Localization Applying An Efficient Neural Network Mapping

Li Li
Communications Research Centre
3701 Carling Avenue
P.O. Box 11490, Stn. H
Ottawa, ON K2H 8S2 Canada
li.li@crc.ca

Thomas Kunz
Carleton University
1125 Colonel By Drive
Ottawa, ON K1S 5B6 Canada
tkunz@sce.carleton.ca

ABSTRACT
Node location information is essential for many applications in Autonomic Computing. This paper presents and evaluates a new cooperative node localization scheme. We apply an efficient non-linear data mapping technique, the Curvilinear Component Analysis (CCA), to produce accurate node position estimates employing only a small number of anchor nodes. Being a light-weight neural network, CCA has the learning ability to self-organize maps of nodes, and to project node coordinates with improved accuracy and efficiency. We present the distributed CCA-MAP scheme that derives node locations in either range-based or range-free scenarios. Unlike other schemes, no further refinement is needed to improve the position estimates generated by the devised CCA projection method. Through extensive simulation studies, we evaluate the performance of our scheme for both regular and irregular networks of different configurations. Comparisons with other related localization schemes are also presented, demonstrating the improved location estimate accuracy and performance efficiency.

Keywords
Localization, nonlinear mapping, simulations, curvilinear component analysis.

1. INTRODUCTION
The vision behind autonomic computing is to reduce system management complexities to drive computing into a new era. We think that the most likely network architectures will have a high percentage of embedded, wireless computing/communication devices. One key piece of infrastructure support in such networks is localization.

In pervasive networks, location information is essential to identify, for example, where sensor readings originated from and to track events and targets. Location information has numerous other applications such as support of context-based routing protocols, geographic routing protocols, location-aware services, and enhanced security protection mechanisms.

AUTONOMICS 2007, 28-30 October 2007, Rome, Italy

DOI 10.4108/ICST.AUTONOMICS2007.2126

As many devices will be embedded in real-world artifacts, localization solutions based on GPS or other external mechanisms will be problematic. Software-based solutions typically require nodes with known locations, called anchor nodes. To reduce the complexity of operating the network infrastructure, we are looking for solutions that do not require a high percentage of such anchor nodes.

Similarly, ranging measurements will be utilized where appropriate to increase the distance reading accuracy, applying spread spectrum or UWB technologies [9]. However, distance measurements may not be relied on in all cases, as complex terrain can still impede the measurement accuracy.. The localization solution of interest here assumes thus the following characteristics: that it requires only a small number of anchor nodes to facilitate the deployment process, and that it achieves high levels of position accuracy with or without the assistance of range measurements.

Recently, various localization schemes have been proposed in the literature. Localization algorithms based on Multidimensional Scaling (MDS) [11][23][24][25][27] are examples of such an approach for deriving sensor node locations, which can be computed in a distributed manner in either range-based or range-free conditions with a small number of anchor nodes, (e.g., at least 3 or 4 anchor nodes in the 2D or 3D space). This class of algorithms also delivers higher node position accuracy (*<20%r* where *r* is the average radio signal radius) when compared with many other approaches. MDS is a non-linear mapping technique applying dimension reduction and data projection that transforms proximity information into a geometric embedding [2]. While preserving the distances between data points and reducing the data dimension from *n* to two (e.g., 2D space) or three (e.g., 3-D space) as required, MDS, similar as other non-linear reduction algorithms, incurs fairly high computational cost of $O(n^3)$ and suffers quite often from local minima. As pointed out in [25], MDS is often good at finding the right general layout of the network, but not the precise locations of nodes. In [11][23][24][25][27], a refinement step using least-square minimization over the results obtained from MDS was applied to achieve better location accuracy.

In this paper, we apply an efficient non-linear mapping technique, the Curvilinear Component Analysis [5], to compute the location coordinates of sensor nodes and other network elements connected wirelessly. CCA is a self-organized neural network performing vector quantization and non-linear projection for dimensionality reduction of multidimensional data sets. Unlike general neural networks, CCA preserves the distances between the

data points of the input space when generating output data space, and exhibits much higher efficiency in its unfolding of the dimensions. The non-linear projection capability of CCA is similar in its goal to other nonlinear mapping methods, such as MDS [2] and Sammon's nonlinear mapping (NLM) [21], in that it minimizes a cost function based on interpoint distances in both input and output spaces. However, CCA outperforms these algorithms in several aspects, making it very useful for node localization. CCA overcomes fairly well the local minima problem and delivers much improved accuracy. In our experiments, CCA was found to map node coordinates with high accuracy without any additional refinement process. Secondly, CCA is more computationally efficient, with a cost of only $O(N)$ per cycle compared to $O(N^2)$ of other non-linear reduction methods and thus results in a much faster data mapping process. In this paper we present a novel approach, applying CCA to localize wireless sensor nodes, and demonstrate the improved accuracy and efficiency compared with other solutions.

The rest of the paper is organized as follows: Section 2 presents a brief overview of related work on localization, in particular in the area of Wireless Sensor Networks. Section 3 discusses CCA and presents our localization scheme that applies CCA; Section 4 presents the performance results and comparisons; and Section 5 concludes the paper.

2. RELATED WORK

Various node localization schemes have been proposed in the literature [3][6][13][15][19][20][22][25]. Based on whether the algorithm uses absolute measurements of node distances or alignment angles in the localization process, the solutions can be classified as range-based [4][15][18][20][22] and range-free [3][10][13][16][23]. While range-based schemes use the absolute measurements in solving the location coordinates, range-free ones do not. It is generally true that range-based solutions often produce finer resolution results when the range errors are kept small. More accurate ranging technologies such as UWB [9] are found to offer promising measurement results. On the other hand, ranging techniques are very environment dependent, e.g., indoor vs. outdoor, path obstacles, etc., and require (additional) hardware cost. When measurements are not accurate, range errors may also have an extended impact when accumulated during the position calculations [16]. In our work, the latest ranging technologies may be explored to generate measurement results as accurately as possible; nevertheless a solution that can be flexibly applied in both range-based and range-free scenarios is our preferred choice.

