

# Localization in Sensor Networks with Limited Number of Anchors and Clustered Placement

King-Yip Cheng, King-Shan Lui and Vincent Tam  
Department of Electrical and Electronic Engineering  
The University of Hong Kong  
Pokfulam Road, Hong Kong, China

**Abstract**—Many localization algorithms have been proposed in recent years. Although different algorithms based on different methodologies, the use of anchors is common to most algorithms. The placement and the density of anchors affect the accuracy of different algorithms to different extent. Location estimates are usually more accurate with a higher density of anchors. When there are only a few anchors, efficient algorithms tend to perform poorly. However, having more anchors will increase the cost of a sensor network. In this paper, we present an algorithm which uses two different localization techniques, multidimensional scaling (MDS) and proximity distance mapping (PDM), in a phased approach. MDS has a high complexity but can give good results when there are only very few anchors. PDM, on the other hand, is a distributed algorithm but performs poorly when anchors are scarce. The phased approach has comparable complexity to PDM but less than MDS. With extensive simulations, we demonstrate that the proposed algorithm gives accurate solution with very few anchors or clustered anchors which is intrinsically a difficult challenge to most existing algorithms.

## I. INTRODUCTION

In many sensor network applications, sensors are deployed in a large physical environment where location information of each sensor is critical to the applications, e.g. habitat monitoring, target tracking, location-based routing, etc. However, localization remains a challenging problem in wireless sensor networks. Although sensors equipping GPS (Global Positioning System) receiver can obtain accurate absolute location information, the energy consumption and cost prohibit large-scale deployment. To mitigate the problem, various localization algorithms [1]- [8] have been proposed. Most algorithms assume the existence of the anchors. Anchors are sensors equipped with GPS receivers or having prior knowledge about their physical locations. The location information of anchors is used for multilateration, transformation of relative coordinates to absolute coordinates or constraints in mathematical programming-based algorithms [5] [8]. Distributed algorithms based on multilateration requires significant amount of anchors to maintain the accuracy of the solution. However the cost of anchors is much higher than a normal sensor. Furthermore, the placement of anchors also affects the performance of multilateration based methods [9] [10] [11]. Better performance can be achieved if anchors are distributed uniformly around the perimeter of the sensor network. On the other hand, centralized algorithms require less anchors but they are less scalable at the same time. The effect of anchor placement is less significant when compared to multilateration-based algorithms.

In this paper, we propose a localization algorithm which requires very few anchors to work well in a decentralized fashion. The result is comparable to centralized algorithms. Moreover, there is no restriction on the anchor placement. Anchors can be distributed randomly over the sensor network or squeezed in the corner of the sensor network. This property is especially good for sensor networks in hostile environment. To ruin a sensor network covering a battlefield, anchors will become the first target. Even though the outlook of an anchor may be indistinguishable from other sensors, leaving it in front line will expose it to the fire of the enemy. The whole network may be malfunctioned because a few anchors are destroyed. However, if anchors are only placed in an area which is under control, a sensor network can still operate when some normal sensors are destroyed in the front line.

Our algorithm is composed of two localization techniques, multidimensional scaling (MDS) [12] and proximity-distance map (PDM) [7], in a phased approach. In the beginning, there are a few nodes that are equipped with GPS receivers and we called these nodes *primary anchors*. In the first phase, a subset of ordinary sensors are selected as *secondary anchors*. Their locations are determined by MDS. The number of secondary anchors is controlled such that MDS can be performed on each selected sensors individually. Other ordinary sensors are localized through PDM.

The paper is organized as follows. In Section II, we review some of the previous works of localization in wireless sensor networks. In Section III, we give the details of our proposed algorithms. Section IV presents the simulation results and we conclude the paper in Section V.

## II. RELATED WORKS

Niculescu *et al.* [1] proposed a distributed localization algorithm called Ad Hoc Positioning System (APS). A sensor node first estimates the distances from at least 3 (4 in 3-D) anchors whose location information is known and accurate. A system of non-linear equations relating the distance estimates and the position of the sensor and anchors is established. The system is approximated as a linear system and is solved iteratively by each sensor. There are three propagation methods to disseminate the distance information, namely DV-Hop, DV-Distance and Euclidean. Hop count and shortest-path distance are used by DV-Hop and DV-distance respectively to estimate the distance between the sensor and anchors. The last

propagation method, Euclidean, propagates the true Euclidean distance.

