IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488735

SmartDistance: A Mobile-based Positioning System for Automatically Monitoring Social Distance

Li Li[1], Xiaorui Wang[2], Wenli Zheng*[3], and Cheng-Zhong Xu[4]

[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]The Ohio State University
[3]Shanghai Jiao Tong University
[4] Faculty of Science and Technology, State Key Laboratory of IoTSC, University of Macau

Abstract—**Coronavirus disease 2019 (COVID-19) has resulted in an ongoing pandemic. Since COVID-19 spreads mainly via close contact among people, social distancing has become an effective manner to slow down the spread. However, completely forbidding close contact can also lead to unacceptable damage to the society. Thus, a system that can effectively monitor people's social distance and generate corresponding alerts when a high infection probability is detected is in urgent need.**

In this paper, we propose SmartDistance, a smartphone based software framework that monitors people's interaction in an effective manner, and generates a reminder whenever the infection probability is high. Specifically, SmartDistance dynamically senses both the relative distance and orientation during social interaction with a well-designed relative positioning system. In addition, it recognizes different events (e.g., speaking, coughing) and determines the infection space through a droplet transmission model. With event recognition and relative positioning, SmartDistance effectively detects risky social interaction, generates an alert immediately, and records the relevant data for close contact reporting. We prototype SmartDistance on different Android smartphones, and the evaluation shows it reduces the false positive rate from 33% to 1% and the false negative rate from 5% to 3% in infection risk detection.

I. Introduction

COVID-2019 is an infectious disease caused by severe acute respiratory syndrome coronavirus 2. It was first identified in December 2019 and has since spread globally, resulting in an ongoing pandemic. As of August 2020, more than 16.6 million cases have been reported across 188 countries and territories, resulting in more than 658,000 deaths [1]. The virus is believed to be primarily spread between people during close contact, often via small droplets produced by coughing, sneezing and talking.

Social Distancing and Close Contact Reporting. Social distancing [2] and close contact reporting have become two important ways to control and slow down the spread of Covid-19, but they are difficult to conduct. Social distancing means that keeping a safe space between ourselves and other people who are not from our household. However, unintentional distancing violation still commonly occurs, as it is inconvenient to measure the distance between people in social interaction. It is also uneasy to determine how much space is large and safe enough in various environments, while a static criteria (e.g., 6 feet) might be sometimes too conservative and prohibit some necessary social interaction. Moreover, when an infection case is reported, it is critical to identify the group of individuals who have been in close contact with the diagnosed patient, in order to understand and mitigate the spread of the pandemic. Previous attempts analyze the close contact networks related to disease transmission mainly through online questionnaire, surveys or recall by the diagnosed patients. However, such kind of manual approaches usually require a significant amount of effort, and can easily miss some individuals who have high probability to get infected. Thus an automatic solution for an individual to take care of her/his social interaction is in urgent need in order to 1) immediately generate an alert when the social distance is too small to avoid infection, and 2) report the individuals who are likely to be infected when an infection case is found.

Existing Solutions and Challenges. Despite the promising benefits, two main obstacles exist which prevent such an infection risk detection system to be viable. First, it is noticed that not only the relative distance but also the relative orientation is critical to determine the infection probability. For instance, a person is much more likely to get infected when he is talking to a diagnosed patient face to face than back to back [3], keeping the same distance. The problem is that the relative orientation cannot be directly sensed by an ordinary mobile device like a smartphone. Previous work has proposed different approaches to monitor the position information, which are unfortunately insufficient to solve the social distancing problem. First, the widely used positioning technologies like GPS may not work well in indoor environment, and its high power consumption severely hurts the battery lifetime of mobile devices [4]. Moreover, although many studies have attempted to estimate the locations of people using radio signals [5]–[7], these solutions usually require extra hardware equipment and are hardly practical in outdoor environment [8], [9]. Thus, existing solutions cannot be easily applied to sense the social distance. Second, while it is already known that the droplet transmission of Covid-19 usually follow a certain pattern and impacts the surroundings in a certain range, different actions (e.g., speaking, coughing, sneezing) during the social interaction usually form totally different droplet transmission patterns that result in different infection spaces. How to dynamically recognize those actions and determine the corresponding infection space is another non-trivial challenge.

Our Contribution. In this paper, we propose SmartDistance, an intelligent mobile-based positioning system that detects the infection risk in social interaction. SmartDistance mainly consists of two parts: 1) Infection Space Determination and 2) Relative Positioning. Specifically, the first part decides the infection space under different action events (e.g., speaking, coughing and sneezing), within which people are highly probable to get infected due to droplet transmission. We use a CNN-based classifier to recognize different actions with the acoustic signals in real time, and a droplet transmission model to characterize the tendency of droplet movement given a specific action. In the relative positioning part, we design a location mapping algorithm that analyzes both the relative orientation and relative distance of two individuals in dynamic movement scenarios. Thus, when an individual is detected to be in the infection space, an alert will be generated immediately, and the corresponding individual will be automatically recorded for later contact networks analysis.

We compare SmartDistance with a state-of-the-art relative positioning system and show that SmartDistance effectively reduces the relative orientation and distance errors. We also compare SmartDistace with a state-of-the-practice infection risk detection approach, and find that the false positive and false negative rates are both reduced (from 33% to 1% and from 5% to 3%, respectively). To our best knowledge, SmartDistance is the *first* work that studies the real-time infection risk detection problem with mobile-based relative positioning. Specifically, our major contributions are as follows:

- We propose SmartDistance, which integrates mobile-based relative positioning with acoustic signal-based infection space determination, for real-time infection risk detection.
- We design a relative positioning system on mobile devices that detects not only the relative distance but also the relative orientation between two individuals, in a timely and energy efficient manner.
- We design an infection space determination system based on a droplet transmission model and an action recognition component, and solve the problem about how to determine the infection space with different relevant actions.
- We evaluate SmartDistance with both testbed experiments and simulation. It reduces the false positive rate from 33% to 1% and the false negative rate from 5% to 3% in infection risk detection.

The rest of the paper is organized as follows. Section II presents the motivation and system architecture of SmartDistance. Section III presents the relative positioning system. Section IV discusses the infection space determination process in different scenarios. Section V presents the evaluation of SmartDistance. Then Section VI discusses prior research that is closely related with SmartDistance. Finally, Section VII concludes the paper.

II. Motivation and System Architecture

In this section, we first present the motivation of this work, and then introduce the motivated design of SmartDistance system architecture.

