Joint Estimation of the Distance and Relative Velocity of Obstacles via Smartphone Active Sound Sensing for Pedestrian Safety

Thilina Dissanayake
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
thilina.dissanayake@ist.osaka-u.ac.jp

Takuya Maekawa
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
maekawa@ist.osaka-u.ac.jp

Takahiro Hara
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
hara@ist.osaka-u.ac.jp

***Abstract*—In this study, we proposed a new method for pedestrian safety named ObsSense that estimates the distance and relative velocity of both static and mobile roadside obstacles using probing signals emitted by an off-the-shelf smartphone carried by a pedestrian. Sound-based estimation of the movement properties of an obstacle has been actively studied in pervasive computing, and generally provides the distance from a smartphone user to an obstacle and the relative velocity of the obstacle. However, most prior studies have focused on estimating the properties of either static or mobile obstacles. Furthermore, they estimated either the distance to obstacles or their relative velocity. Nonetheless, both the distance and relative velocity of the oncoming obstacles are necessary to derive the timing of a possible collision. To address this issue, we proposed a new probing signal composed of sine waves and sine sweeps, designed to acquire information about both the distance and relative velocity of static and mobile obstacles. In addition, we proposed a neural network model that can jointly predict distance and relative velocity from reflected sounds of the probing signal captured by a smartphone. The proposed neural network was designed to consider the mobility status of obstacles, i.e., whether they are mobile or static, in order to select the most appropriate method to estimate their distance and relative velocity. In addition, our method uses the relationship between distance and velocity to increase the precision of its estimations. For example, the distance can be estimated by integrating the velocity-time curve. The performance of ObsSense was evaluated using real-world data, and the experimental results demonstrated the effectiveness of our proposed probing signal and neural network architecture.**

***Index Terms*—Active sound sensing, pedestrian safety, obstacles**

I. Introduction

A. Background

Due to the recent proliferation of smartphones and related applications, the number of pedestrians that use such devices for activities other than voice calls while walking has increased, and uses such as browsing social media, sending and receiving text messages, watching videos, playing games, and navigation are common. These pedestrians tend to concentrate on their smartphones rather than the road, and may collide with various roadside obstacles, which could lead to serious injuries and even death. As an example, Nasar et al. [1] showed that 1506 injuries were estimated to have occurred in the US in 2010 due to distracted pedestrians using their smartphones. Furthermore, the American Academy of Orthopaedic Surgeons (2015) found that 26% of respondents to a survey of 2000 adults had experienced accidents due to using their smartphones while walking, ranging from bumping into something without injury, to falls, sprains or fractures [2].

In the field of pervasive computing, studies have been actively conducted on detecting oncoming obstacles using data from different onboard smartphone sensor modalities. The footage from the rear-camera of a smartphone has been used to recognize obstacles in front of the user [3]–[8]. However, this method involves some inherent problems, such as limited detection range due to the orientation of the device in use, poor performance in dark environments, and privacy issues. To address these problems, more recent works on obstacle detection have focused on methods based on acoustic sensing. Some previous studies have aimed to detect the presence of obstacles in front of the user with which they may collide [9]–[12]. Recent works have not only detected obstacles, but also estimated other properties, such as their types [13], distance from the user [11], [12], and relative velocity [13]. In this study, we also focused on estimating properties of obstacles, particularly the distance and relative velocity, because this information is crucial to estimate the possible timing of collisions and the level of risk of a given obstacle (e.g., high-speed obstacles exhibit higher risk), which is used to issues warnings to user.

B. Problems

Existing methods designed to estimate properties (distance and relative velocity) of obstacles involve some limitations, including i) limitations in types of obstacles and ii) limited types of properties that can be estimated. As for the first limitation, many previous studies have focused only on static or slow-moving obstacles, such as walls, dustbins, donation boxes, and signs. However, sidewalks often include numerous mobile obstacles, such as pedestrians and bicycles, which can be considered as high-risk obstacles. As for the second limitation, most of the existing methods have focused only on either the distance or relative velocity, although both are crucial information for pedestrian safety, as mentioned above.

These limitations are attributed to the types of probing signals used in these studies. Each of these works relied on either sweep or sinusoidal signals. Although the short pulses of sweep signals have been widely used to estimate the distance between the user and a static obstacle by analyzing reflected sounds [11], [14]–[17], estimating the velocity of mobile obstacles with sweep signals is difficult. Some studies have attempted to detect mobile obstacles and estimate their velocity using time-series analysis of sweep signals, i.e., by tracking the distance to an obstacle within each time window [13], [18]. However, to track the mobility of high-speed obstacles with high granularity, the number of sweeps generated by the smartphone per second should be increased. Due to the inherent characteristics of commercial speakers, when a large number of sweeps are generated per second, the amplitude of the sweeps falls, resulting in the limited sensing range [13].

By contrast, continuous sinusoidal signals (sine waves) have been employed to capture the Doppler shifts created by mobile obstacles to estimate their velocity [19], [20]. However, this approach is not suitable to estimate the distance to an obstacle. In addition, estimating the relative velocity of static obstacles with this approach is difficult because they do not create characteristic Doppler shifts.

As noted above, to the best of our knowledge, no study has attempted to estimate both the distance and velocity of both static and mobile obstacles in a single framework.

C. Approach

To overcome the abovementioned limitations, we proposed a new method named **ObsSense**, which is designed to estimate the distance and relative velocity of both static and mobile obstacles using an off-the-shelf-smartphone. The main features of ObsSense include i) novel inaudible probing sound signals composed of sweep signals and sine waves, which enable joint estimation of the distance and velocity of both static and mobile obstacles, and ii) a novel neural network architecture that efficiently fuses features of the reflected sounds of the probing signal to estimate both the distance and velocity of a given obstacle.

As for the first feature, the probing signal was designed such that i) the signal composed of sine wave and sine sweep is continuous in the frequency domain, which prevents audible sound noises in the signal due to signal power leakage caused by discrete frequency generation by the speaker [21], and ii) our probing signal maintains a constant volume over both sweep and sine wave components, which facilitates capturing distant obstacles.

As for the second, the most appropriate strategy to estimate distance and velocity depends on the movement status of an obstacle, i.e., whether it is static or mobile. For example, reflected sine-wave signals, which contain information about Doppler shift, are more useful in predicting the velocity of a mobile obstacle. By contrast, reflected sweep signals are more effective in predicting the velocity of a static obstacle. Therefore, our proposed neural network is equipped with a motion detection module that identifies the movement status of an obstacle as static or mobile. By leveraging the intermediate outputs of the module, we accelerate the training of the neural network so that it learns an appropriate strategy to estimate the distance and velocity of obstacles depending on their movement status. In addition, our network is designed to estimate the distance and velocity by fusing reflected signals of sweep signals and sine waves. For example, reflected signals of sweep signals are useful to predict the distance to an obstacle while they are sparse and noisy. Our network is designed to leverage reflected signals of sine waves, i.e., Doppler shift, as a supplement to predict the distance because the distance at time t can also be estimated based on the distance at time $t-1$ and the velocity at that time, i.e., using integration, which enables us to improve the distance prediction performance.

D. Contributions

i) This study is the first to estimate the distance and relative velocity of both static and mobile obstacles using active acoustic sensing on a smartphone. ii) We proposed a novel probing signal that facilitates both acoustic ranging and velocity estimation. iii) We proposed a novel neural network architecture designed to estimate the mobility of the detected obstacles and used this knowledge to effectively fuse sound features to estimate their distance and relative velocity.