Savarese *et al.* [6] proposed another multilateration-based algorithm called Hop-TERRAIN. The algorithm consists of two phases. The first phase is similar to the DV-Hop, aiming for an initial estimate. In second phase, each sensor refines its estimation iteratively by triangulation with the measured ranges and the location estimates of connected neighbours. Savvides *et al.* [4] proposed a similar two-phased algorithm for which nodes first obtain a rough estimate by utilizing the bounding box constraints. The solution is refined by applying the Kalman filter.

Him *et al.* [7] designed a distributed algorithm for anisotropic topologies. The major difficulty encountered with anisotropy is the dynamic relation between the geographical distance and the proximity measured by sensors in different directions. To cope with this, Him *et al.* devised a proximity-distance map (PDM) to capture the relation in different directions. Sensors localize themselves by multilateration similar to APS except the proximities collected by each sensor is transformed by the proximity-distance map. However, when anchors are clustered together, the proximity-distance map is no longer reliable. The relation between proximity and geographical distance is not characterized by the PDM in the region without any anchor.

Besides the distributed algorithms mentioned above, researchers also designed many centralized algorithms. Doherty *et al.* [8] proposed a centralized algorithm by formulating the localization problem as a convex optimization problem. The problem is solved by a centralized server. The quality of the solution greatly depends on the density and placement of anchors. It is better to place anchors around the outer boundary of the networks so that sensors fall within the convex hull of the anchors. Biswas *et al.* [5] alleviated the placement problem by formulating the localization problem as a semidefinite program (SDP) through relaxation. Though SDP requires few anchors and yields good result, it requires extensive storage and computation when the scale of network is large. Shang *et al.* [2] proposed a centralized algorithm MDS-MAP which employs multidimensional scaling (MDS). In the first phase, all-pair-shortest-path distance or hop count between sensors is collected through distance vector exchange. The path distance provides a rough estimate about the distance between every pair of sensors. In the second phase, classical MDS is applied to yield a relative map. In the final phase, the relative map is transformed to an absolute map given that there are at least 3 anchors. Because of the global information and computation required by MDS-MAP, it has to be executed in a centralized fashion. To enhance its practicality, Shang *et al.* [3] proposed a distributed version, MDS-MAP(P). The network is divided into different regions and the maps of regions are determined locally. The local maps are merged and transformed into a complete map which is subsequently transformed into a global map with the location information of anchors. Ahmed *et al.* [13] proposed SHARP approach which our algorithm is inspired from. Though SHARP also employs two different

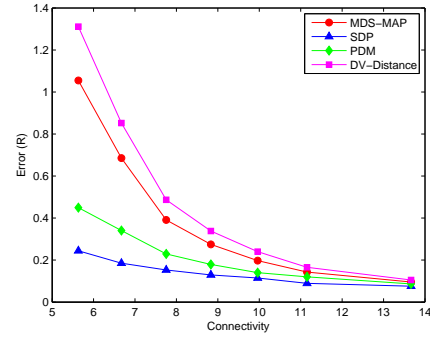


Fig. 1. 50-node network,10 anchors,  $\alpha=0.05$

localization algorithms in a phased approach, it is designed for relative localization. Nodes positions are localized in second phase with reference to the coordinate system determined in the first phase. Our proposed algorithm, on the other hand, can obtain absolute coordinates and we look into the issues of anchor placement.

### III. OUR PHASED APPROACH: MDS+PDM

In this section, we present a distributed localization algorithm MDS+PDM which combines two techniques, multidimensional scaling [12] and proximity distance mapping [7]. The choice of integrating MDS-MAP and PDM is based on our extensive study on existing algorithms. We examine the accuracy obtained by DV-distance(APS), MDS-MAP, SDP and PDM with sufficient anchors (20% of nodes are anchors). Though Figure 1 shows that SDP outperforms the rest of the algorithms, the huge complexity makes it impractical. The next best algorithm is PDM. It is a distributed algorithm and is suitable for the sensor network environment. Unfortunately, it does not perform well when there are only a few anchors as shown in Figure 2. We decided to select MDS and PDM based on the following requirements.

