

RESEARCH

Open Access



# A novel density-based clustering method for effective removal of spurious intersections in bearings-only localization

Zhian Deng<sup>1,2</sup>, Zhiguo Wang<sup>1,2</sup>, Tianbao Zhang<sup>1,2</sup>, Chunjie Zhang<sup>1,2\*</sup> and Weijian Si<sup>1,2</sup>

\*Correspondence:  
zhangchunjie@hrbeu.edu.cn

<sup>1</sup> College of Information  
and communication  
Engineering, Harbin Engineering  
University, Harbin, China

<sup>2</sup> Key Laboratory of Advanced  
Marine Communication  
and Information Technology,  
Ministry of Industry  
and Information Technology,  
Harbin, China

## Abstract

In bearings-only localization, clustering-based methods have been widely used to remove spurious intersections by fusing multiple bearing measurements from different observation stations. Existing clustering methods, including fuzzy C-mean (FCM) clustering and density-based spatial clustering of applications with noise (DBSCAN), must specify the number of clusters and the threshold for defining the neighborhood density, respectively, which are always unknown and difficult to estimate. Moreover, in dense radiation source scenes, existing clustering methods for removal of spurious intersections all deteriorate significantly. Therefore, we propose a novel density-based clustering method called K-M-DBSCAN, which combines the minimum  $K$  distance algorithm with Mahalanobis distance-based DBSCAN clustering. Firstly, K-M-DBSCAN uses minimum  $K$  distance algorithm for preprocessing to remove most of the spurious intersections and reduce the computational complexity of clustering. Mahalanobis distance-based DBSCAN is used for clustering and spurious intersections recognition. In order to adapt the large variations of sample density in clustering, we use Mahalanobis distance to define an explicit neighborhood of DBSCAN instead of traditional Euclidean distance. Simulation results show that the proposed K-M-DBSCAN performs better than FCM and DBSCAN in removing of spurious intersections.

**Keywords:** Passive positioning, Mahalanobis distance, Density clustering

## 1 Introduction

Radar radiation source localization technology based on artificial intelligence and data mining has attracted more and more attentions [1–4]. Bearing only localization (BOL) is the earliest and most mature localization method in passive localization. It estimates the position of radar target by fusing multiple bearing measurements from different observation stations [5]. In passive localization system, due to the lack of distance information, BOL plays an important role in both military and civil applications, such as underwater surveillance [6–8], 3-D passive target tracking [9–11] and unmanned aerial vehicle path planning [12–15]. For multiple radar targets within the surveillance area and two or more observation stations involved in BOL, different bearing intersections without correct data association will cause serious spurious intersections problem [16–18], and the

number of spurious intersections will exponentially grow with the number of the observation stations and radar radiation sources. Recent advances have demonstrated that the clustering method can be utilized to remove the spurious target in BOL system.

Existing approaches for removal of the spurious intersections can be mainly divided into two kinds, including data association and clustering-based approaches. One of the most widely used data association approaches is using  $m$ -best assignment formulation [19] to solve a  $S$  dimension ( $S$ -D) assignment problem. By making appropriate modifications to the cost matrix and solving a series of modified copies of the initial problem, this method can be used to find the  $m$ -best (ranked) solutions to the data association problem [20, 21]. Michael Beard et al. [22] proposed two data association algorithms: Classical Bayesian Thread Association (CBTA) and Monte Carlo Thread Association (MCTA) to solve the problem of thread association in bearings-only tracking of multiple targets. These algorithms used single target bearings-only state estimators to calculate measurement likelihood. Simulation results showed that MCTA algorithm is superior to CBTA algorithm. Some researchers proposed joint probabilistic data association method and multiple hypothesis tracking method, Kirubarajan et al. [23] combined probabilistic data association (PDA) and maximum likelihood method to estimate target motion parameters through batch processing to track low observable targets in passive sonar measurement. However, this method requires a lot of calculation and is only suitable for a small number of radiation sources. Therefore, data association methods [24–26] are not robust and efficient in dealing with complex environments containing multiple targets in real time. They always suffer from the NP-hard problem and are unsuitable for real-time applications.

Another approach for removal of the spurious intersections is the clustering-based method. Two classical clustering methods including FCM clustering and DBSCAN clustering are mostly widely used. Recent related research shows that FCM method combining with a correlation objective function performs well in removal of spurious intersections [27], which allows for class overlaps based on the objective functions. A correlation-based FCM method assigns each sample with degrees of membership to multiple clusters. In order to circumvent the very high complexity of a brute-force solution to the data association problem of multiple-source localization, Reed et al. [28] used the clustering of line of bearing (LOB) to estimate target position estimates and recursively updated the position estimates by matching the LOB measurements between sensors corresponding to the same target. Campello et al. [29] proposed an improved density-based hierarchical clustering method for obtaining a “flat” partition consisting of only the most significant clusters (possibly corresponding to different density thresholds). The method provided a hierarchy of clusters from which can construct a simplified tree of significant clusters. The main drawback of FCM is the number of clusters should be specified as a priori. In order to overcome the drawback of FCM, some researchers have proposed some density-based clustering methods, such as DBSCAN. Duan et al. [30] proposed a local density-based clustering algorithm to overcome the serious defects of DBSCAN in processing the cluster with different local densities. The algorithm took the advantages of local outlier factor (LOF) to detect noise. Experiments showed that their method had better results than Ordering Points To Identify The Clustering Structure (OPTICS). Tran et al. [31] proposed an improved algorithm to overcome the problem

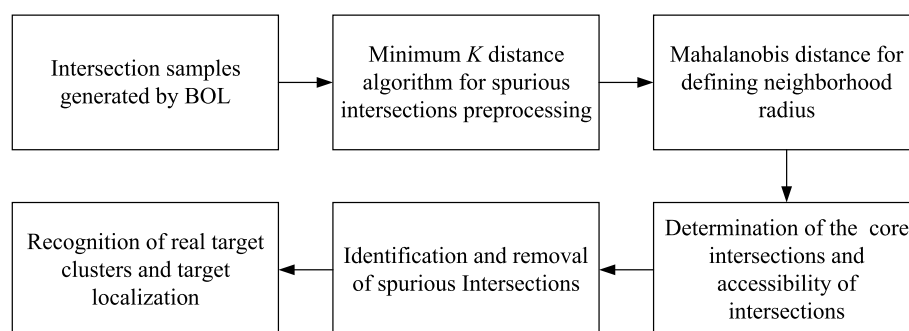
that the DBSCAN algorithm became unstable when detecting border objects of adjacent clusters. The improved DBSCAN has better robustness to data sets with connected clusters, and its clustering results have nothing to do with the order of processing objects. In density-based clustering, Euclidean distance is widely used as a measure of correlation between different samples. DBSCAN is capable of handling and identifying noise, finding clusters with arbitrary shapes, and automatically discovering the number of clusters. However, there are also drawbacks for DBSCAN to remove spurious intersections. Firstly, if the spurious intersections are not preprocessed before DBSCAN, it will increase the computational complexity and introduce much interference. More importantly, the performance may degrade significantly when large variations in sample density occur in real applications.

In order to overcome the drawbacks of DBSCAN method, we propose K-M-DBSCAN method for removal of spurious intersections. The K-M-DBSCAN has two advantages over DBSCAN. Firstly, we deploy the minimum  $K$  distance preprocessing method to reduce the interference caused by spurious intersections and also reduce the computation complexity of clustering. Secondly, M-DBSCAN uses Mahalanobis distance to define density of intersections instead of the traditional Euclidean distance. More specifically, two main parameters, including neighborhood radius and minimum number of neighborhood points, are determined based on Mahalanobis distance. Since Mahalanobis distance considers the probability distribution of intersections, M-DBSCAN has the ability to adapt large variations in intersection sample density, which is critical for effective removal of spurious intersections in a dense radiation source environment.

The remainder of this paper is organized as following. Section 2 gives the principle of BOL and problem formulation for the removal of spurious intersections and then describes the proposed clustering method, called K-M-DBSCAN. Section 3 gives the results and discussion to verify the effectiveness of the proposed method. Section 4 introduces the time complexity of the three algorithms. Section 5 includes the conclusions.

## 2 Methods

An overview of the proposed K-M-DBSCAN method is shown in Fig. 1. Solutions for the removal of spurious intersections using K-M-DBSCAN method can be divided into five steps after the intersections are generated. We first propose the minimum  $K$  distance



**Fig. 1** Overview of the proposed K-M-DBSCAN method for removal of spurious intersections

algorithm to filter out some spurious intersections and thus reduce the interference and the computational complexity of clustering. The second step is to define neighborhood radius ( $eps$ ) and calculate the density of each intersection based on Mahalanobis distance. Besides, we discuss another important parameter, minimum number of neighbor points ( $min_{pts}$ ) required. Thirdly, determine the core and accessibility of the given intersections. The core intersections are obtained if there are at least  $min_{pts}$  intersections within its  $eps$  neighborhood. Intersection accessibility is defined as being within an observed  $eps$  range, or through some dense intersections chains that are within  $eps$  distance from each other. In the fourth step, we regard the clusters formed by the core intersections and accessible intersections as the real target clusters, and the rest intersections as spurious clusters. Finally, use the measurement error of angle to update the cluster center, and achieve the real target localization.

