iThunder: Smartphones-based Crowdsensing for Thunder Localization

Abstract—**Thunder localization is an important solution to lightning location systems. To overcome limitations in existing centralized solutions, this paper formally describes, designs, implements, and evaluates a smartphone-based thunder localization system, i.e., iThunder. In this study, we investigate dual micphones integrated in modern smartphone and leverage the sign of Time Differences of Arrival (TDOA) as measurement information. The key idea behind iThunder is to turn the localization problem into search problem in Hamming space by collecting the dual-microphone data of smartphones via crowdsourcing mechanism. TDOA of thunders pertaining the same smartphone is estimated through the simple generalized cross-correlation method. After the quantization with a bit for the TDOA measurement from the smartphone nodes, thunder localization is performed by minimizing the Hamming distance between the measured binary sequence and the binary vectors in a database. The proposed iThunder system is evaluated through extensive simulations and physical experiments (an test-bed with 30 smartphone nodes). Evaluation results demonstrate that iThunder can effectively localize the visual thunders with good robustness.**

Keywords*—*Thunder Localization, Smartphone Crowedsensing, Hamming Distance

I. Introduction

Thunder is the acoustic shock wave resulting from a sudden and intense heating of the air in the lightning channel [1]. The distance of the lightning can be calculated by the listener depending on how long the sound is heard since the vision of the lightning strikes [2]. Depending on the nature of the lightning and distance of the listener, thunder can range from a sharp, loud crack to a long, low rumble (brontide). The sudden increase in pressure and temperature from lightning produces rapid expansion of the air surrounding and within a bolt of lightning. In turn, this expansion of air creates a sonic shock wave which produces the sound of thunder, often referrs to as a clap, crack, or peal of thunder. After propagating a few meters, the initial shock wave, which travels at a rate faster than the speed of sound, turns into a thunder.

Thunder localization system has a variety of applications, such as lightning channel reconstruction, risk analysis of faults in power systems, and electromagnetic interference studies [3]. The bright image in the sky that entertains us is a direct threat to air and ground-based operations, and is a reflection of other destructive forces associated with thunderstorms and severe weather. Due to the enormous amount of data that can be gathered by means of lightning location systems (LLS), these systems represent a promising source of experimental data to be used for the development of standards related to the protection of power systems against lightning. Cloud-to-ground (CG) lightning is the single largest cause of transients, faults, and outages in electric power transmission and distribution systems in lightning-prone areas. Additionally, lightning is a major cause of electromagnetic interference that can affect all electronic systems.

In the last decade, lightning location systems were based on electromagnetic radiation pulse and the price was high [4] . In recent years, the centralized microphone array-based solution to thunder localization exploited multiple synchronized microphones to simultaneously acquire acoustic signals, which have some limitations with regards to distances between the microphones, and sensing range for the large-scale applications [5] . With the recent advances in mobile computing and communication technology, most mobile devices (e.g. smart phones, tablets and laptops) are equipped with one/multiple microphones onboard. Wireless acoustic sensor networks (WASNs) can overcome these limitations. Thunder localization in WASN is increasingly becoming feasible due to recent advances in personal portable computing devices with the rapid deployment ability. A WASN consists of a set of wireless microphone nodes that are spatially distributed over the environment, usually in an ad-hoc fashion. Due to the wireless communication, the array-size limitations disappear and the microphone nodes can physically cover a much larger area. The acoustic source localization problem in sensor networks has been widely studied in the literature. In previous research on distributed acoustic source localization system, generally each node just has a single microphone element. In the past few years, there has been a growing interest for acoustic nodes made of two or more synchronized microphones. Aarabi, *et al.* [6] used 10 dual-microphone arrays distributed in a room and used their data to locate three speakers. Wu, *et al.* used three two-microphone arrays to locate two sound sources in a distributed way in which only the local DOA estimates are communicated among arrays [7]. Most of the system need to design the special hardware to capture the multiple channel synchronized audio signal.