- **Anchor Density and Placement**

The solution quality of multidimensional scaling is less sensitive to the density and placement of anchors than other algorithms [11]. The role of anchors in MDS is to determine the rotation or shifting of the relative map. The accuracy of the configuration mainly depends on the accuracy of the distance matrix. Thus four or five anchors are sufficient to generate a good solution.

- **Distributed Computation and Complexity**

A sensor network may cover a large geographical area and consists of thousands of nodes, thus a distributed algorithm is more appropriate. Though MDS can minimize the number of anchors needed, running it in a network-wise manner requires huge computation and communication cost. To minimize the computations required, classical MDS is only applied to a subset of sensors so that the problem can be solved by normal sensor individually. Afterwards other normal sensors can

localize themselves through PDM-based which is a truly distributed algorithm.

The major cost of computation of classical MDS and PDM comes from the singular value decomposition (SVD). For classical MDS, the complexity of SVD is  $O(n^3)$  where  $n$  is number of nodes. The complexity for PDM is  $O(m^3)$  where  $m$  is the number of anchors. The complexity of SVD in our phased approach is  $O(m_x^3)$  where  $m_x$  is the total number of primary and secondary anchors. By keeping  $m_x$  as a reasonable number, the complexity of the our approach is similar to the complexity of PDM.

Our algorithm works as follows: In the first phase, some sensors are selected as secondary anchors which are localized through multidimensional scaling. We denote the anchors deployed at the beginning as primary anchors to differentiate them from the secondary one. Nodes which are neither primary anchors nor secondary one are called normal sensors. In the second phase, the normal sensors are localized through proximity distance mapping. The mapping is derived from the primary and secondary anchors altogether and we assume nodes know the number of primary anchors  $k_p$  at deployment. The details of operation are given below:

- 1) The first step is to identify secondary anchors. Each primary anchor sends an invitation packet containing its unique ID, a counter initialized to zero and a value  $k_s$  controlling the number of secondary anchors, to *one* of its neighbors. Normal sensor receiving this packet will perform a Bernoulli trial with a success rate of  $p$ . The success rate  $p$  roughly controls the separation between secondary anchors so that they will not be clustered together. The value of  $p$  can be included in the packet sent by primary anchors or embedded in sensor nodes before deployment. If the outcome is true, the normal sensor increments the counter by one and becomes a secondary anchor. The packet will be forwarded to another neighbour until the counter equals to  $k_s$ . Thus the total number of primary and secondary anchors will be  $k_p \times (k_s + 1)$ . If a secondary anchor receives a packet originated from other primary anchors, the packet will be ignored and forwarded to another node.
- 2) After sending the invitation packet, each primary anchor sends packets containing its unique ID and coordinates to *all* of its neighbours. The packet also bears a field marking the proximity, i.e. the distance or hop count the packet has travelled. The value is initialized to be zero. Secondary anchors will also do what primary anchors do, sending out packets with its unique ID but leaving the coordinates field blank.
- 3) Every node (including anchors) receiving a proximity packet from an anchor (either primary or secondary) will store its ID and the proximity value. If a packet from a particular anchor has been received before, the node examines the proximity and check whether it is larger than the stored proximity. If it is larger than the stored

value, the packet will be discarded. Otherwise, the stored value and the proximity field of the packet will be updated and the packet will be forwarded to other neighbours. Thus the stored proximity always reflects the shortest path distance or hop count from a particular anchor.

