Cooperative Target Localization Method for Heterogeneous Sensor Networks

Qing Yang, Lu Su, Quanlong Li, Xiaofei Xu

Department of Computer Science and Engineering
Harbin Institute of Technology, Heilongjiang, P.R.C, 150001
{yangqing,suluhit,liquanlong,xiaofei}@hit.edu.cn

Abstract. Based on the binary sensor model, a novel method for target localization in heterogeneous sensor networks is presented. With the binary information reported by nodes, target's position is locked into the intersection area of sensing areas of all nodes which detect the same target, and then the estimated position is computed by geometric means. The proposed method adapts to heterogeneous sensor networks, moreover, it can integrate with other target localization approaches easily. Simulation results demonstrate that, in sensor networks composed of the same type of sensors, our method lead to a decrease in average localization errors compared with the traditional method; in heterogeneous sensor networks, the method renders more accurate estimate of the target's location.

1 Introduction

Advances in the fabrication and integration of sensing and communication technologies have facilitated the deployment of large scale sensor networks. A wireless sensor network consists of tiny sensing devices, deployed in a region of interest. Each device has processing and wireless communication capabilities, which enable it to gather information from the environment and to generate and deliver report messages to the remote base station (remote user). The base station aggregates and analyzes the report messages received and decides whether there is an unusual or concerned event occurrence in the deployed area [1].

Because of its spatial coverage and multiplicity in sensing aspect and modality, a sensor network is ideally suited for a set of applications: biomedicine, hazardous environment exploration, environmental monitoring and military tracking. Target localization is the foundation of many sensor networks' applications, so research about target localization in sensor networks has recently attracted much attention. For example, Time of Arrival (TOA) technology is commonly used as a means of obtaining range information via signal propagation time; Maximum Likelihood testing (ML) [2] and minimum square estimation [3], are applied to compute the target's position at one node which in charge of collecting the data captured by other sensors. Some other methods estimated the target location at one sensor by successively computing on the current measurement and the past history at other sensors [4, 5, 6]. With hardware limitations and the inherent energy constraints of sensor devices, all the signal processing technologies present a costly solution for localization in wireless sensor networks. Unlike these approaches, our cooperative target localization method requires only that a sensor be able to determine whether an object is somewhere within its maximum detection ranger. Our proposed method is similar to the algorithm mentioned in [7] which considers the average *x* and *y* coordinates of all reporting nodes as the target location. However, our algorithm can render more accurate target location estimation without losing the briefness and efficiency.

This paper makes three major contributions to the target localization problem in sensor networks. First, though many methods [2, 3, 4, 5, 6] have been proposed to solve this problem, none of them has considered the networks composed of heterogeneous sensors. This paper provides a realistic and detailed algorithm to determine the target's location in heterogeneous sensor networks. Second, compared with the prior algorithm such as that mentioned in [7], the proposed method renders more accurate estimate of target's location. Third, the presented approach can guarantee that the target must be in a small intersection area X which our algorithm works out; that means other methods do not need to search the whole area but only X.

The organization of the rest of this paper is as follows. Section 2 gives brief description of the binary sensor model, preliminaries and assumptions. In Section 3, we present details of the target localization model and VSB (Valid Sensing Border) updating method. Section 4 designs and analyses the Cooperative Target Localization (CTL) Algorithm. In Section 5, we present simulation results, comparing CTL with the traditional method mentioned in [7]. Section 6 concludes the paper and outlines the direction for future work.

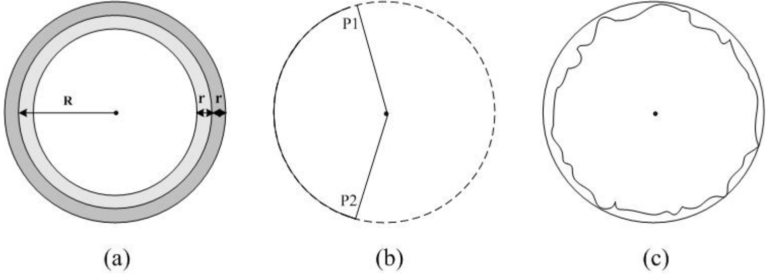
2 The Binary Sensor Network

Suppose a set of m different kinds of sensors $S = \{s_1, s_2, s_3, \ldots, s_m\}$ are deployed within a bounded 2-dimensional area, these sensors compose a binary sensor network. In this binary sensor network, each sensor's result is converted reliably to one bit of information only. This binary information may have different meanings, for example, it means whether an object is approaching or moving away from sensors in [8]. In this paper, we define it as whether an object is somewhere within the maximum detection range of sensors.