II. Related work

In the pervasive computing community, smartphone sounds have been used for various real-world context recognition, e.g., prediction of room-level indoor location semantics [22], [23], event detection of indoor objects such as doors [24], recognition/understanding of indoor activities such as tooth brushing [25], and obstacle detection. Here we introduce prior studies on sound-based obstacle detection. Generally, studies on obstacle detection can be divided into two major groups according to the nature of the targeted obstacles; (i) static obstacle detection methods, (ii) mobile obstacle detection methods.

A. Static obstacle detection and distance estimation

Low-cost ultrasonic sensors have been employed to detect and estimate the distance to obstacles [9], [26], [27]. However, these methods require additional attachments to the smartphone or special devices. Recently, active sound sensing

has been widely adapted to perform static obstacle detection and distance estimation on smartphone devices. Some studies have detected obstacles with a possibility of collision by employing the camera as well as active sound sensing [12] and active sound sensing with two onboard microphones [11]. In addition, the distance between a smartphone user and a static obstacle was estimated using sound probes. Specifically, Tung et al. [12] used a smartphone speaker to emit short pulses of sine waves and record the reflections with the microphone. They then calculated the impulse response of the reflections to estimate the distance to obstacles. This was performed by accurately calculating the traveling time of the emitted signal between the user and the obstacle using a threshold-based method to detect the peaks created by the obstacle in the calculated impulse response. More recently, Wang et al. [11] used a smartphone to emit inaudible sweep signals and estimate the distance to obstacles in front of a user by calculating the correlation between the emitted and received signals. Similarly, they employed a threshold-based method to detect the peaks in the impulse response. By contrast, our study focused on *both* the distance and velocity. Furthermore, rather than using a threshold-based method, we utilized a neural network to extract features related to the distance from the calculated impulse responses.

B. Mobile obstacle detection and velocity estimation

Passive sensing with smartphones has been employed to detect mobile obstacles, such as approaching vehicles [28]–[30]. Some recent works have also aimed to estimate both the distance and relative velocity of the obstacles at the same time [31] by employing acoustic signals emitted by vehicle-mounted speakers. However, these methods only work for moving vehicles that actively emit characteristic noise, and are not suitable to detect other mobile obstacles or static obstacles.

In the field of active sound sensing via smartphones, acoustic ranging techniques using excitement signals, such as sweep signals, have been adapted to detect and estimate the velocity of mobile obstacles. Wu et al. [13] emitted short sweep signals from a smartphone and captured the reflections of vehicles. They estimated the speed of vehicles by capturing the change in the distance between the user and the vehicle over time. Jin et al. [18] used a COTS speaker connected to a smartphone to detect right turns of vehicles and alert cyclists prior to potential accidents. They emitted periodic sweeps from the speaker and analyzed time-series reflections to detect approaching vehicles. By contrast, we designed a new probing signal composed of sweep signals and sine waves, and fuses them to estimate the relative velocity.

Sine waves have also been widely used to capture Doppler shifts created by the movements of obstacles and to estimate their velocities. Zhang et al. [19] utilized an inaudible sine wave emitted by a smartphone to capture the Doppler shifts created by an approaching person. They extracted Doppler profiles to identify trajectories of people approaching each other. Similarly, Liu et al. [20] proposed a method to detect indoor pedestrian encounters and estimate their velocities using a sine wave. However, this method is not suitable to detect static obstacles, as they do not produce characteristic Doppler shifts. By contrast, our proposed method was designed to estimate the velocity of both static and mobile obstacles.

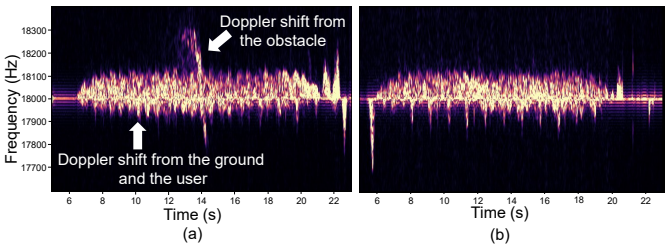

Fig. 1: FFT spectrograms of reflected sounds of sine waves recorded when (a) a smartphone user was walking toward a person while the person was also walking toward the user and (b) a user was walking toward a stationary person.

III. Investigation and Probing Signal Design

ObsSense was designed to estimate the distance and relative velocity of both static and mobile obstacles. To this end, we investigated the characteristics of the different probing signals that can be used to estimate different properties of static and mobile obstacles. Considering the outcomes of our investigations, we proposed a novel probing signal design that facilitates both distance and velocity estimation.

A. Obstacle mobility detection

Obstacles on the sidewalk can be divided into two major groups according to their mobility; static obstacles such as walls, signposts, and parked vehicles, and mobile obstacles such as walking pedestrians and bicycles. As mentioned in the introduction, the proposed neural network was designed to detect the mobility of an obstacle. To do so, we can employ the well-known Doppler shift. Figure 1 shows the Doppler shift created on an 18 kHz sine wave during two different encounters with obstacles. Here, a smartphone user walked towards two different obstacles while the smartphone emitted an 18 kHz sine wave. Simultaneously, the top microphone of the device recorded the reflected waves. Figure 1 (a) shows the characteristic Doppler shift created by a pedestrian (obstacle) who walked towards a user. However, when the user walks towards a stationary person (Figure 1 (b)), a characteristic Doppler shift cannot be observed. This is because the Doppler shifts created by a static obstacle are not clearly visible against the Doppler shifts created by the moving body parts of the user and the ground. These characteristic Doppler shifts can be used to differentiate between mobile and static obstacles.

B. Obstacle distance estimation

To estimate the distance of obstacles, we can employ a short excitement signal emitted by the smartphone's speaker, known as a sweep signal. In the proposed approach, sweep signals are emitted from the speaker periodically and the reflected signals are captured. In this experiment, the length of the sweep length was 0.05 sec and the duration between the sweeps was 0.25 sec. Next, we calculated the impulse response of the reflected waves and extracted their signal envelopes (IR envelopes; [11],

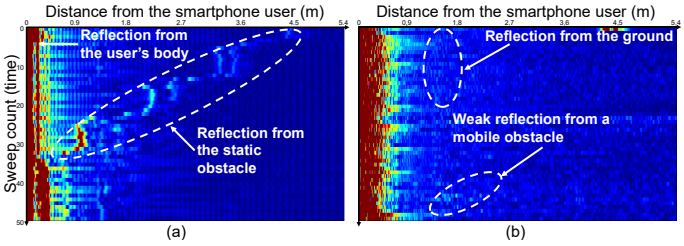

Fig. 2: (a) Reflections of a static obstacle (a wall) and (b) reflections of a mobile obstacle (a jogging person), recorded on IR envelopes. Each IR envelope is 6000 data points long (x-axis). Considering the speed of sound as 340 m/s and sampling rate as 192 kHz, the distance resolution of each IR envelope sample corresponds to $\frac{340\text{m/s}}{192\text{kHz}} \approx 0.18\text{cm}$. Considering the round-trip time, the maximum distance from which the reflections can be captured is $(0.18\text{cm} \times 6000)/2 = 540\text{cm} = 5.4\text{m}$

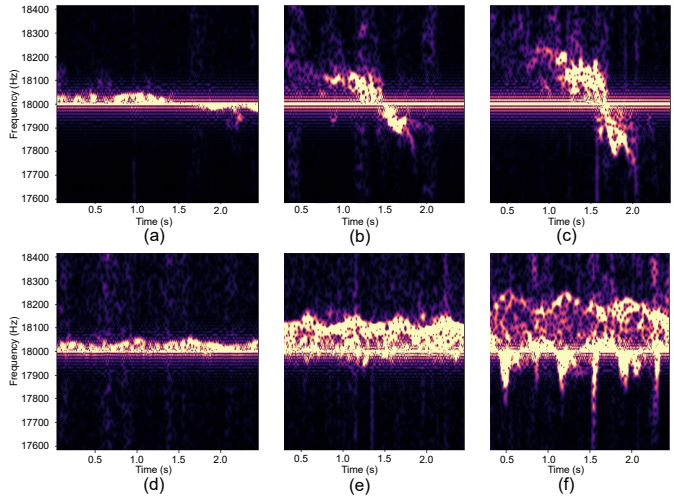

Fig. 3: FFT spectrograms of reflected sounds of sine waves recorded when a pedestrian walked towards a static user at (a) slow, (b) medium, and (c) fast speeds and a user walking towards a wall at (d) slow, (e) medium, and (f) fast speeds.