- 4) After an anchor  $x$  has discovered its proximities to all anchors, it will send the proximities it has collected to other anchors and wait for other anchors to do the same thing. When all anchors have distributed the proximities to their counterparts, each anchor knows the proximity information between every pair of anchors. Each secondary anchor can now determine its location through classical MDS.
- 5) After the first phase, each secondary anchor also knows the position estimates of other secondary anchors as MDS provides a configuration about the primary and secondary anchors. Thus the proximity distance mapping  $\mathbf{T}$  can be calculated immediately as follows:

Let  $\mathbf{P}$  be the proximity matrix that  $p_{ij}$  is the proximity measure between anchors  $i$  and  $j$  where  $p_{ii} = 0$ . The proximity can be hop count or cumulative path distance between anchors. Similarly, let  $\mathbf{L}$  be a geographical distance matrix that  $l_{ij}$  is the geographical distance between anchors  $i$  and  $j$ .  $l_{ij}$  can be calculated by utilizing the position estimates obtained from the first phase.  $\mathbf{P}$  and  $\mathbf{L}$  are square matrices with  $k_p \times (k_s + 1)$  columns. The PDM  $\mathbf{T}$  is defined as a linear mapping that maps matrix  $\mathbf{P}$  to matrix  $\mathbf{L}$  which the following error is minimized:

$$e_i = \|l_i^T - t_i P\|^2 \quad (1)$$

and

$$\mathbf{T} = \mathbf{L}\mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1} \quad (2)$$

The mapping  $\mathbf{T}$  can be calculated by using *singular value decomposition* (SVD). Every secondary anchor calculates the PDM. The mapping and the position estimates of secondary anchors obtained from the first phase are distributed to the normal nodes nearby.

- 6) Normal sensor node  $s$  uses the mapping  $\mathbf{T}$  to process the proximity vector  $\mathbf{p}_s$  it has stored when it aided anchors exchanging proximity information.

$$\mathbf{l}_s = \mathbf{T}\mathbf{p}_s \quad (3)$$

Finally, the node position is calculated by multilateration with the processed proximity vector and the position information of primary and secondary anchors.

#### IV. SIMULATION

To justify our proposal, extensive simulations are conducted to study the performance of MDS+PDM. Simulations are run under MatLab Version 7.2. Two hundred nodes are uniformly distributed in a  $10r \times 10r$  square, where  $r$  is an arbitrary measurement unit. The connectivity is controlled by varying the communication range of sensors. Nodes are capable of measuring the distance away from its one-hop neighbours.

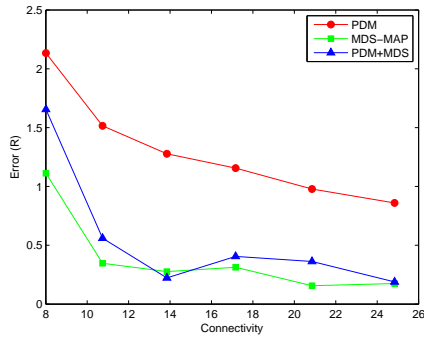


Fig. 2. 3 uniformly distributed anchors,  $\alpha = 0.05$

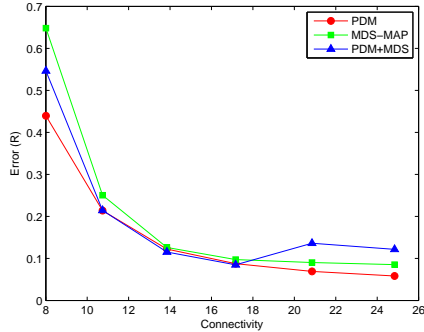


Fig. 3. 10 uniformly distributed anchors,  $\alpha = 0.05$

Measurements are subjected to random errors. The deviation is governed by a normal random process with zero mean and standard deviation equal to  $\alpha$ , i.e.

$$\hat{d} = d * (1 + N(0, s))$$

The estimation error is normalized by the communication range  $R$ . For MDS+PDM, we randomly pick 20 normal sensors as secondary anchors.

#### A. Effects of the number of primary anchors

Figure 2 shows the performance of PDM, MDS-MAP and MDS+PDM under different degrees of connectivity. The standard deviation of noise is equal to 0.05. Three primary anchors are randomly placed in the network. The values presented are the average position error of all nodes obtained from 30 topologies. With merely 3 anchors, MDS+PDM performs consistently better than using PDM alone. Meanwhile, the phased approach gives comparable accuracy to MDS-MAP which is well-known for its performance with very few anchors. It should be noted that the complexity of MDS+PDM is much smaller than MDS-MAP and the latter is a centralized algorithm. With mean connectivity of 13.8, the average position error obtained by MDS+PDM is around  $0.22R$  while the one obtained by PDM is around  $1.27R$ , almost six times of that of MDS+PDM.

