

DroneLoc: Drone Localization with Smartphone Camera Network

Abstract—Drones are increasingly flying in sensitive airspace where their presence may cause harm, such as near airports, forest fires, large crowded events, secure buildings, and even jails. A cost-effective drone detection and localization system is needed to warn of the presence of drones in such cases. Unpredictable computer controlled movements, speed and maneuver abilities of drones, their resemblance to birds in appearance when observed from a distance make it challenging to detect, identify and correctly localize them in complex outdoor environments. To overcome these limitations in existing costly solutions, this paper formally describes, designs, implements, and evaluates a smartphone-based drone localization system, i.e., DroneLoc. The key idea behind DroneLoc is to utilize the classify ability of deep neural network to implement drone detection, then implement drone localization by fusion the detection result of multiple smartphones. In this study, we investigate camera integrated in modern smartphone and leverage the location in the image as measurement information. DroneLoc turns the localization problem into search problem by collecting the detection data of smartphones in the camera network. Location in the image is estimated through deep neural network. Drone localization is performed by minimizing the distance between the measured sequence and the vectors in a database. The proposed DroneLoc system is evaluated through extensive simulations and physical experiments (a test-bed with 30 smartphone nodes). Evaluation results demonstrate that ThunderLoc can effectively localize the drone with good robustness.

Keywords—Drone Localization, Camera Network

I. INTRODUCTION

With the advent of inexpensive commercially available unmanned aerial vehicles (UAV), drones are rapidly rising in popularity as a host of a wide class of applications ranging from commercial delivery, environment monitoring, photography, policing, fire fighting, just to name a few. However, with the rise in drone usage, there has also been a rise in incidents involving drones, such as mid-air collisions, damage to property, and violations of privacy. In particular, drones are increasingly flying in sensitive airspace where their presence may cause harm, such as near airports, forest fires, large crowded events, and even jails. A quadcopter crashed on the White House lawn, raising concerns about the safety of buildings and political leaders. The presence of drones has interfered with and grounded aircraft fighting forest fires. Drone crashes have also disrupted sporting events such as the US Open tennis tournament as well as a World Cup skiing race. In fact, based on FAA data, more than 300 drone incidents were reported in California alone between April 2014 and Jan 2016, which is equivalent of 15 incidents per month on average or 1 incident every two days.

A variety of approaches have been explored to interdict drones. These include shooting nets at the drones to tamper

with their propeller to bring them down, using lasers to shoot down drones, spoofing GPS to confuse a drone localization system, hijacking the software of drones by hacking into them, using other drones to hunt down unauthorized drones, and even training eagles to attack and disable drones. However, these interdiction strategies typically presume that the presence of the drone has already been detected. Geofencing is useful to prevent drones from flying into fixed areas known a priori as sensitive, but requires manufacturers to install such software and is less useful to prohibit drones from flying around temporary event venues. The work utilizing passive RF to identify drones has sought, for example, to detect the frequency of transmission, the MAC address of the drone, and the frequency of packet communication. Recent work has sought to develop drone detection systems that leverage either microphone, camera, or radar to sense the presence of drones. Considering the physical signatures of the drone motion, Mobisys2017 detect the presence drone by passively eavesdropping on the RF communication between a drone and its controller (Wi-Fi standard). All these techniques suffer from various limitations. Audio-based approaches can be confused by other sounds in noisy environments, has limited range, and cannot detect drones that employ noise canceling techniques. Thermal and IR imaging cameras for long distance are prohibitively expensive and have limited coverage. Radio-frequency techniques based on active radar introduce RF interference. RF-based methods often mistake the drone for birds and increase nuisance alarm rate.

In this paper, we propose a new method for thunder localization by leveraging dual-microphone smartphones. With commodity dual-microphone smartphones, the binary left/right data from dual-microphone of each smartphone is effectively collected and utilized to estimate the location of the thunder. The proposed system solely relies on the collaborative effort of the participating users to achieve thunder localization. The ThunderLoc project created such a prototype system with different types of Android based mobile phones, and validate our approach with a virtual thunder study, showcasing the potential of the proposed solution. DroneLoc takes advantage of the sensing capabilities, including the acoustics, position and orientation offered by the microphone, GPS, accelerometer and magnetometer on the smartphones. The key idea of DroneLoc is the division of a 2D localization space into distinct regions by the perpendicular bisectors of lines joining dual-microphone in each smartphone. Each distinct region formed in this manner can be uniquely identified by a binary sequence. We firstly construct the binary sequence table that maps all these feasible binary sequences to the corresponding regions by using the locations and directions information of the smartphone nodes. The smartphone nodes determine the

measured binary sequence based on the sign of TDOA between two microphones of each smartphone node. The location of an acoustic source is estimated by searching through the binary sequence table to determine the nearest feasible sequence to the measured sequence. To our knowledge, drone localization with smartphone-based has not been considered in the literature before. Our contributions are as follows:

1) We designed, implemented, and deployed a smartphone-based drone localization system called DroneLoc. Our DroneLoc app for Android platform has been installed by more than 1,000 smartphones in our university to monitor the drone near our campus. To the best of our knowledge, DroneLoc is the first smartphone-based drone localization service.

The rest of the article is organized as follows. Section II presents an overview of the DroneLoc system. Then, the design of ThunderLoc is introduced in section III. Section IV discusses several issues concerning practical system deployment. Section V presents simulation results and an empirical evaluation on the test-bed. Section VI briefly surveys related work. Section VII concludes the whole article.

II. SYSTEM OVERVIEW

The ThunderLoc system architecture was designed to allow transmission of data from dual-microphone smartphones to the cloud platform with crowdsensing mechanism. ThunderLoc leverages stereo recording with high sampling rate of off-the-shelf mobile devices to estimate the left/right binary information. The perpendicular bisector of dual-microphone divides the localization space into two regions. If the thunder is on the right of the perpendicular bisector, then the binary code of the corresponding half-plane is set as 1, and the code of the left half-plane is set as 0.