An important aspect of any localization scheme is its computing algorithm, which is the focus in this paper. Position computation often applies trilateration, triangulation, or multilateration [22]. In a straightforward way, direct reach of at least three anchor nodes is needed for a node to compute its location coordinates [10][15][22]. In computing the position using any of the above methods, algorithms often employ iterations [14][22], starting from the anchor nodes in the network and propagating their locations to all other free nodes, which use this information to calculate their positions. One of the problems of this approach is its low success ratio when the network connectivity level is not very high or when not enough well-separated anchor nodes exist in the network. To localize all the nodes, these algorithms quite often require that 20-40% of the total nodes in the network be anchor nodes [13][16], unless anchor nodes can increase their signal range [10]. To solve the problem of requiring a large number of anchor nodes, some approaches apply limited flooding to allow nodes to "hear" anchor nodes multiple hops away, and to use an approximation of shortest distances over communication paths as the Euclidean distance [16]. However, such hop based distance approximation works rather poorly in anisotropic networks, introducing large position errors [13][16]. In many scenarios, they do not seem to significantly reduce the number of required anchor nodes [13][16]. High network connectivity levels required for the success of such algorithms also give rise to practical concerns, as dense neighborhoods often severely impede radio network throughputs. Additionally, the accumulated location errors also need to be well dealt with [14] to maintain the accuracy of position estimates. Among such schemes, the one proposed in [14] reported one of the best results, where the position estimation error can be reduced to about $5\%r$ in more than 6 iterations when the network connectivity level (number of node neighbors) exceeds *16* and *10%* of nodes are anchors.

Cooperative localization schemes take a quite different approach, formulating the localization problem as a joint estimation problem. Instead of using only constraints between the sensor nodes and anchor nodes, these solutions consider all constraints on inter-node distances and apply optimization techniques to derive location coordinates. Algorithms based on rigidity theory are one example [8][19]. In [19], a heuristic is employed to create a well-spread, fold-free graph layout that resembles the desired network graph. Then a mass-spring model analogy is used to optimize the localization estimates using the minimum energy stage of the mass-spring model. Such an optimization problem is NP-hard and the heuristic needs further studies to prove its convergence [19]. In [8], the conditions required for networks to be localizable using rigidity theory were investigated and a subclass of the grounded graphs were identified which can be computed efficiently. However, the focus was to find the network formation that can be computed. The overall performance of the algorithm for different network formations was not well reported. In [11][23-25][27], connectivity constrains were used to calculate node positions applying multidimensional scaling (MDS). In [1][6], inter-node distance measurements are modeled as convex constraints, and linear programming and semi-definite programming (SDP) methods were adopted to estimate the location of free nodes. These cooperative localization methods are often quite powerful as they require only a small number of anchor nodes and produce highly accurate results. However, such algorithms are often computationally intensive.

The localization approach proposed in this paper belongs to the class of cooperative localization, making joint location estimation using global formulations of inter-node distance constraints. Compared with the existing cooperative algorithms, our approach delivers improved position accuracy and computational efficiency.

3. DISTRIBUTED CCA LOCALIZATION

We first briefly discuss here the non-linear mapping method of Curvilinear Component Analysis (CCA) [5].

3.1 Non-linear Projection Using CCA

Given N input vectors $\{x_i; i = 1,..., N\}$ where each vector x_i is of n dimensions, CCA looks for N output vectors $\{y_i; i = 1,..., N\}$, where each y_i is of s dimensions ($s < n$).

Additionally, the distance between input vector x_i and x_j is preserved between output pair y_i and y_j. That is, given the Euclidean distance between x_i's as: $X_{ij} = d(x_i, x_j)$ and the corresponding distance in the output space $Y_{ij} = d(y_i, y_j)$, CCA pushes Y_{ij} to match X_{ij} for each possible pair (i, j) while minimizing a cost function

$$E = \frac{1}{2}\sum_i \sum_{j \neq i} (X_{ij} - Y_{ij})^2 F(Y_{ij}, \lambda_y) \qquad (1)$$

where $F(Y_{ij}, \lambda_y)$ is a weighing function, often chosen as a bounded and monotically decreasing function, in order to favor local topology conservation (such as that in neural networks of SOM (Self-Organizing Map)) [5]. Decreasing exponential, sigmoid, or Lorentz functions are all suitable choices for *F*.

The computing efficiency of CCA arises from its minimization process of the cost function (1). Compared to other methods such as the stochastic gradient descent ($\Delta y_i \approx -\nabla_i E$) or the steepest gradient descent where one vector y_i is moved every time according to the sum of every other y_j's influence, CCA temporarily pins one y_i and moves all the other y_j around, without regard to interactions amongst the y_j. This not only converges the computation much faster, but also increases the probability to escape from local minima of *E* [5]. The update for each cycle has the following simple expression:

$$\Delta y_j = \alpha(t) F(Y_{ij}, \lambda_y)(X_{ij} - Y_{ij}) \frac{y_j - y_i}{Y_{ij}} \quad \forall j \neq i \qquad (2)$$

Similar to stochastic gradient methods, $\alpha(t)$ decreases with time. This rule for update in each cycle is much lighter than stochastic gradient as only the distances from node *i* to the others need to be computed, instead of all the $N(N-1)/2$ distances in both the input and output spaces. For an adaptation cycle of all nodes (except *i*), the complexity is only in *O(N)* instead of *O(N²)* as in most of the NLM algorithms.

3.2 Mapping Node Coordinates

From the above descriptions, the CCA algorithm preserves the distances between every two data points in the input data space while generating the output data set with each data point having a reduced dimension. The localization problem can then be formulized as the following:

Given a distance matrix $D_{(N\times N)}$ of *N* nodes, find the coordinates of all the points to achieve:

$$\min \sum_{i,j} (d_{ij} - p_{ij})^2 \qquad \text{for } i,j=1,2,...N$$

where

- d_{ij} is the measured/known distance between node *i* and *j*, and
- p_{ij} is the distance between node *i* and *j* computed using the calculated position coordinates of *i* and *j*.

We take the distance matrix $D_{(N\times N)}$ of *N* nodes as the input data set, i.e., $x_{(N\times N)} = D_{(N\times N)}$. Each vector in the input space is thus of *N* dimensions. The output data set contains the *N* vectors each reduced to a dimension of 2 (or 3). Without losing the generality of applicability to both *2D* and *3D* spaces, we take dimension of 2 in the following discussions. Thus the output data set is denoted as $y_{(N\times 2)}$ which is in fact the coordinate matrix of the *N* nodes.

We force the inter-vector distance in the input data space to be $X_{ij} = D_{(N\times N)}$, even though $D_{(N\times N)}$ is not the real distance between vectors in the input data set $x_{(N\times N)}$. Nevertheless, the algorithm works well with such a forced distance value.