Figure 3 shows the performance when the number of anchors is increased to 10. The accuracy of PDM improves

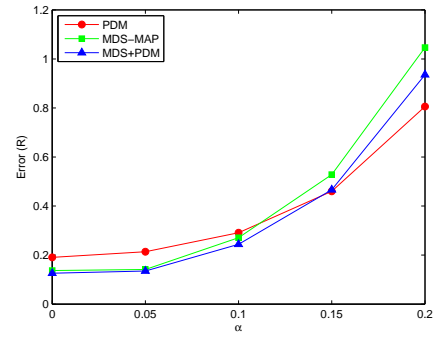


Fig. 4. 5 uniformly distributed primary anchors, Connectivity=13.8

drastically. Though adding extra anchors can improve the performance of MDS-MAP, anchors are only used to determine the linear transformation. MDS-MAP and MDS+PDM enjoy less benefits from further increase of anchors.

#### B. Sensitivity to noise

Figure 4 presents the performance under different degrees of measurement errors. The accuracy of PDM surpassed MDS-MAP and MDS+PDM when  $\alpha$  grows beyond 0.1 and 0.15 respectively. The errors obtained from MDS-MAP, PDM and MDS+PDM for  $\alpha$  equals to 0.15 are  $0.56R$ ,  $0.46R$  and  $0.47R$  respectively. Since the performance of MDS-MAP greatly depends on the distance matrix, MDS-MAP is the least robust algorithm against noise compared with PDM and MDS+PDM. As proximity information is preprocessed by the proximity-distance map  $\mathbf{T}$ , PDM is less susceptible to noise. MDS+PDM gives very good result when noise is minimal. Though it is less immune to larger noise as compared to PDM, our phased approach still performs better than MDS-MAP.

#### C. Effects of anchor placement

Recall that PDM gives the best result when there are 10 anchors scattered in the network. There is no reason to adopt MDS-MAP nor MDS+PDM as PDM gives better accuracy and less complexity. However this is only true when anchors can be uniformly distributed across a network, which is not always feasible. To study the effect of anchor placement on MDS+PDM, we adopt the same topologies generated for previous simulations but anchors are not distributed randomly. They are all randomly placed in a  $3r \times 3r$  square at the corner of the network. From Figure 5, we see that MDS+PDM is capable of providing accurate estimates even though anchors are not uniformly distributed in the network. For connectivity about 13.8, 10 anchors and  $\alpha$  equals to 0.1, the average error of MDS+PDM is  $0.58R$  while the average error of PDM is  $1.05R$  which is almost a double of MDS+PDM one. The corresponding errors of MDS+PDM and PDM when anchors are distributed uniformly are  $0.22R$  and  $0.18R$ , respectively. The secondary anchors spread across the network can mitigate the adverse effects from limited number of anchors and clustered placement of primary anchors. The performance of