In this paper, we propose a new method for thunder localization by leveraging dual-microphone smartphones. With commodity dual-mic smartphones, the thunder's TDOA from 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. We develop a prototype system called 'iThunder' with different types of Android based mobile phones, and validate our approach with a visual thunder study, showcasing the potential of the proposed solution. The iThunder system was created to take advantage of the sensing capabilities, include the acoustics, position and orientation offered by the microphone, GPS, accelerometer and magnetometer on the smartphones. The key idea of iThunder is the division of a 2D localization space into distinct regions by the perpendicular bisectors of lines joining dual-microphone in each smartphone. We show that 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, thunder localization with smartphones based crowdsensing has not been considered in the literature before. Our contributions are as follows:

1) We designed, implemented, and deployed a crowdsourced thunder localization system called iThunder. Our iThunder app for Android platform has been downloaded by more than 1,000 users in our university, and are used on a daily basis by hundreds of users to monitor thunder near our university. To the best of our knowledge, iThunder is the first automated crowdsensing thunder localization service.

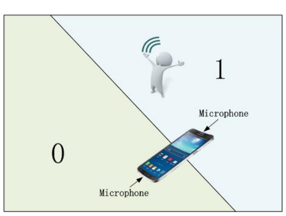
2) Synchronization-free localizationiThunder just uses the sign of TDOA between two synchronized microphones of each smartphone, can reduce the requirement of time synchronization;

3) Error-tolerant localization: binary left/right data and novel localization method make the localization system more robust to the location error,direction error and measurement error of nodes;

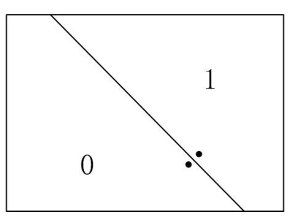
4) Low communication overhead and computational complexity: just 1 bit measurement information is passed in the sensor network, and the simple GCC (Generalized Cross-Correlation) algorithm is good enough to estimate the binary left/right data;

The rest of the article is organized as follows. Section II presents an overview of the iThunder system. Then, the design iThunder of 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



(a) smartphone



(b) microphones divide the space

Fig. 1: The smartphone with two microphones.

In this section, we focus on the system overview of the ThunderLoc system based on dual-mic smartphones with crowdsensing mechanism. As it is shown in Fig.1(a), most of smartphones have two microphones which provide hardware conditions for our paper. The perpendicular bisector of two microphones can divide the localization space into two regions as showed Fig.1(b). The acoustic source is on the right of the perpendicular bisector, then the binary code of the right halfplane is 1, and the code of the left halfplane is 0.

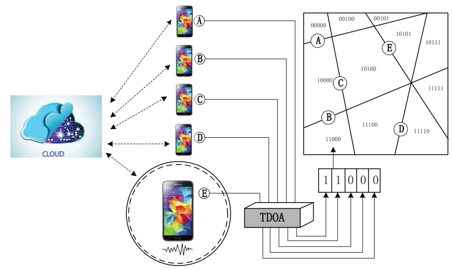


Fig. 2: System overview.

Fig. 2 shows a layout of iThunder system. We use dual-mic smartphones in 4G network as sensor nodes. For simplifying the problem, let us consider an 2D localization space consists of distribution of N dual-microphones smartphones, and the signals acquired by the two microphones of the same smartphone are synchronized. The perpendicular bisectors of all N pairs of microphone 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 iThunder system works as follows. The location information and direction information of smartphone can be estimated by GPS and motion sensors (such as accelerometer, gyroscope) integrated in the smartphone, respectively. When the thunder occurs, each smartphone detects TDOA of signals emitted from the thunder using its dual-microphones, then get the binary codes according to the sign of TDOAs. The APP in each smartphone upload 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 collecttion task in which they are actually involved. The system also encourages participants to actively upload these information to cloud computing platform when thunder occurs. Then 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.2, which embedded relative position relationships among the smartphone nodes and the thunder. Then, with pre-computed map division, the location of the thunder can be estimated by processing the detection sequence.

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

III. DESIGN

iThunder adopts the client-server system architecture to allow transmission of data from the mobile devices to a server station. When a thunder occurs, the devices will be triggered to transmit their location, direction and binary measurement data to the server. The server will subsequently process the received data to implement 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 at client end

An overview of the architecture of the client application 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 reach the micphone that corresponding to the top channel.

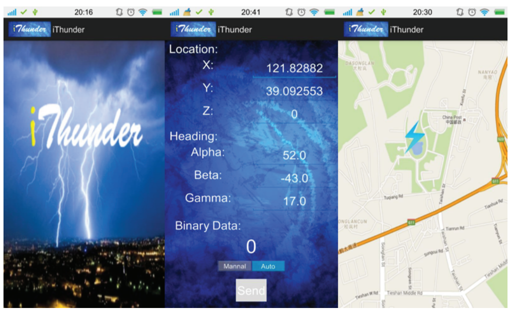


Fig. 3: Screen shots of the iThunder client application.