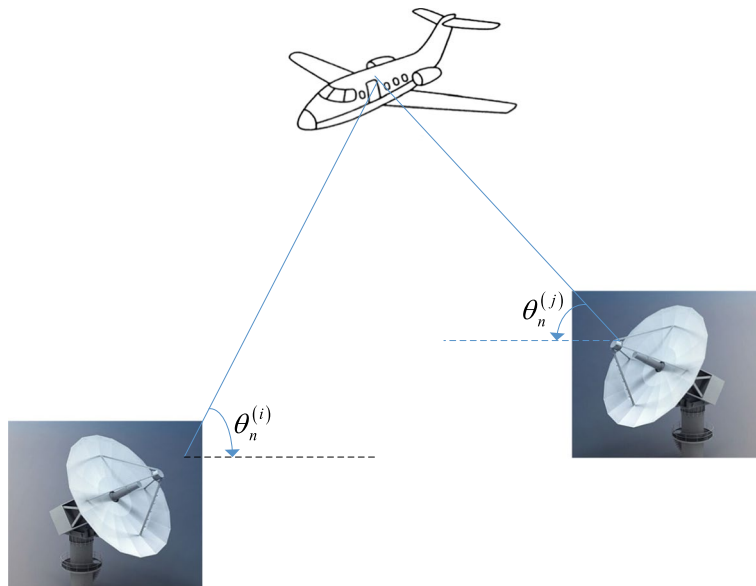
In this section, we first formulate the removal of spurious intersections problem in two-dimensional BOL and then describe the details of the proposed K-M-DBSCAN method.

### 2.1 Problem formulation

For simplicity, we describe target in two-dimensional plane with the bearing measurements received by two observing stations, as shown in Fig. 2.

The position of the radiation target is described by its two-dimensional Cartesian coordinates. The bearing angles measured by observation stations  $S_i(x_i, y_i)$  and  $S_j(x_j, y_j)$  corresponding to the target are  $\theta_n^{(i)}$  and  $\theta_n^{(j)}$ , respectively, which can determine the  $n$ -th intersection  $(x_n, y_n)$ . The target localization [32, 33] can be given as

$$\tan \theta_n^{(i)} = \frac{y_n - y_i}{x_n - x_i} \quad (1)$$



**Fig. 2** Geometry of BOL

$$\tan \theta_n^{(j)} = \frac{y_n - y_j}{x_n - x_j} \quad (2)$$

Equations (3) and (4) can be obtained through simplification:

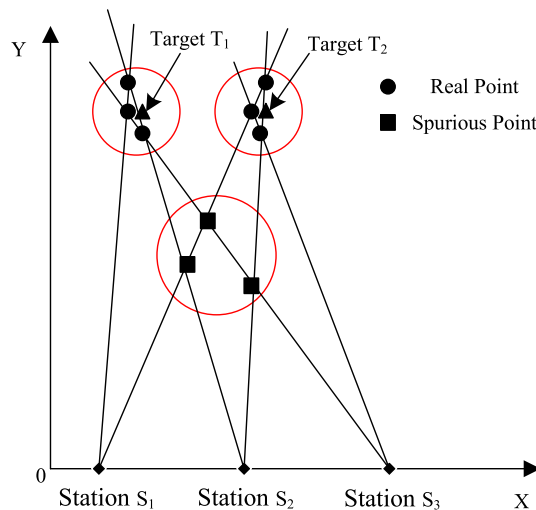
$$x_n = \frac{y_j - y_i + x_i \tan \theta_n^{(i)} - x_j \tan \theta_n^{(j)}}{\tan \theta_n^{(i)} - \tan \theta_n^{(j)}} \quad (3)$$

$$\hat{y}_n = \frac{(x_i - x_j) \tan \theta_n^{(i)} \tan \theta_n^{(j)} + y_j \tan \theta_n^{(i)} - y_i \tan \theta_n^{(j)}}{\tan \theta_n^{(i)} - \tan \theta_n^{(j)}}. \quad (4)$$

In fact, the actual BOL system involves multiple observation stations, whose bearing measurements will produce a large number of spurious intersections, as shown in Fig. 3. Assuming that there are  $s$  observing stations and  $t$  radiation sources, there are a total number of  $C_s^2 * t^2$  intersections ( $C_s^2 = s * (s - 1) / 2$ ), of which  $C_s^2 * t$  intersections are real targets and the rest are spurious intersections. It clearly shows that the density of real intersections near the real target is higher than those of the spurious targets. Therefore, the method based on density clustering is feasible to remove spurious intersections.

## 2.2 Minimum $K$ distance algorithm for preprocessing

In order to delete some spurious intersections initially and reduce the computational complexity of clustering, we deploy the minimum  $K$  distance algorithm before density-based clustering. The minimum  $K$  distance algorithm is inspired by the baseline least distance method [34]. The baseline least distance is a popular method to remove spurious intersections, which has low computational complexity and is insensitive to measurement error of angle. However, the baseline least distance method only keeps one intersection pair for each bearing based on the minimum Euclidean distance. Though a large number of spurious intersections are deleted, it also deletes many real intersections. In contrast, the minimum  $K$  distance algorithm used in proposed K-M-DBSCAN



**Fig. 3** Distribution of intersections for three observation stations and two radiation sources

algorithm retains  $K$  pairs of intersections with the least distance for each bearing. As a result, the minimum  $K$  distance algorithm filters out a considerable proportion of spurious intersections, while little real intersections are deleted.

The bearing line of the  $i$ -th observation station to the  $j$ -th target is recorded as  $L_{ij}$ , and the bearing line of the  $m$ -th observation station to the  $n$ -th target is recorded as  $L_{mn}$ , and the intersection of  $L_{ij}$  and  $L_{mn}$  is recorded as  $X_{mn,ij}$ .

Randomly choose a bearing line  $L_{ij}$  as the reference line, all intersections on the  $L_{ij}$  form a set  $\Lambda_{ij}$ . The intersections of all bearing lines of the  $m$ -th ( $m \neq i$ ) observation station and  $L_{ij}$  form a set  $S_{mn,ij}$  ( $S_{mn,ij} \subseteq \Lambda_{ij}$ ), with fixed  $i, j, m$  and  $n = 1, 2, \dots, NT$ . Then traverse all intersections  $X_{mn,ij}$  ( $n = 1, 2, \dots, NT$ ) in the set  $S_{mn,ij}$ , and calculate the minimum distance between  $X_{mn,ij}$  and the remaining points  $X$  in the set  $\Lambda_{ij}$  ( $X \in \Lambda_{ij}, X \notin S_{mn,ij}$ ). Each  $X_{mn,ij}$  will generate a minimum distance. Next sort all the minimum distances generated by  $S_{mn,ij}$ , and select  $K$  minimum distance intersections to join the set  $P$ . After traversing all bearing lines, repeating above steps, and removing the repeated intersections, the generated  $P$  is the set of intersections after preprocessing. Implementation details of minimum  $K$  distance algorithm are given in Table 1.

### 2.3 M-DBSCAN for removal of spurious intersections

#### 2.3.1 Defining neighborhood radius based on Mahalanobis distance

There are two main parameters for density-based clustering method, including neighborhood radius and minimum number of neighborhood points. The former represents the area of the neighborhood elliptic region, while the latter determines the density of the intersections. For density-based clustering, setting of neighborhood radius  $eps$  is critical for ultimate clustering results. If the value is too small,

**Table 1** Minimum  $K$  distance algorithm

---

**Algorithm 1: Minimum  $K$  Distance Algorithm**

---

**Input:** Number of observation stations  $NS$ ; Number of targets  $NT$ ; Bearing line  $L_{ij}$  ( $i = 1, 2, \dots, NS; j = 1, 2, \dots, NT$ ). Intersections set without preprocessing; Interested intersections pair  $K$ .