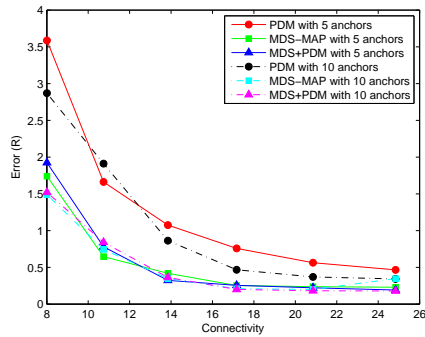


Fig. 5. Clustered anchors, Connectivity=13.8,  $\alpha=0.05$

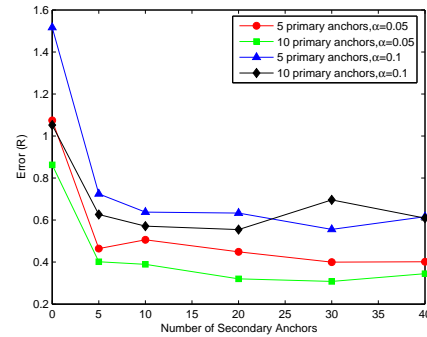


Fig. 7. Clustered primary anchors, Connectivity=13.8

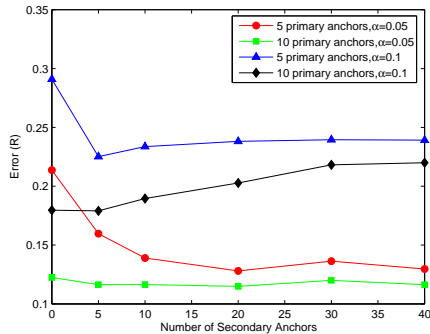


Fig. 6. Uniformly distributed primary anchors, Connectivity=13.8

PDM suffers a lot from the clustered placement of anchors. It is because the accuracy of PDM relies heavily on how much the relationship and characteristic between the proximity and geographical distance can be captured in the mapping  $\mathbf{T}$ . If anchors can be scattered across the network, network-wise characteristic can be captured. On the other hand, if anchors are confined within a particular region, only local characteristic within that region can be captured. Hence, the accuracy drops significantly when anchors are squeezed in a region.

#### D. Effects of the number of secondary anchors

By considering the computation and communication costs of MDS, the number of secondary anchors should be minimal given that the performance can be kept satisfactory. To study the effect of the number of secondary anchors, we vary the number of secondary anchors from 0 (i.e. pure PDM) to 40. Figure 6 and Figure 7 show the corresponding performance for uniformly distributed primary anchor and clustered primary anchor respectively. In general, the average estimation error decreases as the number of secondary anchors increases but further increase of secondary anchors beyond 10 does not give any significant improvement. Furthermore, the increase of secondary anchor hurts the performance when there are 10 primary anchors *uniformly distributed* across the network. It is because PDM outperforms MDS when there are sufficient anchors *uniformly distributed* across the network.

## V. CONCLUSION

In this paper, we presented a phased localization algorithm which performs well with very few anchors. Reducing the number of anchors deployed can reduce the cost of sensor network substantially. Through extensive simulations, we demonstrated that the proposed algorithm provides results as accurate as MDS-MAP even the number of anchors is minimal. Our algorithm also exhibits less complexity by employing secondary anchors and PDM in the second phase of localization. The algorithm can be implemented in a distributed fashion. Since performance of most existing algorithms degrade significantly when anchors are not placed around the perimeter of the network, we also investigated the effect of anchor placement. Simulation results showed that our proposal is less susceptible to anchor placement than other algorithms.

## REFERENCES

- [1] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *IEEE Globecom*, 2001, pp. 2926 – 2931.
- [2] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *ACM MobiHoc*, 2003, pp. 201–212.
- [3] Y. Shang and W. Ruml, "Improved MDS-based localization," in *IEEE INFOCOM*, 2004.
- [4] A. Savvides, H. Park, and M. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *ACM WSN*, 2002, pp. 112–121.
- [5] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *IEEE IPSN*, 2004, pp. 46–54.
- [6] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithm for distributed ad-hoc wireless sensor networks," in *USENIX Technical Annual Conference*, 2002, pp. 317–327.
- [7] H. Lim and J. C. Hou, "Localization for anisotropic sensor networks," in *IEEE INFOCOM*, 2005.
- [8] L. Doherty, K. Pister, and L. Ghaoui, "Convex position estimation in wireless sensor networks," in *IEEE INFOCOM*, 2001.
- [9] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," in *Computer Networks*, vol. 43, no. 4, 2003, pp. 499–518.
- [10] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M. B. Srivastava, "On the error characteristics of multihop node localization in ad-hoc sensor networks," in *IEEE IPSN*, 2003, pp. 317–332.
- [11] Y. Shang, H. Shi, and A. A. Ahmed, "Performance study of localization methods for ad-hoc sensor networks," in *IEEE MASS*, 2004, pp. 184–193.
- [12] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Applications*, 2nd ed. New York: Springer, 2005.
- [13] A. A. Ahmed, H. Shi, and Y. Shang, "Sharp: A new approach to relative localization in wireless sensor networks," in *IEEE ICDCS*, 2005.