In our CCA algorithm, a decreasing exponential function is selected as: $F(Y_{ij}, \lambda) = e^{-\frac{Y_{ij}}{\lambda(t)}}$, where $\lambda(t)$ also decreases with time. Then our CCA algorithm runs in the following two simple steps:

(1) Set the initial output estimation of $y_{(N\times 2)}$ using the mean values of the first two columns of the input data set $x_{(N\times N)}$, adjusted by a uniformly randomized standard deviation of the same column.

(2) In each cycle, select node *i* and compute for each node *j* $(j \neq i)$, the new $y_j(t+1)$ from the current value of $y_j(t)$ using:

$$y_j(t+1) = y_j(t) + \alpha(t) e^{\frac{-Y_{ij}}{\lambda(t)}} (\frac{X_{ij}}{Y_{ij}} - 1)(y_j - y_i) \qquad (3)$$

The following function (equation (4)) can be used to implement (i.e., using $\alpha = \nu$ and $\lambda = \nu$ in (4) below) $\alpha(t)$ and $\lambda(t)$, though other similar functions can be selected instead:

$$\nu(t) = \nu(0) \times \left(\frac{\nu(c)}{\nu(0)} \right)^{\frac{t}{c-1}} \qquad (4)$$

where *c* is the number of total computing cycles, which is often also called the training length in CCA. Sample values for experiments can be chosen as: $\alpha(0) = 0.5$, $\alpha(c) = \alpha(0)/100$, $\lambda(0) = \max\{std_1, std_2, ..., std_N\} \times 3$ and $\lambda(c) = 0.01$, where std_i is the standard deviation for the ith column of the input data set $D_{(N\times N)}$.

Table 1 below illustrates the data dimension reduction power of CCA. Three types of random maps of 100, 200 and 300 nodes in a 10×10 square area are used in the computation. In the maps, all nodes are randomly scattered with the connectivity level of about 10. The input data matrices, i.e., the distance matrices are thus of sizes (100x100), (200x200) and (300x300). The output matrices are of size (100x2), (200x2) and (300x2), respectively, to conform to the point coordinates in the map of 100, 200 and 300 nodes. The CCA algorithm was run using Matlab V7.2 on a Pentium® M processor of 1.60GHz with 1G RAM. More than 10

maps of each size are computed for the experiments. The results show remarkable accuracy for relatively short execution times. The median errors in Table 1 record the average median error of the node coordinates obtained by the algorithm. Here we only study and verify the CCA algorithm for reducing the dimension of the distance matrix to obtain node coordinates. Practically, it is almost impossible to obtain the entire distance matrix in the network to perform such centralized computation. In the next subsection a distributed map algorithm will be applied with the CCA method, which is the more practical solution of our interest.

Table 1: Experiments Using CCA to Compute Node Coordinates from Distance Matrix

100 nodes (input data size 100x100)		200 nodes (input data size 200x200)		300 nodes (input data size 300x300)	
Time(s)	Median Error (r)	Time (s)	Median Error (r)	Time(s)	Median Error (r)
0.358	0.0141	0.685	0.0083	1.124	0.0011
0.796	1.51e-6	1.035	0.74e-4	1.608	0.12e-4

As explained before, CCA runs in iterations/cycles to reduce the data sets. The cost of each cycle is *O(N)*. In Table 1 above, around 12-20 cycles were applied for each input point/data vector. Thus CCA has a complexity of about $O(N^2)$ in total running cost, compared to that of $O(N^3)$ in typical NLM algorithms such as MDS and Sammon's. With the increase of the input data matrix size, the number of cycles required does not increase but often decreases instead to achieve the same resulting accuracy. However, the time spent in each cycle increases with the growing input data size. It should also be noted that when the number of cycles is too low, e.g., at 10 for the above cases in Table 1, though good results can still be obtained, local minima might be experienced. For example, using 10 cycles to run CCA on the maps of 300 nodes for ten times, the results all have median error $<0.01r$ with most of them $<0.001r$ (averaged across all scenarios). But out of the ten scenarios, one had a median error around $0.02r$. As the computing time does not rise significantly with the expansion of the input data size (number of nodes), highly accurate node coordinates can be obtained using CCA with a moderate number of cycles. CCA is thus very powerful in computing the node coordinates using the distance matrix.

The distance matrix for large networks is often not available. Recently, Drineas et al. have proposed algorithms for distance matrix reconstruction for sensor network localization using singular value decomposition [7]. This may provide an option to obtain the distance matrix for the network. We however assume that the distance matrix of the network is unknown. Instead, a distributed map algorithm [23] is adapted in our scheme to compute the node coordinates in the network.

3.3 Distributed Map Algorithm Using CCA

Adopting steps similar to the distributed map algorithm of MDS-MAP [24][25], we propose an alternative distributed map algorithm, the CCA-MAP algorithm. The CCA-MAP scheme, similar to MDS-MAP, builds local maps at possibly each node in the network and then patches them together to form a global map. Differently from MDS-MAP, CCA is employed in computing the node coordinates in the local map. Each node computes its local map using only the local information. If accurate ranging capability is available in the network, local distance between each pair of neighboring nodes is measured and known. Otherwise, only connectivity information is applied to assign a value of 1 to the edge between each neighboring pair of nodes. Then a distance matrix for all the nodes in the R hop neighborhood of node x can be constructed using the shortest distance matrix as approximation. Instead of a fixed $R=2$, as used in MDS-MAP algorithms [24][25], R is variable in CCA-MAP. CCA's reduction technique can generate quite accurate results with a reasonably accurate distance matrix of a small size. In addition, a smaller matrix results in faster computation time. Therefore, we choose R to be 1 if the network is dense; otherwise we set R to 2. In the ranging-based scenarios, a one-hop neighborhood distance matrix of at least 12×12 produced good results. In the range-free cases, the distance matrix is considerably inaccurate using the hop count approximation. Thus a larger distance matrix assists in mapping to the node position coordinates using CCA. Therefore in the range-based CCA-MAP algorithm, for any given node, if its one-hop neighborhood has more than 12 nodes, $R = 1$ can be chosen for improved performance. Otherwise, $R = 2$ is applied. In the range-free CCA-MAP, where the local distance matrix is much less accurate due to the hop count approximation, we set $R = 1$ when the one-hop neighborhood has a very large size (above *30*), otherwise $R = 2$ is chosen. The steps of CCA-MAP are thus as follows:

(1) For each node x, neighbors within R hops are included in building its local map. Compute the shortest distance matrix of the local map and take it as the approximate distance matrix *LD*.
(2) Each node applies the CCA algorithm using local distance matrix *LD* as both the input data set and the distance matrix of the input data set as described in the previous subsection. This generates the relative coordinates for each node in the local map of node x of its R hop neighbourhood.
(3) Merge local maps.
(4) Given sufficient anchor nodes (>=3 for *2D* space and >=4 for *3D*), transform the merged map to an absolute map based on the absolute positions of anchors.