Nevertheless, in heterogeneous sensor networks, detection ranges of sensors are different from one to another. For example, sensors with infrared or ultrasound sensing devices have a circle-like sensing area as illustrated in Fig.1 (a); image sensors have a sector-like sensing area, as illustrated in Fig.1 (b); some other sensors' sensing areas may be irregular as illustrated in Fig.1 (c). It is difficult to use these sensing areas directly, so we uniformly define the sensing border of every sensor as a circle.

Definition 1 Sensing Radius: The sensing radius of one sensor $s_i \in S$ is defined as $Max|s_ip|$, where p is one point of set Q which consists of all the points that can be detected by sensor s_i. We denote s_i's sensing radius as $s_i.R$.

Definition 2 Sensing Border: Consider any sensor $s_i \in S$, the circle centered at this node with radius $s_i.R$ is s_i's sensing border, denoted as $s_i.C$.
Definition 3 Sensing Area: Consider any sensor $s_i \in S$, its sensing border and the insidearea are s_i's sensing area, denoted as $s_i.A$.



(a) (b) (c)

Fig. 1. Sensing area of different kinds of sensors

Consider the sensing area shown in Fig.1 (a), based on the probability-based sensor detection model [9], this kind of sensor's sensing radius is $R+r$. For the image sensor, valid sensing border is not a circle but an arc, such as inner arc p_1p_2 illustrated in Fig.1 (b). We will define the sensor'svalid sensing border in later sections.

In heterogeneous sensor networks, sensors have different sensing radiuses, which may be caused by two reasons. First, sensors have different sensing radiuses initially. Second, sensor's sensing radiuses may change during its lifetime. For example, the power level may have an impact on sensor's sensing range. In this paper, we make the assumption that sensing radiuses of every sensor are known and will not change during the whole lifetime. Second, we suppose that each node knows its own location and nodes are not moving. The node's location information does not need to be precise because we are using conservative sensing area to calculate target's location.

Based on the above assumptions, for any sensor $s_i \in S$, one bit of information '1' will be sent if the distance between a target and itself is less than $s_i.R$. If this distance is no less than s_i's sensing radius, the binary information is '0' and nothing will be sent.

3 Cooperative Target Localization Method in Heterogeneous Sensor Networks

3.1 Target Localization Model for Binary Sensor Networks

In binary sensor networks, target localization problem can be formulated as follows. Within a bounded 2-dimensional area, m binary sensors $S = \{s_1, s_2, s_3, \ldots, s_m\}$ are deployed. Assume a target T moves into the area and is located at (x_0, y_0), there will be a set of sensors $D = \{d_1, d_2, d_3, \ldots, d_n\}$ ($D \subseteq S$) detect a target appearing (for example, 6 nodes detect one target as illustrated in Fig.2) by signal processing approaches such as LRT [6]. These nodes in D then send binary information '1' to base station or Cluster Heads which analyzes the report results and estimates the target's location. With the location information of each node and reported binary results, target can be locked into the intersection area $X = d_1.A \cap d_2.A \cap d_3.A \cap . . . \cap d_n.A$; then the average x and y coordinates of all vertexes of X will be regarded as the target's location.