[18], Section IV-C in detail). These IR envelopes contain the intensity of reflections from an obstacle for each distance.

Figure 2 (a) shows the reflections from a wall recorded on the IR envelopes when the user walks towards a wall. This image was created by time-series stacking of 50 separate IR envelopes. As may be observed, when the user moved towards the obstacle, the reflection appeared closer. This information can be used to estimate the distance to obstacles. However, mobile obstacles with high speeds do not produce reflections with high granularity, as the number of emitted sweeps within a second is limited (Section I-B). Figure 2 (b) shows the reflections from a jogging person recorded on the IR envelopes, which were measured with low granularity because the person was moving rapidly. Hence, we proposed a neural network architecture that incorporates both the IR envelopes and Doppler features to estimate the distance of obstacles, because we can observe clear Doppler shift of the mobile obstacle as shown in Figure 1 (a).

C. Obstacle velocity estimation

The frequency offset created by a mobile obstacle is directly proportional to its relative velocity with respect to the user. Figure 3 (a,b,c) shows the Doppler shifts created by a pedestrian walking toward the user at different speeds. As may be observed, the higher the velocity of the pedestrian, higher the frequency offset of the Doppler shift. However, when an obstacle is static, estimating its relative velocity is difficult, as shown in Figure 1 (b). Although the velocity of a static obstacle can be predicted from the time-series of the distance to the obstacle obtained by sweep signals, reflected sounds of the sweep signals are sparse and noisy, as shown in Figure 2. Here, note that the relative velocity of the static obstacle is equal to the velocity of the user, meaning that we can predict the relative velocity from the velocity of the user. As shown in Figure 1 (b), the reflected sounds contain the Doppler shifts created by the human body as well as the ground, and they can change depending on the velocity of the user. Figure 3 (d,e,f) shows the difference in Doppler shifts created by the user's body and the ground depending on their walking speed. The result indicates that reflected sounds of sine waves can be used to predict the relative velocity of a static obstacle in addition to reflected sounds of sine sweeps.

D. Probing signal design

To implement ObsSense in real-world applications, the probing signal should satisfy the following requirements. i) As shown in the above investigation, to capture the distance and relative velocity of static and mobile obstacles, a probing signal composed of both sweep signal and sine wave components is required. ii) To estimate the distance and relative velocity of high-speed mobile obstacles, each signal component should either be emitted continuously or in rapid succession. iii) To alert the user prior to a collision, oncoming obstacles should be detected from as far away as possible. iv) The probing signal should be inaudible to the human ear. We considered frequencies above 18 kHz as meeting this requirement. However, we found in our preliminary investigations that the frequency response of the smartphone's microphone decreased with frequency (frequencies up to 21 kHz exhibited reliable frequency response). Therefore, using higher frequencies adversely affected the sensing range.

Figure 4 (a) shows an FFT spectrogram of reflected sounds of an ideal probing signal that emits a continuous sine and periodic sweep signals simultaneously in two separate frequency bands. The frequency of the sine wave was 22 kHz and sweep range was 18–21 kHz. Both components were separated by a 1 kHz frequency band to prevent Doppler shifts from the sine wave from overlapping with the sweep signals. However, this probing signal involves two major drawbacks. First, at the beginning and end of each sweep signal, the speaker emits a distinct and audible noise that has a frequency lower than 18 kHz (denoted by white arrows). Furthermore, these noises interfere with the emitted sine wave as well. This issue is related to commercial speakers, and is known as signal power leakage due to frequency-hopping, which is caused by emitting

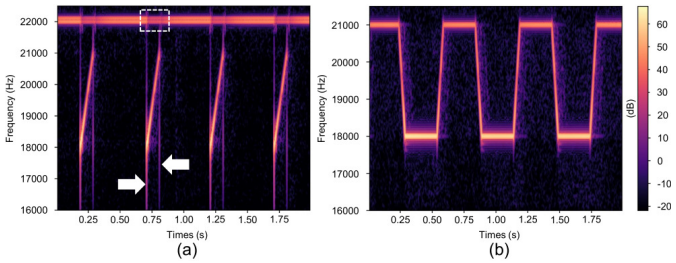

Fig. 4: FFT spectrograms of two probing signal designs

on discrete frequencies consecutively [21]. Second, as periodic sweeps are emitted simultaneously with the sine wave, the speaker power is divided between the two signal components. Hence, each component is emitted at half the maximum power of the speaker, resulting in the reduced maximum sensing range of our method. As may be observed in the rectangle outlines in white dots in Figure 4 (a), the sine wave segments that are being simultaneously emitted with the sweep signals exhibit significantly low amplitudes compared to the remainder of the signal.

To overcome these problems, we proposed a novel probing signal that emits a combination of sweep signals and sine waves in quick succession. Figure 4 (b) shows the FFT spectrogram of reflected sounds of the proposed probing signal. In this signal, every 18 kHz sine wave is followed by an 18–21 kHz sweep signal, followed by a 21 kHz sine wave, which is then followed by a 21 kHz–18 kHz inverse sweep. As may be observed, the proposed probing signal is continuous in the frequency domain, without consecutive discrete frequencies, which eliminates noises due to signal power leakage. Furthermore, this approach solves the speaker power division problem by emitting one signal at a time at full power, which increases the sensing range. In our implementation, each sine wave and sweep/inverse sweep were 0.25 and 0.05 sec long, respectively. The length of the sweep/inverse sweep was selected to minimize system detection delays and multi-path distortions [11]. Furthermore, the interval of sweep/inverse sweep signals (i.e., the length of the sine wave) was determined based on our preliminary experiments. When the number of sweeps emitted per second increased, the overall amplitude of the emitted sweeps decreased, resulting in limited sensing range. Therefore, we selected the interval that did not affect the amplitude of the sweeps. A similar interval between the sweep signals was also used in [18].

Because the probing signal emits sine waves between sweep signals, and ObsSense learns connections between the Doppler and IR features to jointly estimate the distance and velocity, our method can estimate the distance between the sweeps with high granularity (at 100 Hz in our implementation).

IV. ObsSense Method

A. Overview

In this study, we assumed that the user holds the smartphone in their hand such that they can view the content of the screen of the phone clearly while walking on the sidewalk. A probing signal designed in the previous section is emitted from the top speaker of the smartphone and the top microphone records the reflected waves at a sampling frequency of 192 kHz.