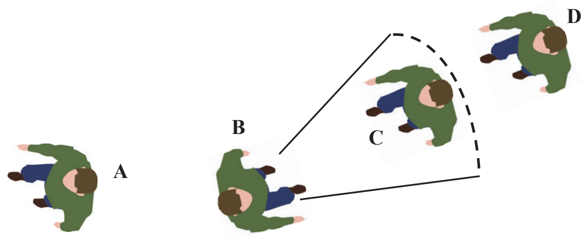


Fig. 1: Different close contacts in social interaction.

A. Motivation

Covid-19 spreads mainly among the people who are in close contact for a period of time. When an infected individual talks, coughs or sneezes, the droplets are sprayed from the mouth or nose into the air. Those droplets can then be inhaled into the mouths or noses of others nearby. The virus is transmitted in this process. Figure 1 shows different close contacts in social interaction. For the individuals B and C who are talking to each other face to face, if B carries the virus, C is highly probable to get infected. This is because C is within the droplet transmission space of B. In this case, D has less probability to get infected, since D maintains a longer distance from B (i.e., out of the droplet transmission space of B). It is important to note that A is also unlikely to get infected, though he/she is close to B. The droplets exhaled by B are hard to reach A in this case due to the relative orientation between A and B. Thus, the following two critical challenges need to be addressed in order to make SmartDistance viable: *1) How to determine the infection space due to droplet transmission in different scenarios? 2) How to determine the relative position (both relative orientation and distance) between two individuals in real time?*

B. System Architecture

Figure 2 shows the system architecture of SmartDistance. The *Infection Space Determination* part contains a well-designed droplet transmission model and an action recognition component to effectively decide the infection space with different actions in real time. The *Relative Positioning* part consists of a motion info retrieval component and a location mapping algorithm to analyze the relative position information. In order to be efficiently deployed on mobile devices, SmartDistance also needs to meet the following two design requirements: 1) it should generate the alert in time in order to effectively prevent the infection, and 2) it should be energy efficient for battery-powered mobile devices. In order to balance the energy efficiency and realtimeness, we design SmartDistance as follows:

- During social interaction, a mobile device first makes connection with the nearby devices. In this work, we select Bluetooth to provide the communication service due to the following reasons: 1) Bluetooth is widely equipped on smartphones, 2) it consumes low energy and 3) it fits the social interaction scenario which is conducted within a short geographical distance (the maximum communication distance of Bluetooth is 30m). In the current

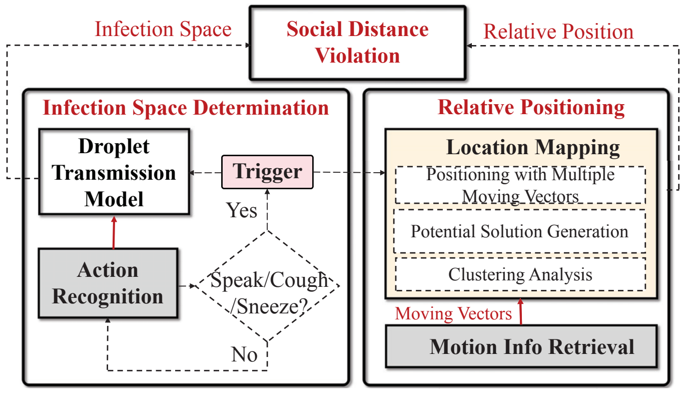


Fig. 2: System Architecture of SmartDistance.

implementation, SmartDistance requires that the users keep their Bluetooth on during the overall social interaction process. This assumption has been widely made by the previous research which focuses on the cooperation and monitoring of mobile devices in proximity [10], [11].

- After the connection is established, the *Motion Info Retrieval* component starts sensing the movement information of the nearby devices. The *Action Recognition* component also starts to constantly recognize the predefined actions (e.g., speaking, coughing and sneezing).
- Once a predefined action is recognized, the *Droplet Transmission Model* is used to estimate the infection space of the corresponding action. At the same time, the *Location Mapping* algorithm is triggered to analyze the relative position.
- If a certain individual is detected to be within the infection space, the alert is generated and the ID of that corresponding individual is automatically recorded.

It is important to note that, in Figure 2, the components in grey background (Action Recognition and Motion Info Retrieval) run constantly during the social interaction process, while the other components are triggered by the predefined events (e.g., speaking/coughing/sneezing). This event-based design helps with both the timeliness and energy efficiency of the system.

III. Relative Positioning

In this section, we propose the method to estimate the relative orientation at first, and then present the system design of the Relative Positioning part, with discussing the detailed problems and solutions at last.

A. Estimation of Relative Position

Following the discussion in Section II, the relative positioning system has one important design requirement: it has to provide not only the distance, but also the orientation relative to each other in different social interaction scenarios (e.g., stationary or moving at different speeds). Figure 3 shows an example that Alice (A) identifies the relative distance d_{AB} and the relative orientation α of Bob (B). α is defined as the angle between the Magnetic north and the vector AB anticlockwise. In the case shown in Figure 3, α is 90°. Here, we use the magnetic north to measure the orientation because it is independent of the various factors local to each member (e.g., smartphone facing up or down, which direction an individual is facing) that may unnecessarily complicate our problem.

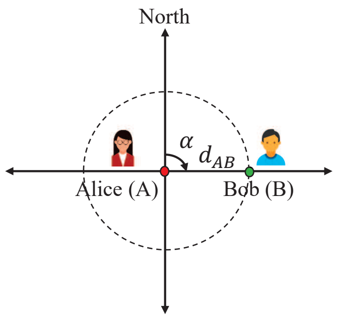


Fig. 3: Illustration of Relative Positioning, with two members in a group. From A' point of view, the relative distance of B is d_{AB}, and the relative orientation of B is α.

While the estimation of relative distance has been widely studied before, there is no appropriate solution to estimate the relative orientation yet, for people in social interaction. We tackle this problem based on the following **Key Observation**: *The distance between two individuals and their moving vectors (e.g., moving distance and direction measured using the local smartphone's sensors) can help them estimate the orientation relative to each other.*

Figure 4a illustrates the above observation. Initially, A and B locate at $A_{(0)}$ and $B_{(0)}$ respectively. Then, A moves to $A_{(1)}$ along the moving vector $V_{A(1)}$ and B moves to $B_{(1)}$ along the moving vector $V_{B(1)}$. The coordinates of the points $A_{(0)}$, $B_{(0)}$, $A_{(1)}$ and $B_{(1)}$ are as follows: $(x_{A(0)}, y_{A(0)})$, $(x_{B(0)}, y_{B(0)})$, $(x_{A(1)}, y_{A(1)})$, and $(x_{B(1)}, y_{B(1)})$, respectively. Similarly, the moving vectors $V_{A(1)}$ and $V_{B(1)}$ can be represented as $(u_{A(1)}, v_{A(1)})$ and $(u_{B(1)}, v_{B(1)})$, where u and v represents the moving vector components along the x-axis and y-axis, respectively. In order to calculate the orientation α, we need to know the coordinates of $B_{(1)}$.