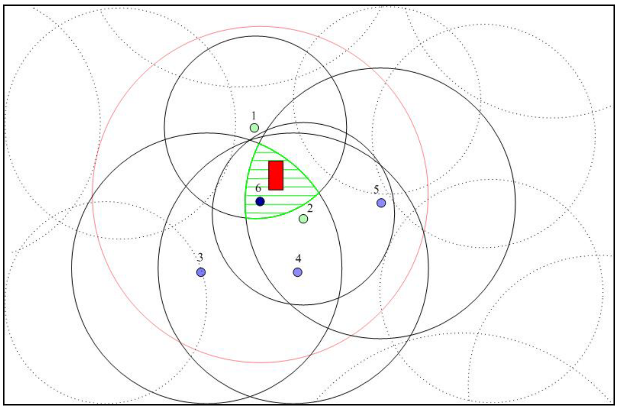


Fig. 2. Detection of one target by multi-sensors

Consider any two nodes d_i and d_j in D, suppose they are located at o_i and o_j respectively. Since they detected the same target, sensing borders of d_i and d_j must intersect. Let $d_i.C$ and $d_j.C$ touch at point p_1 and p_2, then the target must appear in the area enveloped by arc $o_ip_1p_2$ and arc $o_jp_1p_2$. Because sensors' locations are known, the arcs generated by intersections of all nodes' sensing borders will be the most import information to compute area X.

Definition 4 Valid Sensing Border (VSB): Consider any sensor $d_i \in D$, its valid sensing border is defined as all of the arcs which satisfy that for any point p on these arcs and any sensor $d_j \in D$ $(i \neq j)$, p must be in the sensing area of d_j. If no such arc exists, then the valid sensing border of d_i is null. We denote d_i's valid sensing border as $d_i.arc$.

Theorem 1 Suppose the area enveloped by all sensors' valid sensing borders is R, the intersection area of all sensors' sensing areas is X, then $R = X$.

Proof： (1) We will prove that $R \subseteq X$.

Consider any sensor $d_i \in D$ whose VSB is not null. For any point p on $d_i.arc$, from Definition 4, p must be in the area $X = d_1.A \cap d_2.A \cap d_3.A \cap . . . \cap d_n.A$. Since sensor d_i and point p are randomly selected, that means for each $i = 1, 2, . . . , n$, if $d_i.arc$ is not null, it must be in the area X. Moreover, area R is enveloped by VSBs of all sensors, so for any point q in the area R, $q \in X$ must holds. That is to say $R \subseteq X$.

(2) We will show $X \subseteq R$.

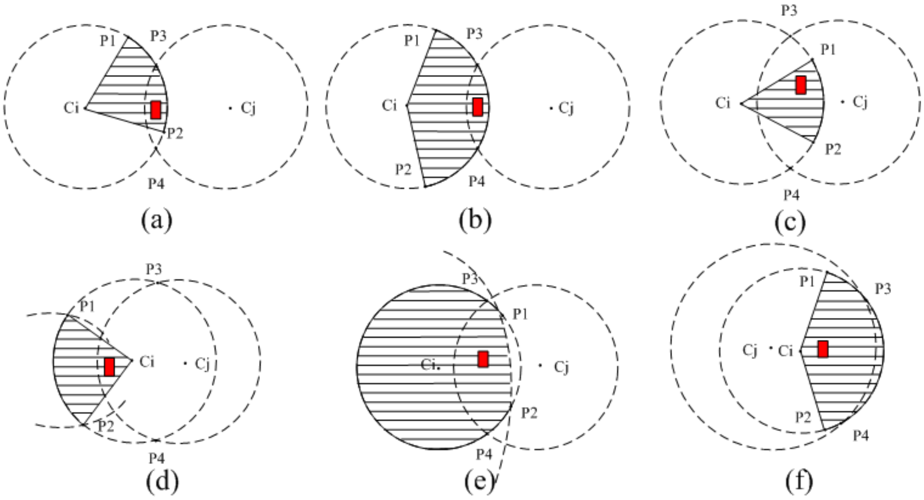
Assume by contradiction that the claim is false. This implies that there exists at least one point p, and p is in the area X but not in R. If p is outside R, then p must not be on $d_i.arc$ (for each $i = 1, 2, . . . , n$). From Definition 4, there must be at least one sensor d_j which satisfies that p is not in $d_j.A$. That means point p is not in X, this is contradictory to our hypothesis. Thus, the expression $X \subseteq R$ must holds.

Combining (1) and (2) completes the proof.

Based on Theorem 1, we can use VSBs of all nodes in D to compute the target's location instead of using the intersection area of all nodes' sensing areas directly. In this way, as Fig.1 (b) shows, we can initialize the VSB of image sensor as arc p_1p_2. In the next section, we will discuss in detail how to update VSB of each sensor. After getting every node's VSB, we regard the average x and y coordinates of vertexes of all sensors' VSBs as the target's location.

