A WEIGHTED FASTMAP ALGORITHM FOR WIRELESS SENSOR NODES LOCALIZATION

Waleed A. Saif [1], *D. C. McLernon* [1] *and M. Ghogho* [1, 2]

[1] School of Electronic & Electrical Engineering, the University of Leeds, Leeds LS2 9JT, UK
[2] International University of Rabat, Rabat, Morocco
email:{eenwas, d.c.mclernon, m.ghogho}@leeds.ac.uk

ABSTRACT

In this paper a weighted Fastmap (WFM) algorithm is proposed in which more than one pair of anchor nodes is used to evaluate the one-dimensional coordinates of the unknown nodes while in the original Fastmap (FM) algorithm only one pair of anchor nodes was employed. However, some nodes might be too far from the anchor nodes thus resulting in a high coordinate estimation error. This motivates the use of the WFM but at a slight increase in the computational complexity. The optimal WFM weights were determined via (constrained) minimization of the mean-squared error (MSE) of the estimated node coordinates. A simplification of the WFM is also introduced, called the averaged FM (AFM), where the complexity is reduced at the expense of the WFM performance. Both the WFM and AFM exhibit improved performance over the original FM algorithm.

1. INTRODUCTION

A wireless sensor network (WSN) is a network of a large number of wireless devices able to cooperatively monitor many applications and tasks. Originally, the development of WSNs was motivated by military applications such as battlefield surveillance. However, WSNs are now proposed in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation and traffic control [1], [2].

One important feature of sensor networks is that the position of the sensor nodes does not need to be engineered or pre-determined and this allows random deployment of sensors in the monitored region. This means that sensor network protocols and algorithms must possess self-organizing capabilities [3].

Since the main purpose of WSNs is gathering information or data for a specific task or application, the data may be useless if the location of the sensor node transmitting the data is not known itself. For some applications the actual location of the sensor node is the required information to be transmitted. That is why WSNs and position location of the sensors are often associated with each other. So, accurate and low-cost sensor localization is a critical requirement for the deployment of wireless sensor networks in a wide variety of applications [4].

Wireless location has received considerable attention over the past 10 years. Research in wireless position location was boosted by the requirements of the U.S. Federal Communications Commission (FCC) that asked for all wireless service providers, including cellular, broadband PCS and wide-area specialized mobile radio (SMR) licensees, to provide location information to Emergency 911 (E-911) public safety services. The wireless carrier should be able to report the location of all E-911 callers with an accuracy of 125 m (410 ft) in 67% of cases [5], [6].

Recently, an algorithm called Fastmap (FM) has been proposed for node localization in WSNs [7]. The main advantage of this algorithm is its low computational cost compared to other algorithms doing the same task - such as metric Multidimensional Scaling (MDS) [7], [8]. This algorithm requires three anchor nodes (in the case of 2D localization) to be located on the vertices of a right-angled triangle. In [7] it has also been stated (without further analysis) that the anchor nodes should be placed on the edge of the network – this was based on simulation results. In [9], the mathematical analysis for the FM algorithm was carried out. Moreover, a modified version of the FM was later proposed and analysed in [9].

It was also observed in [9] that for nodes far located from the anchor nodes, the FM coordinate estimation error is large due to the high distance measurement error. This problem motivates the use of a weighted version of the FM algorithm where now more than one pair of anchor nodes is used to locate the one-dimensional coordinates of unknown nodes. So in the proposed algorithm the result is that we allocate a larger weight to the estimate generated by FM using anchor node pairs closer to the unknown node (compared to the estimates from the anchor node pairs further located from the unknown node). The price paid for this new weighted FM (WFM) algorithm is a slight increase in the complexity. The analysis of the proposed WFM algorithm in terms of the unknown node coordinate MSE is presented. Due to the paper size limitation the analysis is implemented (without loss of generality) on only the x-coordinate (as the procedure is similar in the case of the y-coordinate). Finally, a simplification of the WFM is also presented in which the weights are all constrained to be equal. We will call this approach the averaged FM (AFM).