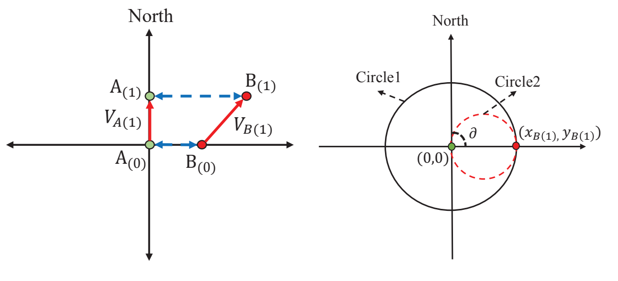
Therefore, the problem becomes to find the coordinates of $B_{(1)}$ using the known moving vectors $V_{A(1)}$, $V_{B(1)}$ and the known distances $d_{A(0)B(0)}$, $d_{A(1)B(1)}$, which are measured in the Motion Info Retrieval component as introduced in Section III-B. We note that the coordinates of $A_{(1)}$ can be represented as a function of the coordinates of $A_{(0)}$ and the components of the moving vector $V_{A(1)}$, as follows:

$$x_{A(0)} + u_{A(1)} = x_{A(1)}, \ y_{A(0)} + v_{A(1)} = y_{A(1)} \qquad (1)$$

To locate B from the point of view of A, we set the origin of the observation system always at the current location of A. It means that after A arrives at A_1, the coordinates of $A_{(1)}$ become (0,0). As a result, we can rewrite Equation 1 as $x_{A(0)} = -u_{A(1)}$ and $y_{A(0)} = -v_{A(1)}$. Thus, the coordinates of $A_{(0)}$ and $A_{(1)}$ are known. On the other hand, using a similar system of equations for B, the coordinates of $B_{(0)}$ and $B_{(1)}$ are unknown. To derive the coordinates of $B_{(0)}$ and $B_{(1)}$, we can use the distance relationship as follows:

$$(x_{A(0)} - x_{B(0)})^2 + (y_{A(0)} - y_{B(0)})^2 = d^2_{A(0)B(0)} \qquad (2)$$

$$(x_{A(1)} - x_{B(1)})^2 + (y_{A(1)} - y_{B(1)})^2 = d^2_{A(1)B(1)} \qquad (3)$$



(a) Moving Scenario Example (b) Relative Position Solution

Fig. 4: Example of Relative Positioning: 1) distance before moving ($d_{A(0)B(0)} = 1m$), 2) moving vectors ($V_{A(1)} = (0,1)$ and $V_{B(1)} = (1,1)$), and 3) distance after moving ($d_{A(1)B(1)} = 2m$). Circle1: center=$(0,0)$, $radius = 2m$, Circle 2: center=$(1,0)$, $radius = 1m$. Intersection point $(2,0)$ is the solution of relative position $(x_{B(1)}, y_{B(1)})$.

By substituting Equation 1, which represents the moving procedure of A, and the similar equations of B, into Equations 2 and 3, we have a system of two equations with only two unknown variables, $x_{B(1)}$ and $y_{B(1)}$. Thus, the position of B relative to A can be obtained by jointly solving Equations 2 and 3. Then the relative orientation α can be easily obtained by leveraging the inverse tangent function.

B. System Design of Relative Positioning

As mentioned above, the relative positioning system needs the distance and the local moving vectors to calculate the relative orientation. Figure 5 shows the architecture of the relative positioning system in SmartDistance, including two main components: 1) Motion Info Retrieval, and 2) Location Mapping. Motion info Retrieval tracks the moving vectors and the distances between two individuals. This information is then input to Location Mapping, which determines the relative distance and orientation between two individuals.

Motion Info Retrieval first retrieves the distance between individuals A and B at time point T_0. The initial distance between the two smartphones d_0 is retrieved using the received signal strength indication (RSSI) of Bluetooth with a mean error of 0.2 meters, according to [12]. After that, the two individuals track their own moving vectors locally. At time point T_1, the two individuals exchange their moving vectors $(u_{A(1)}, v_{A(1)})$ and $(u_{B(1)}, v_{B(1)})$ measured between time points T_0 and T_1. At the same time, they measure the distance (d_1) between each other through the RSSI during the message exchange. In this work, a step detector is implemented to detect each step and the corresponding step length [12]. A heading detector [13] is implemented to get the orientation information of each step with data collected from the gyroscope, accelerometer and magnetometer.

Location Mapping leverages the obtained sensing information to get the relative position between two individuals. Specifically, with the information (1) the distance between individuals A and B before the moving process (e.g., $d_{A(0)B(0)}$), (2) their moving vectors (e.g., $V_{A(1)}$) and $V_{B(1)}$), and (3) the distance after the moving process (e.g., $d_{A(1)B(1)}$), Location Mapping calculates the position of B relative to A through

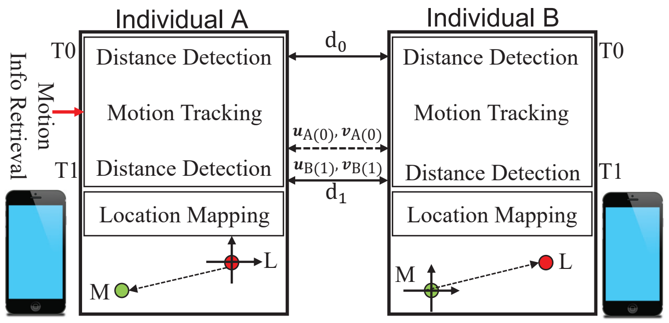


Fig. 5: Architecture of Relative Positioning System.

solving Equations 2 and 3, based on the method in Section III-A.