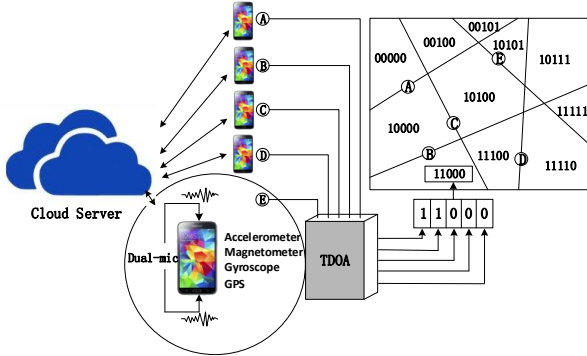


Fig. 1: System overview.

Fig. 1 shows a layout of ThunderLoc system. This research involves the use of dual-microphone smartphones as a ad-hoc sensor array based on crowdsensing in 4G cellular mobile network. To simplify the problem, an 2D localization space consisted of N dual-microphone smartphones is considered, and the signals acquired by the dual-microphone of the same smartphone are synchronized. The perpendicular bisectors of N dual-microphones divide the localization space into some small subregions.

In this paper, the thunder localization problem can be solved by using the map information and the measured binary sequence. Briefly, the ThunderLoc system works as shown in Fig. 2. In the client end, the location information and direction information of the smartphone can be estimated by GPS and the motion sensors (such as accelerometer, gyroscope) integrated in the smartphone, respectively. When a thunder occurs, each smartphone detects TDOA of two synchronized signals recorded by dual-microphone, then achieves the binary left/right data according to the sign of TDOA. The APP in each smartphone uploads the binary measurement data, the location information and direction information of the smartphone to the cloud computing platform together. During the data collection, participants can be even unaware of the collection task in which they are actually involved. Besides, ThunderLoc encourages participants to manually upload the sensing data to cloud computing platform when they hear the thunder. Thunder localization algorithm is run on the cloud computing platform to localize the dominant thunder. With these random distributed smartphones, the area can be divided into subregions, according to the positions and directions of the smartphone nodes. This naturally gives a N-vector binary codes called the detection binary sequence in cloud computing platform, as shown in Fig. 1, which embedded relative position relationships among the smartphone nodes and the thunder. With the pre-computed subregion division, the location of the thunder can be estimated by processing the binary detection sequence.

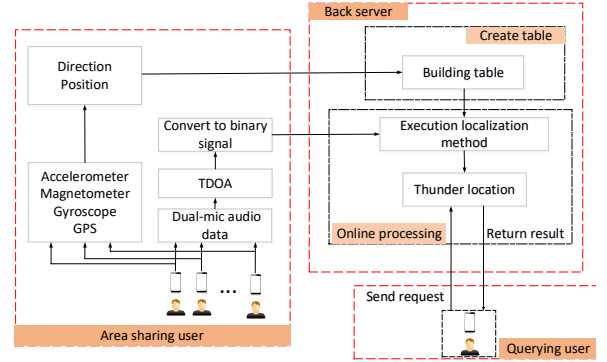


Fig. 2: System architecture.

In the next section, we will provide the basic ThunderLoc system firstly, then the robust version of ThunderLoc is proposed to deal with the adverse condition in the practical application scenario.

III. DESIGN

ThunderLoc adopts the client-server system architecture to allow transmission of data from the mobile devices to a server station or cloud platform. When a thunder occurs, the smartphones will be triggered to transmit their positions, directions and binary measurement data to the server. The server will subsequently process the received data to implement thunder localization. In this section, we firstly introduce the design of client application, then two localization methods in the server end are introduced in the next two subsections.

A. Data collection on client end

Screen shots of the application and visualizations used by the participants of the thunder can be seen in Fig. 3. Fig. 4 shows the two channel sound signals from dual-microphone in a mobile phone. It is easy to see that the thunder firstly reaches the microphone that corresponding to the top channel.

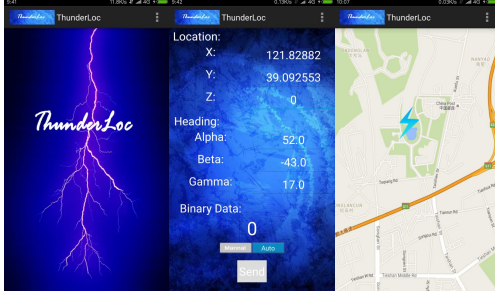


Fig. 3: Screen shots of ThunderLoc client application.

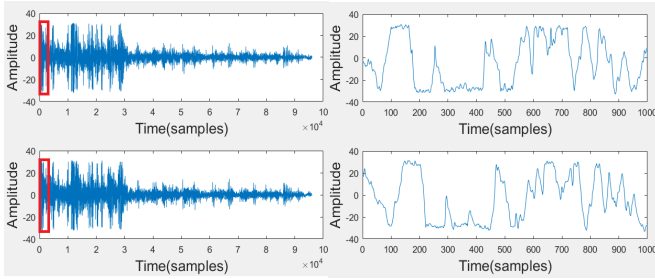


Fig. 4: Dual-microphone acoustic signals.

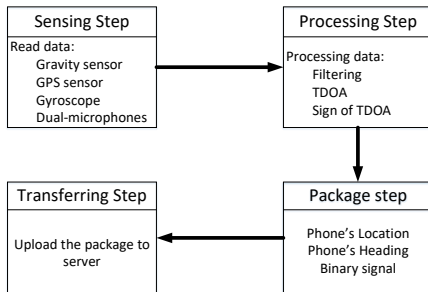


Fig. 5: The diagram of signal processing of acoustic signals.

