1-Bit Compressive Sensing Based Acoustic Source Localization in Dual-Microphone Array

Xin Zhou, Naigao Jin, Mingyi Yang, Xinyang Liu, Lei Wang
School of Software, Dalian University of Technology, China
Email: ngjin@dlut.edu.cn, lei.wang@dlut.edu.cn, zhouxin964@gmail.com

***Abstract*—Acoustic source localization in sensor network is a challenging task because of severe constraints on cost, energy, and effective range of sensor devices. To overcome these limitations in existing solutions, this paper formally describes, designs, implements, and evaluates a 1-bit compressive sensing method to acoustic source localization, i.e., CS-ASL, in distributed smartphone networks. We collect the 1-bit binary left/right data by estimating the sign of TDOA between the synchronized dual microphones. The key idea behind CS-ASL is to turn the localization into sparse recovery problem of 1-bit compressive sensing. The proposed design is evaluated through extensive simulations and physical experiments in an indoor test-bed with 30 smartphone nodes. Evaluation results show that CS-ASL can effectively locate the acoustic source with good robustness.**

***Keywords*—Compressive Sensing, Acoustic Source Localization, Dual-Microphone Array, Sparse Recovery**

I. Introduction

Acoustic source localization (ASL) has extensive applications in civil and military field [1]–[3]. The common array-based method with microphones to ASL utilize several microphones to acquire multiple signals simultaneously, which have some limitations such as regard to strict clock synchronization of the multiple sensors, distances between the microphones, and concern about the large range of sense for specific applications. Wireless acoustic sensor networks (WASNs) with dual microphone are appropriate to solve these questions. In WASNs, multiple wireless microphone sensors are distributed spatially in a localization environment, which communicate by ad-hoc network. The microphone nodes can be physically deployed in a large localization area and long range limitations disappear thanks to the wireless communication. With the popularity of personal portable computing devices, the ability of sensor rapid deployment improves significantly, making acoustic source location in WASN become increasingly feasible in practical applications.

Previous literature has proposed mass of algorithms for acoustic source localization problem in sensor networks. In the last few years, research generally utilized single microphone sensors on distributed acoustic source localization system with a hard work on clock synchronization [4]. Recent years, researchers concentrated their attention on acoustic nodes with two or more synchronized microphones. Aarabi utilized the datas acquired by 10 dual-microphones distributed randomly in a room to locate three speakers [5]. Wu proposed a double acoustic source positioning method in a distributed way with three two-microphone arrays, local DOA estimates are communicated in WASN [6]. However, most of the system need special signal acquisition equipment to collect the multiple synchronized signal. With the improvement of the computing and communication technology in mobile, most mobile devices (e.g. smart phones, tablets and laptops) are equipped with multiple microphones onboard. There have existed some work that adopted the smartphone with double microphones to Snoop the Keystroke [7], [8].

Compressive sensing (CS) has already shown good application prospects in the field of localization. Zhang *et al.* [9] proposed a novel compressive sensing based approach for sparse target counting and positioning in wireless sensor networks. Feng, *et al.* [10] proposed accurate and real-time indoor positioning solutions using compressive sensing. Recently, Jacques, *et al.* [11] proposed 1-bit compressive sensing in mathematics and proposed BHIT algorithm. Yan, *et al.* [12] proposed AOP algorithm reconstructing the 1-bit signals againt bit-flipping. In this paper, we propose a new solution to the acoustic source localization by constructing an ad-hoc communicated by dual-microphone smartphones. it is desirable that only transmitt as little as possible data from multiple sensors to the sink node (processing center). We transmmit only 1-bit left/right binary code as the sensor data in WASNs. 1-bit compressive sensing is particularly attractive in this scenario, due to its capability of reducing the communication and computational costs of local sensors.