C. Problems and Solutions with Location Mapping

The real-world cases are much more complex than the example in Figure 4, requiring to consider the following two problems for Local Mapping to work in practice:

- Problem 1: The system of Equations 2 and 3 is a system with 2 equations and two variables (e.g., $x_{B(1)}, y_{B(1)}$), and it is solvable. However, in general, this type of system may have no solution, one solution or two candidate solutions. As a result, in some cases, we may need to estimate the relative position if there is no solution, or choose between two candidate solutions.
- Problem 2: The noises and errors introduced during the movement tracking and distance estimation can impact the detection accuracy. For instance, the noises and errors can impact the centers (determined by the moving vectors) and radiuses (determined by the distances) of the two circles as shown in Figure 4b. As a result, the intersection points may deviate from the real relative position, leading to inaccurate position estimation.

Thus, we design the following three steps to solve the above problems: 1) positioning with multiple vectors, 2) potential solution generation, and 3) clustering analysis.

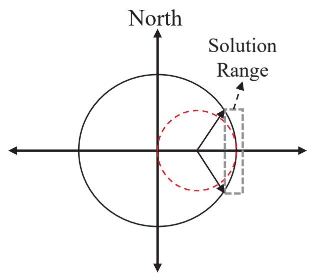
Positioning with Multiple Moving Vectors. In order to solve Problem 1, we propose to calculate the relative position with multiple moving vectors obtained by using the history of moving vectors during a period of time (e.g., $V_{A(1)}...V_{A(n)}$, $V_{B(1)}...V_{B(n)}$) and the calculated relative position for each sample. We can get the current relative position $(x_{B(i)}, y_{B(i)})$ from the previous step $(x_{B(i-1)}, y_{B(i-1)})$ combining with the moving vectors $(u_{A(i)}, v_{A(i)})$ and $(u_{B(i)}, v_{B(i)})$ as follows:

$$x_{B(i)} = x_{B(i-1)} + u_{B(i)} - u_{A(i)} \tag{4}$$

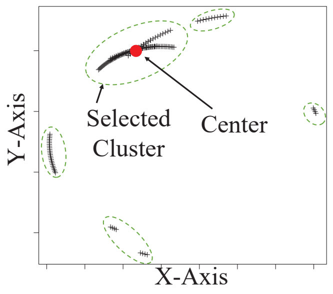
$$y_{B(i)} = x_{B(i-1)} + v_{B(i)} - v_{A(i)} \tag{5}$$

The more previous moving vectors we consider (e.g., $u_{B(i-n)}$ to $u_{B(i-1)}$), the more solution points we can get for the current relative positions. This is because each pair of moving vectors can contribute to a solution of the current relative position through Equations 4 and 5.

In summary, we get the current relative position from two approaches: *Approach 1:* solving Equations 2 and 3 with the



(a) Potential Solution range



(b) Clustering Analysis

Fig. 6: Examples of potential solution generation (a) and clustering analysis (b). In (b), the cluster that contains the most data points is selected and its medoids is reported as the relative position.

latest moving vectors and distances, and *Approach 2:* estimate the current relative position based on the previous relative position combined with the moving vectors with Equations 4 and 5. Then, we save the data obtained through both the two approaches for further analysis.

Specifically, Problem 1 can be solved as follows:

- When there is no solution for Approach 1 (e.g., no intersection points on the two circles in Figure 4b), we can get the current relative position with Approach 2 from the previous solution set.
- In the case that there are two solutions (e.g., the two circles in Figure 4b have 2 intersection points), only one of them is the real relative position. We collect multiple data points with Approach 2 (e.g., considering multiple previous moving vectors), and the majority of data points are supposed to be close to the real relative position, which has been confirmed with our evaluation results.

Potential Solution Generation. In order to solve Problem 2, instead of only considering the intersection points, we select the points located within a certain range of the intersection points and consider them as potential solutions. As a result, the solution set has a higher probability of covering the real relative position. Figure 6a shows an example of the potential solution range. The data points in the box are selected out. Moreover, the range size is configurable, based on the average error and noise introduced by smartphone sensors.

Clustering Analysis. Both the above two steps (i.e., positioning with multiple moving vectors and potential solution generation) generate multiple data points for the current relative position, which can contain the real relative position and the ones caused by other issues (e.g., the two-solution case, sensing errors). Thus, we need to select out those that are close to the real relative location. To solve this problem, we use clustering analysis.

Clustering analysis is used to group the saved potential solution points into different clusters. Figure 6b shows an example. Specifically, SmartDistance adopts K-Medoids Partitioning Around Medoids (PAM) for clustering analysis, which has more tolerance to noise and outliers compared to the K-means algorithm. Moreover, Silhouette analysis is adopted to determine the right number of clusters. Then, the cluster that contains the most data points is selected out. Finally, the medoid (center) of the cluster is reported as the relative position.

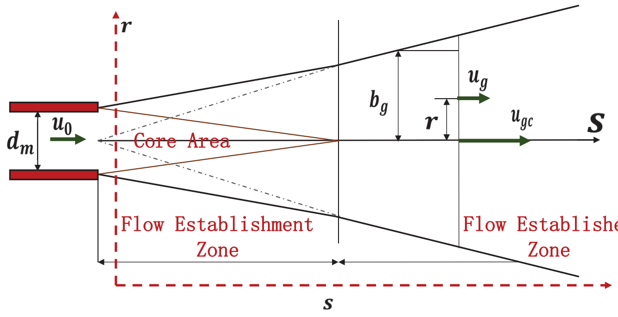


Fig. 7: Illustration of the Jet Model.

IV. Infection Space Determination

In this section, we discuss how SmartDistance determines the infection space. Specifically, we first present the modeling of droplet transmission, and then explain the action differentiation methodology.

A. Droplet Transmission Model

Close contact can create the condition for droplet transmission. When people are talking with each other, droplets are sprayed from mouths. The droplets containing pathogens can be the primary medium for infection transmission, and some droplet nuclei and fine droplets can be directly inhaled by the people who are having a face-to-face interaction. Thus, effectively modeling the droplet transmission route and the corresponding transmission range are critical to determine whether a certain individual is highly probable to get infected during a social interaction event.

According to [3], the exhaled air flow from an infected person can be treated as a turbulent round jet which can be mainly divided into two zones: 1) a flow establishment zone and 2) a flow established zone. Figure 7 shows the illustration of the jet model. The flow rate and the velocity profiles can be obtained by classic jet theories [14]. The corresponding notations are as follows:

- s represents the center line distance travelled by the jet.
- d_m represents the source mouth diameter.
- u_g is the Gaussian velocity.
- u_0 represents the initial velocity at the source mouth outlet.
- r is the radial distance away from the jet center line.
- R represents the radius of the jet's potential core.
- b represents the Gaussian half width.
- Q_{jet} is the jet flow rate.
- u_{gc} represents the Gaussian center line velocity.

