Maximum Likelihood Position Estimation of Network Nodes Using Range Measurements

Anthony J. Weiss*, *Fellow IEEE* and Joseph Picard

Abstract

Given a network of stations with incomplete and possibly imprecise inter-station distance measurements it is required to find the relative stations position. Due to its asymptotic properties Maximum Likelihood estimation is discussed. Although the problem is quadratic, the proposed solution is based on solving a linear set of equations. For precise measurements we obtain explicitly the exact solution with a small number of operations. For noisy measurements the method provides an excellent initial point for the application of the Gerchberg-Saxton iterations. Proof of convergence is provided. The case of planar geometry is coached using complex numbers which reveals a strong relation to the celebrated problem of phase retrieval. Numerical examples are provided to corroborate the results.

Index Terms

Wireless Sensor Network, Location Estimation, Ad Hoc Networks, Maximum Likelihood Estimation, Phase Retrieval, Alternating Projections, Cramér-Rao Bound

---

* The authors are with Tel Aviv University, School of Electrical Engineering, Tel Aviv, 69978, Israel. (email: ajw@eng.tau.ac.il) This research was supported by the Israel Science Foundation (grant No. 1232/04)

I. Introduction

Problem Description and Possible Applications

Location estimation of nodes within a network is considered. It is assumed that incomplete and possibly inaccurate distance measurements are provided. The challenge is to find the best estimates of the nodes position given the distance measurements. The problem can be solved using a few nodes with known location (anchors) or it can be treated with no anchors and then only relative positions are found. Applications of sensor networks include environmental sensing (temperature, barometric pressure, humidity, etc.), water quality monitoring, inventory management, traffic monitoring, herd control, wild life behavior studies and many more.

Related Papers

Sensor network positioning attracts much interest in the research community. Although it is impossible to discuss all previous contributions we summarize the some of the main results. In 1] a bottom-up approach is discussed. Every node communicates with its neighbors and generates a local map using local coordinates. The location of nodes is based on triangulation using distances to at least three neighbors. In the next step local maps are merged using nodes that are common to more than a single local map. A bottom-up approach is also described in [2] using as basic building blocks robust quadrilaterals. Local maps are generated using quadrilaterals that are open enough to resist noisy measurements. Maximum Likelihood Estimation and the Cramér-Rao Bound are discussed in [3]. Thus, [3] is strongly related to the results discussed here. Here we provide a different algorithm for the same cost function and also provide a matrix formulation of the Cramér Rao Bound. In [4] a centralized approach is adopted using LMI (Linear Matrix Inequalities) to solve an optimization formulation of the problem. The method is effective only if the nodes are within the hull of the reference nodes (anchors) as pointed out in [5]. A reformulation of the problem that can be solved using semi-definite programming is presented in [5]. However, since additional unknowns (slack parameters) are introduced the required number of equations (distance measurements) is high as pointed out in [5, proposition 1]. The authors of [6] discuss the important issue of location uniqueness. According to this work a necessary and sufficient condition for generic node localizability is not yet known. In other words, using the technique described in the paper one can find uniquely locatable nodes but not necessarily all such nodes. The authors of [7] propose a two phase algorithm. In the first phase the location of each node is obtained by triangulation of three distances from 3 anchors. The distance from the anchor is obtained by counting the number of hops from the anchor. This is a rough position estimate. In the second phase each node obtains the position of its neighbors and its own distances to the neighbors. This information is used by the node to update its own position using least squares estimates. After several iterations the algorithm hopefully converges. In [8] three different algorithms are compared. All algorithms have three phases; Phase 1: Determine an approximate distance between the node and the anchors; Phase 2: Find the node position via lateration or

similar method; Phase 3: Refine the node position by using the neighbors' position and the distances to the neighbors. No single algorithm was found best under all conditions.
The algorithm in [9] is based on four steps: 1) A node determines its reference points (other nodes with known location) 2) The node collects distance measurements to the reference nodes. 3) The node determines its position 4) The node advertises its position estimate and the corresponding error estimate. Thus, iteratively the nodes find their own locations.

The basic idea in [10] is that each node receives enough beacon transmissions to estimate its position. A beacon is a node with known location which broadcasts its location periodically. The node estimates its position based on the centroid of the received beacons. The authors of [11] discuss an algorithm based on Multidimensional Scaling (MDS). This approach is based on the distance matrix which contains the distances between every pair of nodes. If the distance matrix is complete, *i.e.*, all the inter node distances are measured, the technique yields good results. If some measurements are missing iterations are required.