In step 3, we apply a similar approach to that in MDS-MAP in merging local maps. Use a randomly selected node's local map as the starting current map. Each time, the neighbor node whose local map shares the most nodes with the current map is selected to merge its local map into the current map. Two maps are merged/patched using the coordinates of their common nodes. To merge a new local map B into the current map A, a linear transformation (translation, reflection, orthogonal rotation, and scaling) is determined to ensure that the coordinates of the common nodes in map B after transformation best conform with those in current map A.

In the CCA-MAP scheme, no refinement is applied as the results are often satisfactory without further optimization. It was found in the MDS-MAP algorithm [23][24][25] that the refinement process takes most of the computing time, which can be several orders of magnitude more expensive than MDS calculation. Therefore the computing cost is greatly reduced in CCA-MAP when not applying any of the extra optimization process.

Computing of local maps can be distributed to each local node, or can be carried out at more powerful gateway nodes of each cluster should the sensor network have a hierarchical structure to relieve the resource-constrained sensor nodes from

any of the computing and communication demands imposed by localization. The local maps can be merged in parallel in different parts of the network by selected nodes. There is no need for anchor nodes in patching the maps. When at least three anchor nodes are found in the patched map of a subnetwork, an absolute map of the subnetwork can be computed using the coordinates of the anchor nodes to obtain the absolute coordinate values of all the nodes in the map of the subnetwork.

4. PERFORMANCE EVALUATIONS

Three types of networks are selected for performance experiments and comparisons, as illustrated in Figure 1. In all three types of networks, the network area consists of a $10i \times 10i$ square with $i = 1$ set as the placement unit length. The first networks models uniform random networks. 200 nodes are randomly placed inside the $10i \times 10i$ square. The second network models irregular random networks. 160 nodes are randomly placed inside a C-shaped area within the $10i \times 10i$ square. The third network represents uniform grid networks. 100 nodes are placed according to a $10i \times 10i$ grid with the placement error uniformly distributed in the range of [*-20%i, 20%i*]. For each network type, CCA-MAP was run on 10 randomly generated network examples. The experiments were run in Matlab V7.2 on a Pentium® M processor of 1.60GHz with 1G RAM.

During the experiments, anchor nodes are randomly chosen. Several sets of differently positioned anchor nodes are selected for each network run and the average result is taken. Therefore, the results are not only generated for "well positioned" anchor nodes. We kept the number of anchor nodes to <= 5% of the total number of nodes in the network. The node that does the global network merge is also randomly selected. Both range-based and range-free options were investigated for all the network configurations. When using the range-free option, hop count is used as the distance matrix between a pair of nodes as described before. In the range-based scenarios, the range is modeled as the actual distance combined with Gaussian noise. Thus, the range measured between neighboring nodes is a random value drawn from a normal distribution with actual distance as mean and standard deviation of *5%*.

4.1 Random Uniform Networks

CCA-MAP is applied to random uniform networks of 200 nodes as shown in Figure 1 (a) using 3, 6 and 10 anchor nodes in the network. The network connectivity level increases from 7 to 29, with radio range spanning from *r=1.07i* to *2.4i*. It should be noted that even though a lower Connectivity Level (CL) such as *CL=6* is more favorable to optimize the radio throughput at the MAC layer [12], we did not test at such low connectivity levels because for this type of random network configuration of 200 nodes, the network is frequently not fully connected at *CL=6*. In fact, random placement of sensor nodes results in the neighborhood density being very uneven; a higher connectivity level is often needed for the network to be fully connected.

In the range-based option, with an average of 5% measurement error on local distance between each pair of neighboring nodes, the coordinate's estimation median errors are shown in Figure 2, which depicts the average error on computed coordinates as a function of connectivity level. At Connectivity Level (CL) CL=9, using 3 anchor nodes, the median error is 0.097r. Around CL=12 to 14, the median error decreases to 0.05r. At CL=29, the median error is 0.01r. Increasing the number of anchor nodes improves the results, though not significantly. Using 6 or 10 anchor nodes further decreases the median error to below 0.01r at high connectivity levels (e.g., CL=29). As mentioned before, almost all localization schemes show improved performance when the sensor network has a high connectivity level. However, a large number of neighboring nodes may not be desirable in real deployment as it hampers the radio network throughput. Therefore we are not particularly interested in using high connectivity levels to produce better positioning results, but are more concerned with the performance at more moderate connectivity levels. It is very appealing that CCA-MAP produces more accurate position estimates at relatively low connectivity levels compared with most of the existing methods, as discussed below.

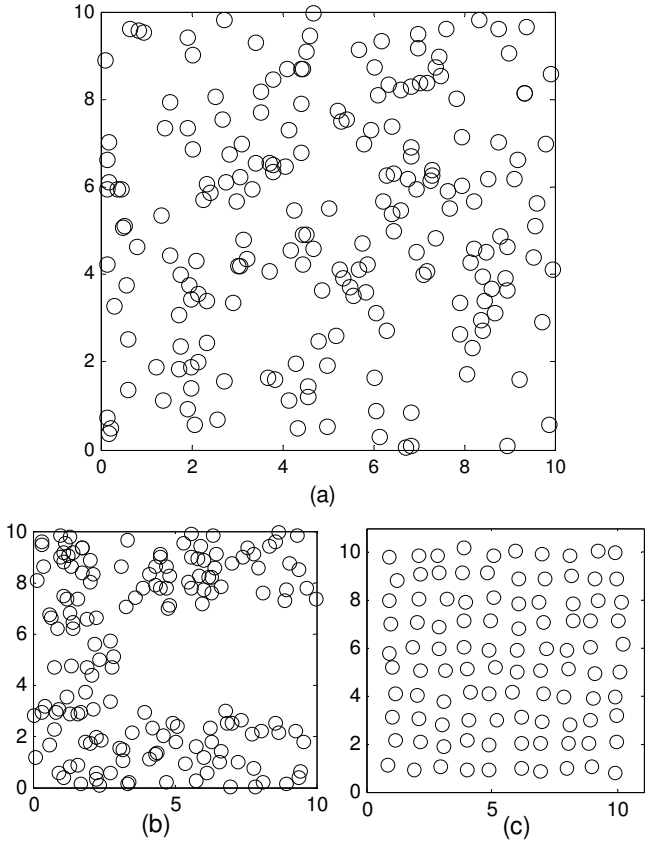

Figure 1: Three example-networks: (a) random uniform network of 200 nodes; (b) random C-shaped network of 160 nodes; (c) grid uniform network of 100 nodes