Fig. 5 depicts the flow of the ThunderLoc client application. ThunderLoc client application has two main independent threads: sensing thread and communication thread. The sensing thread handles the interaction with the main application and the on-board sensors, the communication thread handles local storage, transmission of recorded sensor events and requeues of events in the case of poor network connection. Considering the energy problem, it is not an energy-effective solution for the mobile devices to continuously send data to the servers.

To deal with this issue, a three-state model is used for ThunderLoc client application: Idle Mode, Listening Mode and Communication Mode. Fig. 6 depicts the different modes of sensing of the client end. While continuously recording and sensing for probable thunder events locally on the mobile phones, this model permits minimal transmission of data to the server.

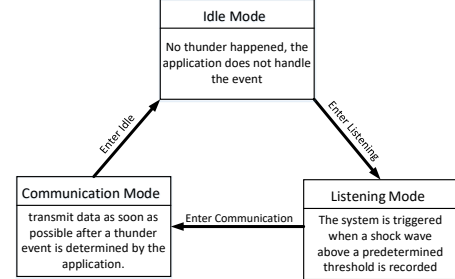


Fig. 6: Modes of sensing of the client end

1) Idle Mode: In order to determine the orientation of a mobile phone, the mobile device must be stationary for a period of time while recording the sensor data. Device movement is characterized by a change in the readings of the accelerometer and gyroscope. If the cumulative movement is under the threshold, the device will be verified as still, then jumps into the listening mode.

2) Listening Mode: Thunder is the acoustic shock wave resulted from a sudden and intense heating of the air in the lightning channel. The system is triggered when a shock wave above a predetermined threshold is recorded. Specially, the trigger is fired when an average energy is rising 4 times experienced by any of the dual microphones.

3) Communication Mode: To reduce the total latency of the ThunderLoc system, it is necessary to transmit data as soon as possible after a thunder event occurs. The application records sensor readings about heading and location of the smartphone, then immediately transmits these data to the server along with the binary measurement data estimated from two channel acoustic signals of dual microphones. In case the communication service is not available during the thunder event, a copy of these recordings is stored locally, then placed into a queue to be sent once communication is available again.

Besides the implicit way to collect data, ThunderLoc also allows user to explicitly input location, direction and binary measurement data by the way of 'humans as sensor' [7].

B. Basic localization on server side

In this section, the position and orientation of a smartphone are assumed to be known, which can be estimated by its GPS(or Wi-Fi based indoor localization) and motion sensors, respectively. Fig. 7 depicts the flow of the ThunderLoc server application. In the cloud computing platform, the map of space division can be built with the position and orientation information of all smartphone nodes. Then, we can turn the acoustic source localization into the searching problem in the Hamming space.

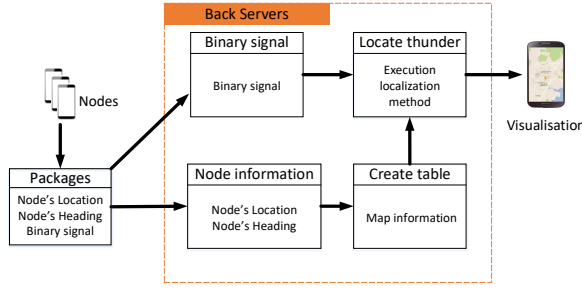


Fig. 7: The diagram at cloud platform.

Firstly, we introduce the basic localization method. Let us consider the 4G network with N dual-microphone smartphones randomly deployed in 2D area. The top-level idea for basic ThunderLoc is to split the whole localization area into some subregions identified by respective binary sequences. Given N smartphone nodes in the localization space, the whole number of combinations of binary sequences is 2^N in theory. However, in the practical system, given N smartphone nodes in the localization space, the possible number of combinations of binary sequences is only $(N^2 + N + 2)/2$. The lower dimensionality of the sequence table enables the correction of errors in the measured sequence. This is one of the reasons that our proposed algorithm performs well in adverse conditions. Localization system can achieve higher localization performance as the number of smartphones increases, such extensibility is an unique advantage compared with dedicated microphone array hardware.

Binary sensor model: We propose a binary sensor model, where the TDOA of dual-channel acoustic signals is converted reliably to one bit of information: the thunder is on the left or right of the perpendicular bisector of dual-microphones. We use the sign of TDOA as 1 bit measurement information, which reveals the left/right information about the direction of the incoming Thunder. Using 1-bit measurement information allows for inexpensive sensing as well as minimal communication load. As shown in Fig. 2, for each dual-microphone smartphone node, TDOA is computed by traditional time delay estimation method (such as generalized cross correlation, GCC), then we can get the binary sequence of the thunder.

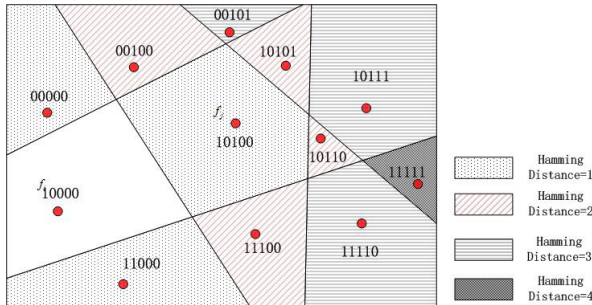


Fig. 8: Hamming distance vs. geographic distance.

Hamming distance: For two faces f_i and f_j in Fig. 8, now there are two types of distance: (i) the geographical distance

$GD(f_i, f_j)$ between the center points of f_i and f_j , and (ii) the Hamming distance $HD(f_i, f_j)$. Hamming distance measures the number of dimensions where two vectors have different values.