Considering that 1-bit compressive sensing is particularly suitable for resource-constrained wireless sensor networks (WSNs), we investigate the ASL from 1-bit measurements obtained by a large number of dual-microphone smartphones. ASL is modeled as the sparse recovery problem based on 1-bit compressive sensing. On the basis of the theory of 1-bit compressive sensing, we proposed some bit-flipping tolerance algorithms to solve the localization problem. To our knowledge, localization with the 1-bit compressive sensing has not been considered in the literature before. The main contributions of this article are in the followings:

(1) 1-bit data is robust to measurement error, as long as they preserve the signs of the measurements.
(2) 1-bit data could reduce the overhead in the sensor network.

The rest of the article is organized as follows. Section II presents the system model. Then, our methods are introduced in section III. Section IV presents simulation results. Finally, section V concludes the whole article.

Naigao Jin is the corresponding author.

II. SYSTEM MODEL

In our work, we use the dual-microphone smartphones to collect the acoustic signals. The binary left/right data is achieved by leveraging the sign of time-difference-of-arrival (TDOA) of the sound at the two phone microphones. Here we bring in the perpendicular bisector of the double microphones, according to the acoustical signal received by smartphone, judging the acoustical source in the left or in the right of the perpendicular bisector. We set that in the clockwise direction if acoustic source in the left we describe the measure data as 0, otherwise if in the right we define it as 1. Compared with traditional method, we use the 1-bit data instead of the concrete TDOA values, which is robust to measurement error.

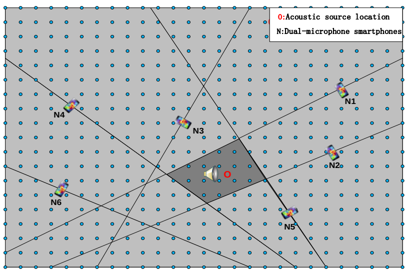


Fig. 1: System model

As is shown in Fig.1, every lines can divide the space into two parts, when the lines increase to n, the space can be divided into $\frac{n^2+n+2}{2}$ parts at best. As long as the number of the smartphones is adequate, the area of the subdomain is adequately accurate to represent the source location. Our target is searching for the points in the target subdomain using the 1-bit measure datas. We make the space gridding with large scale, transform the area into a large number of points. The points inside the target subdomain are sparse compared with the whole points, satisfy the character of compressive sampling. We regard the points in the target subdomain as the final target points, utilizing the average of these points to determine the location of the acoustical source.

Here, we talk about our 1-bit CS framework, considering the scene in the 2D space with n points and m smartphones, n is large that can present the whole localization area, all points are $S = \{point_1, \cdots, point_i, \cdots, point_n\}$, where any node $point_i$ has its location coordinate denoted as $[x_i, y_i]$. The acoustic event occurs at $O_s = [x_s, y_s]$. As for the m smartphones, we utilize the midperpendicular of the two microphones, transform them to m lines, where the $line_i$ is described as $a_i x + b_i y + c_i = 0$, stipulate a_i is a positive number all the time. And the data we measure from every smartphone can be defined as

$$D_i = \begin{cases} 1 & \text{if} \quad a_i x_s + b_i y_s + c_i \geq 0, \\ 0 & \text{if} \quad a_i x_s + b_i y_s + c_i < 0, \end{cases} \tag{1}$$

where $1 \leq i \leq m$, our target is to find few points among all the n points when substituting it to the m lines that most approaches to the D. Here, we confirm the average of the points to be the real source acoustic location.

We describe the localization model in the matrix form. Coefficient matrix A is denoted as

$$A = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ . & . & . \\ . & . & . \\ . & . & . \\ a_m & b_m & c_m \end{bmatrix} \tag{2}$$

which indicates the m lines and the point matrix is expressed as

$$S = \begin{bmatrix} x_1 & x_2 & . & . & . & x_n \\ y_2 & y_2 & . & . & . & y_n \\ 1 & 1 & . & . & . & 1 \end{bmatrix} \tag{3}$$

We represent the unknown target positions on a location grid as a sparse $n \times 1$ vector X, calculating X to determine whether $point_i$ in the target subdomain, for the points in target area, in matrix X we set their index location as 1, as for other points we give their index value 0. To sum up, we conclude our localization model to a formula

$$Sign(ASX) = D \tag{4}$$

Provided we solve the matrix X, then leading to the points in target subdomain, utilize their average value to represent the location of the acoustic source. After all concepts are put forward, we introduce the design of our localization method.