A brief description of the original FM algorithm is given in section 2. The bias and the variance of the estimated x-coordinate are derived in section 3. Our proposed WFM algorithm and its analysis in terms of MSE is presented in section 4. Finally, simulation results and conclusions are given in sections 5 and 6 respectively.

2. THE FASTMAP ALGORITHM

The FM algorithm was first proposed by Falutsos and Lin in 1995 [10]. It is viewed as a distance mapping algorithm or a dimension reduction method. The distance matrix, also known as a (dis)similarity matrix, is the only given input and it represents a map of N points in p-dimensional space. The FM algorithm aims to find the coordinates of the N points (O_i) or objects in a k-dimensional space ($k < p$) whose Euclidean distances will match the distances of the given $N \times N$ distance matrix.

The first step, and a basic element of the FM algorithm, is to select two objects (O_a and O_b) to form the projection line. These objects are called pivots [10]. The two pivots should be selected such that the distance (d_{ab}) between O_a and O_b is maximized. To accomplish such a task Falutsos and Lin proposed a linear heuristic algorithm, based on "*choose-distant-objects*" [10].

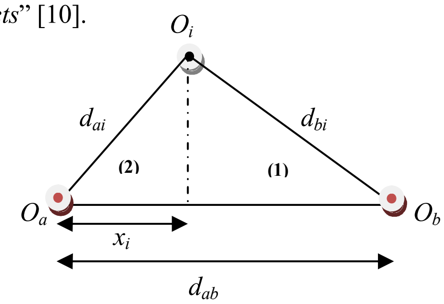

Figure 1: Calculation of coordinate x_i via projection onto the line O_a - O_b.

The second step is to project any other i-th object O_i onto the pivot line and first find its x-coordinate by employing the cosine rule [10]. Mathematically, to get the first coordinate of O_i, we apply Pythagoras' Theorem for the two triangles (as shown in Fig. 1) and subtract the two equations to get the coordinate x_i (measured from O_a on the line connecting O_a and O_b):

$$x_i = \frac{d_{ai}^2 + d_{ab}^2 - d_{bi}^2}{2d_{ab}}, \quad i = 1,2,\cdots,N \tag{1}$$

where d_{ij} is the distance between any two objects i and j.

The third step is to consider an imaginary hyper-plane H that is perpendicular to the line O_a - O_b and project all the objects onto this hyper-plane. If O_i and O_j are two objects and their projections on H are O'_i and O'_j respectively, then it can be shown that the distance d'_{ij} between the projected objects on H is given by:

$$d_{ij}'^{\,2} = d_{ij}^2 - \left(x_i - x_j\right)^2, \; i,j = 1,2,\cdots,N\,. \tag{2}$$

The ability to compute the distance d'_{ij} allows for further projection onto a second line which lies on the hyper-plane, H, and is orthogonal to the first line O_a - O_b. This can solve the problem for the 2-dimensional 'target' space. The same steps above can be applied recursively, k times, thus solving the problem for any k-dimensional space [10].

The main advantage of the FM algorithm is its low computational complexity compared to other dimensional reduction algorithms such as metric MDS [7]. If N is the total number of objects and k is the dimensionality of the target space, then the total FM cost is $O(kN)$ compared to $O(N^3)$ for metric MDS. However, FM is sensitive to outliers and coordinates alignment [7], [11].

3. THE BIAS AND VARIANCE OF THE FASTMAP ALGORITHM ESTIMATE

In this section, the mathematical representation of the bias and variance of the x-coordinate estimate for the FM algorithm will be derived. Without loss of generality the same approach can be applied to the y-coordinate. Recall from [7] and [9] that the pairwise distance estimate will contain some measurement errors which are amplified with the increasing distance between the nodes. Such an error can be modelled as multiplicative normal noise with zero mean and a certain standard deviation. Thus the measured distance $\hat{d}_{ij}$ will be given by:

$$\hat{d}_{ij} = d_{ij} + d_{ij} n_{ij}, \quad i,j = 1,2,\cdots,N \tag{3}$$

where n_{ij} is a zero-mean, normal, random variable with variance σ_n^2.