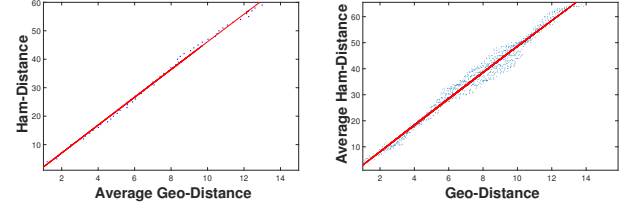


Fig. 9: Relationship between Hamming distance and geographic distance.

From Fig. 9 we can see that the closer the two faces are in geographical distance, the smaller their Hamming distance is. In other words, geographical distance is positively correlated with Hamming distance. For these two distances, we have the following observation:

$$GD(f_i, f_j) \propto HD(f_i, f_j) \quad (1)$$

Equation (1) indicates that the Hamming distance between two faces is approximately proportional with their geographical distance. This is because longer geographical distance creates more chances to cross bisectors, resulting in more flipped bit. So when the Hamming distance is close to zero, then the locations are very close to each other. Moreover, we do lots of simulation experiments to prove it. The result of the experiments can be seen in Fig.7 and Fig.8. It proves that Hamming distance is almost linear growth with geographical distance increasing. Given a query binary vector from multiple smartphones, we can estimate the location by retrieving the vectors in the beforehand database and find the smallest Hamming distance. In other words, we can find the target location through by querying the smallest Hamming distance.

The basic ThunderLoc contains the three steps:

(1) Dividing the space into p grid points : Suppose there is an acoustic source at the point $P_i, i = 1, \dots, p$, the j -th smartphone $n_j, j = 1, \dots, N$ can compute the TDOA and get the sign of the TDOA. Then we can get the binary code $C_{i,j} \in \{0, 1\}$ by the sign of TDOA. Combining the binary code of all N smartphones, we can get a N -vector binary sequence $D_i, i = 1, \dots, p$. The database about p discrete points is obtained, and an item in the database is $S_i = \{D_i, P_i\}$.

(2) Computing the binary sequence T of the target: When the acoustic source emits sound, all the smartphone nodes compute the TDOA, then get the sign of TDOA and determine the binary code. $TDOA_i$ of the i -th smartphone can be computed by using time delay estimation algorithm(such as GCC), and the binary code of the i -th smartphone can be defined by computing the sign of the TDOA:

$$Binary_data_i = \begin{cases} 1, & \text{if } TDOA_i \geq 0; \\ 0, & \text{if } TDOA_i < 0. \end{cases} \quad (2)$$

(3) Processing the binary sequences T : Computing the Hamming distance between T and each $D(i)$, the acoustic

source position can be found by searching the minimum of Hamming distance. However, it should be noted that in general there are several source points with the same minimum value of Hamming distance. This is due to the finite estimate resolution which creates areas with the same Hamming distance. The final position estimate is the mean of all points with the minimum Hamming distance.

We can calculate time complexity according to the three steps. In step 1, we build the database, including calculating TDOA of a pair of microphones and getting its sign just use the linear time. For each grid point we need to calculate TDOA of N pairs of microphones and there exists P grid points, so the time complexity in step 1 is $O(PN)$. The time complexity of step 2 is $O(N)$ similarly. In step 3, the time complexity of the search processing is $O(PN)$. So the overall time complexity is $O(PN)$.

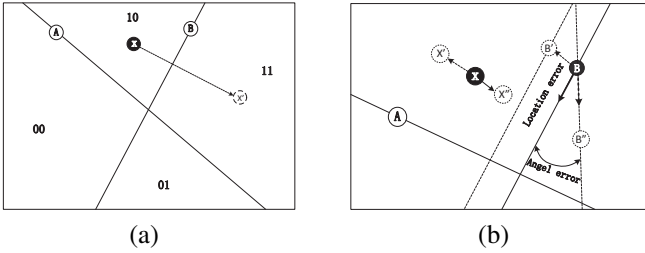


Fig. 10: The impact of variety of errors.

The bit flip problem for measurement data and the location and direction parameter errors have impact on the localization performance as shown in Fig. 10. As shown in Fig. 10(a), In case the measurement binary data from node B is flipped, the binary sequence '10' is turned into '11', reducing the accuracy of localization system. In Fig. 10(b), if the position measurement or angle measurement of sensor B is inaccurate, there is a certain distance between actual position and estimated position. According to the above, the basic Thunderloc is not suitable for the practical application. In the next section, we propose a robust localization solution against the measurement errors, position errors and orientation errors of smartphones.

C. Robust localization on server side

In the practical application, the bit-flip problem mentioned before should be considered. Moreover, the measurement error and the uncertainty of node parameters have effects on the localization performance. In this subsection, the robust version of ThunderLoc is devised to circumvent these problems. The results we have obtained empirically indicate that the robust localization method can dramatically reduce the localization error under the practical application.

In order to improve the robustness of the localization system, the possible coordinates for the thunder are computed through selecting the K smallest Hamming distance instead of the minimum Hamming distance. Then, the centroid estimation method is utilized to set the center of gravity of all the possible points as the estimated location of the thunder.

However, the selection of the parameter K is a difficult task. Deeply thinking the relation between the Hamming

distance and the 2D geographic distance, the smaller Hamming distance is, the closer the coordinate region is to the target. For each discrete point in localization space, the weight of each coordinate point is assigned based on Hamming distance. Specially, the weight of each coordinate point is monotonic decreasing function of Hamming distance.

Gaussian function is adapted to weight discrete points according to the respective Hamming distance:

$$w_i = \exp(-HD(T, D_i)/N\sigma^2) \quad (3)$$

where σ is the parameter to decide the weighting strategy.