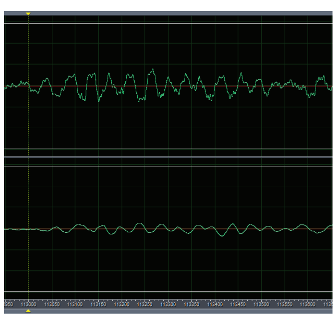


Fig. 4: Dual microphone sound signals.

The iThunder client application has two main independent thread: 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 energy-effective solution for the mobile devices to be continuously sending data to the servers. To deal with this issue, a three-state model was used for iThunder client application: Idle Mode, Listening Mode and Communication Mode. This model permits minimal transmission of data to the server, while continuously recording and sensing for probable thunder events locally on the mobile phones. Fig. 5 depicts the flow of the iThunder client application.

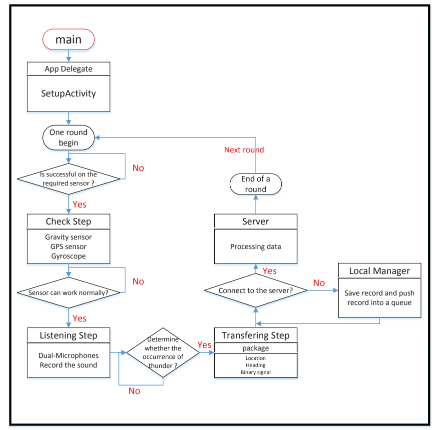


Fig. 5: The diagram of signal processing of sound signals.

1) Idle Mode: In order to determine orientation of mobile phones, the mobile device must be stationary for a period of time prior to record the sensor data. Device movement is characterized by a change in the accelerometer/gyroscope reading. If the cumulative movement is under the threshold, the device would be verified as still, then moves onto the Listening mode.

2) Listening Mode: Thunder is the acoustic shock wave resulting from a sudden and intense heating of the air in the lightning channel. The application is able to capture the shock wave by always keeping in a circular memory buffer a segment of the signal recorded before the threshold triggering of the system. 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 iThunder system, it is necessary to transmit data as soon as possible after a thunder event is determined by the application. The application records sensor readings about heading, location and immediately transmits these data to the server along with the binary measurement data estimated from two channel sound signals of dual microphones [8]. In case the device does not have communication service during the thunder event, a local copy of the recordings are stored locally, and placed into a queue to be sent once communication is available again.

Besides the implicit way to collect data, iThunder also allows user to explicitly input location, direction and binary measurement data by the way of 'human as sensor' [9], [10].

B. Basic localization in server side

In this section, the position and orientation of smartphones can be estimated by its GPS and motion sensors, respectively. 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.

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, for N reference 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 the adverse conditions. Localization system can benefit from higher performance (in terms of localization granularity) as the number of smartphones increases, such extensibility is an unique advantage compared with dedicated microphone array hardware.

Firstly, we introduce the basic localization method. Let us consider the 4G network with N dual-microphones smartphones randomly deployed in an area of size S. The top-level idea for basic localization is to split the whole localization area into some subregions identified by respective binary sequences.

Binary sensor model: We propose a binary sensor model, where the TDOA of each sensor is converted reliably to one bit of information: the object is left or right of the perpendicular bisector of dual-microphones. Using a single bit information allows for inexpensive sensing as well as minimal communication load. We use the sign of TDOA as measurement information, which we can easily distinguish the target at the left/right of the perpendicular bisector of the pair of microphone. As shown in Fig. 2, for each smartphone node, TDOA is computed by time delay estimation method firstly, then we can get the binary sequence for the target.

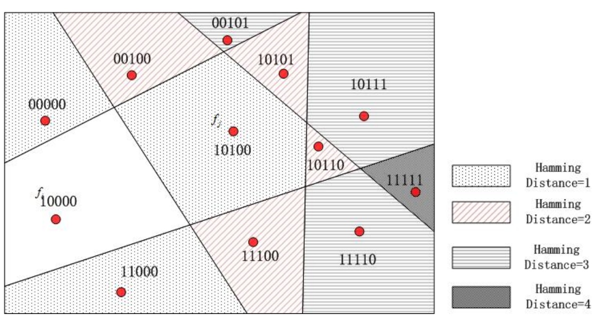


Fig. 6: Hamming distance vs. geographic distance.