III. DESIGN

In this section, we introduce the design of CS-ASL system. The key idea behind CS-ASL is to turn the localization into sparse recovery problem of 1-bit compressive sensing. Firstly, we introduce a CS-ASL-BF method with brute force. Then we describe a CS-ASL-GD method with gradient descent. After the two methods are proposed, we do further local refinement to improve the accuracy of the localization.

A. *CS-ASL-BF with brute force*

CS-ASL-BF is based on the idea of brute force searching. Grid meshing leading to a large number of points, we checkout them one by one with brute force searching, consider whether the point we are searching for is the one in target subdomain. Every point can reach an assumed measure vector by the relative location to m lines, we believe the point with its assumed measure vector most approach to matrix D is what we search for. In our CS framework, for the $point_i$ searching in matrix X we set its index location as 1 and other location as 0, leading to an error matrix

$$error_i = sign(ASX_i) - D \tag{5}$$

where $1 \leq i \leq n$, every point of n maps an error matrix, in that matrix defines none-zero numbers $\boldsymbol{nnz_i}$ as the cost function of that point. Finally the points with the minimum of cost function are in the target subdomain and their average value is the location we ultimately determine.

Algorithm 1: CS-ASL-BF with brute force

Input: Coefficient matrix A
Point matrix S
Measure Data D

Output: Acoustic source location

Step 1: Initializing the calculated matrix $X \leftarrow 0$;
Step 2: Computing the the cost function(nnz) of the whole points:
for $i \leftarrow 1$ ***to*** n **do**
    Set the ith position of X to 1: $X_i = X(i) \leftarrow 1$;
    $error_i = sign(ASX_i) - D$;
    $nnz_i \leftarrow$ none-zero number of $error_i$;
    $X \leftarrow 0$;
end
Step 3: Find the points with the minimum nnz value.
Step 4: Determine the target location by the average of points' location in Step 3.

B. CS-ASL-GD with gradient descent

CS-ASL-BF performs extremely well in practical test. It can even obtain a high precision location when bit-flipping or other measure errors occur. However, when we get an accurate location, meanwhile the huge number of the point searchings lead to high time complexity, which may slow down our system. To strengthen our method, we put forward a CS-ASL-GD method using gradient descent to confirm the real source acoustic point instead of searching for one point by one point.

We can turn the location problems to solve the linear equation with one unknown. One classical method in signal processing fields is bringing in gradient, searching for the points with maximum probability in every iteration process by gradient descent, in result, determine the real source acoustic point in the final time. CS-ASL-GD method is based on it.

Here our estimated matrix X represents the probability of the n points in target subdomain. Each iteration, we calculate the gradient at first, then update the probability of the n points according to the step-size that gradient decreases by.

$$\varDelta = sign(ASX) - D \tag{6}$$

Eq.6 means the increment of the functional value.

$$\triangledown = sign((AS)' * \varDelta) \tag{7}$$

Eq.7 gets the gradient $\triangledown$ in Eq.4, aiming to get a solution to vector X.

$$X = X - \mu * \triangledown \tag{8}$$

Eq.8 figures out a new probability of the n points. μ present the gradient descent step, in traditional gradient descent process choosing an appropriate step-size u is essential. If that value is large, we cannot obtain a more accurate probability. Otherwise if the value is too small, the change of the probability will be unapparent. One of the advantages of our 1-bit CS framework is that we introduce the sign function, utilizing the 1-bit 0/1 information instead of concrete measure data, not only make iteration rapidly convergence but provides adequate slacks to construct gradient descent without an extremely accurate step-size. We choose the points with the maximum probability in vector X, confirming other points outsides the target area. Every iteration process excludes a large number of unnecessary points, which can rapid convergence to the area with acoustic source.