Figure 3 shows the effects of increased range errors on the localization results produced by CCA-MAP. When the range error increases, so do the errors on the node position estimates. It can also be seen that CCA-MAP sustains its performance quite well under the increased range errors.

Using only the connectivity information in range-free scenarios, the localization results for the same set of random networks are presented in Figure 4. Without ranging capabilities, the performance is degraded as expected. Using 3 anchor nodes, CCA-MAP achieves a median error of position estimation of

0.17*r* at a connectivity level (CL) of 12. At CL=19 the median error decreases to 0.15*r* and to 0.12*r* at CL=29. Increasing the number of anchors improves the results especially for low connectivity levels. However, the difference between using 6 anchor nodes (3% of total nodes) and 10 anchor nodes (5% of total nodes) is not significant when CL increases to more than 10. For example, at CL=12, 6 anchor nodes produce a median error of 0.1438*r* while 10 anchor nodes generate coordinates with a median error of 0.1407*r*. At CL=29, 10 anchor nodes can reduce the median error to 0.094*r* while 6 anchor nodes result in an error of 0.10*r*.

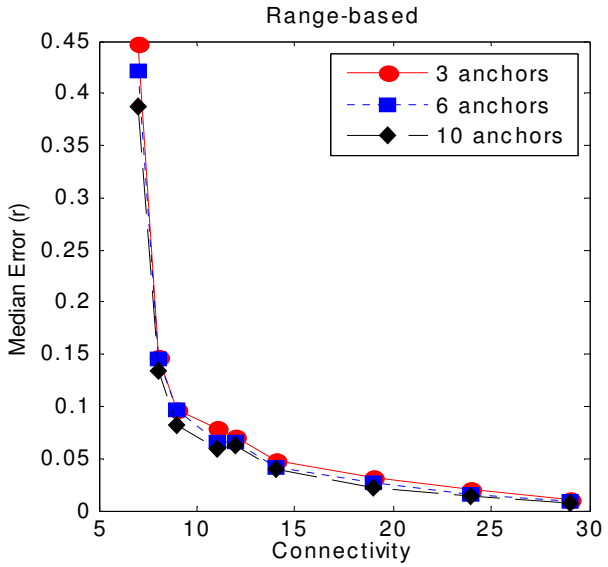

Figure 2: Results of CCA-MAP on random uniform networks of 200 nodes with 5% error on local distance measurements

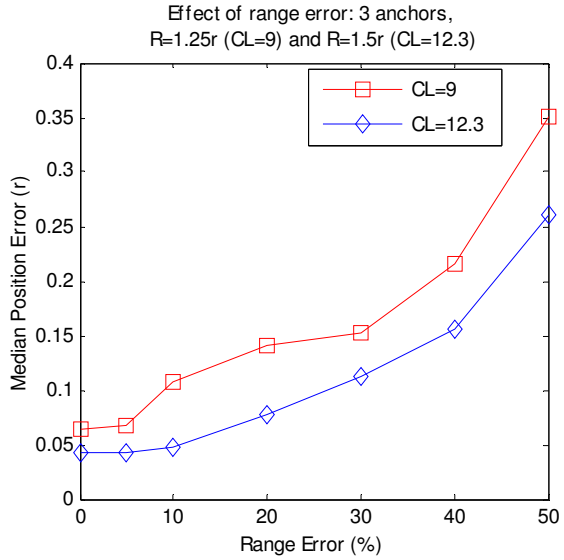

Figure 3: Effect of range errors on the position estimation for random uniform networks of 200 nodes

Figure 5 compares CCA-MAP with one of the best MDS algorithms, MDS-MAP(P,R), using range-based option where 5% local measurement error is assumed. Both algorithms work fairly well for this type of network using the range-based option, providing position estimation with less than 10%*r* of median error for most cases. CCA-MAP outperforms MDS-MAP(P,R) by about 1-2% lower position estimation error in most cases. The improvement of CCA-MAP compared to MDS-MAP(P,R) is considerable (about 5%) in the low connectivity area, e.g., when CL < 10. MDS-MAP(P,R) applies an additional refinement process using non-linear optimization which can be very costly, e.g., about 50 seconds are needed for refining both local maps and the global map for the network of 200 nodes. CCA-MAP on the other hand can achieve better position accuracy without this refinement process. CCA calculation of local maps on each node takes only a few hundred milliseconds. The computational overhead of CCA-MAP is thus much lower.

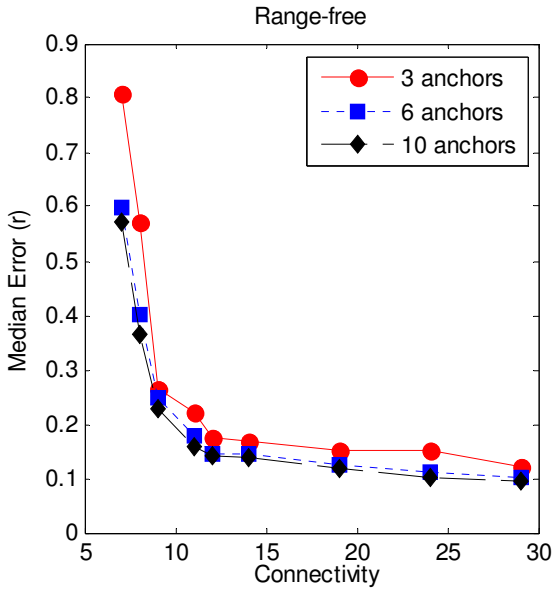

Figure 4: Results of CCA-MAP on random uniform networks of 200 nodes using connectivity information only

Figure 6 compares the performance between CCA-MAP and MDS-MAP(P,R) for range-free cases, where the node coordinates were calculated using only the network connectivity information. CCA-MAP shows more improvement compared with MDS-MAP(P,R) when fewer anchor nodes are employed in the network, and when the connectivity levels are low.