Now, consider a network with N nodes whose positions are uniformly distributed between 0 and A, i.e., $x_i \sim U(0,A)$ and $y_i \sim U(0,A)$, where A is the length of the side of the network, and hence the mean position of all nodes in the network will be $A/2$.

If no measurement error is incurred in the distance measurement, then the x-coordinate of node i is given by (1). However, due to the measurement error, (1) can be re-written as follows (i.e., for the noisy estimate):

$$\hat{x}_i = \frac{\hat{d}_{ai}^2 + d_{ab}^2 - \hat{d}_{bi}^2}{2d_{ab}}\,. \tag{4}$$

Note that in [9], it was assumed the distance between the two anchor nodes does not contain any measurement error since their positions are exactly known with respect to each other. This assumption will give the following result:

$$\hat{x}_i = x_i + \frac{(2n_{ai} + n_{ai}^2)d_{ai}^2 - (2n_{bi} + n_{bi}^2)d_{bi}^2}{2d_{ab}}\,. \tag{5}$$

However, if noise among anchor nodes is also to be considered, then (4) can be re-written to give the following estimate for x_i:

$$\tilde{x}_i = \frac{\hat{d}_{ai}^2 + \hat{d}_{ab}^2 - \hat{d}_{bi}^2}{2\hat{d}_{ab}} = \hat{x}_i K_{ab} + \frac{d_{ab}}{2}\left(1 + n_{ab} - K_{ab}\right) \tag{6}$$

where

$$K_{ab} = \frac{1}{1+n_{ab}} . \quad (7)$$

The mean of the estimate $\tilde{x}_i$ in (6) is given by:

$$\mu_{\tilde{x}_i} = \left(x_i + \sigma_n^2 \frac{\left(d_{ai}^2 - d_{bi}^2\right)}{2d_{ab}} \right) E\left[K_{ab}\right] + \frac{d_{ab}}{2}\left(1 - E\left[K_{ab}\right]\right) \quad (8)$$

and the bias of (6) can be evaluated as follow:

$$\text{bias}[\tilde{x}_i] = b_{\tilde{x}_i} = E[\tilde{x}_i] - x_i = \mu_{\tilde{x}_i} - x_i . \quad (9)$$

Substituting (8) into (9) we get:

$$b_{\tilde{x}_i} = \frac{d_{ai}^2 - d_{bi}^2}{2d_{ab}} \left\{ \sigma_n^2 + \left(\sigma_n^2 + 1\right)\left(E[K_{ab}] - 1\right) \right\} . \quad (10)$$

To evaluate $E\left[K_{ab}\right]$ we can use a Taylor's series approximation:

$$\begin{aligned} K_{ab} &= \frac{1}{1+n_{ab}}, \qquad \left|n_{ab}\right| < 1 \\ &\approx 1 - n_{ab} + n_{ab}^2 - n_{ab}^3 + n_{ab}^4 - n_{ab}^5 + \ldots \end{aligned} \quad (11)$$

$$\Rightarrow E\left[K_{ab}\right] \approx 1 + \sigma_n^2 + 3\sigma_n^4 + 15\sigma_n^6 + \ldots . \quad (12)$$

Note that K_{ab} will be 1 if the noise measurement in the anchor nodes is neglected and in this case (10) becomes:

$$b_{\hat{x}_i} = \sigma_n^2 \frac{d_{ai}^2 - d_{bi}^2}{2d_{ab}} . \quad (13)$$

The variance of estimate $\tilde{x}_i$ in (6) can also be evaluated as follows:

$$\text{var}[\tilde{x}_i] = \sigma_{\tilde{x}_i}^2 = E\left\{\left(\tilde{x}_i - \mu_{\tilde{x}_i}\right)^2\right\} = E[\tilde{x}_i^2] - \mu_{\tilde{x}_i}^2 \quad (14)$$

where:

$$E[\tilde{x}_i^2] = E\left[\left(\hat{x}_i K_{ab} + \frac{d_{ab}}{2}(1 - K_{ab} + n_{ab})\right)^2\right] \quad (15)$$

and $\mu_{\tilde{x}_i}$ is defined in (8).