Based on the normalized weight \bar{w}_i , the final estimated location of the thunder is estimated by computing the weight sum of all grid points:

$$\bar{w}_i = \frac{w_i}{\sum_{i=1}^P w_i} \quad (4)$$

$$\hat{x} = \sum_{i=1}^P \bar{w}_i \cdot p_i \quad (5)$$

Algorithm 1 illustrates the pseudo code of robust ThunderLoc method.

Algorithm 1: Robust ThunderLoc

- 1 **Input:** The location coordinates of all smartphones
 - 2 The orientations of all the smartphones
 - 3 Information about microphones *measure_data*
 - 4 **Output:** Location of the thunder
 - 5 **Step 1:** Database building:
 - 6 **Step 2:** Computing the binary code T of the thunder from the dual-channel acoustic signals:
 - 7 **Step 3:** Processing the binary sequence:
 - 8 (1) Computing Hamming distance for each point:
 - 9 **for** $i \leftarrow 1$ **to** p **do**
 - 10 $HD(T, D(i)) = \sum_{j=1}^N (T(j) \oplus D_i(j))$
 - 11 **end**
 - 12 (2) Computing the weight of each grid point using Eq. 3 :
 - 13 (3) Normalizing the weights and Estimating the source location using Eq. 4 and Eq. 5, respectively.
-

Compared with the basic ThunderLoc, robust ThunderLoc leverages all possible points to locate the thunder. Although the computing load increases, robust ThunderLoc has two benefits: (i) Avoiding the problem of selecting parameter K ; (ii) More robust to measurement errors and node parameter errors. Our evaluation in sections V shows that robust localization method considerably enhances the robustness of localization system.

We can calculate time complexity of robust ThunderLoc according to the three steps. From the algorithm we can see the time cost is the same as basic localization method in step 1 and step 2. Besides, in step 3, we should calculate the normalized

weighting of each point, and the corresponding time complexity is $O(P)$. At last step, we should calculate the location of the thunder by the weight sum of each grid point, and the corresponding time complexity is $O(P)$. The time complexity of robust ThunderLoc method is also $O(PN)$. At the cost of $O(P)$ time complexity more than basic ThunderLoc method, robust ThunderLoc method enhances system robustness. The computing of Hamming distance is the core of the proposed ThunderLoc system, which can be effectively implemented via the Boolean XOR operation in the cloud computing platform with GPU and FPGA support.

IV. DISCUSSION

A. Humans as sensors

Humans as sensors denotes a new measurement model for crowdsensing. In the model, measurements are not only taken by smartphone, but also come from the personal observations of participants. As a complement of automatically acquiring the localization data from smartphones, ThunderLoc provides the interface for participants to manually upload personal observations. The users of smartphones can manually input the personal localization data in the ThunderLoc app, including the binary left/right data, the position and direction when they hear the thunder.

However, how to motivate enough participants for ThunderLoc is a problem in reality. Here a possible solution is given as follows. ThunderLoc app can be integrated into a weather forecast app. When the thundershower is coming, the weather forecast app pushes a notice to inquire whether the user to join the ThunderLoc project. If the user agrees to participate in, ThunderLoc app will be downloaded and run. Once the cloud platform finishes the localization operation, the position of thunder will be push to the user. The well-informed discussions about the incentive mechanisms for smartphone collaboration are detailed in Ref. [8].

B. The impact of background noises

Above, we put forward basic ThunderLoc and robust ThunderLoc to implement thunder localization by using dual-microphone smartphones. The binary left/right measurement data of each smartphone is from the sign of TDOA between two audio signals recorded with dual-microphone. However, in the practical localization system, bit flip problem may occur because of the effect of background noises in the phase of computing binary left/right for each smartphone, then lead to localization error.

The lower SNR is, the measurement of this smartphone is more unreliable. We use the SNR of each smartphone node as the reliability metric, then compute Hamming distance with the reliability information.

The new localization method is the same as robust localization method described in Algorithm 1, the only difference is that using of the weighting Hamming distance instead of traditional Hamming distance:

$$WHD = \sum_{i=1}^N F(i) (T(i) \oplus D_j(i)) \quad (6)$$

C. System scalability and multiple thunder localization

Our system is evaluated in the small campus area, but it can extend to large area. The Fig.11 describes system scalability. In Fig.11, the signal from thunder in large-scale system only can cover limited area which means when we measure the binary sequence, just a small portion of microphones close to the thunder are effective. So the binary sequence of thunder is shorter than N (there exist N dual-microphone smartphones). We can use a range R to put the large-scale system down to a smaller area which is large enough to cover the coverage of the thunder's signal. And the small area can be handled as our algorithm proposed by preceding part of the paper. By this way, the length of the binary sequence is shorter and matching is more efficient.

Localizing multiple, simultaneously active acoustic sources is a more difficult task. In order to avoid conflict of several acoustic sources, multiple source localization must be able to uniquely identify the signature of each thunder, which is beyond the capability of this paper, and we will not do too much discussion about it in this paper. For example, if the several thunders are set as shown in Fig. 11, we can calculate the positions of the two thunders because the thunder I is far enough from thunder II. The smartphones close to effective area I could handle thunder I, and another smartphones close to effective area II could handle thunder II.

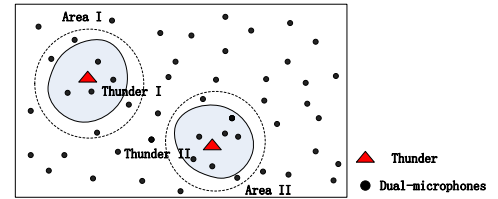


Fig. 11: System scalability for large-scale systems.

D. Time Synchronization and Energy Efficiency

In our system, we calculate the sign of TDOA by utilizing two synchronized audio signals from dual-microphone of each smartphone. High-accuracy of time synchronization among smartphones is not required.