3.2 VSB Updating Method

Suppose every node in D has the same sensing radius, the case with different sensing radius will be discussed later. Consider any sensor $d_i \in D$, assume its VSB is arc p_1p_2. Since the sensing radius of each sensor is identical, the arc p_1p_2 must be an inner arc. In the following, if we do not indicate specially, the word ‘arc’ means inner arc. All nodes in D detected the same target, so $d_i.C$ must intersect other sensors' sensing borders. Suppose $d_i.C$ touches with $d_j.C$ (another node’ssensing border) at point p_3 and p_4, and then four cases will appear as shown in Fig.3 (a, b, c, d).



(a) (b) (c)

(d) (e) (f)

Fig. 3. One sensor’s VSB intersectsanother’ s sensing border. The arc drawn by real line is d_i’s VSB and the rectangle denotes the target.

Case 1: As Fig.3 (a) illustrated, if only one point of p_3 and p_4 is on arc p_1p_2 (without loss of generality, suppose p_3 is on arc p_1p_2), then we must have only one point of $p_1$and p_2 is on arc p_3p_4 (suppose p_2 is on arc p_3p_4). Thus, the new VSB of d_i is arc p_2p_3 on $d_i.C$.

Case 2: As Fig.3 (b) illustrated, if both p_3 and p_4 are on arc p_1p_2, at the same time, both p_1 and p_2 are not on arc p_3p_4, then the new VSB of d_i is arc p_3p_4.

Case 3: As Fig.3 (c) illustrated, if both p_3 and p_4 are not on arc p_1p_2, both p_1 and p_2 are on arc p_3p_4, then d_i's VSB is still arc p_1p_2.

Case 4: AsFig. 3 (d) illustrated, if both p_3 and p_4 are not on arc p_1p_2, at the same time, both p_1 and p_2 are also not on arc p_3p_4, based on Theorem 2, VSB of d_i is null.

Theorem 2: Consider any two sensors d_i and d_j in the D, suppose $d_i.arc$ is arc p_1p_2 and $d_i.C$ intesects $d_j.C$ at point p_3 and p_4. Thus, if both p_3 and p_4 are not on arc p_1p_2, both p_1 and p_2 are not on arc p_3p_4, then VSB of d_i is null.

Proof: Assume by contradiction that VSB of d_i is not null. Then, from Definition 4, for any point p on $d_i.arc$ and any node $d_k \in D$ ($i \neq k$), p must be in the sensing area of d_k. If we use p_1 and d_j to replace p and d_k, then p_1 must be in $d_j.A$. Since p_1 is on circle $d_i.C$, that means p_1 is in the $d_i.A \cap d_j.A$. Because $d_i.C$ and $d_j.C$ touch at p_3 and p_4, p_1 is on $d_i.C$, then p_1 must be on arc p_3p_4 of $d_i.C$. On the other hand, p_1 is not on arc p_3p_4, which is contradictory; thus, the claim $d_i.arc$ is null holds.

All the cases discussed above will happen if each node in D has the same sensing radius. If sensing radiuses of sensors in D are different, there will be some new cases. Firstly, we will define the distance of two sensors in S. Consider any two sensor s_i and s_j in S, suppose they are located at (x_i, y_i) and (x_j, y_j) respectively; then the distance between them is defined as:

$$dis(s_i, s_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{1}$$

Since sensing radiuses of s_i and s_j are different, s_i's sensing border will completely contains $s_j.C$, which happens whenever $s_j.R+dis(s_i, s_j) < s_i.R$ holds.

Case 5: If $d_j.R+dis(d_i, d_j) < d_i.R$, without further calculation, VSB of d_i is null and $d_j.arc$ will not change. This is because target T can only be in the sensing area of d_j, and then there will be no VSB on $d_i.C$.

Notice that if sensing radiuses are different from one to another, VSB of each sensor will not always be an inner arc. If one sensor's VSB is an outer arc, the updating method mentioned above will still work for Case 1, 3 and 4. But for Case 2, the original VSB will be cut into two pieces.