Once again, for noiseless distance measurements among anchor nodes, then (14) can be simplified to give:

$$\sigma_{\hat{x}_i}^2 = \frac{\sigma_n^2\left(\sigma_n^2 + 2\right)\left(d_{ai}^4 + d_{bi}^4\right)}{2d_{ab}^2} . \quad (16)$$

The MSE of the x_i estimate is obtained as follows using (13) and (16):

$$\begin{aligned} \text{MSE}(\hat{x}_i) &= E[(\hat{x}_i - x_i)^2] = \text{var}(\hat{x}_i) + \left(\text{bias}(\hat{x}_i)\right)^2 \\ &= \frac{(4\sigma_n^2 + 3\sigma_n^4)(d_{ai}^4 + d_{bi}^4) - 2\sigma_n^4 d_{ai}^2 d_{bi}^2}{4d_{ab}^2} . \end{aligned} \quad (17)$$

Note that in section four, our proposed algorithm will require the calculation of the bias $(b_{\hat{x}_i})$ from (13) and the variance $(\sigma_{\hat{x}_i}^2)$ from (16). However, these need the exact distance measurements (i.e., d_{ai} and d_{bi}) which we will not have in a practical scenario. So in our algorithm, both (13) and (16) will be approximated using the estimated (or measured) distances (i.e., $\hat{d}_{ai}$ and $\hat{d}_{bi}$). Finally, we will assume that the noise variance (σ_n^2, in both (13) and (16)) is also known a-priori.

4. WEIGHTED FASTMAP ALGORITHM (WFM)

The MSE in (17) may sometimes be greater than we can tolerate. This motivates us to examine an alternative way to reduce this MSE, and so we propose to use more than one pair of anchor nodes (with appropriate weighting to combine the different estimates).

To begin with, assume that we have M pairs of anchor nodes and that each pair is placed as shown in Fig. 2 (red circles). Let $\hat{x}_{im}$ be the estimate of the x_i-coordinate obtained using FM in (4) with the m-th anchor node pair. Equation (6) can be used if the noise in the distance measurements among the anchor nodes is also to be considered. But, for ease of analysis, the noise amongst the anchor nodes will be neglected and we will assume that each anchor node knows exactly its position as well as other anchor nodes positions. Then, for M anchor node pairs, we propose the following weighted estimate of the x_i-coordinate:

$$\hat{x}_i = \sum_{m=1}^{M} \alpha_{im} \hat{x}_{im} , \quad i = 1,2,\cdots,N \quad (18)$$

where α_{im} is the weight value applied to the $\hat{x}_{im}$ estimate subject to the constraint:

$$\sum_{m=1}^{M} \alpha_{im} = 1 , \quad i = 1,2,\cdots,N . \quad (19)$$

In order to evaluate the MSE of the x_i-coordinate estimate for the WFM, both the variance and the bias of (18) should be evaluated. Since the MSE was defined in (17), it is clear that (subject to (19)):

$$\begin{aligned} \text{MSE}\left(\hat{x}_i\right) &= \sum_{m=1}^{M} \alpha_{im}^2 \text{var}(\hat{x}_{im}) + \left(\sum_{m=1}^{M} \alpha_{im} \text{bias}(\hat{x}_{im})\right)^2 \\ &= \sum_{m=1}^{M} \alpha_{im}^2 \sigma_{im}^2 + \left(\sum_{m=1}^{M} \alpha_{im} b_{im}\right)^2 , \sum_{m=1}^{M} \alpha_{im} = 1 \end{aligned} \quad (20)$$

where for simplicity of notation we have written $\sigma_{im}^2 = \text{var}[\hat{x}_{im}] = E[(\hat{x}_{im} - E[\hat{x}_{im}])^2]$ and $b_{im} = \text{bias}[\hat{x}_{im}] = E[\hat{x}_{im}] - x_i$.