Hamming distance: For two faces f_i and f_j in Fig.3, 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. From the Fig.6 we can see that the two faces are closer in geographical distance then their Hamming distance is smaller. In other words, geographical distance and Hamming distance is positively proportional function. For these two distances, we have the following observation:

$$GD(f_i, f_j) \propto HD(f_i, f_j) \tag{1}$$

Equation (1) indicates that the Hamming distance between two faces is approximately proportional with their geographical distance. This is because longer geographical distance creates

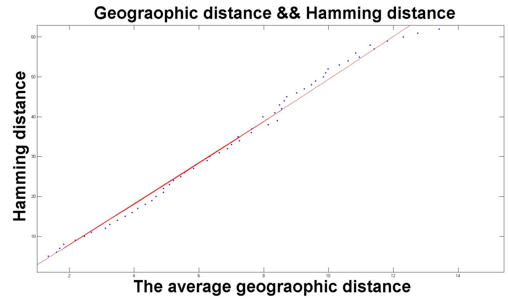


Fig. 7: Hamming distance vs. Average geographic distance.

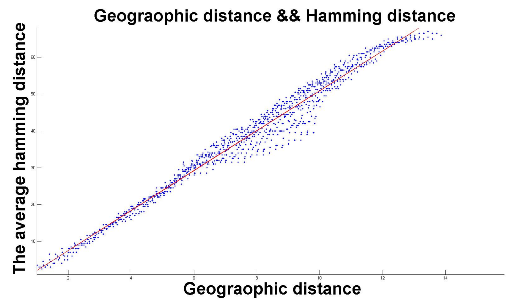


Fig. 8: Average Hamming distance vs. geographic distance.

more chances for crossing more bisectors, resulting in more flipped node pairs. 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 increases as geographical distance. 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 from the query vector. In other words, by searching the smallest Hamming distance, we can find the target location through querying in Hamming space.

The basic localization contains following three steps:

(1) Discreting the space into P points grid: Supposing there is an acoustic source at the point $p_i, i = 1, ..., P$, the $j-th$ smartphone $n_j, j = 1, ..., N$ can compute the TDOA and get the sign of the TDOA, according to the sign we can get the binary code $C_{i,j} \in \{0, 1\}$. Combining the binary code of all N smartphones, we can get a N-vector binary sequence $D_i, i = 1, ..., P$. 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 $i-th$ smartohone 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 the sign of the TDOA:

$$Binary\_data_i = \begin{cases} 1, & \text{if } \ TDOA_i \geq 0; \\ 0, & \text{if } \ TDOA_i < 0. \end{cases} \tag{2}$$

(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. Furthermore, in order to improve the robustness of the localization system, the possible coordinates for the target are computed through selecting the K smallest Hamming distance instead of the minimum Hamming distance. Finally, Centroid estimation set the center of gravity of the all possible points as the estimated location of the target node.

We can calculate time complexity according to the three steps. In step 1 we build the database, calculating TDOA of a pair of microphones and getting its sign just use the linear time, but for each grid point we need calculate TDOA of N pairs of microphones and there exists P grid points. So in step1 we will cost PN and in step 2 we just need to cost N. In step3 we search in database which will cost PN time. So the overall time complexity is $O(PN)$.

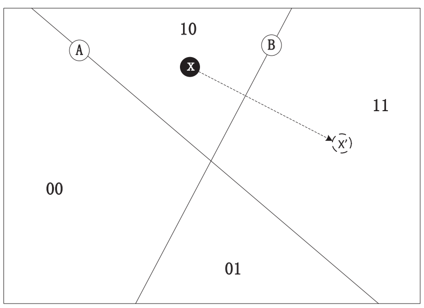


Fig. 9: Measurement error of node B.

It is worth noting that the parameter K in basic localization is hard to select. Moreover, the measurement error and parameter errors can make a big error about target location as shown in Fig. 9. Especially, if the node B is error, for example, the binary sequence about target should be $'10'$, but the error of sensor B makes the sequence into $'11'$. Fig. 10 show that if the position measurement or angle measurement of sensor B is error, there is a certain distance between actual position and measuring position. In this case, the estimated position is far away from the target. According to the above we know that the basic localization method has some problems, so we propose an robust localization solution in the next subsection.

C. Robust localization in server side

In the basic localization method, the number of neighbours K in step 3 is an important parameter for localization performance. Morover, the measurement error and node parameters uncertainty have effect on the localization performance of baisc localization. In this subsection, a robust localization method 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.