Case 2a: As Fig.3 (e, f) illustrated, if both p_3 and p_4 are on arc p_1p_2, both p_1 and p_2 are on arc p_3p_4, then the new VSB of d_i are arc p_1p_3 and arc p_2p_4.

In this case, the updating method mentioned above need some modifications since $d_i.arc$ has more than one arc. Suppose $d_i.arc$ is composed of arc_1, arc_2, arc_3, . . . , arc_k (later we will proof that k is less than n). Obviously, all these k arcs are inner arcs. We firstly consider $d_i.arc$ has only one arc such as arc_1, and then update this VSB according to the method mentioned above. If Case 4 or Case 5 occurs, then wipe arc_1 out from the original VSB of d_i; otherwise, new VSB will replace arc_1. In succession, let $d_j.arc$ is arc_2, arc_3, . . . , arc_k, and then update the VSB.

Theorem 3: Every sensor in set D has at most $n - 1$ valid sensing borders.

Proof: Consider any node d_i in D, $d_i.C$ will intersect at most $n - 1$ sensing borders of other sensors in D, this can produce at most $2n - 2$ points on the sensing border of d_i. Since every VSB of sensor d_i must be the arc between two points produced by sensing borders' intersection, these $2n - 2$ points can build at most $n - 1$ arcs. That means $d_i.arc$ has at most $n - 1$ arcs, so we complete the proof.

4.Cooperative Target Localization (CTL) Algorithm

Based on the above analysis, it is very important to find out VSBs of all node in D. For the image sensor, as shown in Fig1 (b), we initialize its VSB as the arc p_1p_2; for the other types, as Fig.1 (a, c) illustrated, their initial VSBs are null. Consider any sensor $d_i \in D$, our algorithm aims to calculateits new VSB after $d_i.C$ intersects the sensing border of every other node d_j ($i \neq j$) in D. If $d_i.arc$ is null, the arc produced by the intersection of $d_i.C$ and $dj.C$ is d_i's new VSB; if $d_i.arc$ is not null, we use the VSB updating method introduced in section 3.2 to get d_i's the new VSB. In some unusual cases, sensing borders of d_i and d_j may touch at one point. That means the target is located at this point, so we need no more calculations. After getting the new VSB of d_i, we apply the same method to d_j and then update $d_j.arc$. Based on the VSB of each node in D, we use formula 2 to calculate the target's location.

$$\begin{cases} x = \sum_{i=1}^{n'} \sum_{j=1}^{ki} (d'_i.arc_j.p_1.x + d'_i.arc_j.p_2.x) \Big/ 2(k1 + k2 + \cdots + kn') \\ y = \sum_{i=1}^{n'} \sum_{j=1}^{ki} (d'_i.arc_j.p_1.y + d'_i.arc_j.p_2.y) \Big/ 2(k1 + k2 + \cdots + kn') \end{cases} \quad (2)$$