Now re-write (20) as:

$$\begin{aligned}\text{MSE}(\hat{x}_i) &= \boldsymbol{\alpha}_i^T \mathbf{V}_i \boldsymbol{\alpha}_i + \boldsymbol{\alpha}_i^T \mathbf{b}_i \mathbf{b}_i^T \boldsymbol{\alpha}_i \\ &= \boldsymbol{\alpha}_i^T (\mathbf{V}_i + \mathbf{b}_i \mathbf{b}_i^T) \boldsymbol{\alpha}_i \qquad (21) \\ &= \boldsymbol{\alpha}_i^T \mathbf{A}_i \boldsymbol{\alpha}_i, \quad \text{with} \quad \boldsymbol{\alpha}_i^T \mathbf{e} = 1\end{aligned}$$

where

$\mathbf{e}$ is a column vector of ones, $\mathbf{A}_i = \mathbf{V}_i + \mathbf{b}_i \mathbf{b}_i^T$, $\boldsymbol{\alpha}_i = [\alpha_{i1} \quad \alpha_{i2} \quad \dots \quad \alpha_{iM}]^T$, $\mathbf{b}_i = [b_{i1} \quad b_{i2} \quad \dots \quad b_{iM}]^T$ and

$$\mathbf{V}_i = \begin{bmatrix} \sigma_{i1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{i2}^2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma_{iM}^2 \end{bmatrix}.$$

To find the values of $\boldsymbol{\alpha}_i$ such that (20) or (21) is minimized we form the Lagrangian:

$$L(\boldsymbol{\alpha}_i, \lambda) = \boldsymbol{\alpha}_i^T \mathbf{A}_i \boldsymbol{\alpha}_i + \lambda(\boldsymbol{\alpha}_i^T \mathbf{e} - 1). \qquad (22)$$

Taking the partial derivative of (22) with respect to $\boldsymbol{\alpha}_i$ gives:

$$\frac{\partial l(\boldsymbol{\alpha}_i, \lambda)}{\partial \boldsymbol{\alpha}_i} = 2\mathbf{A}_i \boldsymbol{\alpha}_i + \lambda \mathbf{e} = \mathbf{0} \qquad (23)$$

$$\Rightarrow \boldsymbol{\alpha}_i = -\frac{\lambda}{2} \mathbf{A}_i^{-1} \mathbf{e}, \quad i = 1, 2, \cdots, N. \qquad (24)$$

Combing the constraint ($\boldsymbol{\alpha}_i^T \mathbf{e} - 1$) with (24) gives:

$$\lambda = \frac{-2}{\mathbf{e}^T \mathbf{A}_i^{-1} \mathbf{e}} \qquad (25)$$

and substituting back into (24) we get:

$$\boldsymbol{\alpha}_i = \frac{\mathbf{A}_i^{-1} \mathbf{e}}{\mathbf{e}^T \mathbf{A}_i^{-1} \mathbf{e}}, \qquad i = 1, 2, \cdots, N. \qquad (26)$$

Now, the terms σ_{im}^2 and b_{im} that make up (26) follow from (13) and (16) and so to get the minimum MSE (MMSE) of the WFM estimate, substitute (26) into (21) to give:

$$\text{MMSE}(\hat{x}_i) = \frac{1}{\mathbf{e}^T \mathbf{A}_i^{-1} \mathbf{e}}. \qquad (27)$$

Finally, one last case is worth consideration. If all the weights are chosen as equal (i.e., $\alpha_{im} = \frac{1}{M}, \forall m, i$) then from (21) the MMSE becomes:

$$\text{MMSE}(\hat{x}_i) = \frac{1}{M^2} \mathbf{e}^T \mathbf{A}_i \mathbf{e}. \qquad (28)$$

As mentioned in the introduction, this scenario is called the averaged FM (AFM) and its advantage is the reduced complexity, i.e., the weighting coefficients do not need to be calculated via (26).

5. SIMULATION RESULTS

We consider a network of N=100 sensor nodes, uniformly distributed within a square of unit area. The term n_{ij} in (3) is Gaussian with zero mean and variance σ_n^2. We consider five different placements of the anchor nodes as shown in Fig. 2. All simulations were obtained with 100 independent Monte Carlo runs.