$$\hat{X}_i = \begin{cases} 1 & \text{if} \quad X_i = max(X), \\ 0 & \text{if} \quad X_i \neq max(X), \end{cases} \tag{9}$$

where $1 \leq i \leq n$, Eq.9 introduces a vector $\hat{X}$, the points with maximum probability are all set 1, consider they has the same probability to exist in target subdomain in this iteration, no more determine the source location with other points. Then we use Eq.10 to make normalization processing, guarantee vector X record the probabilities of the whole points, as input to next interation.

$$X = \hat{X} / \sum_{i=1}^{n} \hat{X}_i \tag{10}$$

Several iterations later, the probability of few points in vector X keeps a large and stable value. Eventually, we regard the index points with a nonzero value in vector X as the target points in our localization area.

However, in real situations the kinds of error especially bit flipping seriously affect the performance of the system. Since what the gradient descent method search for is optimal points in one part, when bit-flipping occurs, the points we obtain often include several points outside the target sub-region without any handling. Here we make the points gathering from gradient decease method as suspicious points, then adapt the CS-ASL-BF method to suspicious points for further searching, believe the points with nonzero value in matrix X as the final points in real localization area.

C. CS-ASL-LR with local refinement

In our 1-bit CS framework, we utilize the sampling points in target subdomain to present the acoustic source. CS-ASL-BF with brute force can find the whole points in target subdomain even occurs bit-filpping, as a result may slow down the system efficiency. CS-ASL-GD with bit fipping tolerance brings in gradient descent, although can get results fast, only determine several points in the target area not the whole points, leading to localization error in certain degree, which makes its accuracy inferior to CS-ASL-BF in high error conditions especially bit-flipping. Here we regard the CS-ASL-BF as the best accuracy standard, making more efforts to make the accuracy of the CS-ASL more approach to CS-ASL-BF with high efficiency. Aiming at above question, we have once considered finding the bit flipping nodes in the process of iterations, recover them and localize again by CS-ASL-GD method. However, CS-ASL-GD has determine a more precise location, repeat work only make the system more complicated. Owing to the area of the target subdomain is adequate small, only make further amelioration searching by CS-ASL-BF method for the points neighbor to the location determined by CS-ASL-GD method can overcome

Algorithm 2: CS-ASL-GD with gradient descent

Input: Coefficient matrix A
Point matrix S
Measure Data D
Maximum number of iterations $maxiter$

Output: Acoustic source location

Step 1: Initializing the probability vector X:
$\hat{X} \leftarrow 1$;
$X = \hat{X} / \sum_{i=1}^{n} \hat{X}_i$;
Step 2: Searching for the points with maximum probability in target subdomain:
for $i \leftarrow 1$ ***to*** $maxiter$ **do**
  $\Delta = sign(ASX) - D$;
  $\bigtriangledown = sign((AS)' * \Delta)$;
  $X = X - \mu * \bigtriangledown$;
  $index = find((X_i == max(X))$;
  $\hat{X}(1:n) = 0$;
  $\hat{X}(index) = 1$;
  Normalization processing to vector X:
  $X = \hat{X} / \sum_{i=1}^{n} \hat{X}_n$;
  $i = i + 1$;
end
Step 3: Find the points with the maximum probability as the suspicious points in target area.
Step 4: Utilize the CS-ASL-BF method for further filtrating among suspicious points. Believing the average of final filtrated points' location as the target location.

this limit without raising much complexity. Results collect more points belong to the target subdomain leading to a more accurate localization.

IV. EVALUATION

A. Simulation

In order to verify our methods, demonstrating their behaviors in different conditions, we do a great deal of Monte Carlo simulator experiments randomly that design and implement the 1-bit CS method including CS-ASL-BF, CS-ASL-GD and CS-ASL-LR using MATLAB. In our experiments, we simulate a 10m$\times$10m area with some smartphones as sensors deployed randomly and choose a point as acoustic source. Considering the uncertain conditions in real localization environment, we add different kinds of error including node location error, node angle error and the number of bit-flipping nodes aiming to show their good performances in different situations. We set the step of gradient descent μ as 0.5. Table 1 shows the major default parameter configurations in our simulations. Results are demonstrated in the followings.