In sensor set D, these sensors whose valid sensing borders are not null compose a new set, we denote it as D'. For any node $d'_i \in D'$, assume its VSB has ki pieces of arcs ($1 \leqslant ki \leqslant n - 1$). Then we can use $d'_i.arc_j.p_1$ and $d'_i.arc_j.p_2$ ($1 \leqslant i \leqslant n'$, $1 \leqslant j \leqslant ki$) to denote two vertexes of the jth valid sensing border of d'_i. Formula 2 aims to calculate the average x and y coordinates of all vertexes of X.

```
Procedure CTL(D)
for $i = 1$ to $n$ do
  for $j = i + 1$ to $n$ do
    if $d_i$ and $d_j$ cross then
      $p_3, p_4 \leftarrow$ The points of intersection between $d_i$ and $d_j$;
      for every arc $arc_k$ of $d_i.arc$ do
        if only one of $p_3$ and $p_4$ is on $arc_k$ then                /*Case 1*/
          Wipe $arc_k$ out; Add the overlapped part of $arc_k$ and arc $p_3p_4$ to $d_i.arc$;
        end if
        if both $p_3$ and $p_4$ are on $arc_k$ then
          if both $arc_k.p_1$ and $arc_k.p_2$ are not on arc $p_3p_4$ then  /*Case 2*/
            $arc_k.p_1 \leftarrow p_3$ and $arc_k.p_2 \leftarrow p_4$;
          end if
          if both $arc_k.p_1$ and $arc_k.p_2$ are on arc $p_3p_4$ then      /*Case 2a*/
            Wipe $arc_k$ out; Add arc $p_1p_3$ and arc $p_2p_4$ to $d_i.arc$;
          end if
        end if
        if both $p_3$ and $p_4$ are not on $arc_k$ and                   /*Case 4*/
          both $arc_k.p_1$ and $arc_k.p_2$ are also not on arc $p_3p_4$ then
```

```
19:         Wipe $arc_k$ out from $d_i.arc$;
20:       end if
21:     end for
22:   else                                               /*Case 5*/
23:     if $d_i.R > d_j.R$ then $d_i.arc \leftarrow null$;
24:     end if
25:   end if
26:   update $d_j.arc$ using the same method;
27:  end for
28: end for
29: for $i = 1$ to $n$ do            /*average $x$ and $y$ coordinates of all vertexes*/
30:  while $d_i.arc$ != $null$ do   /* The initial values of $x$, $y$ and $num$ are 0*/
31:   $x \leftarrow d_i.p_1.x + d_i.p_2.x + x$;
32:   $y \leftarrow d_i.p_1.y + d_i.p_2.y + y$;
33:   $num \leftarrow num + 1$;
34:  end while
35: end for
36: $x \leftarrow x/(2\times num)$; $y \leftarrow y/(2\times num)$;
37: return $(x, y)$
```

Based on Theorem 1, the region enveloped by all VSBs must contain the target; so our CTL algorithm is right. In the following, we will discuss the running time of this algorithm. Line 4, 6-8, 9-16, 17-20, 23-24 can be performed in $O(1)$ time. Because one sensor's VSB has at most $n - 1$ arcs, line 3-25 is executed at most $O(n)$ time. The "for loop in 2-3 requires $O(n^2)$ time, then line 1-28 takes at most $O(n^3)$ running time. Line 29-35 contributes $O(n^2)$ to the running time. Thus, the total running time of this algorithm is at most $O(n^3)$. Obviously, running time only depends on the number of sensors which detected the same target.

5. Simulation

We implemented a simulator for CTL in order to examine the accuracy of estimates. Let networks cover a 1000m×1000m rectangle area which was divided into 1m×1m grids. Suppose the target is located on each grid, we record the distance between the estimated and real target's position. AvgXY denotes the method mentioned in [7].

5.1 Results

(1) We firstly define the Node Density (ND) as the average number of nodes per $R \times R$ area where R is the sensing radius. If all sensors have the same sensing radius, let $R = 20m$, then Fig. 4 explores the localization estimation accuracy of two methods. From this picture, we can easily find that the average localization errors of CTL are about 1 meter less than those of AvgXY.

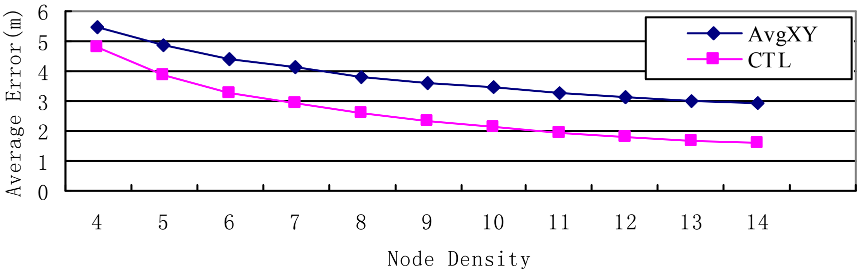


Fig. 4. Average localization errors of two methods with the same sensing radius

(2) In heterogeneous sensor networks composed by 10*m* and 30*m* sensors, we define the ND as the average number of nodes per 20*m* ×20*m* area. Fig5 shows that the average localization errors of CTL are 2 meters less than AvgXY.