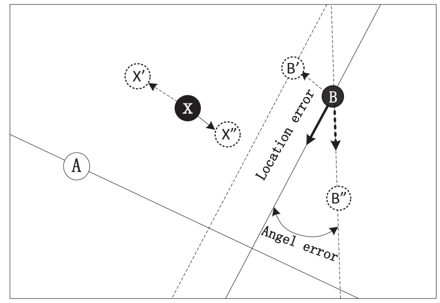


Fig. 10: Angle error and position error.

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:

$$w_i = exp(-HD(T, D_i)/2\sigma^2) \tag{3}$$

Compute the normalized weight $\bar{w_i}$, and the final estimated location of the acoustic source is the weight sum of all possible points:

$$\bar{w_i} = \frac{w_i}{\sum\limits_{i=1}^{P} w_i}, \hat{x} = \sum_{i=1}^{P} \bar{w_i} \cdot p_i \tag{4}$$

Algorithm **1** illustrates the pseudo code of robust localization method.

Algorithm 1: Robust localization method

Input: The location coordinates of all smartphones
    The direction of all the smartphones
    Information about microphones $measure\_data$
Output: Location of the Target
Step 1: Database building:
Step 2: Get the binary code T of the target from the measurement information:
Step 3: Processing the binary sequence:
(1) Computing Hamming distance for each point:
for $i \leftarrow 1$ ***to*** P **do**
    $HD(T, D(i)) = \sum\limits_{j=1}^{N} (T(j) \oplus D_i(j))$
end
(2) Computing the weight for each point using Eq. (3):
(3) Normalizing the weights and Estimating the source location using Eq. (4):

Compared with the basic localization, robust localization method leverages all possible points to locate the target. Although the computing load is increased, robust localization method has two benefits: (1) overcoming the requirement of parameter K selection; (2) 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 according to the three steps. From the algorithm we can see in step1 and step2 the time cost is the same as basic localization method. Besides, in step 3, we should calculate the normalized weighting of each point, and the corresponding time complexity is $2P$. At last step, we should calculate the location of target by using each points, and the corresponding time complexity is P. The time complexity of robust localization method is also about $O\left(PN\right)$. At the cost of more about $3P$ time complexity than basic localization method, robust localization method enhances system robustness.

IV. Discussion

A. The impact of background noises

Above, we put forward Basic ThunderLoc and Advanced ThunderLoc to implement thunder localization by using dual-microphone smartphones. However, in a practical localization system, the effect of background noises maybe occur bit flip problem when computing binary code for each smartphone, then lead to localization error.

In the presence of background noise, some of the peaks in related to reflective paths might have a magnitude that turns out to be comparable to or larger than that of the direct path. For this reason, we use the reliability index that computes the ratio between the energy in a window containing the highest peak and the energy of the remaining samples in GCC-PHAT:

$$F(i) = \frac{\sum_{\tau \in W} R^2_{m_i^1, m_i^2}(\tau)}{\sum_{\tau \notin W} R^2_{m_i^1, m_i^2}(\tau)} \tag{5}$$

where W is an interval around the highest peak in the GCC-PHAT algorithm.

We measure the reliability index at each smartphone node, then compute Hamming distance with the reliability information. When the reliability is more lower, the measurement at this smartphone is more unreliable. The new localization method is the same as robust localization method described in Algorithm **1**, the only difference is that the weighting Hamming distance is used instead of traditional Hamming distance:

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

B. System Scalability for large-scale systems

Our system is suitable for small area, but it can extend to large area. There exist some questions when we extend the system. As it is shown 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-microphones smartphones). As shown in Fig.11, we can use a range R to put the large-scale system down to a smaller area that is large enough to cover the coverage of the target'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.

C. Multiple Thunder Localization

Localizing multiple, simultaneously active sources is a more difficult problem. In order to avoid conflict of several source, multiple source localization must be able to uniquely identify the signature of each thunder, which is beyond the capability of this paper. So we will not do too much discussion about it in this paper. We just take an example, if the several thunders are set as shown in Fig. 11 then we can calculate the location of the thunders because the thunder I is far enough from thunder II and the single from them can be distinguished. So the smartphones close to effective area I could handle thunder I, and another smartphones close to effective area II could handle thunder II. The Fig.6 also describes system scalability. we can use a range R to put the large-scale system down to a smaller area that 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.