4.2 Random C-shaped Networks

The random irregular networks used in the experiments consist of 160 nodes scattered in an irregular area in the shape of a C, as shown in Figure 1 (b). Random irregular networks of anisotropic topology are particularly challenging for node position estimations as the anisotropic topology breaks the approximation of inter-node distances. Previous methods reported very poor results on such a topology [16]. In our experiments, the network connectivity level increases from 9 to 24, with radio range spanning from *r=1.28i* to *2.36i*. Again, the connectivity levels were selected to ensure that the networks are connected. The performance results are illustrated in Figure 7.

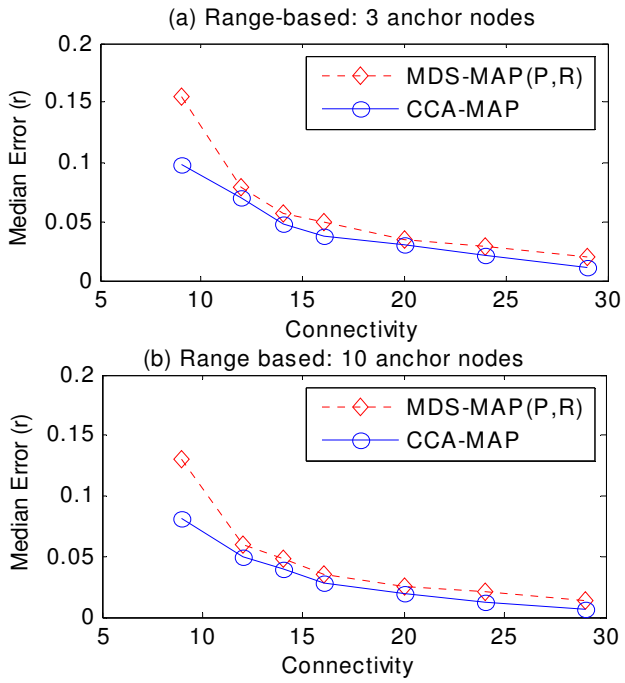

Figure 5: Comparison between range-based CCA-MAP and MDS-MAP(P,R) on random 200 node networks with 5% local measurement error

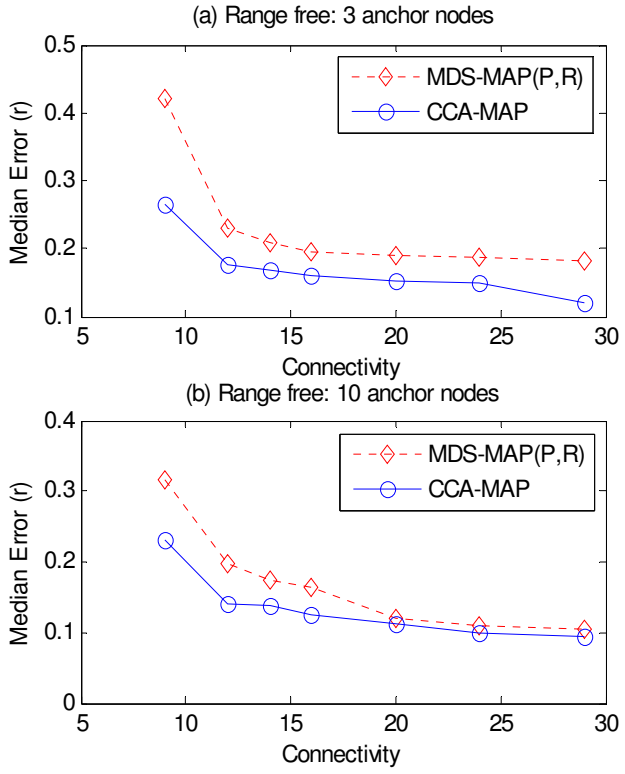

Figure 6: Comparison of range-free CCA-MAP and MDS-MAP(P,R) on random 200 node networks

Using only connectivity information in the range-free option, the median error of node locations is below *20%r* in the best cases, when 3 or 6 anchor nodes are deployed in the network. In some cases, the extended radio range even increases the estimation errors. With increased radio range and connectivity level, more nodes are included in the local map while more of them use the sum of the distances over multiple hops to approximate their true distance. In the irregular C-shaped area, certain corner nodes make such approximations more erroneous. This then introduces the additional errors to the position estimation.

With ranging capability (Fig. 7(b)), the position estimation median error can be contained under *10%r* when connectivity level is at about 12.5, using 3 or 6 anchor nodes in the network. More anchor nodes do not improve significantly the performance.

Compared with the MDS-MAP(P,R) algorithm, CCA-MAP improves considerably the position estimation accuracy, especially when the connectivity level is low. Figure 8 and Figure 9 present the comparison of CCA-MAP and MDS-MAP(P,R) for both the range-free option, where only the connectivity information is applied, and the range-based option, where 5% local distance measurement error is assumed. For both range-based and range-free options, CCA-MAP outperforms MDS-MAP(P,R), especially in the lower connectivity range.

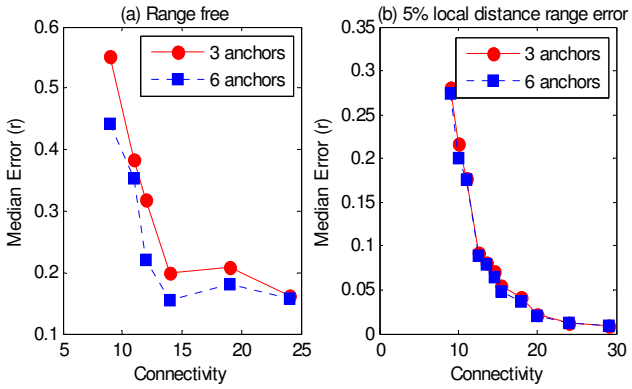

Figure 7: CCA-MAP on C-shaped random networks of 160 nodes

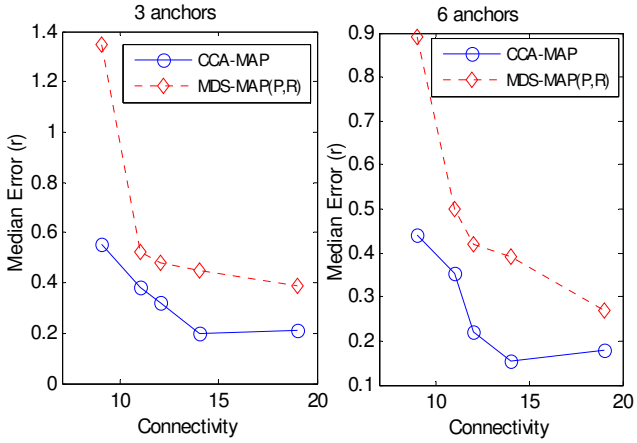

Figure 8: Comparison between CCA-MAP and MDS-MAP(P,R) over random irregular C-shaped networks of 160 nodes using range-free option