**Output:** Set of intersections after preprocessing  $P$ .

```

1 Mark all bearing lines as unprocessed
2 Randomly select an unprocessed bearing line  $L_{ij}$  as the reference line, all intersections on  $L_{ij}$ 
  form a set, which is record as  $\Lambda_{ij}$ 
3 Mark  $L_{ij}$  as processed
4 for all  $m, m = 1, 2, \dots, NS, m \neq i$  do
5 The set  $S_{mn,ij} = (X_{m1,ij}, X_{m2,ij}, \dots)$  of intersections formed by  $L_{ij}$  and
   $L_{mn}$  ( $n = 1, 2, \dots, NT$ )
6 for all  $X_{mn,ij} \in S_{mn,ij}$  do
7 Calculate the distance  $d_{mn,ij}$  between point  $X_{mn,ij}$  and all other intersections  $X$ 
  ( $X \in \Lambda_{ij}, X \notin S_{mn,ij}$ ),  $d_{mn,ij} = \min \{ \|X_{mn,ij} - X\| \}$ 
8 end
9 Each loop in step 6 will generate a  $d_{mn,ij}$ , sort all  $d_{mn,ij}$  generated in step 6, then select
  the first  $K$  pairs of intersections with the smallest distance  $d_{mn,ij}$  to add to  $P$ 
10 end
11 Repeat steps 2-10, until there are no unprocessed bearings
12 Remove the repeated intersections in  $P$ 
13 Output  $P$ 

```

---

intersections of the same cluster will be divided into multiple clusters, while too large value will make multiple clusters merged into one.

For neighborhood radius setting, the distance measurement criterion is a key affecting factor. Mahalanobis distance criterion is effective to measure the similarity of two unknown sample sets and it can eliminate the interference of correlation between variables. Compared with traditional Euclidean distance criterion, we construct Mahalanobis distance criterion based on the probability distribution of intersections, which adapt the large variations in sample density. Since the measurement error of angle, positions of the observation stations and radiation sources are all considered, so the constructed Mahalanobis distance to calculate  $eps$  is more robust and effective to adapt the variations in the environment and measurement error of angle.

In order to define the neighborhood radius, we firstly give the derivation of the covariance matrix of localization error. Secondly, we use the covariance matrix to calculate the Mahalanobis distance and define an explicit neighborhood. Finally, we derive neighborhood radius based on the spatial probability density distribution of the intersections.

In passive localization systems, the localization error covariance matrix is an important indicator of localization accuracy. Assuming that the measurement error of angle following a Gaussian distribution with zero mean, the localization error equation can be derived from Equations (1) and (2),

$$\begin{bmatrix} d\theta_n^{(i)} \\ d\theta_n^{(j)} \end{bmatrix} = \begin{bmatrix} -\frac{\sin^2 \theta_n^{(i)}}{y_n - y_i} & \frac{\cos^2 \theta_n^{(i)}}{x_n - x_i} \\ -\frac{\sin^2 \theta_n^{(j)}}{y_n - y_j} & \frac{\cos^2 \theta_n^{(j)}}{x_n - x_j} \end{bmatrix} \begin{bmatrix} dx_n \\ dy_n \end{bmatrix} + \begin{bmatrix} k_{\theta_n^{(i)}} \\ k_{\theta_n^{(j)}} \end{bmatrix} \quad (5)$$

Among them,  $\theta_n^{(i)}$  and  $\theta_n^{(j)}$  are the bearings of the observation station  $S_i(x_i, y_i)$  and station  $S_j(x_j, y_j)$  to the  $n$  target, respectively, and  $d\theta_n^{(i)}$ ,  $d\theta_n^{(j)}$  represent the angle measurement error of the observation station  $S_i$  and  $S_j$ , respectively. In addition,

$$k_{\theta_n^{(i)}} = \frac{\sin^2 \theta_n^{(i)}}{y_n - y_i} dx_i - \frac{\cos^2 \theta_n^{(i)}}{x_n - x_i} dy_i \quad (6)$$

$$k_{\theta_n^{(j)}} = \frac{\sin^2 \theta_n^{(j)}}{y_n - y_j} dx_j - \frac{\cos^2 \theta_n^{(j)}}{x_n - x_j} dy_j \quad (7)$$

The localization error equation is further expressed as

$$dV = CdX_t + dX_s \quad (8)$$

where  $dV = [d\theta_n^{(i)}, d\theta_n^{(j)}]^T$  is the observation error vector,  $dX_t = [dx_n, dy_n]^T$  is the target localization error vector, and  $dX_s = [k_{\theta_n^{(i)}}, k_{\theta_n^{(j)}}]^T$  is the site error vector. The coefficient matrix  $C$  can be expressed as

$$C = \begin{bmatrix} -\frac{\sin^2 \theta_n^{(i)}}{y_n - y_i} & \frac{\cos^2 \theta_n^{(i)}}{x_n - x_i} \\ -\frac{\sin^2 \theta_n^{(j)}}{y_n - y_j} & \frac{\cos^2 \theta_n^{(j)}}{x_n - x_j} \end{bmatrix} \quad (9)$$

The covariance matrix of localization error is

$$\Sigma = E \left[ dX_t dX_t^T \right] = C^{-1} \left\{ E \left[ dV dV^T \right] + E \left[ dX_s dX_s^T \right] \right\} C^{-T} \quad (10)$$

Among them,

$$E \left[ dV dV^T \right] = \text{diag} \left[ e_{\theta_n^{(i)}}^2, e_{\theta_n^{(j)}}^2 \right] \quad (11)$$

$$E \left[ dX_s dX_s^T \right] = \text{diag} \left[ \frac{e_s^2}{\sqrt{(x_n - x_i)^2 + (y_n - y_i)^2}}, \frac{e_s^2}{\sqrt{(x_n - x_j)^2 + (y_n - y_j)^2}} \right] \quad (12)$$

where  $e_s$  represents the standard deviation of the site error. The covariance matrix of localization error is

$$\Sigma = \begin{bmatrix} e_x^2 & e_{xy} \\ e_{xy} & e_y^2 \end{bmatrix} \quad (13)$$

The Mahalanobis distance is defined according to the covariance matrix of localization error, which can be expressed as the following formula:

$$r = \sqrt{[\mathbf{v} - \boldsymbol{\mu}]^T \Sigma^{-1} [\mathbf{v} - \boldsymbol{\mu}]} \quad (14)$$

where  $\mathbf{v}$  represents two-dimensional coordination vector of a certain intersection, the mean vector  $\boldsymbol{\mu}$  is the coordination vector of the assumed real intersection, and  $\Sigma$  represents the covariance matrix of the localization error.

Assuming that a certain intersection is the real target, Mahalanobis distance can define an elliptical neighborhood of the intersection. We construct the spatial probability density distribution function of intersections by Mahalanobis distance. The distribution function of spatial probability density can be expressed as

$$p(x) = (2\pi)^{-D/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2[\mathbf{v} - \boldsymbol{\mu}]^T \Sigma^{-1} [\mathbf{v} - \boldsymbol{\mu}]} \right\} \quad (15)$$

where  $D$  is the Gaussian distribution dimension, the two-dimensional vector  $D = 2$ . And the probability density distribution function can be rewritten as

$$p(r) = \frac{1}{2\pi} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2r^2} \right\} \quad (16)$$

Then, we can define the neighborhood radius by the probability density function. If the probability that a certain intersection falls outside the ellipse neighborhood is  $\beta$ , the neighborhood radius  $r_0$  can be calculated through the spatial probability distribution. According to the distribution function of probability density, the probability  $P$  is equivalent to doing a double integral in the ellipse neighborhood ( $\frac{x^2}{a^2} + \frac{y^2}{b^2} = r^2$ ),

$$P = \int_0^{2\pi} \int_0^{r_0} \frac{1}{2\pi} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2r^2} \right\} (abr) d\theta dr \quad (17)$$

where the neighborhood radius is  $r_0 = \sqrt{-2 \log(1 - \beta)}$  when  $\beta(0 < \beta < 1)$  is fixed.



Minimum number of neighborhood points  $min_{pts}$  is another crucial parameter of density clustering, which has a great influence on clustering results. We deploy the mathematical expectation based on self-decay term [35, 36] to automatically determine the  $min_{pts}$  parameter, and the expression is given as

$$MinPts = \left\lceil \frac{1 - \lambda}{n} \right\rceil \sum_{i=1}^n P_i \quad (18)$$

where  $\lambda (0 \leq \lambda \leq 1)$  is the self-attenuation coefficient. In this paper, the value of  $\lambda$  is between 0.1 and 0.3,  $n$  is the total number of intersections (including all real and spurious intersections remained after preprocessing), and  $P_i$  is the number of intersections in ellipse neighborhood.

### 2.3.2 Implementation of M-DBSCAN

M-DBSCAN algorithm does not require prior information about the number of clusters. Moreover, two important parameters including  $min_{pts}$  and  $eps$  can be well obtained according to the algorithm mentioned above. We construct the Mahalanobis distance criterion to measure the distance between intersections, which enables the adaption of large variations in cluster density and measurement error of angle.