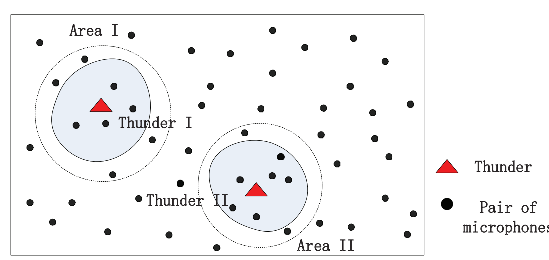


Fig. 11: System scalability for large-scale systems.

D. Time Synchronization and Energy Efficiency

In our system, we just use a pair of microphones of each smartphone to calculate the TDOA. So we just need to ensure the time synchronization on each smartphone which is easy to implement by traditional time synchronization protocol, such as RBS, TPSN, and FTSP.

Energy efficiency is another issue in smartphone networks. We can depoly some smartphones to record the sound with low sample rate, keep running when thunder may be occur, we call them demon smartphones. The demon smartphones near the thunder detect the acoustic signal, then alert its neighbour smartphones to prepare for receiving the signal.

V. RESULTS

A. 2D Simulation

In this section, we evaluated our localization methods in 2D scenario using MATLAB. In the simulation, we randomly dual-microphone mobile phones on a 10km $\times$ 10km area, the distance within a pair of microphone nodes is 14cm. Mobile phones are deployed randomly with uniform distribution. In default, there are 100 mobile phones and just one thunder. To simulate the impact of the uncertainty of the position and orientation to microphone nodes on the accuracy of localization, and with no loss of generality, all the simulations are added a certain amount of angle error and position error. The unit of the angle error range is degree, and the location error range is meter. In default, the angle error range is 5 degree and location error range is 100m. 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 (each node contains a pair of microphone), the number of fault nodes, the impact of the location error of nodes, the impact of the angle error of nodes. All the statistics are running more than 3000 times for high confidence, and reported by Root Mean Square Error (RMSE).

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 with a different number of nodes from 50 to 250 in steps of 10. We run the simulation with the measure error of TDOA is 2 sample, 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 iThunder is better than basic iThunder. 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 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 iThunder is little smaller than that of basic iThunder.

3) Impact of the location error of nodes: In the experiment, we perform the impact of the location error of nodes for the basic iThunder and robust iThunder with the range from 0 to 1000m in steps of 50. Other simulation parameters keep default. Fig. 14 shows that when the error of node location increases, the average localization error for robust iThunder is much more smaller than that of basic iThunder.

4) Impact of the angle error of nodes: In the experiment, we perform the impact of the angle error of nodes for the basic iThunder and robust iThunder with the range from 0 to 20 degree in steps of 2. Other simulation parameters keep default. As the direction of nodes are uncertain, we can guess the error of node angle 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 iThunder is much more smaller than basic iThunder, so the localization accuracy of robust iThunder is much better than that of basic iThunder.

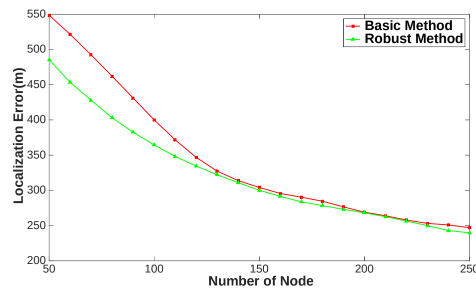


Fig. 12: Impact of number of nodes

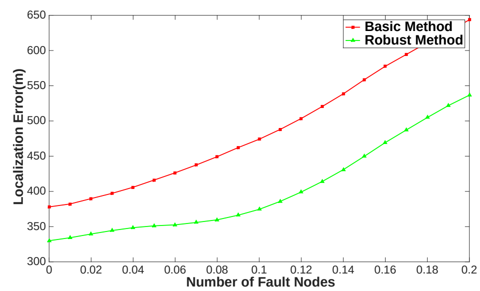


Fig. 13: Impact of number of error nodes.

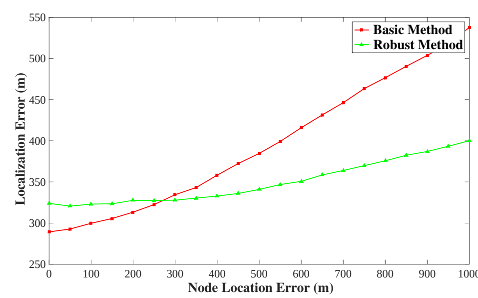


Fig. 14: Impact of the location error.