Note that the estimation of $\boldsymbol{\alpha}_i$, via (26) requires both b_{im} and σ_{im}^2 for m=1, 2... , M, which come from (13) and (16) respectively. This requires knowledge of σ_n^2, $d_{a_m i}$ and $d_{b_m i}$, where a_m refers to the m-th anchor node O_a, etc. We will assume that σ_n^2 is known in advance but in practice we do not have a-priori knowledge of the exact $d_{a_m i}$ and $d_{b_m i}$. Thus, both $\hat{d}_{a_m i}$ and $\hat{d}_{b_m i}$ will be used instead.

Figure 3 shows the (sample) mean absolute error (MAE), $\frac{1}{N}\sum_{i=1}^{N} |\hat{x}_i - x_i|$, (where $\hat{x}_i$ in (18) uses α_{im} from (26)), versus the noise variance σ_n^2, for the original FM, WFM and AFM with M=2 and 5 pairs of anchor nodes. We can clearly see the substantial improvement of the WFM over the original FM algorithm. Also, we can see the difference between WFM (M=2) and WFM (M=5), but the improvement is not as substantial. This leads us to ask: how many pairs of anchor nodes should be used for WFM?

Figure 4 shows the MAE versus the number of anchor node pairs (M) for the WFM and AFM (in which all weight coefficients are equal) for three different noise variances. With M=1, we get the original FM. In general, the WFM algorithm outperforms the AFM, especially at higher noise variance. However, as the number of pairs increases, the improvement in both WFM and AFM flattens out.

One final remark is on the computational complexity of the WFM algorithm compared to the original FM. In the original FM the complexity is of the order of $O(kN)$ while it is $O(kMN)$ for the WFM, where k is the dimensionality of the target space. It is clear that the complexity of the WFM is still linear in the total number of unknown nodes N. However, the complexity increases by a factor M - the number of pairs of anchor nodes used in the localization process.

6. CONCLUSION

In this paper a WFM algorithm for sensor node localization in a wireless sensor network was proposed. The motivation for the WFM algorithm comes from the poor performance of the original FM compared to some other node localization methods.

It was observed that the original FM produces bad coordinate estimation for nodes that are distant from the anchor nodes. This is due to the high measurement error incurred in the distance estimation between the sensor nodes and the anchor nodes. To solve this problem a weighted version of the FM algorithm produces a better coordinate estimate of the unknown nodes at the cost of a slight increase in the complexity. The optimum weighting coefficients of the WFM algorithm in terms of the MMSE have been presented.

It has been shown that the WFM algorithm improves the overall performance substantially compared to the original FM algorithm but with a slight increase in the complexity.

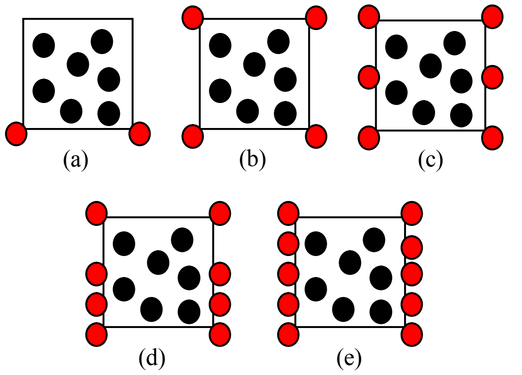

Figure 2: ● (100 unknown nodes-black) ● (Anchor nodes-red) (a) configuration 1, *M*=1; (b) configuration 2, *M*=2; (c) configuration 3, *M*=3; (d) configuration 4, *M*=4; and (e) configuration 5, *M*=5. Note that all "squares" have sides of unit length.