M-DBSCAN can be described with respect to the directed  $eps$  neighborhood graph, which is designed to discover clusters of arbitrary shape with a fixed  $eps$  and a density threshold  $min_{pts}$ . Some concepts and terms to explain M-DBSCAN can be defined as follows:

**Definition 1** An intersection  $p$  is a core intersection which is defined as intersections whose  $eps$  neighborhood size is greater than or equal to  $min_{pts}$ .

**Definition 2** An intersection  $p$  is accessible intersection which is defined as being within an observed  $eps$  range, or through some dense intersections chains that are within  $eps$  distance from each other.

**Definition 3** An intersection  $p$  is noise if it is neither a core intersection nor an accessible intersection. This means that noise is not part of any clusters.

Implementation of proposed M-DBSCAN algorithm is given in Table 2. First, we randomly choose an intersection  $p$  and traverse all points in its  $eps$  neighborhood. If  $p$  is a core intersections, it will be marked as a new cluster. This cluster is extended by retrieving all accessible intersections corresponding to  $p$  and further merges these intersections into the same cluster. This process is repeated until no clusters are found which means all intersections are observed. Another possibility is that  $p$  is an accessible intersection grouped into other clusters. If the density of  $p$  is less than  $min_{pts}$ , it will be marked as noise.

## 2.4 Target localization

Based on the previous section, we can distinguish real target clusters and spurious target clusters. Target localization estimation is based on real target clusters. We first

**Table 2** M-DBSCAN algorithm**Algorithm 2:M-DBSCAN algorithm**

**Input:** A data set  $X$  containing  $n$  intersections, neighborhood radius  $eps$ , minimum number  $min_{pts}$ , Mahalanobis distance matrix  $D$ .

**Output:** Clustering set  $C$ , noise parameter  $is\ noise$ .

```

1 Mark all intersections as untraversed
2 Randomly select an unobserved intersection  $p$ 
3 Mark point  $p$  is processed
4 If the  $eps$  neighborhood of  $p$  has at least  $min_{pts}$  intersections (core intersections) based on  $D$ 
5 Create a new cluster  $C$  and add  $p$  to  $C$ 
6 Create a set  $N$  that contains all intersections in the  $eps$  neighborhood of  $p$ 
7 for all  $q \in N$  do
8 If  $q$  is unprocessed
9 Mark  $q$  as processed
10 If the  $eps$  neighborhood of  $q$  has at least  $min_{pts}$  points, add these intersections to  $N$ 
11 If  $q$  is not yet a member of any cluster, add  $q$  to  $C$ 
12 end
13 Output  $C$ 
14 Otherwise, mark  $p$  as noise (spurious target cluster)
15 Until there are no unobserved intersection

```

calculate the cluster center, but it cannot be directly used as the localization result due to some intersections with large errors. Hence, the measurement error of angle is used to update the cluster center and achieve the real target localization.

Suppose that there are a total of  $w$  real target clusters. That is, the  $M$  intersections are divided into  $w$  categories  $\{C_1, C_2, \dots, C_w\}$ , and each type  $C_i$  contains a lot of intersections. The cluster center can be given as follows [37, 38]

$$(\tilde{x}_i, \tilde{y}_i) = 1/\text{card}(C_i) \sum_{e=1}^M u_{ie}(x_e, y_e), i = 1, 2, \dots, w \quad (19)$$

where  $\text{card}(C_i)$  represents the total number of samples in the cluster, and  $u_{ie}$  is used to represent whether the intersection  $(x_e, y_e)$  is in  $C_i$ . And  $u_{ie}$  is given by

$$u_{ie} = \begin{cases} 1, & (x_e, y_e) \in C_i \\ 0, & (x_e, y_e) \notin C_i \end{cases} \quad (20)$$

However, cluster  $C_i$  may contain noisy intersections with large errors. Measurement error of angle is used to eliminate the noisy intersections and then update the cluster center. The implementation process is given as follows.

First, according to the localization equation, the angle  $\tilde{\theta}$  between the estimated localization  $(\tilde{x}_i, \tilde{y}_i)$  and the observation station  $(\tilde{x}_i, \tilde{y}_i)$  can be calculated with

$$\tilde{\theta} = \arctan \left( \frac{\tilde{y}_i - y}{\tilde{x}_i - x} \right) \quad (21)$$

So  $\theta_{ie}$  between the intersection  $(x_{ie}, y_{ie})$  and the observation station  $(\tilde{x}_i, \tilde{y}_i)$  in the cluster can be expressed as

$$\theta_{ie} = \arctan \left( \frac{\tilde{y}_i - y_{ie}}{\tilde{x}_i - x_{ie}} \right), i = 1, 2, \dots, k; e = 1, 2, \dots, M \quad (22)$$

Secondly, compare the calculated  $\tilde{\theta}$  with  $\theta_{ie}$ , if the absolute value of the difference is greater than the measurement error of angle  $\sigma$ , then remove  $(x_{ie}, y_{ie})$  from this cluster. We can use the formula to express as

$$\begin{cases} |\tilde{\theta} - \theta_{ie}| > \sigma, \text{remove}(x_{ie}, y_{ie}) \\ |\tilde{\theta} - \theta_{ie}| \leq \sigma, \text{retain}(x_{ie}, y_{ie}) \end{cases} \quad (23)$$

Finally, update the intersections in the cluster, and calculate the cluster center again according to formula (17), and use it as the target localization result. This algorithm can recognize and delete the noisy intersection samples of the cluster and improve the localization accuracy.

### 3 Results and discussion

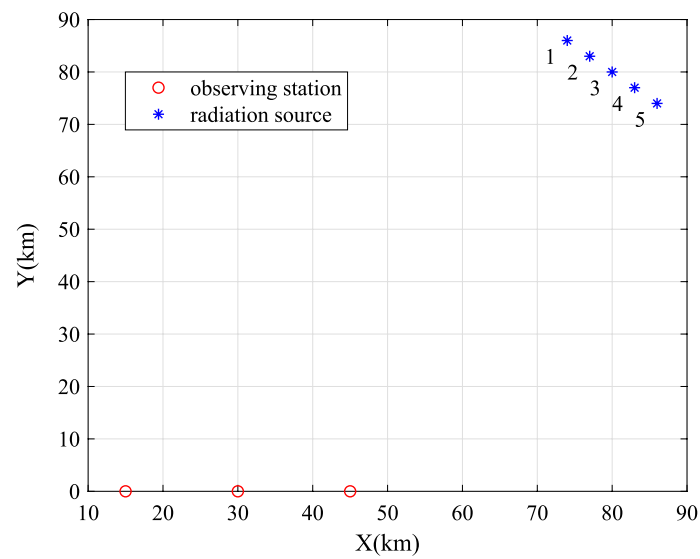
In this section, performance of the proposed K-M-DBSCAN is compared with DBSCAN and FCM algorithm via Monte Carlo (MC) simulations. The localization precision threshold mentioned in this section is defined as a localization error distance threshold. The localization error distance is defined as Euclidean distance between the estimated target localization results and the true target positions. We consider target localization result of a cluster is a real intersection if localization error distance is less than the localization precision threshold. Otherwise, the target localization result of a cluster is a spurious intersection. We adopt the recall and precision as the performance metrics. Precision is the fraction of relevant instances among the retrieved instances, while recall is the fraction of relevant instances that were retrieved. When clustering returns  $N_i$  clusters that we estimate via  $i$  th MC simulation, only  $R_i$  of which are relevant, while failing to return  $T_i$  additional relevant clusters. We can define the recall and precision obtained from  $M = 10000$  MC simulations as follows:

$$\text{recall} = \frac{\sum_{i=1}^M R_i}{\sum_{i=1}^M (T_i + R_i)}, \text{precision} = \frac{\sum_{i=1}^M R_i}{\sum_{i=1}^M N_i} \quad (24)$$