Energy efficiency is another issue in smartphone networks. We can activate the ThunderLoc app in the smartphone according to the thundershower alert from the weather forecast app. Only if it is raining with thunder, ThunderLoc app will be awake.

V. RESULTS

A. 2D Simulation

In this section, we evaluated our localization methods in 2D scenario using MATLAB. In the simulation, we randomly deployed dual-microphone mobile phones on a $10\text{km} \times 10\text{km}$ area with uniform distribution, and the distance between dual-microphones is 14cm, and audio sampling rate is 48kHz. All the statistics are run more than 3000 times for high confidence, and reported by Root Mean Square Error (RMSE).

The parameter σ in robust ThunderLoc has little impact to the localization performance. In default, $\sigma = 1$. We intend to make a comprehensive assessment and comparison for the two proposed localization methods from different aspects, including the impact of the number of nodes, the number of fault nodes, the impact of the position error of nodes, the impact of the angle error of nodes. In default, there are 100 mobile phones and just one thunder. To simulate the impact of the position and orientation of smartphone nodes on the accuracy of localization, all the simulations are added a certain amount of position and angle error. In default, the angle error range is 5 degree and the position error range is 100m. The results of simulation evaluation are as follows.

1) Impact of the number of nodes: In this experiment, we investigate the localization error over number of nodes from 50 to 250 in steps of 10. We run the simulation with 2 samples TDOA error, and other simulation parameters are default. As shown in Fig. 12, with the increasing of the number of mobile phones, the whole area could be divided into more grids, thus more accurate localization estimation were achieved for both of two localization methods. When the number of mobile phones is low, the performance of robust ThunderLoc is better than basic ThunderLoc. With the increasing of the number of participants, two localization methods could get almost the same localization performance.

2) Impact of the number of error nodes: In the practical application, if dual-microphone is very close to each other along the direction of event propagation, they would detect the event almost simultaneously. In this case, the binary left/right data in the sequence may occur to flip. We called this smartphone as the error node. In this experiment, we try to compare the two methods with different percentage of error nodes, and it ranges from 0 to 0.2 in steps of 0.01. Other simulation parameters remain default. The Fig. 13 indicates that the localization error increases as the number of error nodes increases for both of the two methods. And it also shows as the number of error nodes increases, the average localization error of robust ThunderLoc is little smaller than that of basic ThunderLoc. robust ThunderLoc provides a effective solution to deal with the bit flip problem mentioned before.

3) Impact of the position error of nodes: In the experiment, we evaluated the impact of the position error of nodes for the basic ThunderLoc and robust ThunderLoc with the range from 0 to 1000m in steps of 50. Other simulation parameters keep default. Fig. 14 shows that with the increasing of the position error of smartphone nodes, the average localization error of robust ThunderLoc is much smaller than that of basic ThunderLoc.

4) Impact of the orientations error of nodes: In the experiment, we evaluated the impact of the angle error of nodes for the basic ThunderLoc and robust ThunderLoc with the range from 0 to 20 degree in steps of 2. Other simulation parameters keep default. As the orientations of nodes exist errors, we can guess the error of orientations may influence the localization accuracy. As shown in Fig. 15, the graph to the localization error for the two methods is rising as the angle error of nodes increases. When the angle error of nodes is relative large, the average localization error for the robust

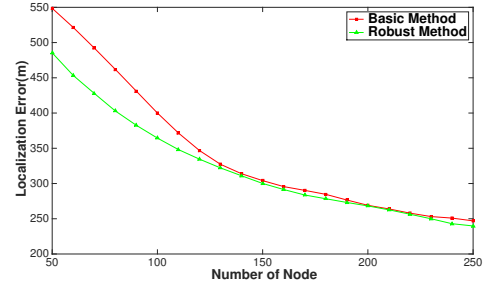


Fig. 12: Impact of number of nodes

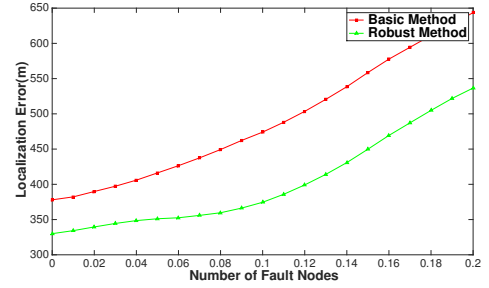


Fig. 13: Impact of number of error nodes.

ThunderLoc is much smaller than basic ThunderLoc, so the localization accuracy of robust ThunderLoc is much better than that of basic ThunderLoc.

B. 3D Simulation

The height of the thunder should be considered in some applications, such as lightning channel reconstruction. In this section, we evaluated our localization methods in the 3D scenario. In the simulation, we randomly deployed 100 dual-microphone smartphones to cover a $10\text{km} \times 10\text{km}$ region. The altitude of mobile phones is about 10m and the height of the thunder is about 3km. The number of nodes is 100, and the angle error range of smartphones is 5. Due to limitations on space, we just investigate the impact of the number of smartphones with the range from 80 to 160. The localization result of the experiment is showed in Fig. 16. Comparison with the 2D scenario, the localization errors are much larger in 3D space for the same number of smartphones. The reason is that the localization errors are related to the size of space grids divided by the perpendicular bisector of dual-microphone. In the 3D scenario, the more smartphones is needed to achieve high-accuracy localization.