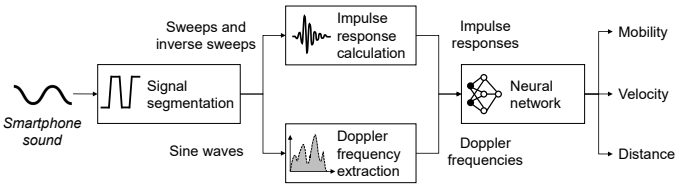

Fig. 5: Overview of ObsSense

Figure 5 shows an overview of the proposed approach. The recorded audio is first fed into the signal segmentation module, which separates the reflected signal into reflections of sine waves, sweeps, and inverse sweeps. Next, the reflections of sine waves are fed into the Doppler frequency extractor and the reflections of sweeps and inverse sweeps are fed into the impulse response calculator. The resulting impulse responses and Doppler frequency features are then fed into a specially tailored neural network that jointly estimates the distance and the relative velocity as well as the mobility of the obstacle. Note that the velocity can be calculated from the distance through derivation. However, because the information about the velocity is important in the distance prediction when sine sweeps are not emitted, this architecture, which explicitly predicts the velocity, can achieve precise distance prediction with high granularity. In addition, we assumed a module that detects an obstacle in front of the user with the possibility of collision, which has been extensively studied in prior works based on probing signals, such as sweep signals and sine waves [9]–[12]. When an obstacle is detected, ObsSense predicts its attributes. We explain the modules of ObsSense as follows.

B. Signal segmentation

Here, we segmented the recorded reflected sound into sine waves, sweeps, and inverse sweeps because sound features are extracted from each of the segments in the following procedure. We extracted segments corresponding to reflections of sine waves, sweeps, and inverse sweeps from the FFT spectrogram of the received signal (as shown in Figure 4 (b)) calculated using a Hanning function on a 90% overlapping sliding window with 16,384 samples. Ideally, the smartphone can record the starting and ending times of the sine waves, sweeps, and inverse sweeps because the smartphone generates and emits the probing signals. However, due to several factors, such as noise and delays in sound emission and reflection, the actual starting and ending times of the reflections differed slightly from the times recorded by the smartphone. To acquire exact starting and ending times, we generated a pseudo-FFT spectrogram of the probing signal by using the recorded starting and ending times of the sine waves, sweeps, and inverse sweeps, and then fine-tuned the starting/ending times to find the times that yield the best match between the pseudo-spectrogram and recorded spectrogram.

This is done by simply shifting the starting and ending times of the pseudo-spectrogram until the Euclidean distance between the two spectrograms is minimized. Note that when

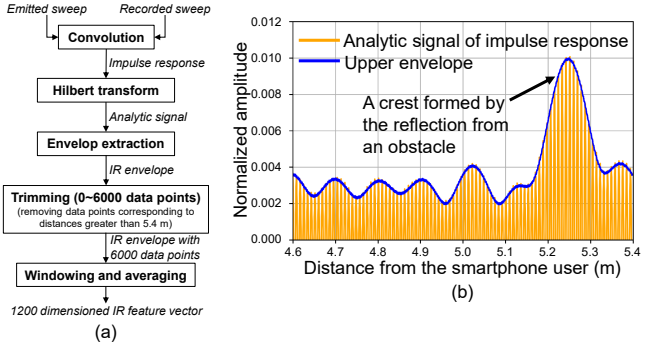

Fig. 6: (a) IR feature extraction pipeline (b) extracted analytic signal and its upper envelope

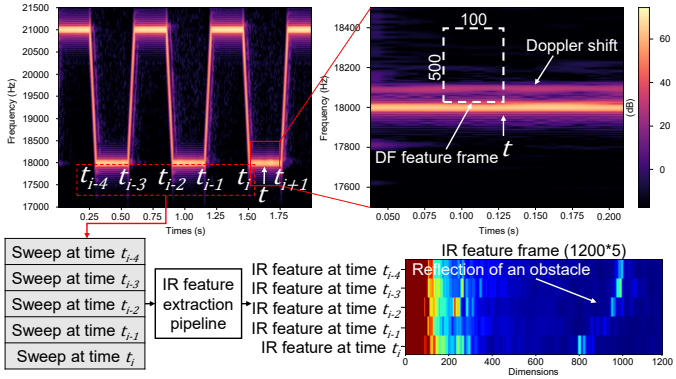

Fig. 7: DF and IR feature preparation for ObsSense neural network

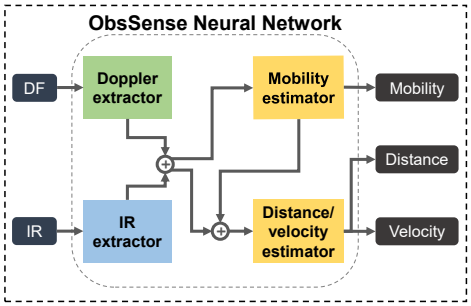

Fig. 8: Structure of ObsSense neural network. Both the Doppler extractor and the IR extractor include three convolutional layers, each followed by max-pooling layers with a kernel size and step size of 3. Six kernels are included in all 3 convolutional layers in the Doppler extractor, with sizes of 11, 3, and 3, respectively. The number of kernels of the IR extractor are 6, 16, 6 and their step size is 3. Both the mobility estimator and the velocity/distance estimator include five fully connected layers with 500, 100, 50, 30, and 2 nodes, respectively. Each fully connected layer uses the ReLU activation function, except for the last fully connected layer of the mobility detector, which uses the sigmoid function.

we calculated the Euclidean distance, the distance for each time step was calculated and the sum over all the time steps within the time window was used as the value of the distance metric. Based on the predicted starting and ending times, the reflected signal can be separated into sweeps, inverse sweeps, and sine waves.

C. Impulse response calculation

Figure 6 (a) shows the IR feature extraction pipeline. Here, we calculated the impulse response using the segmented sweep (or inverse sweep) using the convolution between the recorded signal and the time-reversed transmitted signal [11]. Subsequently, we calculated the analytic representation of the impulse response using the Hilbert transform and extracted the positive amplitudes of the analytic signal. Figure 6 (b) shows the analytic signal of the impulse response and the extracted upper envelope of the analytic signal. Thereafter, we separated the first 6000 data points from the envelope. Using the extracted 6000 data points, we can acquire obstacle distance data from as far as 5.4 m (Figure 2), which would be sufficient to achieve the goal of this study. We further reduced the dimensionality of this envelope up to 1200 by employing a rolling window with 100 samples with a stride of 5 samples and replaced the values of each window with the average value within the window, weighted by the standard deviation. This amplified the crests contained in the envelopes, caused by the reflections of the obstacle. As above, a vector of 1200 dimensions was computed from each envelope. We used these vectors as inputs to the neural network.

D. Doppler frequency extraction

We extracted 500 frequency bins from the top of the bandwidth of the 18 or 21 kHz sine wave from the FFT spectrograms, as shown in the top right panel of Figure 7. In this case, we were only interested in the positive Doppler shifts that occur when an obstacle approaches the user. We used these Doppler shifts as inputs to our neural network.

E. ObsSense neural network architecture

The input for the ObsSense neural network is composed of two features: i) time-series Doppler frequency features (DF features) extracted from the sine waves, and ii) the upper envelopes of the impulse responses (IR features) extracted from periodic sweep signals. Figure 7 shows the preparation of DF and IR feature frames. For a given time t, we use the previous 100 Doppler samples to form a DF feature frame of size 500×100. We further stacked the 5 closest previous IR envelopes to form an IR feature frame of size 1200×5. When sweeps were emitted at times t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1}, and t_i and will be emitted at time t_{i+1}, to estimate the distance and the relative velocity of the obstacle at time t ($t_i < t < t_{i+1}$), we employed the envelopes of the sweeps emitted at times t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1}, and t_i. By time-series stacking of the IR features, we captured information regarding the mobility of the obstacles.