In the **flow establishment zone** ($s \leq 6.2d_m$, Gaussian profile):

$$u_g = u_0; \quad r \leq R \tag{6}$$

$$u_g = u_0 exp\left[-\frac{(r-R)^2}{{b_g}^2}\right]; \quad r \geq R \tag{7}$$

$$Q_{jet} = \pi {b_g}^2 u_0 \tag{8}$$

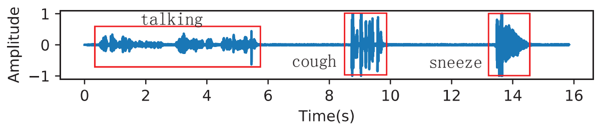


Fig. 8: Amplitude of acoustic signal of different events.

$$b_g = 0.5d_m + 0.033355s \tag{9}$$

In the **established flow zone** ($s > 6.2d_m$, Gaussian profile):

$$u_{gc} = 6.2u_0(d_m/s) \tag{10}$$

$$Q_{jet} = \pi {b_g}^2 u_{gc} \tag{11}$$

$$b_g = 0.114s \tag{12}$$

From the equations above, we can calculate the radius of the jet b_g through the length of s as follows:

$$b_g = \begin{cases} 0.5d_m + 0.033355s & s \leq 6.2d_m \\ 0.114s & s > 6.2d_m \end{cases} \tag{13}$$

We can see from Equation 13 that in the flow establishment zone, the radius of the jet b_g is determined by d_m and the center line distance travelled by the jet. In the established flow zone, the radius of the jet b_g is only decided by the corresponding center line distance travelled by the jet. Thus, when the source mouth diameter d_m is treated as fixed, each center line distance maps to a corresponding b_g. Hence in order to determine the infection space, we need to decide the range of the center line distance and the corresponding radius of the jet b_g. Equation 10 shows the relationship among the Gaussian center line velocity u_{gc}, the initial velocity at the source mouth outlet u_0 and the center line distance travelled by the jet. We assume that the droplet does not transmit any more when the Gaussian center line velocity is below a certain threshold $u_{threshold}$. Since d_m is usually a fixed value for a particular individual, the maximum center line distance is determined by the initial velocity at the source mouth outlet u_0.

However, u_0 can be totally different with different actions. For instance, the average velocity at the mouth for speaking is 3.9 m/s while it is 11.7 m/s for coughing [3]. A higher initial velocity leads to longer center line distance and larger infection space. Thus, recognizing different actions in real-time is critical for SmartDistance to estimate the infection space.

B. Action Recognition

The action recognition component differentiate actions (e.g., speaking, sneezing and coughing) in real time based on acoustic signals. Specifically, we follow the approach in [15] to design the action recognition component, which mainly consists of an audio sampler and an action detector.

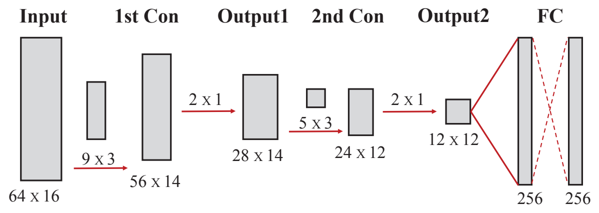


Fig. 9: Architecture of the CNN-based Classifier.

Audio Sampler reads acoustic samples from the microphone for further analysis. Figure 8 shows corresponding examples of relevant acoustic signals. Specifically, audio signals are continuously sampled, and each sample is converted into a 16-bit binary value. After that, the stream of acoustic data is then segmented into frames. In the current implementation, the length of each frame is 4ms. Finally, multiple consecutive frames are grouped together as a window. Each window is then send as input to the action detector for detection analysis.

Action Detector mainly contains the following two steps: 1) preprocessing the acoustic signals and 2) training a classifier to effectively detect different events.

- **Step 1:** The preprocessing step conducts the frame admission process. Specifically, for each window (64ms), the RMS energy is first retrieved and compared with a predefined threshold. The windows whose energy is below the threshold are treated as ambient noise. These windows are then discarded. For the windows whose energy is above the threshold, a 128-bin ShortTime Fourier Transform (SIFT) is applied to generate a 64∗16 spectral segment. These segments are then fed into the following classifier for event differentiation.
- **Step 2:** In this step, we train a CNN-based classifier in order to effectively differentiate various events, which has been widely used in action recognition, speech recognition and natural language processing. Figure 9 represents the concrete architecture of the classifier [16]. It consists of five layers including two convolutional layers, two fully connected layers and a softmax classification layer. The last layer takes the outputs of the second fully connected layer and differentiates the corresponding inputs as speaking, coughing or sneezing.

V. Evaluation

In this section, we first introduce the experimental methodology and baselines, and then discuss the results.

A. Experimental Setup

We build a prototype of SmartDistance as the testbed, using Android smartphones with different hardware configurations including Nexus 4, Nexus 5, Galaxy S2, Galaxy S4, Galaxy mini and LG v10. A Monsoon Power Monitor [17] is used to measure the energy overhead of SmartDistance on different mobile devices. In addition to the testbed experiments, we conduct simulations with different traces of human mobility and action events, to evaluate SmartDistance with more diverse scenarios. We generate the mobility traces based on a human mobility model [18] which contains different statistical patterns of human mobility including fractal way-points, pause-times, inter-contact times and truncated power-law distributions. The mobility model [18] effectively characterizes the social interaction scenarios in a specific community such as parks, university campus and companies. Moreover, the action event traces are created through randomly generating the predefined events (e.g., speaking/coughing) at different time points. The sensing errors collected on the testbed smartphones are adopted in the simulations.

We adopt the following solutions as the baselines, for the comparison with SmartDistance.

- **Dis-Only** determines whether an individual is highly probable to get infected only based on the relative distance, without considering the orientation and relevant action events. Specifically, if the distance between two individuals is less than 6 feet [19], the alert will be generated.
- **AMIL** is a state-of-the-art relative positioning technique [20] that uses the acoustic information to detect the relative positions of nearby devices. AMIL assumes that the nearby devices are stationary during the detection process and does not track their motion information, because it treats the nearby devices as the anchors and do triangulation for positioning. SmartDistance differs from it for tracking the motion information (moving distance and direction) of each individual locally in the social interaction process, which enables a new positioning algorithm with higher accuracy, since the mobile devices are not likely to be always stationary during social interaction.
- **One-Step** finds the optimal solution meeting the constraints represented by Equations 2 and 3, constructed by the latest moving vectors and distance information. In comparison, SmartDistance not only uses multiple moving vectors, but also applies potential solution generation, and hence performs more comprehensive analysis.