C. Virtual Thunder Emulation

The virtual thunder test was useful as a proper verification of the functionality of our ThunderLoc project. We use 30 dual-microphone Xiaomi smartphones as sensor nodes and connect them through CISCO CVR328W-K9-CN wireless router. The distance between the two microphones is 14cm. We set the 30 nodes in a size of $1\text{km} \times 1\text{km}$ space and there just one target during an experiment. In order to simulate the thunder, we broadcast the recorded sound signal of thunder by speaker at the top of the three buildings. Smartphones

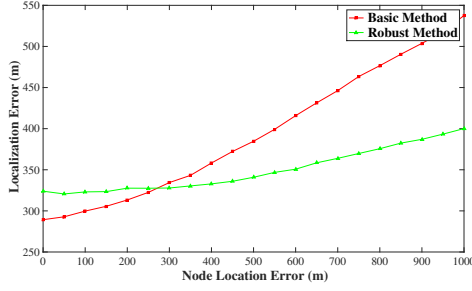


Fig. 14: Impact of the position error.

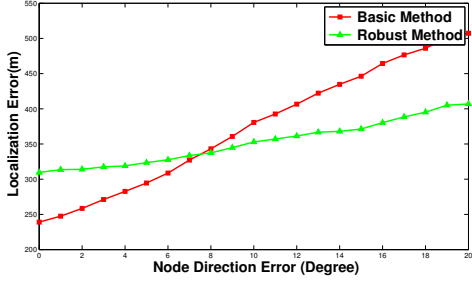


Fig. 15: Impact of orientation error.

are randomly deployed around the building. The localization results are shown in Fig. 17, where blue squares stand for dual-microphone smartphones, and blue circles and red circles are the ground truth and the estimated position by robust localization method. An arrow originates from the estimated position to the real position of the virtual thunder. As the results shown in Fig. 17, the estimated position is close to real position and the localization error is within the expected range, which means that our proposed ThunderLoc system can effectively locate the thunder by the way of crowdsensing.

VI. RELATED WORK

Thunder localization system can use infrasonic signals, supersonic signals, and audible signals as the measurement data. Due to space constraints, we can only mention a few related works using audible sound data. Most of the existing thunder localization systems using audible sound data are based on centralized architecture. Arechiga, *et al.* [9] studied acoustic reconstruction of lightning channel geometry using microphone arrays, improved upon thunder localization by using compact acoustic arrays with 1kS/s sample rate. Akiyama, *et al.* [10] used rhombus array with 100k sample rate and 3.50m base line. Few, *et al.* [11] implemented lightning channel reconstruction using 'Y' 4-element array with 1k sample rate and 30m base line. MacGorman, *et al.* [12] considered the effect of temperature and wind, presented a lightning localization system with square array with 0.5k sample rate and 50m base line. Johnson, *et al.* [13] used a network of broadband microphones, including a 4-element array with 1k sample rate, to locate the sources of thunder occurring during an electrical storm. Qiu, *et al.* [14] used 3-element array with 100k sample rate and 12.5m base line to estimate the location of thunders.

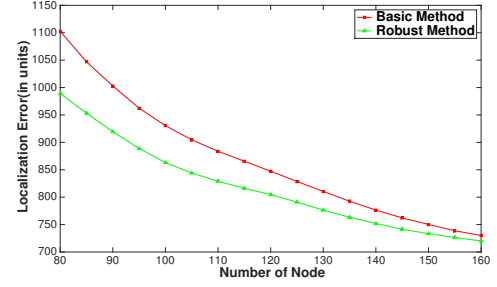


Fig. 16: Thunder localization in 3D space

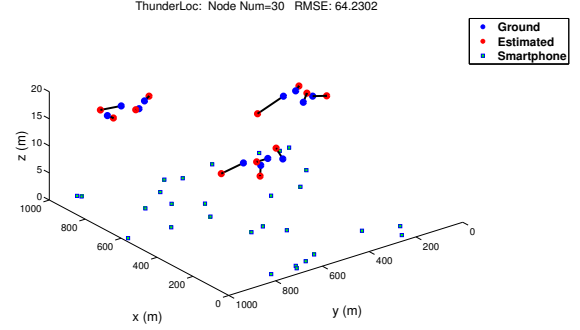


Fig. 17: Experiment result.

In the past few years, there has been a growing interest for acoustic localization based on independent (unsynchronized) acoustic sensors, each made of two or more synchronized microphones. UbiK leveraged the dual-microphone on a mobile device to accurately localize the keystrokes [15]. Zhu, *et al.* [16] proposed to utilize dual-microphone on three phones to identify keystrokes of a nearby keyboard based on time difference of arrival (TDOA) measurements. Liu, *et al.* [17] achieved keystroke recognition by discriminating keystrokes based on TDOA of the keystroke sound at the dual-microphone of the off-the-shelf smartphone. Wang, *et al.* [18] described a system having static cluster architecture, the system experienced a problem in that the accuracy decreased when an acoustic source occurred between the clusters. Chen, *et al.* [19] showed that nodes in the system did not need to recognize their cluster head, reducing the constraints on deployment of the localization system. Hu, *et al.* [20] design the system based on 2-tier architecture, which experienced cost and deployment problems especially in the very large target area. Canclini, *et al.* [21] proposed a method for localizing an acoustic source with distributed microphone networks based on TDOA between microphones of the same sensor. Different from these range-based methods, our proposed ThunderLoc is a range-free method, which can achieve robust localization performance.