TABLE I: **Default configuration parameter**

Parameter	**Description**
Field Area	10m $\times$10m
Number of Anchors	30 (Default)
Node Location Error	0.2m (Default)
Node Angle Error	5°(Default)
Number of Bit-Flipping	3 (Default)
Random-Seed Loop	1000 times

1) Impact of the number of anchors: Here we choose the the number of anchor nodes from 10 to 40 in steps of 3, other parameter set the defaut values. The results in Fig. 2(a) show that, with more node numbers, the whole area is divided into more multiple parts, and the subdomain we finally determine is more approach to the real acoustic source. CS-ASL-BF is better for brute force searching for all points, but degrade the system efficiency. We utilize its localization accuracy as the best bound, compared with other two method to demonstrate their performance. CS-ASL-LR enhances system efficiency in condition of localization accuracy guaranteed.

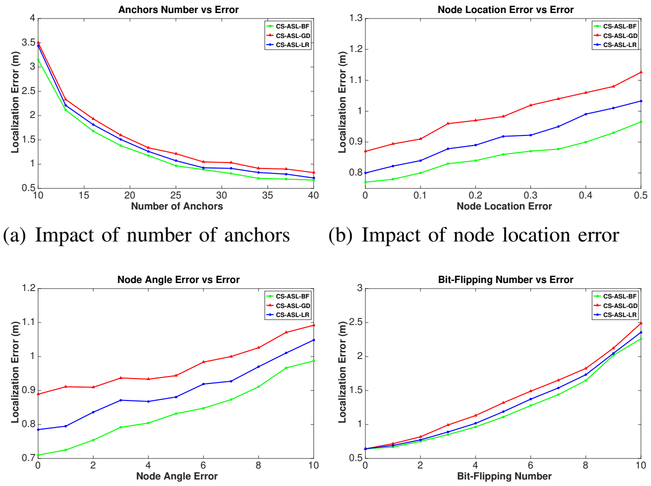
2) Impact of the node location error: In this experiment we choose the location error from 0 to 0.5m in steps of 0.05m, other parameters remain the default. With the raisement of the node location error, Fig. 2(b) shows that the localization error increases slowly, even the error add to 0.5m our methods still have a good performance.

3) Impact of the node angle error: In localization system, node angel is also an essential parameter that influences the property. No matter how the sensor capability improves, the node angle error still exists due to magnetic interference or users' different operations. Here we choose the node angle error from 0 to 10° in steps of 1°, other parameters still the default values. As is shown in Fig. 2(c), results demonstrate that our methods have a good node angle fault tolerance.

4) Impact of bit-flipping number: In WASN, bit-flipping severely influences the localization performance. To prove our methods can refine bit-fllipping, we reversal the 1-bit signals on purpose. In the condition of 30 nodes, we range the bit-flipping number from 0 to 10 in steps of 1, other parameter remain default. Result in Fig. 2(d) shows that CS-ASL-BF method performs well when bit-flipping occurs, on the basis of this, CS-ASL-GD also have a certain ability to refine bit-flipping with the idea of final partly searching. CS-ASL-LR behaves more approach to CS-ASL-BF method, even the bit-flipping number adds to 10, it can still determine the source location. Therefore, our methods are bit-flipping tolerance.