Summary: From these experiments, we can get the following conclusions:

(1) The increase of the number of nodes can improve the localization accuracy, especially when we have many participants holding the mobile phones with different postures.

(2) The increase of the number of fault nodes can effect the localization accuracy, and robust iThunder is better than basic iThunder when the percentage of fault nodes is above 0.06.

(3) The errors of node location and angle can impact the localization error, the localization performance of robust iThunder is better than basic iThunder.

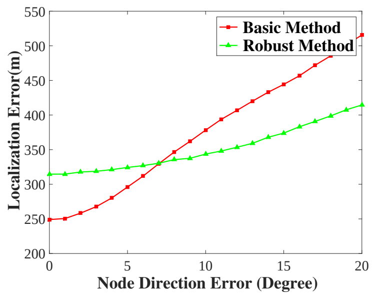


Fig. 15: Impact of angle error.

B. 3D Simulation

In this section, we evaluated our localization methods in 3D scenario. In the simulation, we randomly depolyed 100 dual-microphone phones to conver a 10km $\times$ 10km region. The height of mobile phones is about 10m and the height of thunder is about 3km. The number of node is 100 and the angle error range is 5. Due to limitations on space, we just investigate the localization error over number of nodes from 60 to 200. The localization result of the experiment is showed in Fig. 16. When the number of nodes is 120, the average localization error of robust iThunder is about 800m.

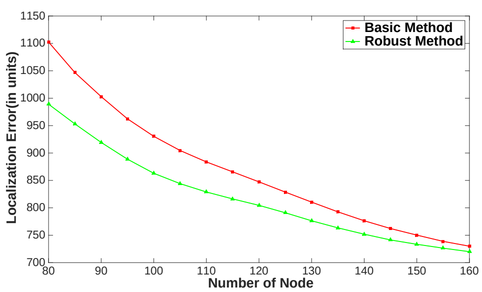


Fig. 16: Impact of number of nodes

C. Visual Thunder Emulation

we use 30 Samsung mobile phones which have two microphones as 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 14m$\times$14m space and there just one target during an experiment. In order to simulate the thunder, we broadcast the recodred sound signal of thunder by speaker at the top of the building. Mobile phones are randomly deployed around the building, and the localization result can be seen in Fig. 17. In the figure, blue squares stand for anchor nodes, black squares are the target nodes, and red circles are the estimated location by robust localization method. An arrow origins from the estimated location of each target and points to its real position. As the results showed in Fig.17, the estimated location is close to real location and the localization error is within the expected range, which means that our proposed iThunder system can effectively locate the thunder by the way of crowdsensing.

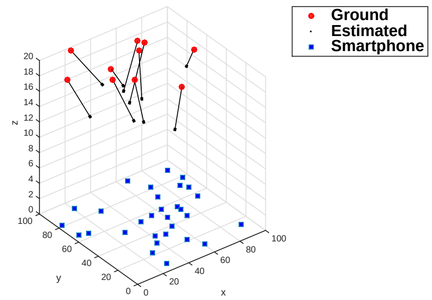


Fig. 17: Experiment result.

VI. Related Work

As a case study of acoustic source localization technology, thunder localization has drawn more attention for lightning location systems in recent years. According to the different frequency, thunder localization system can use infrasonic signals, supersonic signals, and audible signals. Due to space constraints, we can only mention a few related works using audible sound data. Audible thunder is thought to come from the expansion of the rapidly heated lightning channel. Most of the existing thunder localization system using audible sound data are based on centralized architecture. Arechiga, *et al.* [5] 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.* [11] used rhombus array with 100kS/s sample rate and 3.50m base line. Few, *et al.* [12] implemented lightning channel reconstruction using 'Y' 4-element array with 1kS/s sample rate and 30m base line. MacGorman, *et al.* [13] considered the effect of temperature and wind, presented a lightning localization system with saquare array with 0.5kS/s sample rate and 50m base line. Johnson, *et al.* [14] used a network of broadband microphones, includinga 4-element array with 1kS/s sample rate, to locate the sources of thunder occurring during an electrical storm. Qiu, *et al.* [15] used 3-element array with 100kS/s sample rate and 12.5m base line.