The structure of the ObsSense neural network is shown in Figure 8. The Doppler extractor and IR extractor are composed of 2D convolution layers that process the DF features and IR features, respectively. The outputs of the Doppler extractor and IR extractor are then concatenated and fed into the mobility estimator. The mobility estimator has one output, which is the mobility label of the obstacle, i.e., static or mobile. Next, the intermediate output of the mobility extractor (4th layer output) is aggregated with the DF and IR features and fed into the

distance/velocity estimator. By aggregating the intermediate output of the mobility detector, we allowed the neural network to select the most suitable method to predict the distance and the relative velocity of the obstacle based on its mobility. The distance/velocity estimator has two outputs, including the distance between the user and the obstacle and the relative velocity of the obstacle.

When the neural network is trained, we minimized the following loss function using backpropagation based on the Adam optimizer. [32].

$$E(\theta_D, \theta_I, \theta_m, \theta_{d,v}) = \mathcal{L}_m(\theta_D, \theta_I, \theta_m) + \mathcal{L}_d(\theta_D, \theta_I, \theta_{d,v}) + \mathcal{L}_v(\theta_D, \theta_I, \theta_{d,v}) + \mathcal{L}_{int}(\theta_D, \theta_I, \theta_{d,v}), \quad (1)$$

where θ_D, θ_I, θ_m, and $\theta_{d,v}$ represent the network parameters of the Doppler extractor, IR extractor, mobility estimator, and distance/velocity estimator, respectively. $\mathcal{L}_m()$ denotes the loss of the mobility estimation, and calculates the binary cross-entropy loss of the prediction of the mobility classes, i.e., static or mobile. $\mathcal{L}_d()$ and $\mathcal{L}_v()$ denote the losses of the distance estimation and relative velocity estimation, respectively. $\mathcal{L}_d()$ and $\mathcal{L}_v()$ calculate the RMSLE (Root Mean Squared Log Error) losses of distance and velocity estimations, respectively. $\mathcal{L}_{int}()$ is introduced to leverage the relationship between the distance and velocity to supplement the estimations provided by ObsSense. Particularly, we employed the following relationship between the distance and velocity.

$$d_{t-1} = d_t + v_t \Delta t, \quad (2)$$

where d_t is the distance to the obstacle at time t, v_t is the relative velocity at time t, and Δt is the time difference between t and $t-1$. $t-1$ is the timestamp of the closest previous DF and IR feature frames. $\mathcal{L}_{int}()$ calculates the RMSLE loss of d_{t_i-1} prediction as follows.

$$\mathcal{L}_{int}() = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(d_{t_i-1}^{truth}+1) - \log(\tilde{d}_{t_i-1}+1))^2}. \quad (3)$$

Here, t_i denotes the time of the i^{th} instance, $\tilde{d}_{t_i-1}$ is a calculation result of d_{t_i-1} using Eq. 2 with distance/velocity estimates at t_i, $d_{t_i-1}^{truth}$ is the ground-truth of d_{t_i-1}, and n is the number of training instances.

V. Evaluation

A. Dataset

We collected data from five different static obstacle classes and three mobile obstacle classes that can be commonly found on sidewalks, as shown in Table I, selected based on prior studies. Data were collected from obstacles situated inside the campus and in residential areas around the campus that were relatively busy during the daytime. The participant (smartphone user) was asked to walk toward an obstacle, starting 5 m from the obstacle, and randomly varying their walking speed while looking at the smartphone. The mobile obstacles started from much further (to build speed) and moved toward and past the user as close as possible to simulate a near-miss scenario. For this experiment, we used a Samsung S20 smartphone. The smartphone emitted the above probing signal and recorded the reflected waves. Simultaneously, the participant carried a TFMini Plus LiDAR module connected to an M5Stack Basic model to collect ground-truth data on the distance to obstacles at 1kHz. We used this time-series distance information to calculate the ground truth of the relative velocity of the obstacles through derivation. Table I shows the number of encounters of each obstacle class.

TABLE I: Number of encounters per obstacle type

	Label	Encounters
static	parked vehicle	38
	billboard	36
	telephone pole	57
	wall	44
	standing person	27
mobile	walking person	30
	jogging person	30
	bicycle	65
Total		327

Here we explain the rationale of using the TFmini Plus LiDAR for ground-truth collection. We experimented with three methods to collect the distance ground-truth using sensors: HC-SR04 acoustic sensor, which performed poorly in outdoor environments, TFmini Plus LiDAR, and RealSense D455 depth camera. The acoustic sensor and the LiDAR measures distance between the user and a point on the obstacle. While the depth camera can be used to capture the depth to the obstacle as a whole, it only provides a frame rate of 30 Hz with $<$2% depth accuracy within 4 m, while the TFmini Plus provides up to 1000 Hz frame rate at 5 mm distance resolution and up to 12 m sensing range. As ObsSense provides a sensing range of 5.4 m and a frame rate of 100 Hz, the preferred ground-truth collection method was the TFmini Plus LiDAR. When collecting data, the user held the smartphone by one hand and the LiDAR was pointed towards the obstacle. Both the smartphone and the LiDAR was held at same height from the ground and was pointed approximately at the same point on the obstacle to ensure the consistency of the measurement. However, the radius of the light spot of the LiDAR at 5 m is 12-18 cm, therefore the ground-truth distance may slightly differ when the light hits an inclined surface, such as the windshield of a vehicle. Even though it is possible to place a big-enough vertical target on the obstacle and point both the smartphone and the LiDAR at it, the target itself will reflect the acoustic signals and this will deviate the collected data from real-life scenarios.

B. Evaluation methodology

To evaluate the performance of ObsSense, we first randomly separated our dataset (encounters) into training and testing datasets using 70% and 30%, respectively, of the total data. No data from the training encounters (obstacles) were used to test the model. Furthermore, data from only one encounter per each obstacle was used to construct the dataset. We prepared the following methods to investigate the effectiveness of ObsSense.

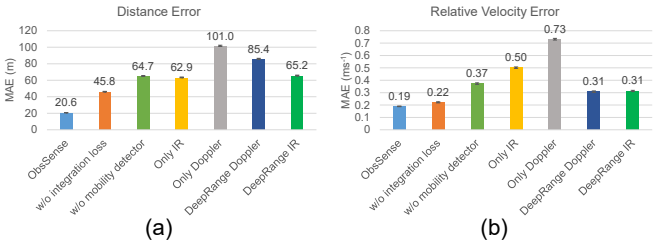

Fig. 9: Comparison of (a) distance and (b) relative velocity prediction errors of ObsSense and the other methods

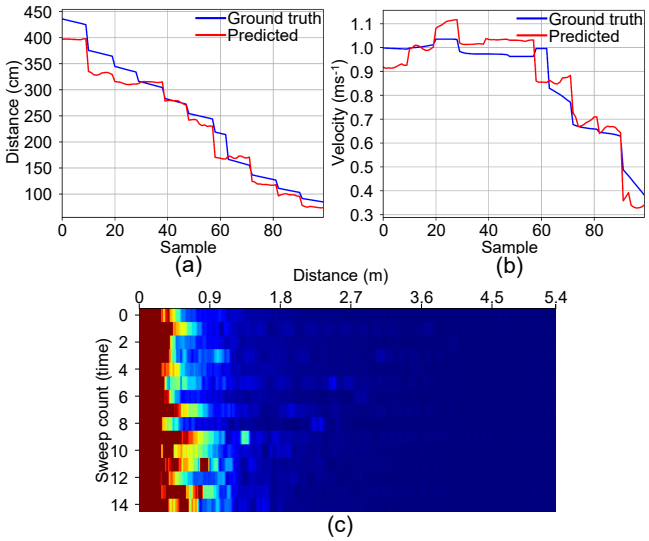

Fig. 10: Comparison of (a) predicted relative velocity and (b) predicted distance against the ground-truth when the participant walked towards a wall. (c) the reflections by the wall recorded on the IR envelopes