B. Performance of SmartDistance

In this section, we evaluate SmartDistance through comparing with Dis-Only. Ten sample cases with different relative distances, relative orientations, face orientations and actions events are presented in Figure 10. We test SmartDistance and Dis-Only with the ten sample cases, and compare their decisions on whether to make alerts or not to the Ground Truth, as shown in Figure 11. The Ground Truth is the theoretically optimal decision given the exact position information, action events, and the infection space determined by the droplet transmission model. Since the Dis-Only scheme uses a fixed distance (6 feet or 1.83 meters here) as a threshold, it generates an alert if and only if the distance is within the threshold. However, the orientation information can actually make a difference. If two individuals are back to back, the possibility for any of them to be infected via droplet transmission by each other is low, even with a relatively short distance. We can see a similar instance in Sample Case 1: although the relative distance is short (0.97m), the alert should not be generated as the relative orientation is $63.9°$ and the face orientation is $115°$. On the other hand, some actions such as coughing usually have a larger infection range. SmartDistance addresses those issues through jointly considering the relative position and action information, and therefore effectively minimizes the false positive rate (1%) and false negative rate (3%), as shown in Figure 12, which presents the statistics of 100 samples. In comparison, the Dis-Only scheme has higher false positive rate (33%) and false negative rate (5%), so the alert can be mistakenly generated and an individual is mistakenly marked with high possibility to get infected, or an individual can become highly possible to get infected without being alerted or recorded.

C. Performance of Relative Positioning

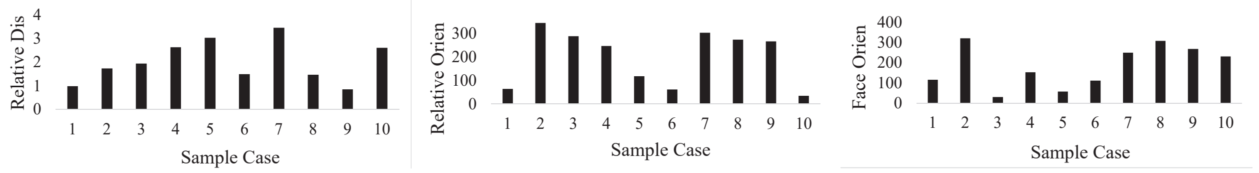
We evaluate the performance of relative positioning from two perspectives: 1) the orientation error which is defined as the angle between the real orientation and the estimated orientation, and 2) the distance error which is defined as the distance between the real position and the estimated position.

We first compare SmartDistance with the baseline One-Step. The two schemes are evaluated with 20 moving test cases, in which the testbed smartphones are taken by the testers walking around in a library. We can see from Figure 13a that with SmartDistance, the orientation errors are less than $22°$ in all the cases, and 80% of them are less than $15°$. However, Figure 13b shows the orientation error achieves up to $200°$ with the One-Step scheme. This is caused by the following two reasons: 1) with the potential solution generation step, SmartDistance leverages more data points and thus is more robust to the noise and errors introduced in moving vector tracking and distance estimation, 2) the clustering analysis step reports the right solution through selecting the cluster that contains the most potential data points. Figures 13c and 13d present similar comparison results about distance errors.

Then we compare SmartDistance with AMIL [20]. From Figure 14, we can find that when the moving distance is constrained within 0.5m, SmartDistance and AMIL have similar errors on average. For AMIL, the distance and orientation errors increase as the the group members move over longer distances during the detection process, and the worse-case error can be much larger than the average error. The moving distance does not significantly impact SmartDistance, because SmartDistance keeps tracking the motion information of each device. Moreover, the motion information is exchanged during the detection, which means a device integrates both the motion information of itself and those of the nearby devices. Thus, the motion-based relative positioning technique of SmartDistance is more effective when people have different motion behaviors at any time, which is common in daily social interaction.

D. Impact of The Potential Solution Range

As discussed in Section III, in order to reduce the impact of noise and errors caused by the moving vectors and distance



(a) Relative Distance. (b) Relative Orientation. (c) Face Orientation.

Fig. 10: Relative distance, relative orientation and face orientation in different sampling cases. The corresponding events are S, S, C, C, S, S, S, C, C, C (S stands for speaking and C stands for coughing.)

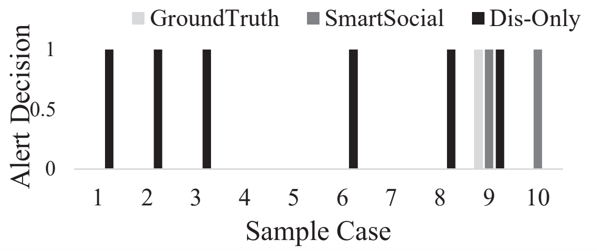
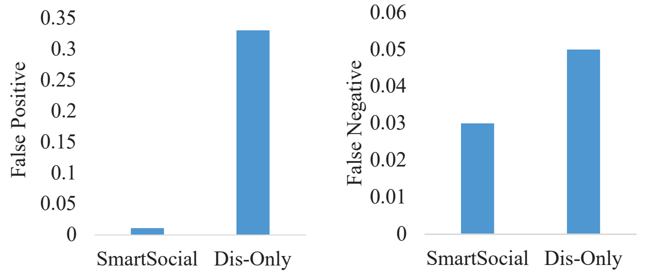
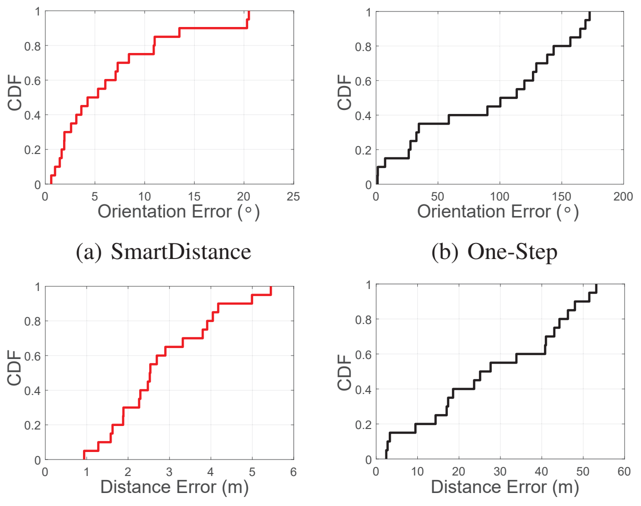


Fig. 11: Comparison on generating alerts.