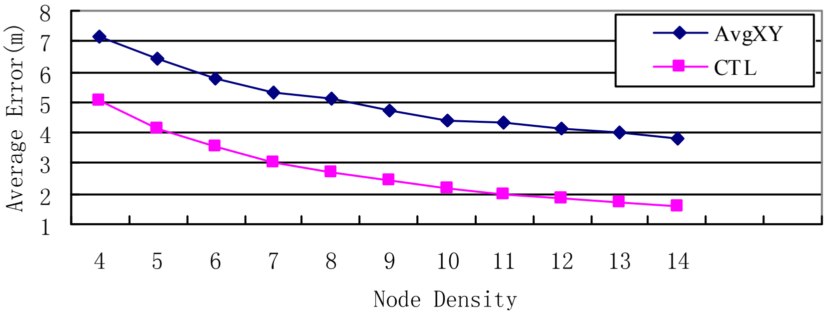


Fig. 5. Average localization errors of two methods in heterogeneous sensor networks

(3) Given ND being a constant (assume ND = 6), Fig. 6 shows that estimation errors of two methods increase as sensing radius become larger. However, CTL always renders a less localization error than AvgXY .

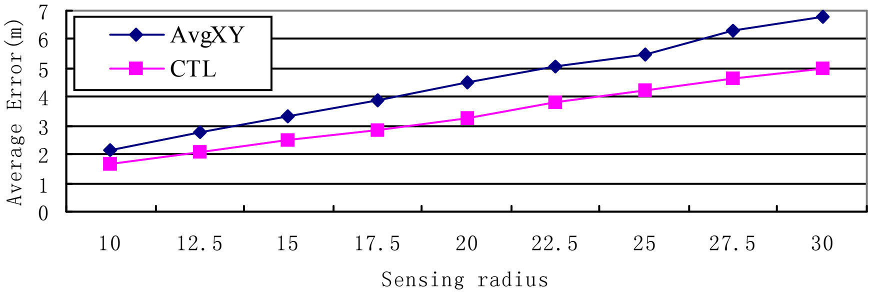


Fig. 6. Average localization errors varying sensing radiuses

6. Conclusion

In this paper, we described a cooperative target localization algorithm for heterogeneous sensor networks. Based on the binary sensor model, we presented the definition of sensing radius, sensing border, sensing area and valid sensing border; then give the target localization model for heterogeneous sensor networks. Simulation results have demonstrated that, not only in sensor networks consist of same types of sensors but also in heterogeneous sensor networks, the proposed method lead to a decrease in average localization errors compared with the traditional methodIn addition, the proposed approach can guarantee that the target is in a small region; this implies that other target localization methods need only to consider this region instead of the whole area. Moreover, if distances between the target and sensors are added into our method, the estimation accuracy will be improved. If consider the target classification information, we can implement the multiple targets localization method.

Reference

1. Tian, Di, Georganas, Nicolas D: "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*, 2002, p 32-41
2. X. Sheng, and Y-H Hu: "Energy Based Acoustic Source Localization *Proc. of 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.
3. D. Li, K.Wong, Y. Hu and A. Sayeed: "Detection, Classification, Tracking of Targets *IEEE Signal Processing Magazine*, pp. 17-29, March 2002
4. F. Zhao, J. Shin and J. Reich: "Information-Driven Dynamic Sensor Collaboration for Tracking Applications, *IEEE Signal Processing Magazine*, March 2002, 19(2):61-72
5. M. Chu, H. Haussecker and F. Zhao: "Scalable Information-driven Sensor Querying and Routing for Ad Hoc Heterogeneous Sensor Networks, *Int'l J. High Performance Computing Applications*, vol. 16, no. 3, Fall 2002.
6. J. Liu, J. Liu, J. Reich, P. Cheung, and F. Zhao: "Distributed Group Management for Track Initiation and Maintenance in Target Localization Applications, *Proc. 2nd Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003:113-128.
7. Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, Tarek Abdelzaher "Range-free localization schemes for large scale sensor networks *MobiCom '03,* September 2003, San Diego, CA, SA:81-95
8. Javed Aslam, Zack Butler, Florin Constantin, Valentino Crespi, George Cybenko, Daniela Rus: "Tracking a Moving Object with a Binary Sensor Network SenSys'03, November 5–7, 2003, Los Angeles, California, USA.
9. Yi Zou and Krishnendu Chakrabarty: "Energy-Aware Target Localization in Wireless Sensor Networks Proceedings of the First IEEE International Conference on Pervasive Computing and Communications 2003:60-67