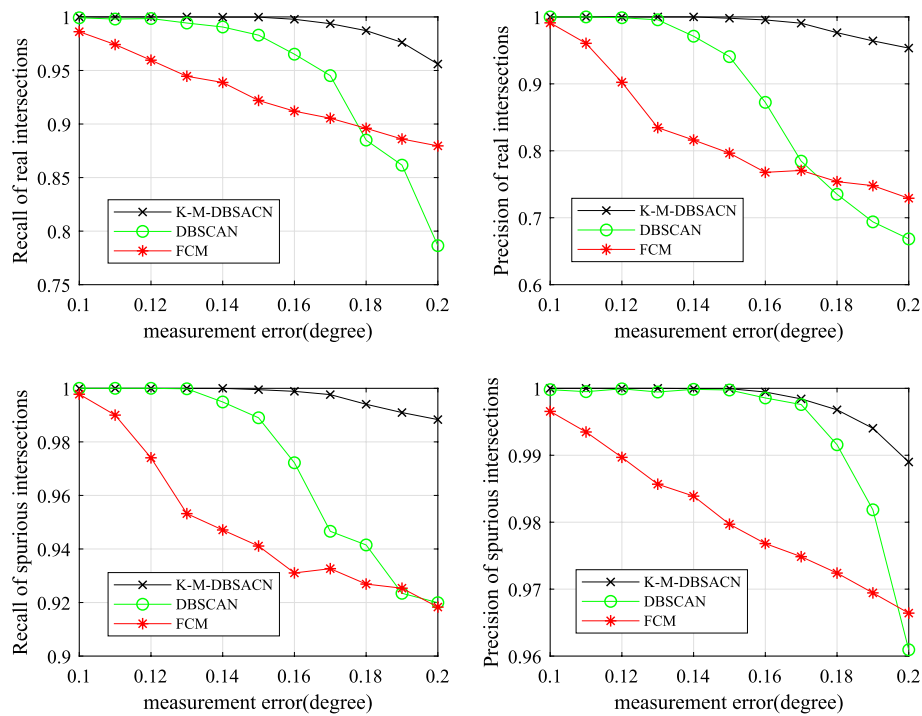
Four scenarios are carried and discussed to demonstrate the superiority of the proposed algorithm. Scenario 1 discusses dense radiation source; Scenario 2 proves that our algorithm adapts variations in sample density caused by coexistence of far and near radiation sources due to the contribution of Mahalanobis distance criterion. Variation in measurement error of angle is also taken into account in scenario 3. In scenario 4, variation in observation station layout is discussed to prove the robustness and adaptability of the proposed algorithm.

#### 3.1 Scenario 1: dense radiation source

The distribution of radiation sources and observation stations on the two-dimensional plane is shown in Fig. 4. The coordinates of five radiation source are (86,74), (83,77), (80,80), (77,83), (74,86) with the unit km. The coordinates of three observation stations



**Fig. 4** Coordinates of observation stations and radiation sources



**Fig. 5** Comparison of recall and precision versus the measurement error of angle

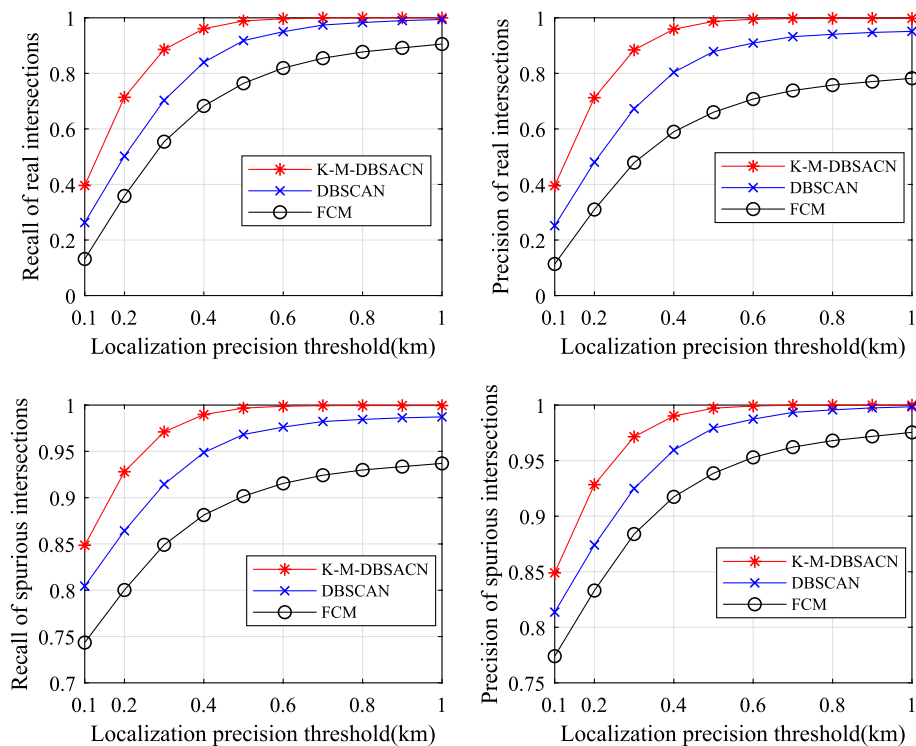
are (15,0), (30,0), (45,0), and assume that each station measure five bearing samples each time.

Figure 5 compares recall and precision performance obtained by K-M-DBSCAN, DBSCAN and FCM, respectively. When the measurement error of angle is 0.2 degrees, the recall and precision of spurious intersections in K-M-DBSCAN can still reach more

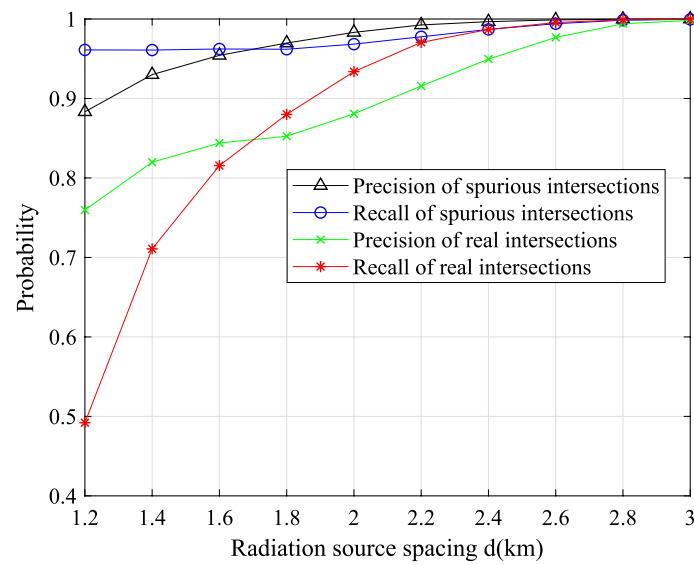
than 98.5%, while recall and precision of real intersections can reach more than 95%. In contrast, the precision of real intersections in DBSCAN is less than 70%, while FCM is less than 75%. When the measurement error of angle is 0.14 degrees, the proposed algorithm can delete all spurious intersections and identify all real targets. In contrast, the precision in K-M-DBSCAN of real intersections is 0.5% and 15% higher than those of DBSCAN and FCM, respectively. The performance of the K-M-DBSCAN in removal of spurious intersections is significantly better than the other two compared algorithms.

Figure 6 compares localization precision threshold of different clustering when measurement error of angle is 0.15 degrees. As the localization precision threshold decreases, the performance of all the three algorithms gradually increases. When the localization precision threshold is 0.6 km, the four performance indicators of the K-M-DBSCAN are all close to 100%. Taking precision of real intersections as an example, precision of K-M-DBSCAN is higher than DBSCAN by nearly 10% and higher than FCM by about 30%.

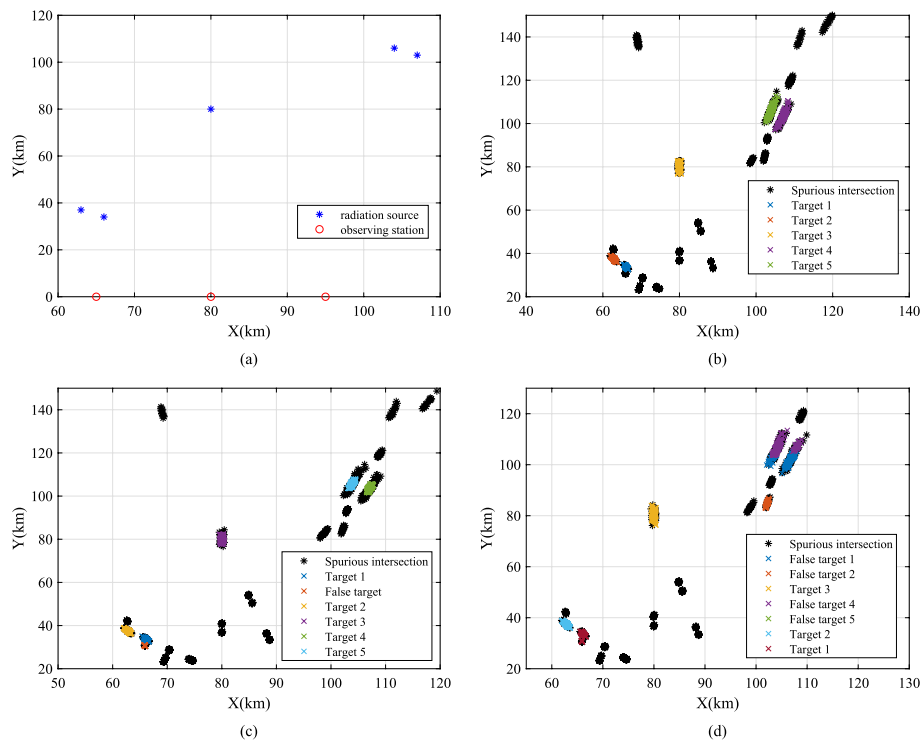
In order to evaluate effects of the radiation source spacing on the removal of spurious intersections and localization performance, we set the coordinates of five radiation source to  $(80 + 2 * d, 80 - 2 * d)$ ,  $(80 + d, 80 - d)$ ,  $(80 - d, 80 + d)$  and  $(80, 80)$ , respectively, with  $d = 1.6 : 0.2 : 3$  (km). Figure 7 plots four performance indicators of the proposed K-M-DBSCAN. When  $d=2.2$  km, both the recall and precision can reach more than 90%; when  $d=2.6$  km, both the recall and precision can reach more than 96%. It is worth mentioning that the distance between the observation station and the radiation source is about 100 km. When  $d=2.2$  km, the proposed algorithm can perform well. We can find that, although the radiation source is relatively dense, the proposed algorithm can still have superior performance.