(a) False Positive. (b) False Negative.

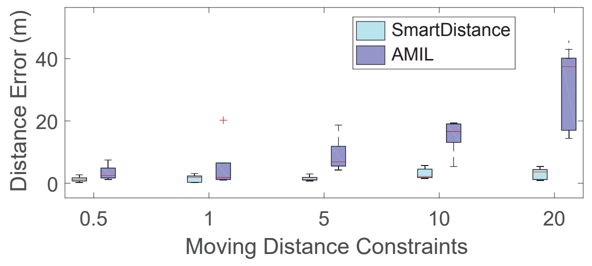
Fig. 12: Statistics of 100 samples of alert decisions.



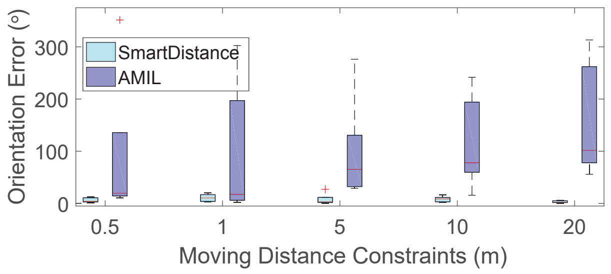
(a) SmartDistance (b) One-Step

(c) SmartDistance (d) One-Step

Fig. 13: Comparison between SmartDistance and One-Step on the orientation error (a and b) and distance error (c and d).



(a) Distance Error



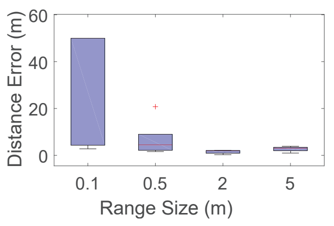
(b) Orientation Error

Fig. 14: Performance Comparison between SmartDistance and AMIL [20]. The moving distance constraint is the longest distance that a member can move during a detection period of relative positioning.

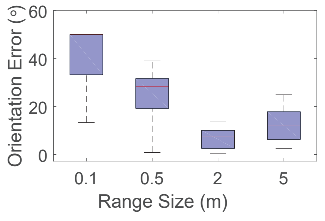
estimation, we select the data points within a certain range of the intersection points (e.g., shown in Figure 6a) as the potential solutions. Here we evaluate the impact of different sizes of this range on the performance of relative positioning. Figure 15a shows the distance errors with different range sizes. When the range size is 0.1m, the highest distance error is observed, because there are no solutions meeting the constraints represented by Equations 2 and 3, due to the distance and moving vector estimation error. In this way, no position can be reported. (We set the distance error as 50m to represent those no-solution cases in the figure.) As the range size increases to 2m, the distance error decreases, because more potential solution points are collected and there is a higher probability to cover the real position during this process. Then, the distance error increases again when the range size increases to 5m, because too many points that are far from the real position are selected as potential solutions in this case, and thus the solution reported by the clustering analysis can deviate from the real position. The results in Figure 15b show a similar trend for the same reasons.

E. Impact of The Number of Vectors

As discussed in Section III, the previous moving vectors can contribute to the potential solution sets of the current relative position. We evaluate the impact of the number of moving vectors on the system performance. Figure 16 shows the distance and orientation errors with different numbers of moving vectors for the location mapping. We can see that

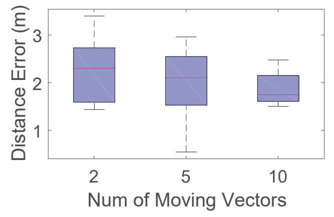


(a) Distance Error

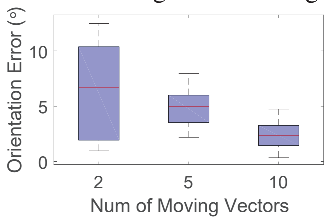


(b) Orientation Error

Fig. 15: Impact of different potential solution generation range.



(a) Distance Error



(b) Orientation Error

Fig. 16: Impact of different number of moving vectors.

the average distance and orientation errors decrease as more vectors are used to determine the current relative position, so do the variance of errors. More potential solutions are collected when more moving vectors are leveraged. This improves the system performance from two aspects: 1) the clustering analysis gets more data points and makes it easier to select the right cluster (e.g., the cluster that contains the real position), and 2) more data points make the system more robust to the errors caused by motion sensing and distance estimation.

F. System Overhead

We evaluate the overhead of SmartDistance based on an LG V10 smartphone. The average power consumption of relative positioning is 119.178mW, which includes step detection, heading estimation, distance estimation, and the location mapping. It accounts for 7.9% of the total power consumption of the whole smartphone. Moreover, the delay of relative positioning is 418ms which is mainly consumed by the clustering analysis. Power consumption and delay of other components are negligible. We can see that the system overhead of SmartDistance is acceptable in real-world cases.

VI. Related Work

Infection Risk Estimation has attracted a lot of attention recently due to the spread of COVID-19 [21]–[23]. The work that is most related to SmartDistance is conducted by Guo et al., who propose an automatic method to identify people who are potentially-infected by droplet-transmitted disease through the captured surveillance videos [21]. However, this approach has the following three limitations that negatively impacts its effectiveness. First, the surveillance video is not efficient in dark environment. Second, it requires extra hardware devices (e.g., camera). Third, it cannot alert people in real-time. SmartDistance effectively solve these problems from a new perspective, based on smartphones with non-visual sensors.

Localization with Smartphones has been widely adopted in different mobile services. A variety of positioning techniques have been proposed to improve the localization performance in different scenarios [24]–[27]. Specifically, WiFi [28]–[31], FM radio [32], magnetic fingerprinting [33] and Motes [34] have been widely adopted in indoor localization through analyzing the received RF signal strength or location fingerprinting [32]–[35]. However, these existing approaches cannot be directly applied in the social interaction monitoring scenario for the following two reasons: 1) extra hardware devices (e.g., antenna arrays) are required which are not suitable for outdoor social interaction; 2) they usually cannot provide the orientation information which is important to determine whether a person has high probability to get infected. Moreover, existing research also investigates relative positioning on mobile phones. For instance, Banerjee et al. [27] propose virtual compass in order to sense mobile social interactions. Though virtual compass can effectively sense the relative distance, it still cannot efficiently obtain the relative orientation information. In addition, there are also commercial solutions that conduct real-time location tracking with integrated GPS, internet, and Wi-Fi technology [36]. However, lacking of orientation and infection range information prevents them to be directly applied in the infection risk detection scenario.