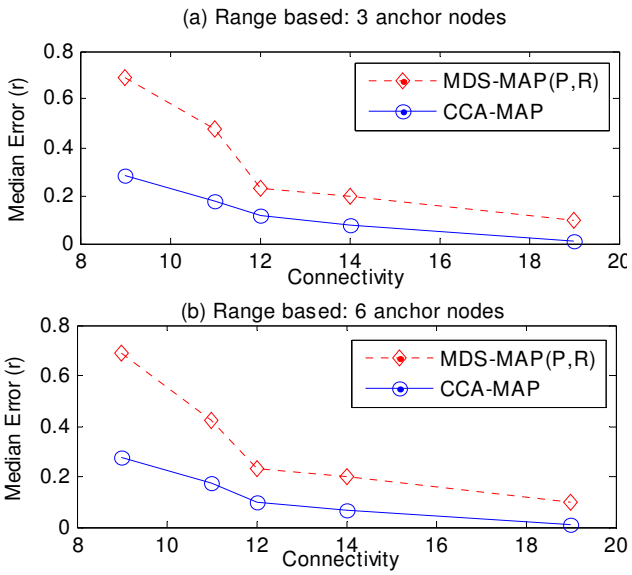

Figure 9: Comparison between CCA-MAP and MDS-MAP(P,R) over random irregular C-shaped networks of 160 nodes using range-based option with 5% local measurement error

4.3 Grid Uniform Networks

The performance results for grid networks that consist of 100 nodes with node placement error uniformly distributed in [*-20%i, 20%i*] as shown in Figure 1 (c) are illustrated in Figure 10. The connectivity levels in the networks increase from 4.5 to 7.5, with radio radius spanning from *1.336i* to *1.86i*. Using connectivity information only, at CL=5.5, the median error is less than *10%r* using 3 anchor nodes. Using 6 anchor nodes brings about *1%r* performance improvement, except when connectivity level is very low, e.g., CL=4.5. With ranging capability measuring local distance between neighboring nodes, the median errors are under *10%r* when CL=4.5 and approach *1%r* at CL=7.5 using merely 3 anchor nodes (Fig. 10(b)). More anchor nodes do not improve significantly the performance in the range-based scheme. The comparisons between CCA-MAP and MDS-MAP(P,R) over the grid networks are summarized in Figure 11 and Figure 12.

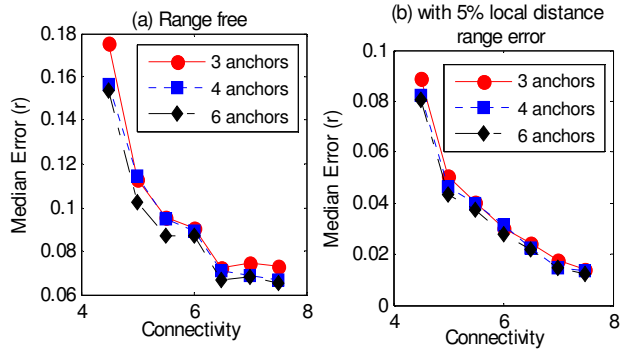

Figure 10: CCA-MAP performances on grid networks of 100 nodes with 10% placement errors

In range-free scenarios, using connectivity information only, both CCA-MAP and MDS-MAP(P,R) achieve good position estimates for the nodes. CCA-MAP outperforms MDS-MAP(P,R) when CL increases from 5 to 7.5. Again the improvement seems to be more pronounced when fewer anchor nodes are used. This is also true for the range-based cases where the local distances between neighboring nodes are known with an average of 5% measurement error.

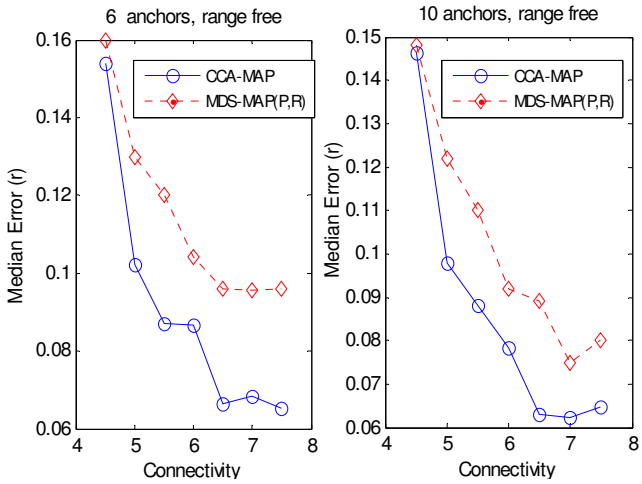

Figure 11: Comparison of CCA-MAP and MDS-MAP(P,R) for grid networks of 100 nodes using range-free option

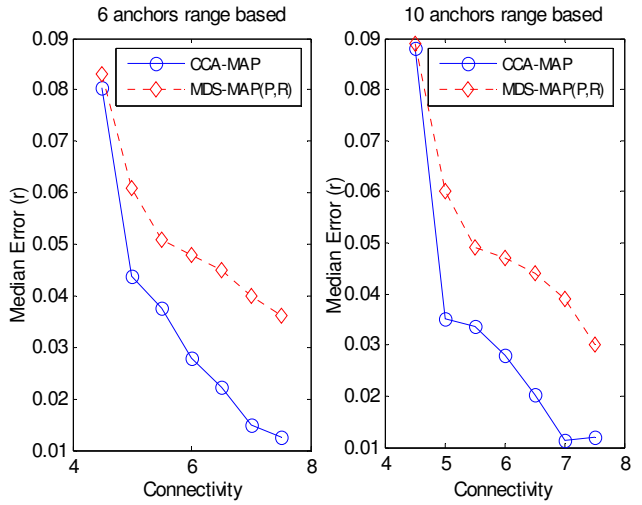

Figure 12: Comparison of CCA-MAP and MDS-MAP(P,R) for grid networks of 100 nodes using range-based option with 5% local distance measurement error