**Fig. 6** Comparison of recall and precision versus the localization precision threshold



**Fig. 7** Comparison of recall and precision versus radiation source spacing



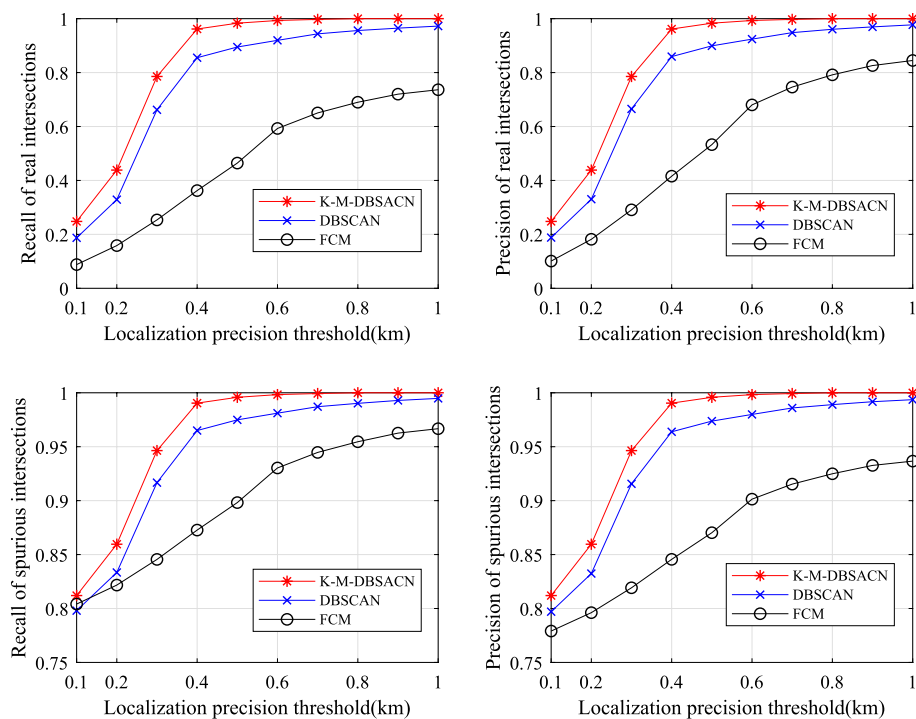
**Fig. 8** **a** Distribution of observation stations and radiation sources; **b** clustering results of K-M-DBSCAN; **c** clustering results of DBSCAN; **d** clustering results of FCM

### 3.2 Scenario 2: coexistence of far and near radiation sources

Change the position of radiation sources and observing stations, and the coexistence of far and near radiation sources is shown in Fig. 8a. In this scenario, the identification of distant targets is more difficult than nearby targets due to the scattered

intersections. The intersections of nearby targets are densely distributed, which makes spurious clusters to be confused with each other. According to the clustering results in Fig. 8b–d, the K-M-DBSCAN can correctly identify all five target clusters including almost all real intersections. Although the DBSCAN algorithm also recognizes five targets, each target cluster only contains part of the real intersections, such as target 4 and target 5. Obviously, the performance of DBSCAN and FCM becomes worse when near and far radiation sources coexist, but the proposed K-M-DBSCAN almost avoids this problem. Because Mahalanobis distance takes into account the probability distribution of real intersections, the clusters generated by K-M-DBSCAN are elliptic and adaptively include almost all true intersections. In contrast, the clusters of DBSCAN are circular based on Euclidean distance and cannot adapt well the density variations caused by coexistence of far and near radiation sources.

The recall and precision of the three algorithms are given in Fig. 9. When the localization precision threshold is 0.5 km, the recall and precision of spurious intersections in the K-M-DBSCAN can reach more than 99%, while DBSCAN is lower than 97%. The recall and precision of real intersections in the K-M-DBSCAN reach about 97%, while those of DBSCAN can only reach about 87%. The performance of FCM is significantly worse than that of DBSCAN algorithm. Simulations verify that the proposed algorithm has superior performance in an environment where far and near radiation sources coexist. This can be contributed that the constructed Mahalanobis Distance-based neighborhood radius definition adapts variation in intersections density, which is caused by the coexistence of far and near radiation sources.



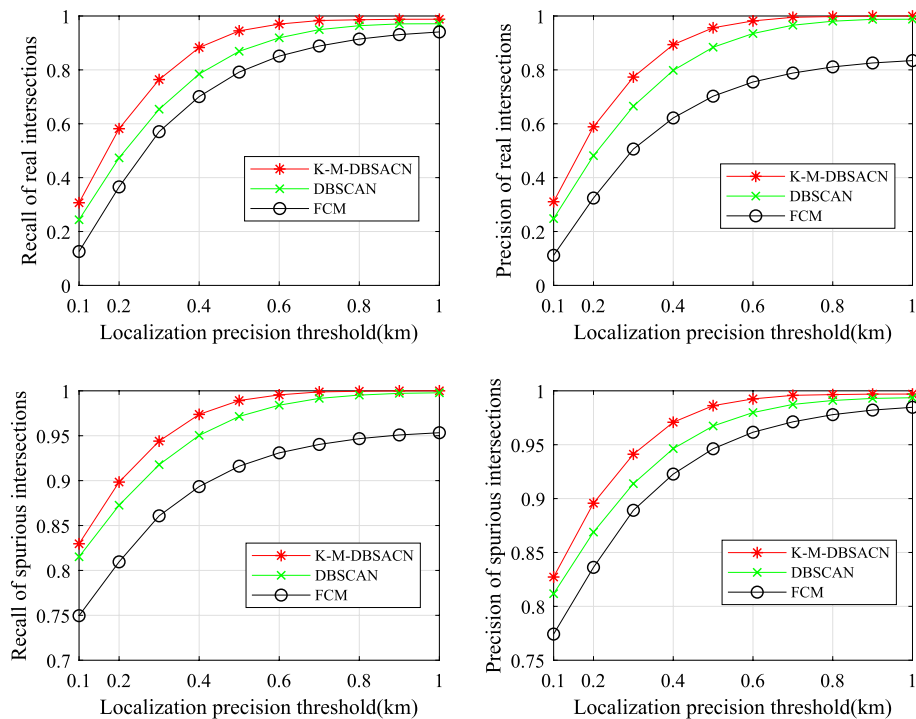
**Fig. 9** Comparison of recall and precision versus the localization precision threshold

### 3.3 Scenario 3: variation in measurement error of angle

In this subsection, we deploy the radiation source and observation station according to Fig. 4, and the measurement errors of three observation stations are 0.25, 0.10 and 0.10 degrees, respectively. Variation in measurement error of observation stations will result in different intersection densities in different directions. According to the previous statement, Mahalanobis distance is more suitable for dealing with density variations by following the probability distribution of intersections. As shown in Fig. 10, when the localization precision threshold is 0.5 km, the recall and precision of spurious intersections in the K-M-DBSCAN can reach more than 98%, and the recall and precision of real intersections can reach 95%; for FCM, when the localization precision threshold is 0.5 km, the recall and precision of spurious intersections are lower than 95%, and the recall and precision of real intersections are lower than 80%. These prove the superiority of Mahalanobis distance in K-M-DBSCAN, which can better deal with large variations in cluster density. With the increase in localization precision threshold, K-M-DBSCAN curve and DBSCAN curve become close to each other, while FCM curve is different from them. It can be seen that the increase in localization precision threshold can hardly overcome the defect of FCM, whose clustering number needs to be estimated and corrected. Compared with the DBSCAN and the FCM, our algorithm overcomes large variations in cluster density caused by different measurement error of angle.