- ObsSense: This was our proposed method.
- w/o integration loss: This was a variant of the proposed method that did not employ the loss term $\mathcal{L}_{int}()$.
- w/o mobility detector: This method did not employ the intermediate-layer output of the mobility detector.
- Only IR: This was a variant of the proposed method that did not employ the Doppler features.
- Only Doppler: This method did not employ the IR features.
- DeepRange Doppler: This method was designed based on the state-of-the-art deep learning method [14]. The method was originally designed to recognize hand gestures and output the distance to the hand by processing the raw reflected sounds of the sine sweeps. However, because it exhibited poor performance in our preliminary experiment when using raw sounds, we used the Doppler features as inputs. The network architecture used in this experiment was identical to [14], except for the output layer, where our version had two output nodes to estimate the distance and relative velocity.
- DeepRange IR: This method was also based on [14]. This method uses IR features as the input.

The performance of the distance and relative velocity estimations was measured in terms of the mean absolute error (MAE) between the predictions and the ground truth.

C. Results

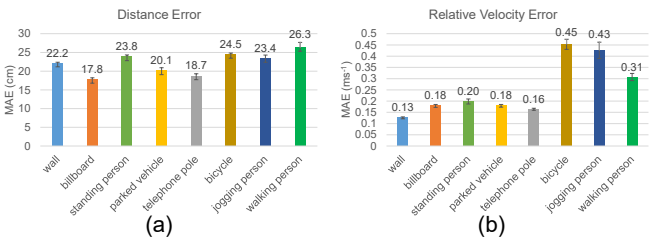

Fig. 11: (a) Distance and (b) relative velocity prediction errors of ObsSense for different obstacle types

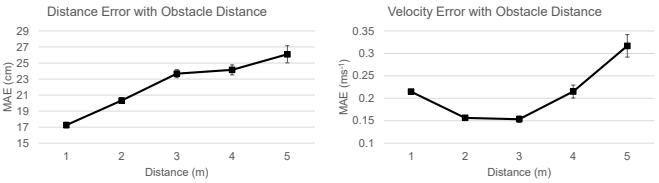

Fig. 12: Distance and relative velocity prediction errors according to different obstacle distances

1) Performance of ObsSense: Figure 9 shows the distance and velocity prediction performances of ObsSense and the other methods. The MAEs of the distance and velocity prediction by ObsSense were significantly smaller than those of the other methods, and ObsSense achieved MAEs of only 20.6 cm and 0.19 ms^{-1}. As shown in Figure 10 (a,b), ObsSense precisely predicted the distance and velocity, even though the participant randomly changed the walking speed. In addition, as shown in Figure 10 (c), the recorded IR envelopes are very noisy, and the reflections from the wall are not clearly visible, indicating the difficulties of applying threshold-based distance prediction to noisy real-world data. (The performance of threshold-based distance prediction is investigated below.)

Figure 11 shows the MAEs of ObsSense for the different obstacle types. The distance errors for persons are relatively larger than those for the other obstacles because the size of person is smaller than that of the other obstacles. Figures 12 and 13 show the MAEs of ObsSense for different ground truth distances and relative velocities. As can be seen, when the distance from the obstacle increased, the distance and relative velocity prediction errors increased. This is because when the distance to the obstacle increases, the amplitudes of the reflections from the obstacles become weak. Furthermore, when the relative velocity increased, the distance and relative velocity prediction errors increased. This result is natural because the intervals between the sweeps are relatively large compared with the speed. At a distance of 3 m, the relative velocity error was approximately 0.15 ms^{-1}. When the relative velocity of the obstacle was 2 ms^{-1}, the collision time estimation produced an average error of 0.1 s. This accounted for only 6.6% of the total collision time.

Moreover, Figure 9 shows the performance of **DeepRange Doppler** and **DeepRange IR**. While the neural networks of these methods were designed based on state-of-the-art methods, the MAEs of these methods were much poorer than those of ObsSense, indicating the effectiveness of our proposed probing signals and the design of our neural network.

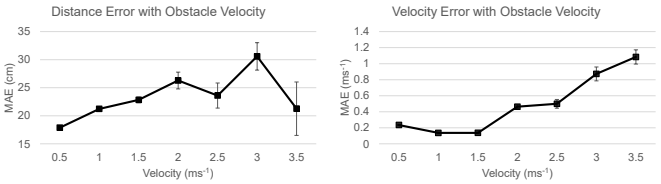

Fig. 13: Distance and relative velocity prediction errors according to different obstacle velocities

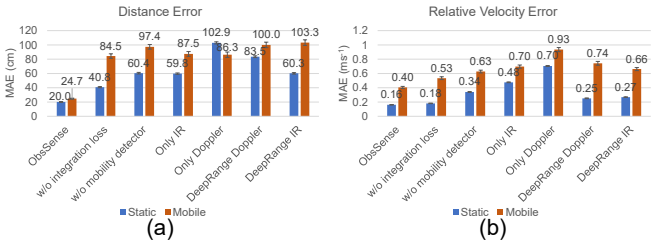

Fig. 14: Comparison of (a) relative velocity and (b) distance prediction errors of different methods according to obstacle mobility

2) Contributions of $\mathcal{L}_{int}()$ and mobility detector: Figure 9 shows the performance of **w/o integration loss** and **w/o mobility detector**. The distance error of **w/o integration loss** is much larger than that of ObsSense. Specifically, as shown in Figure 14, the distance estimation error in **w/o integration loss** for mobile obstacles is significantly higher than that of ObsSense, indicating the effectiveness of $\mathcal{L}_{int}()$ by enhancing the distance prediction with the velocity information.

As suggested by the performance of **w/o mobility detector**, the contribution of the mobility detector was significant. As shown in Figure 14, the mobility estimator improves the accuracy of the distance and relative velocity estimations in both static and mobile obstacles, indicating the effectiveness of knowledge regarding the mobility of the obstacle. Because the classification performance of the mobility detector is high (F-measure: 94%), ObsSense seems to adaptively predict the distance and velocity by referring to the intermediate outputs of the mobility detector, which precisely describes the mobility status of an obstacle.

3) Contributions of DF and IR features: Figure 9 shows the performance of **Only IR** and **Only Doppler**, indicating the significant contribution of IR features. Contrary to our expectations, the velocity error of **Only Doppler** was larger than that of **Only IR**. This can be because $\mathcal{L}_{int}()$ negatively affects velocity prediction. Because it is impossible to predict the distance by using only the DF features, erroneous distance prediction can cause a conflict regarding the distance-velocity relationship.

D. Discussion

1) Testing on other users: When we predict the relative velocity of a static obstacle, ObsSense relies on sound reflections from moving body parts of the user. To test the generalizability of our method across different users, we collected data on fifteen encounters from three additional participants. We fed the extracted features into a trained ObsSense neural network. Note that no data from the above participants were used to train ObsSense. The MAE values of the distance and relative velocity predictions were 34.1 cm and 0.29 ms^{-1}, respectively. The MAEs of the distance and velocity predictions of the data of the additional participants are somewhat poorer than that of the primary user. We believe that the reason for this is the body sizes of the additional users were different from the body size the primary user. However, ObsSense also achieved reasonable performance in this setting.

