley. 2017. Self-driving cars have a bicycle problem. (2017). https://spectrum.ieee.org/transportation/self-driving/selfdriving-cars-have-a-bicycle-problem

[15] Chiung-Yao Fang, Jui-Hung Liang, Chiao-Shan Lo, and Sei-Wang Chen. 2013. A real-time visual-based front-mounted vehicle collision warning system. In *Proceedings of the IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems.* 1–8.

[16] Centers for Disease Control and Prevention. 2021. What noises cause hearing loss? (2021). https://www.cdc.gov/nceh/hearing_loss/what_noises_cause_hearing_loss.html

[17] Google. 2021. Phone device metrics. (2021). https://material.io/blog/device-metrics

[18] Google. 2021. Pixel phone hardware diagram - Pixel Phone Help. (2021). https://support.google.com/pixelphone/answer/7157629?hl=en#zippy=%2Cfeatures-of-pixel-phones

[19] S. Hisaka and S. Kamijo. 2011. On-board wireless sensor for collision avoidance: Vehicle and pedestrian detection at intersection. In *Proceedings of the International IEEE Conference on Intelligent Transportation Systems.*

[20] JE Hummer, NM Rouphail, JL Toole, RS Patten, Robert J Schneider, JS Green, RG Hughes, and SJ Fain. 2006. Evaluation of safety, design, and operation of shared-use paths—Final report. *FHWA-HRT-05-137* (2006).

[21] Realtime Technologies Inc. 2020. Research Bike Simulator. (2020). https://www.faac.com/realtime-technologies/products/research-bike-simulator/

[22] Woongsun Jeon and Rajesh Rajamani. 2018. Rear vehicle tracking on a bicycle using active sensor orientation control. *IEEE Transactions on Intelligent Transportation Systems* 19, 8 (2018), 2638–2649.

[23] Woongsun Jeon and Rajesh Rajamani. 2019. Active sensing on a bicycle for simultaneous search and tracking of multiple rear vehicles. *IEEE Transactions on Vehicular Technology* 68, 6 (2019), 5295–5308.

[24] S. Kawanaka, Y. Kashimoto, A. Firouzian, Y. Arakawa, P. Pulli, and K. Yasumoto. 2017. Approaching vehicle detection method with acoustic analysis using smartphone for elderly bicycle driver. In *Proceedings of the International Conference on Mobile Computing and Ubiquitous Network.*

[25] Young-Jun Kweon. 2014. Development of an assessment guide for bicycle use of right shoulders on controlled access facilities in Virginia. (2014). https://www.virginiadot.org/programs/resources/bike/BikeGuideReport_final.pdf

[26] Sugang Li, Xiaoran Fan, Yanyong Zhang, Wade Trappe, Janne Lindqvist, and Richard E. Howard. 2017. Auto++: Detecting cars using embedded microphones in real-time. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3, Article 70 (2017).

[27] Kaikai Liu, Xinxin Liu, and Xiaolin Li. 2013. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services.* 235–248.

[28] Cycliq Products Pty. Ltd. 2021. Cycliq - Fly6. (2021). https://cycliq.com/bike-cameras/fly6ce/

[29] Garmin Ltd. 2021. Garmin- Varia™ Rearview Radar. https://buy.garmin.com/en-US/US/p/518151overview. (2021).

[30] Shachar Maidenbaum, Shlomi Hanassy, Sami Abboud, Galit Buchs, Daniel-Robert Chebat, Shelly Levy-Tzedek, and Amir Amedi. 2014. The "EyeCane", a new electronic travel aid for the blind: Technology, behavior & swift learning. *Restorative neurology and neuroscience* 32, 6 (2014), 813–824.

[31] Wenguang Mao, Wei Sun, Mei Wang, and Lili Qiu. 2020. DeepRange: acoustic ranging via deep learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020).

[32] Wenguang Mao, Mei Wang, and Lili Qiu. 2018. AIM: Acoustic imaging on a mobile. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services.* 468–481.

[33] Mobileye. 2021. Mobileye-autonomous driving and ADAS. (2021). https://www.mobileye.com

[34] R. Mobus and U. Kolbe. 2004. Multi-target multi-object tracking, sensor fusion of radar and infrared. In *Proceedings of the IEEE Intelligent Vehicles Symposium.*

[35] FACTHUM.lab. Antena 3. CESV. Fundación AXA Montoro L. 2014. Análisis de la siniestralidad en ciclistas (Analysis of accidents in cyclists) 2008–2013. (2014). http://go.uv.es/jWkiU0E

[36] Akshay Uttama Nambi, Adtiya Virmani, and Venkata N. Padmanabhan. 2018. FarSight: A Smartphone-based vehicle ranging system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4, Article 181 (2018).

[37] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. 2015. Contactless sleep apnea detection on smartphones. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services.* 45–57.

[38] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using active sonar for fine-grained finger tracking. In *Proceedings of the CHI Conference on Human Factors in Computing Systems.* 1515–1525.

[39] Rajalakshmi Nandakumar, Alex Takakuwa, Tadayoshi Kohno, and Shyamnath Gollakota. 2017. CovertBand: Activity information leakage using music. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3, Article 87 (2017).

[40] NHTSA. 2018. Bicyclists and other cyclists - 2018 data. (2018). https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812884

[41] The State of California. 2020. Cal. Veh. Code §§670; 21200. (2020). https://leginfo.legislature.ca.gov/faces/codes_displaySection.xhtml?lawCode=VEH&sectionNum=21200.