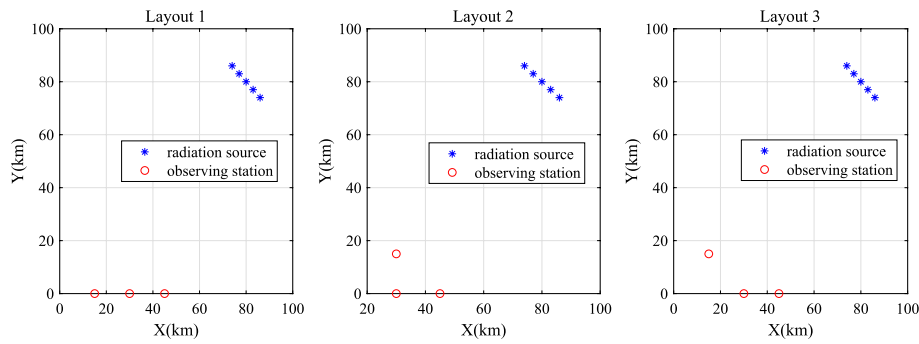
### 3.4 Scenario 4: variation in observation station layout

The Cartesian coordinates of the observation stations in the first three scenarios remain fixed. In order to better verify the robustness and adaptability of our algorithm, three

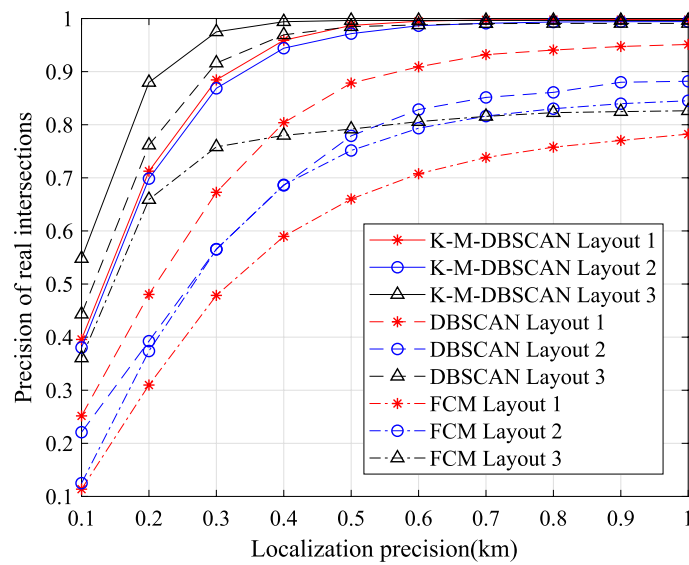


**Fig. 10** Comparison of recall and precision versus the localization precision threshold





**Fig. 11** Distribution of observation stations in three ways



**Fig. 12** Change of the precision of real intersections with localization precision threshold

different observation station layouts are given and compared, as shown in Fig. 11. For layout 1, the coordinates of the observation station are (15,0), (30,0) and (45,0); for layout 2, the coordinates of the observation station are (30,15), (30,0) and (45,0); for layout 3, the coordinates of the observation station are (15,15), (30,0) and (45,0), respectively.

The shape of the intersection distribution is different for different layouts, and this is a phenomenon of large variations in cluster density. The K-M-DBSCAN algorithm is still effective in solving this problem as shown in Fig. 12. When the measurement error of angle is 0.15 degrees and the localization precision threshold is 0.5 km for K-M-DBSCAN, the precision of the three station layouts can reach more than 96%, and the difference between them is less than 5%; when the localization precision threshold is 0.4 km, the precision of layout 3 is close to 98%, and the other two layouts also exceed 93%. When the localization precision threshold of DBSCAN is 0.5 km, only the layout 3 has a precision of more than 96%, and the other two algorithms are less than 90%. The difference between layout 1 and layout 2 is close to 10%; when the localization precision threshold is 0.4 km, the precision of layout 3 is about 96%, and the other two layouts are both less than 80%. When localization precision threshold

of FCM is 0.5 km, the precision is all less than 80%. These results suggest that the performance of the proposed algorithm is better than the FCM and DBSCAN in the same layout. Moreover, K-M-DBSCAN can better adapt to the variations in the layout of observation stations.

#### 4 Time complexity analysis

In Sect. 3, we compared the localization performance of three algorithms: K-M-DBSCAN, DBSCAN and FCM. The following is a comparison of the calculation time of a single location estimation of the three positioning algorithms.

Let the number of observation stations is  $NS$  and the number of targets is  $NT$ . The number of bearing lines is  $NS * NT$ . The total number of intersections is  $C_{NS}^2 * NT^2 = \frac{NS(NS-1)}{2} NT^2$ .

First analyze the minimum  $K$  distance algorithm. From Table 1, we can get that the number of cycles in step 4 is  $NS - 1$ . The maximum cycles in step 6 is  $NT$ , and the number of cycles in step 11 is  $NS * NT$ , so the time complexity of the algorithm is about  $O[(NS - 1) * NT * NS * NT] = O[NS(NS - 1)NT^2]$ .

Preprocess retains  $K$  pairs of intersections for each bearing. Therefore, there are at most  $K * NS * NT$  intersections in the input of M-DBSCAN algorithm. From Table 2, we can get the number of cycles in step 7 is at least  $Min_{pts}$  (Minimum neighbor points of the cluster). And the number of cycles in step 15 is  $K * NS * NT$ . The time complexity of the M-DBSCAN is about  $O(K * NS * NT * Min_{pts})$ .

The input of DBSCAN has  $\frac{NS(NS-1)}{2} NT^2$  intersections. And the time complexity of M-DBSCAN and DBSCAN can be regarded as the same. So the time complexity of the DBSCAN is about  $O\left(\frac{NS(NS-1)}{2} NT^2 Min_{pts}\right)$ .

The input of FCM has  $\frac{NS(NS-1)}{2} NT^2$  intersections. The clusters number of FCM is  $NT$ , the minimum number is  $Min_{pts}$ , the maximum number of iterations is  $NI$  [39]. The time complexity of the FCM is about  $O\left(\frac{NS(NS-1)}{2} NT^3 NI\right)$ .

The summary of algorithm time complexity is in Table 3.

In Sect. 3,  $NS=3$ ,  $NT=5$ , repeat measurement 50 times for each bearing line. Set  $K=3$  in the minimum  $K$  distance algorithm.  $Min_{pts}$  can be calculated by the Mahalanobis distance criterion in the M-DBSCAN algorithm. Set  $Min_{pts}=125$  in the DBSCAN algorithm. Set the number of iterations  $NI$  to 15 under the premise of ensuring that  $J_m$  can converge in the FCM of algorithm. The simulation computer configuration is as follows: Intel Core i7-12700F 2.1 GHz, 48.0 GB of DDR5 RAM, Windows 10 OS. Average time required for each position estimation is shown in Table 4.

**Table 3** The time complexity of algorithm

Algorithm	Time complexity
Minimum $K$ distance	$O[NS(NS - 1)NT^2]$
M-DBSCAN	$O(K * NS * NT * Min_{pts})$
DBSCAN	$O\left(\frac{NS(NS-1)}{2} NT^2 Min_{pts}\right)$
FCM	$O\left(\frac{NS(NS-1)}{2} NT^3 NI\right)$

**Table 4** Average time required for each position estimation

Algorithm	Time required (ms)
Minimum $K$ distance	43
M-DBSCAN	168
DBSCAN	315
FCM	185

Preprocessing by the minimum  $K$  distance algorithm reduces the number of intersections, so the consumption time of the M-DBSCAN algorithm is less than that of DBSCAN.

## 5 Conclusion

To handle the spurious intersections problem in BOL, this paper proposes a novel clustering method called K-M-DBSCAN. Compared with existing FCM and DBSCAN clustering methods, the proposed K-M-DBSCAN performs much better for removal of spurious intersections. The proposed method can not only adapt to the distribution of dense radiation source and the coexistence of far and near radiation sources, but also solve the problem of large variations in cluster density caused by different measurement error of angle. There are two reasons for this improvement. The first one is we construct Mahalanobis Distance-based density clustering, which follows the spatial probability distribution of intersections and enables to adapt the variations in cluster sample density. The second reason is that the deployed minimum  $K$  distance algorithm for preprocessing to filter out some spurious intersections, which reduce the interference and calculation complexity to clustering to a certain extent. Simulation results show that K-M-DBSCAN can effectively remove spurious intersections and still has a high recognition for the target when the localization precision threshold is 0.5 km. The precision of spurious intersections can reach more than 98%, and the precision of real intersections can reach more than 95%. Simulation results also verify the robustness of K-M-DBSCAN with different layout of observation stations.

### Abbreviations

BOL	Bearings-only localization
FCM	Fuzzy C-mean
DBSCAN	Density-based spatial clustering of application with noise
CBTA	Classical Bayesian Thread Association
MCTA	Monte Carlo Thread Association
PDA	Probabilistic Data Association
DOA	Direction of arrival
LOB	Line of bearing
LOF	Local outlier factor

### Acknowledgements

Not applicable.