2) Multiple obstacle encounters: To observe the behavior of our method in the case of a multiple obstacle encounter, we used three billboards situated side-by-side to each other, the rightmost one being the closest to the user and the left being the furthest. The distance between the right and middle billboards was 160 cm, and the distance between the middle and left billboards was 160 cm towards the walking direction of the user. We collected data from five encounters where a participant walked towards the center of the billboards, i.e., the middle billboard, holding the smartphone, and the LiDAR module was directed towards the right billboard to acquire the ground truth of the closest obstacle. We tested the collected data using ObsSense. Although we assumed that ObsSense outputs the distance to the closest billboard, i.e., the rightmost billboard, ObsSense seems to output the distance to the middle billboard. Therefore, the MAE of the distance estimation was 191.4 cm. As the distance to the middle billboard was 160 cm greater than the distance to the ground truth, i.e., the distance to the right billboard, the MAE of the distance estimation of the middle billboard was as 31.4 cm. This could be because the smartphone was directed toward the middle billboard. ObsSense seems to output the distance to an obstacle situated in front of the user because sound reflections from such obstacles could be stronger than from other obstacles.

3) Obstacle's angle of arrival: To investigate the effect of mobile obstacle's angle of arrival on ObsSense's predictions, we collected data when a walking person arrived at a smartphone user at angles of 0°, 45°, and 90°to the user's walking direction. The MAE of the distance estimations of the above angles were 26.3 cm, 26.5 cm, and 29.1 cm, respectively, while the MAE of the relative velocity estimations were 0.31 ms^{-1}, 0.36 ms^{-1}, and 0.42 ms^{-1}, respectively. As can be seen, even though the MAE slightly increased with the angle of arrival, the change is not significant. Because the relative velocity of the mobile obstacles change with the angle of arrival, the Doppler features can be expected to change slightly. However, it appears that ObsSense can generalize between these features and effectively predict the distance and the relative velocity of the obstacles.

4) Performance under different noise levels: To test our method under different noise levels, we artificially injected noise from urban areas under different (10, 6, 3, and 1 dB) signal-to-noise ratios (SNRs) [33] into five sessions of test data. We used noise from the MAVD-traffic dataset [34] that contains noise from different types of vehicles recorded in urban environments. The MAEs of the distance predictions for different SNRs were 16.7 cm, 16.7 cm, 16.8 cm, and 16.8 cm, respectively. The MAEs of velocity predictions for all SNRs were stable at 0.12 ms^{-1}. When we tested the above five

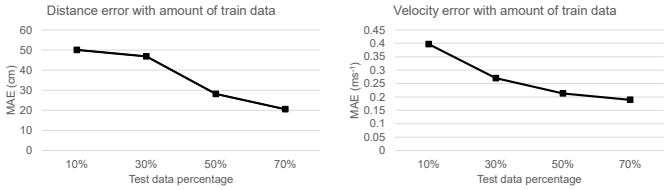

Fig. 15: Distance and relative velocity prediction errors of ObsSense according to different amounts of train data

sessions without adding any noise, the MAEs of distance and velocity predictions were 16.6 cm and 0.12 ms^{-1}, respectively. Similar to prior studies using inaudible sounds [12], [31], [33], the effect of the noises was limited.

5) Amount of training data: Figure 15 shows the effect of the amount of training data on ObsSense. As the amount of training data increased, the MAE of the distance and relative velocity predictions decreased. With 50% of the training data, ObsSense surpasses the performance of all other methods, achieving a distance error of 30 cm and a relative velocity error of 0.2 ms^{-1}.

6) Performance of threshold-based method: Here, we investigated a threshold-based distance estimation method using IR features. The threshold-based method employs peaks (crests) that are greater than the threshold created on the IR features to estimate the distance to the obstacle, as shown in Figure 6 (b). However, because of the semi-naturalistic data collection protocol that we followed, multiple crests created by multiple small obstacles in front of the user are unavoidable. Because it was not possible to distinguish the crest created by the targeted obstacle from that of other obstacles, we calculated the MAE between the ground truth and the largest crest created on the IR envelope. The MAE of the distance estimation of the threshold-based method was 154.8 cm, which is much higher than that of ObsSense. This is because, as Figure 10 (c) demonstrates, the data we have collected for this experiment while the user was walking is very noisy.

7) Comparison with other studies: A prior work related to threshold-based methods [11] achieved distance estimation error of 3 cm from a distance of 5 m. This error was smaller than that of ObsSense. However, unlike experiments of distance prediction in [11], the ObsSense experiment used noisy data collected while the user walked. As can be seen from the evaluation of the threshold-based method, the average distance estimation error was significantly larger.

Note that ObsSense requires training data with distance labels, which are not required in threshold-based methods. We collected the ground truth using the TFMini Plus LiDAR module, which uses a laser to measure distance. Hence, distance ground-truth errors due to the location of the obstacle at which the laser is pointed are inevitable. This is a limitation of our proposed method.

CycleGuard [18] employed a threshold-based method with an external speaker connected to a smartphone to estimate the distance between a cyclist and an approaching vehicle. This method achieves an average distance error of 10 cm. However, this study does not clearly describe the method in which the ground truth is acquired. If ObsSense is to be used to measure the distance between a user and a car, the obtained ground truth of the distance could change from more than 10 cm according to the location where the laser of the LiDAR hits the car, for example, the bumper or windscreen. However, ObsSense, which is based on supervised learning, yields reasonable performance even when we used noisy reflected signals such as Figure 10 (c) obtained by off-the-shelf smartphone speakers.

8) Feasibility of ObsSense: ObsSense employs the user's smartphone to record reflected sound waves at 192 kHz sampling rate and process the recorded sound using the neural network model. Most recent smartphones such as LG V10, ZTE Axon, Xiaomi Mi Note, Samsung Galaxy S20 and Galaxy Note20, and their operating systems support the 192 kHz audio recording. Furthermore, the size of the neural network of ObsSense is as small as 4.4 MB, which is sufficiently small when it is deployed on a neural processing unit in modern smartphones such as Snapdragon Neural Processing Engine. Hence, ObsSense can be conveniently deployed on most of current smartphones.

To ObsSense to be adaptable in real-world applications, it should be able to estimate the distance and the relative velocity of obstacles in real-time, while minimizing the battery consumption. Hanhirova et al. [35] showed that the time taken for a single image to pass through (inference) Inception V2 [36] and MobileNet [37] networks with Snapdragon Neural Processing Engine with GPU acceleration is 51 ms and 41 ms, respectively. As ObsSense is less complex compared to the MobileNet architecture (1.1 M parameters in ObsSense compared to above 2 M parameters in MobileNet), less inference latency can be expected when ObsSense is being deployed on smartphones. Investigating the effect of the inference latency on the user experience is a major part of our future work. Furthermore, the battery consumption of ObsSense can be minimized by only performing sensing and estimation when the user is walking while looking at the smartphone, which can be easily detected using the data from the accelerometer and the gyroscope of the smartphone.

VI. Conclusion

We presented ObsSense, a method for estimating the distance and relative velocity of oncoming obstacles using smartphone active sound sensing. We proposed a novel probing signal that facilitates the distance and relative velocity estimation of both static and mobile obstacles. Additionally, we proposed a novel neural network architecture that considers the mobility of obstacles to jointly predict the distance and relative velocity of obstacles. As part of our future work, we plan to extend our method to audio-based obstacle class prediction for precise risk-level assessment.