B. Testbed Experiment

In this part, we utilize the testbed experiment to evaluate the performance of our CS-ASL-LR method. We use 30 Redmi smartphones with double microphones as sensors to collect 1-bit datas and connect them through TP-LINK TL-WDR4310 wireless router. We do the experiments for 100 times, every time we deployed the 30 smartphones randomly in an area of 16m$\times$10m, results are demonstrated in Fig.3,



(a) Impact of number of anchors (b) Impact of node location error

(c) Impact of angle error (d) Impact of bit-flipping number

Fig. 2: The results of simulation.

blue squares stand for anchor nodes, red circle squares denote the real acoustic sources position and black dots are the estimated location by CS-ASL-LR. An arrow origins from the estimated location of each acoustic source and points to its real position. As shown in Fig.3, most of the estimated locations are close to the ground truth and the errors between them are very small. In our experiment, the acoustic sources got localized with average and maximum error of 0.63 feet and 2.74 feet, respectively. Fig.3 tells that the proposed CS-ASL-LR successfully accomplishes acoustic source localization.

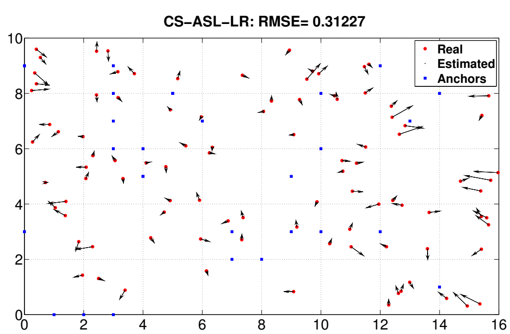


Fig. 3: Testbed experiment

Summary: Considering the parameters including the number of anchors, node location error, angle error and bit-flipping number, simulation results and testbed experiment prove that our methods have a great performance. CS-ASL-BF obtain an accuracy localization result in all fault environment, however, it searches all points every time which slows down the efficiency. CS-ASL-GD based on gradient descent, we make further searching that decrease the final localization error and guarantee the system efficiency even with bit-flipping. CS-ASL-LR takes more step to improve the localization accuracy. Above all, our methods can accomplish localization with robustness.

V. Conclusions

In this paper, we have designed and implemented CS-ASL, which leverages dual-microphone smartphone array to achieve acoustic source localization using unreliable binary node sequences. The proposed design formulates acoustic source localization as the sparse recovery problem of 1-bit compressive sensing by making use of the binary sequence from the smartphone array. Since our system runs on COTS smartphones and supports spontaneous setup, it has potential to enable a wide range of distributed acoustic sensing. Besides the CS-ASL-BF, CS-ASL-GD and CS-ASL-LR are proposed for further enhancing system robustness. Our methods are verified and evaluated through analysis and experiments. According to the results, our method can effectively implement robust acoustic source localization with light communication overhead.

References

[1] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," in *IEEE Transactions on Signal Processing*, vol. 56, no. 1, 2008, pp. 365–337.

[2] J. Sallai, A. Lédeczi, and P. Völgyesi, "Acoustic shooter localization with a minimal number of single-channel wireless sensor nodes," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2011, pp. 96–107.

[3] H. Liu, B. Yang, and C. Pang, "Multiple sound source localization based on tdoa clustering and multi-path matching pursuit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[4] A. Canclini, F. Antonacci, A. Sarti, and S. Tubaro, "Acoustic source localization with distributed asynchronous microphone networks," in *EEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, 2013, pp. 439–443.

[5] P. Aarabi, "The fusion of distributed microphone arrays for sound localization," in *EURASIP Journal on Advances in Signal Processing*, 2003, pp. 338–347.

[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] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *ACM SIGSAC Conference on Computer and Communications Security (CSS)*, 2014, pp. 453–464.

[8] 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 International Conference on Mobile Computing and Networking (MobiCom)*, 2015, pp. 142–154.

[9] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang, "Sparse target counting and localization in sensor networks based on compressive sensing," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2011, pp. 2255–2263.

[10] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Compressive sensing based positioning using rss of wlan access points," in *IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 1–9.

[11] J. N. Laska, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, "Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors," in *IEEE Transactions on Signal Processing*, 2013, pp. 3868–3875.

[12] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Robust 1-bit compressive sensing using adaptive outlier pursuit," in *IEEE Transactions on Information Theory*, 2012, pp. 2082–2102.