Contributions

In this paper the problem is coached using complex numbers for planar geometry, *i.e*. all nodes are contained within a single plane. Obviously, all the results can be extended to three dimensions using real numbers. However, using complex numbers reveals that the problem is strongly related to the family of phase retrieval problems that show up in optics [12, 13] and communications [14]. We show that the celebrated Gerchberg-Saxton algorithm [12] can be applied to the Maximum-Likelihood cost function. Furthermore, we show that the algorithm is guaranteed to converge.

It is well known that often a set of exact quadratic equations can be solved using a set of linear equations. This was shown in [8] and previously in the context of hyperbolic localization in [15,16]. We take advantage of this technique and extend it to the localization of multiple nodes at a time. Thus, we offer a complete network localization solving only sets of linear equations. The method yields precise node positions for precise distance measurements. If the measurements are noisy, the estimation of the node position accuracy deteriorates. However, the estimated locations provide excellent starting point for the Gerchberg-Saxton algorithm (GSA).

It is well known that, under mild regularity conditions, Maximum Likelihood estimates are statistically efficient and asymptotically approach the Cramér Rao Lower Bound (CRLB) [17]. We show that this is the case for the problem discussed here.

II. Problem Formulation for Planar Geometry Using Complex Numbers

Consider a network consisting of n stations with unknown positions and m stations with known positions, termed "anchors". If there are no stations with known positions, it is possible to select three inter-connected stations and use them as anchors. See [2] for more details. Assume that all stations are confined to a plane. The location of each station can than be represented by a complex number where the real part is the *x* coordinate and the imaginary part is the *y* coordinate. Thus the location of all stations can be described by $(n+m)\times 1$ complex column vector, $\mathbf{x}=[x_1,x_2,\cdots x_{n+m}]^T\in\mathbb{C}^{n+m\times 1}$ and the location of the anchors can be described by $m\times 1$ column vector, $\mathbf{z}=[x_{n+1},x_{n+2},\cdots x_{n+m}]^T\in\mathbb{C}^{m\times 1}$. The location difference between any two stations can be described by $(n+m)\times 1$ column vector, $\mathbf{e}_{ij}$, whose *i*-th entry is 1 and whose *j*-th entry is -1, while all other entries are zero. Thus,

$$\mathbf{e}_{ij}^T\mathbf{x}=x_i-x_j \tag{1}$$

Assume that a set of *L* distance measurements between pairs of stations are available. It is emphasized that not all possible distance measurements are available. The distance measurement between the *i*-th station and the *j*-th station is given by d_{ij}. Thus, we have the relation $d_{ij}=\left|x_i-x_j\right|$ if the measurements are precise.

Collecting all the differences associated with available distance measurements we get from (1),

$$\mathbf{Ex}=\mathbf{y}, \tag{2}$$

where the rows of the real matrix **E** are the vectors $\mathbf{e}_{ij}^T$ and **y** is a complex vector. The absolute value of each entry in **y** is given by the appropriate distance, d_{ij}, and the angle of each entry is unknown. Denote by **d** the vector consisting of the absolute values of the entries of **y**.

The problem we are addressing can be concisely described by:

Estimate the *n* unknown elements of $\mathbf{x}\in\mathbb{C}^{n+m\times 1}$ given $\mathbf{E}\in\mathbb{R}^{L\times n+m}$, $\mathbf{z}\in\mathbb{C}^{m\times 1}$ and possibly noisy version of $\mathbf{d}\in\mathbb{R}^{L\times 1}$.

III Solution for Precise Distance Measurements

In this section we discuss the case of incomplete but precise range measurements.
As a first step towards a solution we partition the vector **x** into two sub-vectors, one sub-vector includes the unknown node positions, denoted by **u** and one sub-vector includes the anchor positions, denoted by **z**. Thus, $\mathbf{x}=[\mathbf{u}^T,\mathbf{z}^T]^T$. We also partition the matrix **E** into two sub-matrices, one sub-matrix with *n* columns and one sub-matrix with *m* columns, $\mathbf{E}=[\mathbf{E}_1,\mathbf{E}_2]$, thus (2) becomes,

$$\mathbf{E}_1\mathbf{u}+\mathbf{E}_2\mathbf{z}=\mathbf{y} \tag{3}$$

Since $\mathbf{E}_2\mathbf{z}$ is a known vector it is convenient to replace it by $\mathbf{v}\triangleq\mathbf{E}_2\mathbf{z}$, Thus,

$$\mathbf{E}_1\mathbf{u}+\mathbf{v}=\mathbf{y};\qquad \left|\mathbf{y}\right|=\mathbf{d}, \tag{4}$$

where $\left|\mathbf{y}\right|$ denotes the vector whose entries are the absolute values of the entries in **y**.