[42] The State of Nevada. 2020. Nev. Rev. Stat. §§484A.320; 484B.763. (2020). https://www.leg.state.nv.us/nrs/nrs-484b.html#NRS484BSec763

[43] The State of New York. 2020. N.Y. Veh. Traf. Law §§159; 1231. (2020). https://www.nysenate.gov/legislation/laws/VAT/123

[44] The State of Pennsylvania. 2020. 75 Penn. Cons. Stat. §§102; 3501. (2020). https://www.legis.state.pa.us/cfdocs/legis/LI/consCheck.cfm?txtType=HTM&ttl=75&div=0&chpt=35&sctn=1&subsctn=0

[45] Colorado Department of Transportation. 2013. Roadway design guide. (2013). https://www.codot.gov/business/designsupport/bulletins_manuals/roadway-design-guide/archive-chaps/ch-14-bicycles-pedestrian-facilities

[46] Florida Department of Transportation. 2016. Plans preparation manual. (2016). https://www.fdot.gov/docs/default-source/roadway/PPMManual/2016/Volume1/Chap08.pdf

[47] World Health Organization. 2020. *Cyclist safety: an information resource for decision-makers and practitioners.* World Health Organization.

[48] CROW Platform. 2016. Design manual for bicycle traffic. (2016). https://crowplatform.com/product/design-manual-for-bicycle-traffic/

[49] Swadhin Pradhan, Ghufran Baig, Wenguang Mao, Lili Qiu, Guohai Chen, and Bo Yang. 2018. Smartphone-based acoustic indoor space mapping. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–26.

[50] Kun Qian, Chenshu Wu, Fu Xiao, Yue Zheng, Yi Zhang, Zheng Yang, and Yunhao Liu. 2018. Acousticcardiogram: Monitoring heartbeats using acoustic signals on smart devices. In *Proceedings of the IEEE Conference on Computer Communications.*

[51] X Qian and C Ye. 2013. NCC-RANSAC: A fast plane extraction method for navigating a smart cane for the visually impaired. In *Proceedings of the IEEE International Conference on Automation Science and Engineering.*

[52] A Rodríguez Valiente, A Trinidad, JR García Berrocal, C Górriz, and R Ramírez Camacho. 2014. Extended high-frequency (9–20 kHz) audiometry reference thresholds in 645 healthy subjects. *International journal of audiology* 53, 8 (2014), 531–545.

[53] Samsung. 2021. Specifications | Samsung Galaxy S8 and S8+. (2021). https://www.samsung.com/global/galaxy/galaxy-s8/specs/

[54] Sharang Sharma, Manind Gupta, Amit Kumar, Meenakshi Tripathi, and Manoj Singh Gaur. 2017. Multiple distance sensors based smart stick for visually impaired people. In *Proceedings of the IEEE Computing and Communication Workshop and Conference.*

[55] Tushar Sharma, Tarun Nalwa, Tanupriya Choudhury, Suresh Chand Satapathy, and Praveen Kumar. 2017. Smart cane: Better walking experience for blind people. In *Proceedings of the IEEE International Conference on Computational Intelligence and Networks.*

[56] Stephen Smaldone, Chetan Tonde, Vancheswaran K. Ananthanarayanan, Ahmed Elgammal, and Liviu Iftode. 2011. The Cyber-physical bike: A step towards safer green transportation. In *Proceedings of the Workshop on Mobile Computing Systems and Applications.*

[57] Tesla. 2021. AutoPilot. (2021). https://www.tesla.com/autopilot

[58] Libby Thomas, Krista Nordback, and Rebecca Sanders. 2019. Bicyclist crash types on national, state, and local levels: A new look. *Transportation Research Record* 2673, 6 (2019), 664–676.

[59] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication.* Cambridge university press.

[60] Yu-Chih Tung and Kang G Shin. 2018. Use of phone sensors to enhance distracted pedestrians' safety. *IEEE Transactions on Mobile Computing* 17, 6 (2018), 1469–1482.

[61] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of the international conference on mobile computing and networking.*

[62] Zi Wang, Sheng Tan, Linghan Zhang, and Jie Yang. 2018. ObstacleWatch: Acoustic-based obstacle collision detection for pedestrian using smartphone. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4, Article 194 (2018).

[63] Jiaqi Wen, Jiannong Cao, and Xuefeng Liu. 2015. We help you watch your steps: Unobtrusive alertness system for pedestrian mobile phone users. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications.*

[64] ATC; Arabian Sandra Strack BS CSTR CAISS; Breeze Janis L. MPH; Salzler Matthew J. MD Wolfe, Elizabeth Suzanne CAGS. 2016. Distracted biking. *Journal of Trauma Nursing* 23, 2 (2016), 65–70.

[65] Woongsun Jeon and R. Rajamani. 2016. A novel collision avoidance system for bicycles. In *Proceedings of the American Control Conference.*

[66] Z. Xie, W. Jeon, and R. Rajamani. 2021. Low-density Lidar based estimation system for bicycle protection. *IEEE Transactions on Intelligent Vehicles.* 6, 1 (2021), 67–77.

[67] Cang Ye, Soonhac Hong, Xiangfei Qian, and Wei Wu. 2016. Co-robotic cane: A new robotic navigation aid for the visually impaired. *IEEE systems, man, and cybernetics magazine* 2, 2 (2016), 33–42.

[68] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-grained acoustic-based device-free tracking. In *Proceedings of the international conference on mobile systems, applications, and services.*

[69] Yunting Zhang, Jiliang Wang, Weiyi Wang, Zhao Wang, and Yunhao Liu. 2018. Vernier: Accurate and fast acoustic motion tracking using mobile devices. In *Proceedings of the IEEE Conference on Computer Communications*.

[70] Shifeng Zhou. 2018. A Smart cane to help the blind people walk confidently. *Materials Science and Engineering* 3 (2018), 032121.