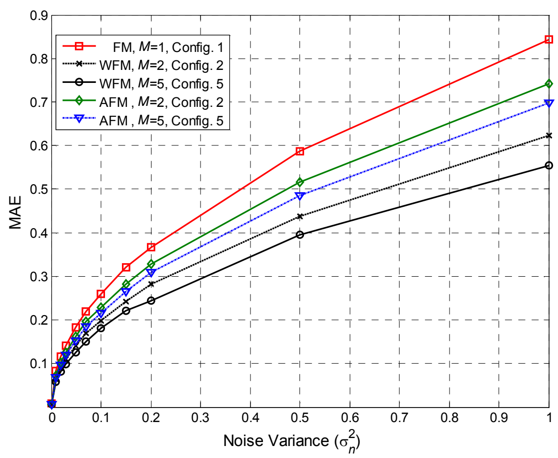

Figure 3: The mean absolute error (MAE) for the *x*-coordinate estimate versus the noise variance (σ_n^2) for the original Fastmap, weighted Fastmap and averaged Fastmap, where *M* is the number of anchor node pairs.

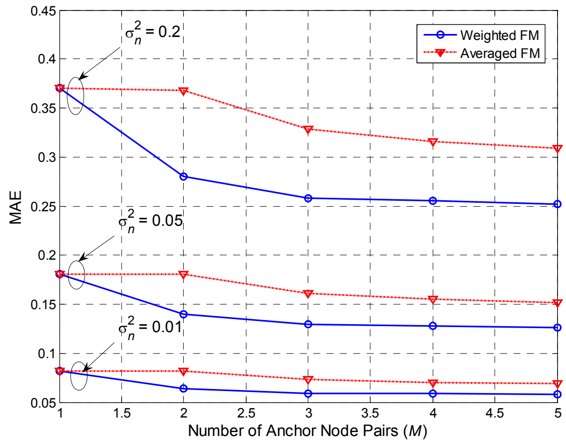

Figure 4: The mean absolute error (MAE) for the *x*-coordinate estimate versus the number of anchor nodes pairs (*M*) for the weighted and averaged Fastmap with different noise variances (σ_n^2).

REFERENCES

[1] Ivan Stojmenović (Editor), *Handbook of Sensor Networks: Algorithms and Architectures*, John Wiley & Sons, Inc. 2005.

[2] K. Romer and F. Mattern, "The design space of wireless sensor networks", *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54 – 61, Dec. 2004.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey", *Computer Networks (Elsevier)*, 38 (4), pp. 393–422, 2002.

[4] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses and N. S. Correal, "Locating the Nodes: Cooperative Localization in Wireless Sensor Network", *IEEE Signal Processing Magazine*, vol.22, no.4, pp. 54-69, July 2005.

[5] J.J. Caffery, *Wireless Location in CDMA Cellular Radio Systems*. Norwell, MA: Kluwer, 1999.

[6] J.J. Caffery and G. L. Stuber, "Overview of Radiolocation in CDMA Cellular Systems", *IEEE Communication Magazine*, vol. 36, no. 4, pp. 38–45, Apr. 1998.

[7] G. Latsoudas and N.D. Sidiropoulos, "A Fast and Effective Multidimensional Scaling Approach for Node Localization in Wireless Sensor Networks", *IEEE Transactions on Signal Processing*, vol. 55, issue 10, pp.5121-5127, Oct. 2007.

[8] T. Cox and M. Cox. *Multidimensional Scaling*, Chapman & Hall CRC, 2nd edition, 2001.

[9] W.A. Saif, M. Ghogho and D.C. McLernon, "A Modified Fastmap Algorithm for Node Localization in Wireless Sensor Networks", *IEEE 9th Workshop on Signal Processing Advances in Wireless Communications*, pp. 251 – 255, 2008.

[10] C. Faloutsos and K. Lin, "Fastmap: A Fast Algorithm for Indexing, Data-mining and Visualization of Traditional and Multimedia Datasets", *ACM SIGMOD Conference* (San Jose, CA), pp. 163–174, May 1995.

[11] G. Ostrouchov and N. F. Samatova, "On Fastmap and the Convex Hull of Multivariate Data: Toward Fast and Robust Dimension Reduction", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, issue 8, pp. 1340 – 1343, Aug. 2005.