Acknowledgment

This work is partially supported by JSPS KAKENHI Grant Number JP21H03428, JP21H05299, and JP21J10059, and JST-Mirai Program Grant Number JP21473170, JAPAN.

REFERENCES

[1] J. L. Nasar and D. Troyer, "Pedestrian injuries due to mobile phone use in public places," *Accident Analysis & Prevention*, vol. 57, pp. 91–95, 2013.

[2] M.-I. B. Lin and Y.-P. Huang, "The impact of walking while using a smartphone on pedestrians' awareness of roadside events," *Accident Analysis & Prevention*, vol. 101, pp. 87–96, 2017.

[3] T. Wang, G. Cardone, A. Corradi, L. Torresani, and A. T. Campbell, "Walksafe: a pedestrian safety app for mobile phone users who walk and talk while crossing roads," in *the Twelfth Workshop on Mobile Computing Systems & Applications*, pp. 1–6, 2012.

[4] A. Caldini, M. Fanfani, and C. Colombo, "Smartphone-based obstacle detection for the visually impaired," in *International Conference on Image Analysis and Processing*, pp. 480–488, Springer, 2015.

[5] S. T. Payne, "Transparent texting," Mar. 27 2014. US Patent App. 13/627,959.

[6] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia, "A smartphone-based obstacle detection and classification system for assisting visually impaired people," in *the IEEE International Conference on Computer Vision Workshops*, pp. 444–451, 2013.

[7] E. Peng, P. Peursum, L. Li, and S. Venkatesh, "A smartphone-based obstacle sensor for the visually impaired," in *International Conference on Ubiquitous Intelligence and Computing*, pp. 590–604, Springer, 2010.

[8] H. Kang, G. Lee, and J. Han, "Obstacle detection and alert system for smartphone ar users," in *25th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–11, 2019.

[9] J. Wen, J. Cao, and X. Liu, "We help you watch your steps: Unobtrusive alertness system for pedestrian mobile phone users," in *the 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 105–113, IEEE, 2015.

[10] B. Thiel, K. Kloch, and P. Lukowicz, "Sound-based proximity detection with mobile phones," in *the Third International Workshop on Sensing Applications on Mobile Phones*, pp. 1–4, 2012.

[11] Z. Wang, S. Tan, L. Zhang, and J. Yang, "Obstaclewatch: Acoustic-based obstacle collision detection for pedestrian using smartphone," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–22, 2018.

[12] Y.-C. Tung and K. G. Shin, "Use of phone sensors to enhance distracted pedestrians' safety," *IEEE Transactions on Mobile Computing*, vol. 17, no. 6, pp. 1469–1482, 2017.

[13] Y. Wu, F. Li, Y. Xie, S. Yang, and Y. Wang, "Hdspeed: Hybrid detection of vehicle speed via acoustic sensing on smartphones," *IEEE Transactions on Mobile Computing*, 2020.

[14] W. Mao, W. Sun, M. Wang, and L. Qiu, "Deeprange: Acoustic ranging via deep learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–23, 2020.

[15] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: a high accuracy acoustic ranging system using cots mobile devices," in *the 5th International Conference on Embedded Networked Sensor Systems*, pp. 1–14, 2007.

[16] W. Mao, J. He, and L. Qiu, "Cat: high-precision acoustic motion tracking," in *the 22nd Annual International Conference on Mobile Computing and Networking*, pp. 69–81, 2016.

[17] R. Tang, G. Duan, L. Xie, Y. Bu, M. Zhao, Z. Lin, and Q. Lin, "Static obstacle detection based on acoustic signals," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–2, IEEE, 2022.

[18] W. Jin, S. Murali, Y. Cho, H. Zhu, T. Li, R. T. Panik, A. Rimu, S. Deb, K. Watkins, X. Yuan, *et al.*, "Cycleguard: A smartphone-based assistive tool for cyclist safety using acoustic ranging," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–30, 2021.

[19] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra, "Dopenc: Acoustic-based encounter profiling using smartphones," in *the 22nd Annual International Conference on Mobile Computing and Networking*, pp. 294–307, 2016.

[20] C. Liu, S. Jiang, S. Zhao, and Z. Guo, "Infrastructure-free indoor pedestrian tracking with smartphone acoustic-based enhancement," *Sensors*, vol. 19, no. 11, p. 2458, 2019.

[21] Y. Xie, F. Li, Y. Wu, S. Yang, and Y. Wang, "Hearsmoking: Smoking detection in driving environment via acoustic sensing on smartphones," *IEEE Transactions on Mobile Computing*, 2021.

[22] T. Dissanayake, T. Maekawa, T. Hara, T. Miyanishi, and M. Kawanabe, "Indolabel: Predicting indoor location class by discovering location-specific sensor data motifs," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5372–5385, 2021.

[23] M. Tachikawa, T. Maekawa, and Y. Matsushita, "Predicting location semantics combining active and passive sensing with environment-independent classifier," in *the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 220–231, 2016.

[24] T. Dissanayake, T. Maekawa, D. Amagata, and T. Hara, "Detecting door events using a smartphone via active sound sensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–26, 2018.

[25] J. Korpela, R. Miyaji, T. Maekawa, K. Nozaki, and H. Tamagawa, "Evaluating tooth brushing performance with smartphone sound data," in *the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 109–120, 2015.

[26] Y. Tange, S. Takeno, and J. Hori, "Development of the obstacle detection system combining orientation sensor of smartphone and distance sensor," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 6696–6699, IEEE, 2015.

[27] Y. Tange, T. Konishi, and H. Katayama, "Development of vertical obstacle detection system for visually impaired individuals," in *the 7th ACIS International Conference on Applied Computing and Information Technology*, pp. 1–6, 2019.

[28] M. Takagi, K. Fujimoto, Y. Kawahara, and T. Asami, "Detecting hybrid and electric vehicles using a smartphone," in *the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 267–275, 2014.

[29] S. Li, X. Fan, Y. Zhang, W. Trappe, J. Lindqvist, and R. E. Howard, "Auto++ detecting cars using embedded microphones in real-time," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–20, 2017.

[30] S. Kawanaka, Y. Kashimoto, A. Firouzian, Y. Arakawa, P. Pulli, and K. Yasumoto, "Approaching vehicle detection method with acoustic analysis using smartphone for elderly bicycle driver," in *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, pp. 1–6, IEEE, 2017.

[31] W. Jin, M. Xiao, H. Zhu, S. Deb, C. Kan, and M. Li, "Acoussist: An acoustic assisting tool for people with visual impairments to cross uncontrolled streets," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–30, 2020.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] T. Amesaka, H. Watanabe, M. Sugimoto, and B. Shizuki, "Gesture recognition method using acoustic sensing on usual garment," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–27, 2022.

[34] P. Zinemanas, P. Cancela, and M. Rocamora, "Mavd: A dataset for sound event detection in urban environments," *Detection and Classification of Acoustic Scenes and Events, DCASE 2019, New York, NY, USA, 25–26 oct, page 263–267*, 2019.

[35] J. Hanhirova, T. Kämäräinen, S. Seppälä, M. Siekkinen, V. Hirvisalo, and A. Ylä-Jääski, "Latency and throughput characterization of convolutional neural networks for mobile computer vision," in *the 9th ACM Multimedia Systems Conference*, pp. 204–215, 2018.

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *Computing Research Repository*, 2017.