5. CONCLUSION

We have devised the CCA-MAP algorithm using an efficient non-linear reduction technique CCA to obtain wireless node coordinates. Our CCA-MAP scheme requires only a small number of anchor nodes to translate node coordinates to absolute values, and can be applied in either range-based or range-free scenarios. The simulation results were compared with another highly accurate solution, MDS-MAP(P,R). In all the network configurations simulated where ranging capability is employed, the CCA-MAP algorithm can achieve accurate node position estimates with an average error of less than *10%r* at relatively low network connectivity levels. Using connectivity information only, CCA-MAP can also contain the position estimate error to within

20%r. Compared with MDS-MAP(P,R), CCA-MAP achieves better position estimate results in most scenarios. The proposed approach is very promising for autonomic applications which require a high degree of accuracy for nodes' positions but having only a small number of anchor nodes in the network. We are currently studying the characteristics of the algorithm for more varieties of irregular network configurations and refining the algorithm parameters to handle better the anisotropic network topologies.

ACKNOWLEDGEMENT

This work was supported by Defence Research and Development Canada. Part of this work was also undertaken in conjunction with the EU IST project RUNES.

References

[1] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proceedings of IPSN 2004*, Berkeley, CA, April 2004

[2] I. Borg and P. Groenen, *Modern Multidimensional Scaling, Theory and Applications*, New York: Springer-Verlag, 1997

[3] N. Bulusu, J. Heidemann, D. Estrin and T. Tran, "Self-Configuring Localization Systems: Design and Experiemental Evaluation", *ACM Transactions on Embedded Computer Systems*, May 2003, vol. 3, no.1, pp.24-60, Feb. 2004

[4] K.K. Chintalapudi, A. Dhariwal, R. Govindan and G. Sukhatme, "Ad-hoc Localization using Ranging and Sectoring", in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, March 7-11, 2004

[5] P. Demartines and J. Herault, "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets", *IEEE Transactions on Neural Networks*, Vol. 8, No. 1, January 1997

[6] L. Doherty, K. Pister and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks", in *Proceedings of IEEE InfoCom 2001*. April 2001.

[7] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangan, R. Virrankoski and A. Savvides, "Distance Matrix Reconstruction from Incomplete Distance Information for Sensor Network Localization", in *Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks* (SECON'06) , September 2006, Reston, VA, USA.

[8] T. Eren, D.K. Goldenberg, W.Whiteley, Y.R. Yang, A.S. Morse, B. D. O. Anderson and P.N. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization", in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, March 7-11, 2004

[9] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor and Z. Sahinoglu, "Localization via Ultra-Wideband Radios", *IEEE Signal Processing Magazine*, Vol. 22, no.4, pp.70-84, July 2005.

[10] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher. "Range-free Localization Schemes for Large Scale Sensor Networks", in *Proceedings of MobiCom 2003*.

[11] X. Ji and H. Zha (2004) 'Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling', in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 7-11, 2004

[12] L. Kleinrock and J. Silvester, Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number," in *Proceedings of the IEEE National Telecommunications Conference*, Birmingham, Alabama, December 1978.

[13] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison." *Computer Networks*, 43(4): 499-518, 2003

[14] J. Liu, Y. Zhang and E. Zhao, "Robust Distributed Node Localization with Error Management", in *Proceedings of MobiHoc'06*, May 2006, Florence, Italy.

[15] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA", in *Proceedings of INFOCOM' 03*, San Francisco, CA, 2003

[16] D. Niculescu and B. Nath. "DV Based Positioning in Ad hoc Networks". *Kluwer Journal of Telecommunication Systems*. Vol 22:1-4, pp267-280, 2003.

[17] P. Pathirana, N. Bulusu, A. Savkin and S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks", *IEEE Transactions on Mobile Computing,* Vol. 4, No. 4, July/August, 2005

[18] N. Patwari, J. N. Ash, S. Kyperountas, A.O. Herro, R.L. Moses and N.S. Correal, "Locating the nodes*", IEEE Signal Processing Magazine*, Vol. 22, no.4, pp.54-69, July 2005.

[19] N.B. Priyantha, H. Balakrishnan and S. Teller, "Anchor-free distributed localization in sensor networks", in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems* (SenSys 2003), Los Angeles, CA, Nov. 5-7, 2003, pp340-341

[20] N.B. Priyantha, A. Chakraborty and H. Balakrishnan, "The Cricket Location-Support System", in *Proceedings of MOBICOM'00*, New York, August 2000.

[21] J.W. Sammon, "A nonlinear mapping for data structure analysis", *IEEE Transactions on Computers,* vol. C-18, pp.401-409, 1969.

[22] A. Savvides, C.-C. Han and M. Srivastava "Dynamic fine-grained localization in ad-hoc networks of sensors", in *Proceedings of ACM MobiCom*'01, Rome, Italy, July, 2001, pp.166–179.

[23] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz. "Localization from Mere Connectivity", in *Proceedings of MobiHoc 2003*. June 2003

[24] Y. Shang and W. Ruml, "Improved MDS Based Localization", in *Proceedings of IEEE INFOCOM 2004*, March, 2004

[25] Y. Shang, W. Ruml, Y. Zhang and M. Fromherz, "Localization from Connectivity in Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, No. 10, October 2004

[26] K.-F. Ssu, C.-H. Ou and H. Jiau (2005) "Localization with mobile anchor points in wireless sensor networks", *IEEE Transactions on Vehicular Technology*, Vol. 54, No. 3, May. 2005

[27] V. Vivekanandan and V.W.S. Wong, "Ordinal MDS-based Localization for Wireless Sensor Networks", *Int. J. Sensor Networks*, Vol. 1, No. 3/4, pp.169-178.

[28] Y,Zhang, Q. Huang and J. Liu, "Sequential Location Algorithm for Active Sensor Network Deployment", in Proceedings of IEEE PCAP 2006, April 2006, Vienna