Initially we want to locate the subset of stations each of which is connected to at least 3 anchors. Thus, we reduce equation (4) using three steps.

1) Sum the columns of $\mathbf{E}_1$. The entries of the resulting column vector that are different from zero indicate nodes connected with the anchors. All the other equations can be removed so the number of rows in $\mathbf{E}_1, \mathbf{v}, \mathbf{y}, \mathbf{d}$ can be reduced and the result is denoted by $\mathbf{E}_1^{'}, \mathbf{v}', \mathbf{y}', \mathbf{d}'$.
2) Sum the absolute value of the rows in $\mathbf{E}_1^{'}$. The entries of the resulting row vector indicate the number of connections with anchors each station has. We keep only stations that have at least 3 connections, and reduce the number of columns in $\mathbf{E}_1^{'}$ and the number of elements in **u**, the result is denoted by $\mathbf{E}_1^{''}, \mathbf{u}'$
3) Finally, we remove all zero rows in $\mathbf{E}_1^{''}, \mathbf{v}', \mathbf{y}'$. The reduced set is denoted by $\mathbf{E}_1, \mathbf{v}, \mathbf{y}, \mathbf{d}$ to keep the notation simple.

Note that the *j*-th row of equation (4) can be described by,

$$\left| \mathbf{q}_j^T \mathbf{u} + v_j \right|^2 = \left| y_j \right|^2 = d_j^2 \tag{5}$$

where $\mathbf{q}_j^T$ is the *j*-th row of $\mathbf{E}_1$, and v_j, y_j, d_j are the *j*-th entry of $\mathbf{v}, \mathbf{y}, \mathbf{d}$, respectively. Using the Kronecker product, denoted by $\otimes$, equation (5) can be rewritten as,

$$(\mathbf{q}_j^T \otimes \mathbf{q}_j^T)(\mathbf{u} \otimes \mathbf{u}^*) + v_j^* \mathbf{q}_j^T \mathbf{u} + v_j \mathbf{q}_j^T \mathbf{u}^* = d_j^2 - \left| v_j \right|^2 \tag{6}$$

Collecting the equations for all rows, we get,

$$\mathbf{Q}(\mathbf{u} \otimes \mathbf{u}^*) + \mathbf{G}\mathbf{u} + \mathbf{G}^* \mathbf{u}^* = \mathbf{b}, \tag{7}$$

where the rows of $\mathbf{Q}$ are $(\mathbf{q}_j^T \otimes \mathbf{q}_j^T)$, the rows of $\mathbf{G}$ are $v_j^* \mathbf{q}_j^T$ and the entries of the real column vector $\mathbf{b}$ are $d_j^2 - \left| v_j \right|^2$. More compactly, using the Khatri-Rao product, $\mathbf{Q} = (\mathbf{E}_1^T \circ \mathbf{E}_1^T)^T$; $\mathbf{G} = \operatorname{diag}\{\mathbf{v}^*\}\mathbf{E}_1$.

In order to solve (7) we find the projection matrix, **P**, on the space orthogonal to the columns space of **Q**. Most of the columns in **Q** are zero and the non-zero columns are equal to the absolute value of the columns of $\mathbf{E}_1$. Let $\mathbf{A} \triangleq \left| \mathbf{E}_1 \right|$. The columns of **A** are orthogonal. Thus, $\mathbf{A}^T \mathbf{A}$ is diagonal and easy to invert. Therefore, it easy to evaluate $\mathbf{P} = \mathbf{I} - \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$. Left-multiply equation (7) by **P**, and get,

$$\mathbf{PGu} + \mathbf{PG}^* \mathbf{u}^* = \mathbf{Pb} \tag{8}$$

Thus, equation (8) becomes,

$$[\mathbf{PG}, \mathbf{PG}^*] \begin{bmatrix} \mathbf{u} \\ \mathbf{u}^* \end{bmatrix} = \mathbf{Pb} \tag{9}$$

This linear set of equations has a unique solution as long as the matrix $[\mathbf{PG}, \mathbf{PG}^*]$ is full rank. Since we assumed precise range measurements the result is the exact stations locations.

In order to solve for the remaining stations we repeat the procedure using the located stations as additional anchors. Thus, if all nodes in the network are connected to the anchors the locations of all nodes are solved in a single step. Otherwise a few steps are required to obtain all node locations. The number of steps depends on the network connectivity.

For precise distance measurements the solution of (9) is precise. However, for the more frequent case of imprecise distance measurements the solution can be improved using iterations as shown in the next section.

IV Maximum Likelihood Solution

In this section we consider range measurements with random errors. The errors are assumed independent, zero-mean, Gaussian with identical variance.