VII. Conclusion

Coronavirus disease 2019 has resulted in an ongoing pandemic. Social distancing and close contact reporting have become the effective manners to slow down the spread. In this paper, we propose SmartDistance, a mobile based system that intelligently detects the infection risk in an effective manner. SmartDistance can generate alerts in time and records the corresponding individuals whenever a high infection risk is detected. Specifically, SmartDistance dynamically senses both the relative distance and orientation during social interaction with a well-designed relative positioning system. In addition, it recognizes different events (e.g., speaking, coughing, sniffing) and determines the infection space accordingly through a droplet transmission model. We prototype SmartDistance on Android devices with different hardware configurations, and evaluate its performance with both simulation and hardware testbed. The experiment results show that SmartDistance reduces the false positive rate from 33% to 1% and the false negative rate from 5% to 3% in infection risk detection, compared with the state-of-the-practice social distancing solution. Moreover, it effectively improves the positioning accuracy compared with the state-of-the-art relative positioning scheme.

VIII. Acknowledgement

This work is supported by the National Key R&D Program of China (No. 2019YFB2102100), Science and Technology Development Fund of Macao S.A.R (FDCT) under number 0015/2019/AKP, Guangdong Key R&D Project (No. 2020B010164003), Shenzhen Discipline Construction Project for Urban Computing and Data Intelligence. Please ask Dr Wenli Zheng (zheng-wl@cs.sjtu.edu.cn) for correspondence.

REFERENCES

[1] "Coronavirus Disease 2019," https://www.cdc.gov/coronavirus/2019-ncov/index.html.

[2] "SocialDistancing," https://www.hopkinsmedicine.org/health/conditions-and-diseases/coronavirus/coronavirus-social-distancing-and-self-quarantine.

[3] W. Chen *et al.*, "Short-range airborne route dominates exposure of respiratory infection during close contact." *Building and Environment*, 2020.

[4] J. Liu *et al.*, "Energy efficient gps sensing with cloud offloading." in *SenSys*, 2012.

[5] K. Subbu *et al.*, "Locateme: Magnetic-fields-based indoor localization using smartphones." *ACM Transactions on Intelligent Systems and Technology.*, 2013.

[6] C. Wu *et al.*, "Automatic radio map adaptation for indoor localization using smartphones." *IEEE Transactions on Mobile Computing.*, 2018.

[7] H. Xie *et al.*, "A reliability-augmented particle filter for magnetic fingerprinting based indoor localization on smartphone." *IEEE Transactions on Mobile Computing.*, 2016.

[8] V. Radu *et al.*, "Pazl: A mobile crowdsensing based indoor wifi monitoring system," in *CNSM*, 2013.

[9] K. Qian *et al.*, "Widar: Decimeter-level passive tracking via velocity monitoring with commodity wi-fi." in *Mobihoc*, 2017.

[10] Y. Lee *et al.*, "Comon: cooperative ambience monitoring platform with continuity and benefit awareness," in *Mobisys*, 2012.

[11] ——, "Socialphone: Everydat face-to-face interaction monitoring platform using multi-phone sensor fusion," in *Mobisys*, 2013.

[12] Q. Chen *et al.*, "Cooperation among smartphones to improve indoor position information." in *WoWMoM*, 2015.

[13] S. Madgwick *et al.*, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays." in *Report x-io and University of Bristol (UK)*, 2010.

[14] J. Lee *et al.*, "Turbulent jets and plumes: a lagrangian approach." *Springer Science and Business Media*, 2015.

[15] X. Sun *et al.*, "Symdetector: Detecting sound-related respiratory symptoms using smartphones." in *SenSys*, 2012.

[16] J. Amoh *et al.*, "Deep neural networks for identifying cough sounds." *IEEE Transactions on Biomedical Circuits and Systems*, 2017.

[17] Monsoon Solutions Inc, "Monsoon Power Monitor," http://www.msoon.com/LabEquipment/PowerMonitor/.

[18] K. Lee *et al.*, "Slaw: A new mobility model for human walks," in *INFOCOM*, 2009.

[19] "How to Social Distance During COVID-19," https://www.redcross.org/about-us/news-and-events/news/2020/coronavirus-what-social-distancing-means.html.

[20] H. Han *et al.*, "Amil: Localizaing neighboring mobile devices through a simple gesture." in *INFOCOM*, 2016.

[21] S. Guo *et al.*, "Droplet-transmitted infection risk ranking based on close proximity interaction." in *frontiers in Neurorobotics*, 2020.

[22] C. Harper *et al.*, "Functional fear predicts public health compliance in the covid-19 pandemic." in *International Journal of Mental Health and Addiction*, 2020.

[23] D. Yang *et al.*, "A vision-based social distancing and critical density detection system for covid-19." in *arXiv:2007.03578v2 [eess.IV] 8*, 2020.

[24] K. Liu *et al.*, "Guoguo: Enabling fine-grained indoor localization via smartphone." in *MobiSys*, 2013.

[25] C. Wu *et al.*, "Smartphones based crowdsourcing for indoor localization." *IEEE Transactions on Mobile Computing*, 2015.

[26] M. Murata *et al.*, "Smartphone-based indoor localization for blind navigation across building complexes." in *PerCom.*, 2018.

[27] N. Banerjee *et al.*, "Virtual compass: Relative positioning to sense mobile social interactions." in *Pervasive.*, 2010.

[28] P. Bahl *et al.*, "Radar: An in-building rf-based user location and tracking system." in *INFOCOM*, 2000.

[29] K. Chintalapudi *et al.*, "Indoor localization without the pain." in *MobiCom*, 2010.

[30] E. Martin *et al.*, "Precise indoor localization using smartphones." in *International conference on Multimedia*, 2010.

[31] J. Xiong *et al.*, "Arraytrack: A fine-grained indoor location system." in *NSDI*, 2013.

[32] Y. Chen *et al.*, "Fm-based indoor localization." in *MobiSys*, 2012.

[33] J. Chung *et al.*, "Indoor location sensing using geo-magnetism." in *MobiSys*, 2011.

[34] K. Lorincz *et al.*, "Motetrack: A robust, decentralized approach to rf-based location tracking." in *Personal Ubiquitous Computing*, 2007.

[35] Z. Yang *et al.*, "Locating in fingerprint space: Wireless indoor localization with little human intervention." in *Mobicom*, 2012.

[36] "Real-Time Location Tracking: How It Works," https://findmykids.org/blog/en/best-real-time-location-tracking-apps-2019.