The most related work to our ThunderLoc project is Lightning@SG, an App developed by National Environment Agency (NEA) in Singapore [22]. By utilizing the four sensors located in different cities, Lightning@SG monitors the lightning area based on current location of user, and pushes notifications of the lightning situation. With the surging of smartphone sensing

and wireless networking techniques, crowdsensing has become a promising paradigm for cross-space and large-scale sensing. The iShake system uses smartphones as seismic sensors to measure and deliver ground motion intensity parameters produced by earthquakes [23]. Hu, *et al.* presented SmartRoad, a crowd-sourced road sensing system that detects and identifies traffic regulators, traffic lights, and stop signs [24]. Zhou, *et al.* presented a novel bus arrival time prediction system based on crowd-participatory sensing [25]. Motivated by these works, we use dual-microphone smartphones as sensors to estimate the position of the thunders by crowdsensing mechanisms.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have designed DroneLoc, a novel drone localization system that leverages smartphones to achieve drone localization using unreliable detection data by deep neural network. The proposed design formulates the drone localization as the searching problem by making use of the data from the smartphones. Since our DroneLoc system runs on COTS smartphones and supports spontaneous setup, it has potential to enable a wide range of drone localization systems. Besides the basic design, a robust localization method is proposed to enhance the localization performance in the practical application. Our system is verified and evaluated through analysis, extensive simulation as well as the test-bed experimentation. The test results have shown that the proposed method can effectively implement drone localization with smartphone camera networks. Our next step is to study the distributed implement of the proposed DroneLoc.

REFERENCES

- [1] K. L. Cummins and M. J. Mur, "An overview of lightning locating systems: History, techniques, and data uses, with an in-depth look at the us nldn," *IEEE Transactions on Electromagnetic Compatibility*, vol. 51, no. 3, pp. 499–518, 2009.
- [2] A. Few, "Acoustic radiations from lightning," *Handbook of atmospheric electrodynamics*, vol. 2, pp. 1–31, 1995.
- [3] M. Dayeh, N. Evans, S. Fuselier, J. Trevino, J. Ramaekers, J. Dwyer, R. Lucia, H. Rassoul, D. Kotovsky, and D. Jordan, "First images of thunder: Acoustic imaging of triggered lightning," *Geophysical Research Letters*, vol. 42, no. 14, pp. 6051–6057, 2015.
- [4] S. Vahabi-Mashak, Z. Abdul-Malek, K. Mehranzamir, H. Nabipour-Afrouzi, B. Salimi, and C.-L. Wooi, "Modeling of time of arrival method for lightning locating systems," *Advances in Meteorology*, 2015.
- [5] P. Aarabi, "The fusion of distributed microphone arrays for sound localization," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 4, pp. 338–347, 2003.
- [6] T. Wu, L. Sun, Q. Cheng, and P. K. Varshney, "Fusion of multiple microphone arrays for blind source separation and localization," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM'12)*, 2012, pp. 173–176.
- [7] W. Dong, A. M. Tanvir, and e. a. Li Shen, "Using humans as sensors: An estimation-theoretic perspective," in *IEEE/ACM IPSN'14*, 2014, pp. 35–46.
- [8] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *IEEE INFOCOM'12*, 2012, pp. 1701–1709.
- [9] R. Arechiga, J. Johnson, E. Badillo, J. Michnovicz, R. Thomas, H. Edens, and W. Rison, "Acoustic network localization and interpretation of infrasonic pulses from lightning," in *AGU Fall Meeting Abstracts*, vol. 1, 2011, p. 0273.
- [10] H. Akiyama, K. Ichino, and K. Horii, "Channel reconstruction of triggered lightning flashes with bipolar currents from thunder measurements," *Journal of Geophysical Research: Atmospheres (1984–2012)*, vol. 90, no. D6, pp. 10 674–10 680, 1985.
- [11] A. Few, "Lightning channel reconstruction from thunder measurements," *Journal of Geophysical Research*, vol. 75, no. 36, pp. 7517–7523, 1970.
- [12] D. R. Macgorman, "Lightning location in a storm with strong wind shear," Ph.D. dissertation, Rice University, Houston, Texas, 1978.
- [13] J. Johnson, R. Arechiga, R. Thomas, H. Edens, J. Anderson, and R. Johnson, "Imaging thunder," *Geophysical Research Letters*, vol. 38, no. 19, 2011.
- [14] S. Qiu, B.-H. Zhou, and L.-H. Shi, "Synchronized observations of cloud-to-ground lightning using vhf broadband interferometer and acoustic arrays," *Journal of Geophysical Research: Atmospheres (1984–2012)*, vol. 117, no. D19, 2012.
- [15] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization," in *ACM MobiSys'14*, 2014, pp. 14–27.
- [16] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *ACM CSS'14*, 2014, pp. 453–464.
- [17] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *ACM MobiCom'15*, 2015, pp. 142–154.
- [18] Q. Wang, W.-P. Chen, R. Zheng, K. Lee, and L. Sha, "Acoustic target tracking using tiny wireless sensor devices," in *IEEE/ACM IPSN'03*, 2003, pp. 642–657.
- [19] W.-P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing (TMC)*, vol. 3, no. 3, pp. 258–271, 2004.
- [20] W. Hu, N. Bulusu, C. T. Chou, S. Jha, A. Taylor, and V. N. Tran, "Design and evaluation of a hybrid sensor network for cane toad monitoring," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 4, 2009.
- [21] A. Canclini, E. Antonacci, A. Sarti, and S. Tubaro, "Acoustic source localization with distributed asynchronous microphone networks," *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, vol. 21, no. 2, pp. 439–443, 2013.
- [22] "Lightning information service," <http://www.weather.gov.sg/lightning/lightning/alertinformationssystem.jsp>.
- [23] J. Reilly, S. Dashti, M. Ervasti, J. D. Bray, S. D. Glaser, and A. M. Bayen, "Mobile phones as seismologic sensors: Automating data extraction for the isake system," *IEEE Transactions on Automation Science and Engineering (TASE)*, vol. 10, no. 2, pp. 242–251, 2013.
- [24] H. Shaohan, S. Lu, L. Hengchang, W. Hongyan, and A. Tarek, "Smartroad: a crowd-sourced traffic regulator detection and identification system," in *IEEE/ACM IPSN'13*, 2013, pp. 331–332.
- [25] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *ACM MobiSys'12*, 2012, pp. 379–392.