### Author contributions

ZA proposed the framework of the whole method; ZG performed the simulations, analysis and interpretation of the results. ZG and TB have participated in the conception and design of this research. CJ have supervised the research of the paper. WJ have revised the manuscript. All authors read and approved the final manuscript.

### Funding

The authors greatly appreciate the editor and reviewers for their work. This work was financially supported by the National Natural Science Foundation of China (Grant No. 61971155), the Fundamental Research Funds for the Central Universities (Grant No. 3072022TS0802).

### Availability of data and materials

Please contact author for data requests.

### Declarations

#### Ethics approval and consent to participate

Not applicable.

#### Consent for publication

The picture materials quoted in this article have no copyright requirements, and the source has been indicated.

#### Conflict of interests

The authors declare that they have no competing interests.

Received: 4 August 2022 Accepted: 11 January 2023

Published online: 02 February 2023

### References

1. A. Achroufene, Y. Amirat, A. Chibani, Rss-based indoor localization using belief function theory. *IEEE Trans. Autom. Sci. Eng.* **16**(3), 1163–1180 (2018)
2. Z. Shi, H. Wang, C.S. Leung, H.C. So, M. EURASIP, Robust mimo radar target localization based on Lagrange programming neural network. *Signal Process.* **174**, 107–574 (2020)
3. S. Tang, L. Chen, K. He, J. Xia, L. Fan, A. Nallanathan, Computational intelligence and deep learning for next-generation edge-enabled industrial iot. *IEEE Trans. Netw. Sci. Eng.* **99**, 1–15 (2022)
4. Y. Wang, Y. Shu, X. Jia, M. Zhou, L. Xie, L. Guo, Multifeature fusion-based hand gesture sensing and recognition system. *IEEE Geosci. Remote Sens. Lett.* **19**, 1–5 (2022)
5. A.G. Lingren, K.F. Gong, Position and velocity estimation via bearing observations. *IEEE Trans. Aerosp. Electron. Syst.* **14**(4), 564–577 (1978)
6. G. Ferri, A. Munafò, A. Tesei, P. Braca, F. Meyer, K. Pelekanakis, R. Petrocchia, J. Alves, C. Strode, K. LePage, Cooperative robotic networks for underwater surveillance: an overview. *IET Radar Sonar Navig.* **11**(12), 1740–1761 (2017)
7. B. Omkar Lakshmi Jagan, S. Koteswara Rao, Underwater surveillance in non-gaussian noisy environment. *Meas. Control.* **53**(1–2), 250–261 (2020)
8. B. Liu, X. Tang, R. Tharmarasa, T. Kirubarajan, R. Jassemi, S. Hallé, Underwater target tracking in uncertain multipath ocean environments. *IEEE Trans. Aerosp. Electron. Syst.* **56**(6), 4899–4915 (2020)
9. N. Modalavalasa, G.S.B. Rao, K.S. Prasad, L. Ganesh, M. Kumar, A new method of target tracking by ekf using bearing and elevation measurements for underwater environment. *Robot. Autom. Syst.* **74**, 221–228 (2015)
10. S. Xu, K. Doğançay, H. Hmam, 3d aoa target tracking using distributed sensors with multi-hop information sharing. *Signal Process.* **144**, 192–200 (2018)
11. L. Chen, S. Tang, Physical-layer security on mobile edge computing for emerging cyber physical systems. *Comput. Commun.* **99**, 1–12 (2022)
12. C. YongBo, M. YueSong, Y. JianQiao, S. XiaoLong, X. Nuo, Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm. *Neurocomputing* **266**, 445–457 (2017)
13. H. Li, J. Chen, F. Wang, M. Bai, Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: a review. *Eur. J. Oper. Res.* **294**(3), 1078–1095 (2021)
14. A. Rutkowski, A. Kawalec, Some of problems of direction finding of ground-based radars using monopulse location system installed on unmanned aerial vehicle. *Sensors* **20**(18), 5186 (2020)
15. L. Chen, R. Zhao, K. He, Z. Zhao, L. Fan, Intelligent ubiquitous computing for future uav-enabled mec network systems. *Clust. Comput.* **25**, 1–11 (2021)
16. A. Farina, Target tracking with bearings-only measurements. *Signal Process.* **78**(1), 61–78 (1999)
17. D.-T. Pham, Some quick and efficient methods for bearing-only target motion analysis. *IEEE Trans. Signal Process.* **41**(9), 2737–751 (1993)
18. X. Lai, J. Xia, L. Fan, T.Q. Duong, A. Nallanathan, Outdated access point selection for mobile edge computing with cochannel interference. *IEEE Trans. Veh. Technol.* **99**, 1–10 (2022)
19. R. Danchick, G.E. Newnam, A fast method for finding the exact n-best hypotheses for multitarget tracking. *IEEE Trans. Aerosp. Electron. Syst.* **29**(2), 555–560 (1993)
20. G.S. Bauer, J.B. Greenblatt, B.F. Gerke, Cost, energy, and environmental impact of automated electric taxi fleets in Manhattan. *Environ. Sci. Technol.* **52**(8), 4920–4928 (2018)
21. R.L. Popp, K.R. Pattipati, Y. Bar-Shalom, m-best sd assignment algorithm with application to multitarget tracking. *IEEE Trans. Aerosp. Electron. Syst.* **37**(1), 22–39 (2001)
22. M. Beard, S. Arulampalam, Comparison of data association algorithms for bearings-only multi-sensor multi-target tracking, in *2007 10th International Conference on Information Fusion*, pp. 1–7 (2007)
23. T. Kirubarajan, Y. Bar-Shalom, Probabilistic data association techniques for target tracking in clutter. *Proc. IEEE* **92**(3), 536–557 (2004)

24. A. Iqbal, N.R. Gans, Localization of classified objects in slam using nonparametric statistics and clustering, in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 161–168 (2018)
25. M. Strecke, J. Stuckler, Em-fusion: dynamic object-level slam with probabilistic data association, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5865–5874 (2019)
26. S. Yang, S. Scherer, Cubeslam: monocular 3-d object slam. *IEEE Trans. Robot.* **35**(4), 925–938 (2019)
27. Y. Kanzawa, S. Miyamoto, Generalized fuzzy c-means clustering and its property of fuzzy classification function. *J. Adv. Comput. Intell. Intell. Inform.* **25**(1), 73–82 (2021)
28. J.D. Reed, C.R. da Silva, R.M. Buehrer, Multiple-source localization using line-of-bearing measurements: approaches to the data association problem, in *MILCOM 2008-2008 IEEE Military Communications Conference*, pp. 1–7 (2008)
29. R.J. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 160–172 (2013)
30. L. Duan, L. Xu, F. Guo, J. Lee, B. Yan, A local-density based spatial clustering algorithm with noise. *Inf. Syst. J.* **32**(7), 978–986 (2007)
31. T.N. Tran, K. Drab, M. Daszykowski, Revised dbscan algorithm to cluster data with dense adjacent clusters. *Chemom. Intell. Lab. Syst.* **120**, 92–96 (2013)
32. Z. Huang, S. Chen, C. Hao, D. Orlando, Bearings-only target tracking with an unbiased pseudo-linear Kalman filter. *Remote Sens.* **13**(15), 2915 (2021)
33. S. Nardone, A. Lindgren, K. Gong, Fundamental properties and performance of conventional bearings-only target motion analysis. *IEEE Trans. Autom. Contr.* **29**(9), 775–787 (1984)
34. G. Dai, Z. Xiao, L. Cai, Y. Le, Phantom excludability algorithm based on speciality of sensor and target, in *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSE)*, pp. 683–686 (2017)
35. W. Lai, M. Zhou, F. Hu, K. Bian, Q. Song, A new dbscan parameters determination method based on improved mvo. *IEEE Access* **7**, 104085–104095 (2019)
36. J.-H. Kim, J.-H. Choi, K.-H. Yoo, A. Nasridinov, Aa-dbscan: an approximate adaptive dbscan for finding clusters with varying densities. *J. Supercomput.* **75**(1), 142–169 (2019)
37. K. He, L. He, L. Fan, X. Lei, Y. Deng, G.K. Karagiannidis, Efficient memory-bounded optimal detection for gsm-mimo systems. *IEEE Trans. Commun.* **99**, 1–15 (2022)
38. J. Wang, Q. Gao, H. Wang, P. Cheng, K. Xin, Device-free localization with multidimensional wireless link information. *IEEE Trans. Veh. Technol.* **64**(1), 356–366 (2014)
39. W. Peizhuang, Pattern recognition with fuzzy objective function algorithms (James C. Bezdek). *SIAM Rev.* **25**(3), 442 (1983)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)