In the past few years, there has been a growing interest for spatial distributions of independent (unsynchronized) acoustic sensors, each made of two or more synchronized microphones. Wang, *et al.* [16] 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.* [17] 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. [18] design the system based on 2-tier architecture, which experienced cost and deployment problems especially in the very large target area. Rabbat, *et al.* [19] proposed a decentralized algorithm based on the distributed ML estimation technique using token ring architecture. Kim, *et al.* [20] proposed to identify the node closest to the acoustic source, based on TOA comparisons between all nodes, thus incurring high communication cost and requiring global synchronization between all sensor nodes. Lightning is a method proposed in [21] to identify the sensor closest to the acoustic source, also based on expensive broadcasting/flooding. Canclini, *et al.* [22] proposed a method for localizing an acoustic source with distributed microphone networks based on TDOA between microphones of the same sensor. Most of the existing acoustic source localization methods in sensor networks are based on range-based measurement. In contrast, our work is a range-free method and robust to the errors of node locations, the errors of node directions and the errors of measurements.

With the surging of smartphone sensing, wireless networking, and mobile social networking techniques, Mobile Crowdsensing has become a promising paradigm for cross-space and largescale 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].

VII. Conclusions and Future work

In this paper, we have designed iThunder, a novel thunder localization system that leverages dual-microphone smartphones to achieve thunder localization using unreliable binary node sequences. The proposed design formulates the localization problem as searching in Hamming space by making use of the binary sequence from the smartphones based crowdsensing. Since our system runs on COTS smartphones and supports spontaneous setup, it has potential to enable a wide range of thunder localization systems. Besides the basic design, a robust localization method is proposed for further enhancing system robustness. Our system is verified and evaluated through analysis, extensive simulation as well as the test-bed experimentation. Our test results have shown that the proposed method can effectively implement thunder localization with crowdsensing mechanism. Our next step is to study motivate mechanism to increase the participation rate of crowdsensing.

References

[1] A. Few, "Acoustic radiations from lightning," *Handbook of atmospheric electrodynamics*, vol. 2, pp. 1–31, 1995.

[2] M. Dayeh, N. Evans, S. Fuselier, J. Trevino, J. Ramaekers, J. Dwyer, R. Lucia, H. Rassoul, D. Kotovsky, D. Jordan *et al.*, "First images of thunder: Acoustic imaging of triggered lightning," *Geophysical Research Letters*, vol. 42, no. 14, pp. 6051–6057, 2015.

[3] 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," *Electromagnetic Compatibility, IEEE Transactions on*, vol. 51, no. 3, pp. 499–518, 2009.

[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*, vol. 2015, 2015.

[5] 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.

[6] 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, 1900.

[7] 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.

[8] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[9] D. Wang, M. T. Amin, S. Li, T. Abdelzaher, L. Kaplan, S. Gu, C. Pan, H. Liu, C. C. Aggarwal, R. Ganti *et al.*, "Using humans as sensors: An estimation-theoretic perspective," in *IEEE/ACM IPSN'14*, 2014, pp. 35–46.

[10] B. Resch, "People as sensors and collective sensing-contextual observations complementing geo-sensor network measurements," in *Progress in Location-Based Services*. Springer, 2013, pp. 391–406.

[11] 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.

[12] A. Few, "Lightning channel reconstruction from thunder measurements," *Journal of Geophysical Research*, vol. 75, no. 36, pp. 7517–7523, 1970.

[13] D. R. Macgorman, "Lightning location in a storm with strong wind shear," Ph.D. dissertation, Rice University, Houston, Texas, 1978.

[14] J. Johnson, R. Arechiga, R. Thomas, H. Edens, J. Anderson, and R. Johnson, "Imaging thunder," *Geophysical Research Letters*, vol. 38, no. 19, 2011.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] M. Rabbat, R. Nowak, and J. A. Bucklew, "Robust decentralized source localization via averaging." in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, 2005, pp. 1057–1060.

[20] Y. Kim, J. Ahn, and H. Cha, "Locating acoustic events based on large-scale sensor networks," *Sensors*, vol. 9, no. 12, pp. 9925–9944, 2009.

[21] Q. Wang, R. Zheng, A. Tirumala, X. Liu, and L. Sha, "Lightning: a hard real-time, fast, and lightweight low-end wireless sensor election protocol for acoustic event localization," *IEEE Transactions on Mobile Computing (TMC)*, vol. 7, no. 5, pp. 570–584, 2008.

[22] 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.

[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 ishake system," *IEEE Transactions on Automation Science and Engineering (TASE)*, vol. 10, no. 2, pp. 242–251, 2013.

[24] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "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.