The probability density function of L independent identically distributed (*i.i.d.*) zero-mean Gaussian distance measurements is given by

$$f\left(\mathbf{d}|\mathbf{u}\right) = \prod_{j=1}^{L} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(|y_j| - d_j)^2\right\} \tag{10}$$

The Maximum Likelihood estimator is given by,

$$\hat{\mathbf{u}} = \arg\min \sum_{j=1}^{L} \left| |y_j| - d_j \right|^2 = \arg\min \sum_{j=1}^{L} \left| |\mathbf{q}_j^T \mathbf{u} + v_j| - d_j \right|^2 \triangleq \arg\min e(\mathbf{u}) \tag{11}$$

In other words, we are interested in minimizing the cost function $e(\mathbf{u})$.

Consider now the following three step minimization procedure for the cost function, where k indicates the iteration number,

$$
\begin{aligned}
&i)\ \ y_j^{(k)} = \mathbf{q}_j^T \mathbf{u}^{(k)} + v_j \\
&ii)\ \ \overline{y}_j^{(k)} = \frac{d_j y_j^{(k)}}{\left| y_j^{(k)} \right|} \\
&iii)\ \ \mathbf{u}^{(k+1)} = \left(\mathbf{E}_1^T \mathbf{E}_1 \right)^{-1} \mathbf{E}_1^T \left(\overline{\mathbf{y}}^{(k)} - \mathbf{v} \right)
\end{aligned} \tag{12}
$$

In words, given a node location vector $\mathbf{u}^{(k)}$ compute the vector $\mathbf{y}^{(k)}$ using the first line of (12). Adjust the modulus of $\mathbf{y}^{(k)}$ to be equal to the measured distances, $\mathbf{d}$ as sown in the second line. Use the third line in (12) to obtain a new estimate of the nodes locations.

Proof of Convergence

The error function defined in (11) can be written using the definitions of the procedure as

$$
\begin{aligned}
e(\mathbf{u}^{(k)}) &\triangleq \sum_{j=1}^{L} \left| \left| \mathbf{q}_j^T \mathbf{u}^{(k)} + v_j \right| - d_j \right|^2 = \sum_{j=1}^{L} \left| \left| y_j^{(k)} \right| - d_j \right|^2 \\
&= \sum_{j=1}^{L} \left| \left| y_j^{(k)} \right| - \left| \overline{y}_j^{(k)} \right| \right|^2 = \sum_{j=1}^{L} \left| y_j^{(k)} - \overline{y}_j^{(k)} \right|^2 = \left\| \mathbf{y}^{(k)} - \overline{\mathbf{y}}^{(k)} \right\|_F^2 \\
&= \left\| (\mathbf{y}^{(k)} - \mathbf{v}) - (\overline{\mathbf{y}}^{(k)} - \mathbf{v}) \right\|_F^2
\end{aligned} \tag{13}
$$

where $\|\bullet\|_F$ stands for the Frobenius norm.
Also, according to the procedure

$$
\mathbf{y}^{(k+1)} - \mathbf{v} = \mathbf{E}_1 \mathbf{u}^{(k+1)} = \mathbf{E}_1 \left(\mathbf{E}_1^T \mathbf{E}_1 \right)^{-1} \mathbf{E}_1^T \left(\overline{\mathbf{y}}^{(k)} - \mathbf{v} \right) \tag{14}
$$

Note that $\mathbf{y}^{(k+1)} - \mathbf{v}$ is the projection of $\overline{\mathbf{y}}^{(k)} - \mathbf{v}$ on the column space of $\mathbf{E}_1$. Thus, $\mathbf{y}^{(k+1)} - \mathbf{v}$ is the closest vector in the $\mathbf{E}_1$ column space to $\overline{\mathbf{y}}^{(k)} - \mathbf{v}$. Similarly $\mathbf{y}^{(k)} - \mathbf{v}$ is also within the column space of $\mathbf{E}_1$ and therefore,

$$
\overline{e}^{(k)} \triangleq \left\| (\mathbf{y}^{(k+1)} - \mathbf{v}) - (\overline{\mathbf{y}}^{(k)} - \mathbf{v}) \right\|_F^2 \le \left\| (\mathbf{y}^{(k)} - \mathbf{v}) - (\overline{\mathbf{y}}^{(k)} - \mathbf{v}) \right\|_F^2 = e(\mathbf{u}^{(k)}) \tag{15}
$$

Note that because the modulus of $\overline{\mathbf{y}}^{(k+1)}$ is the same as the modulus of $\overline{\mathbf{y}}^{(k)}$ and the phase of $\overline{\mathbf{y}}^{(k+1)}$ is equal to the phase of $\mathbf{y}^{(k+1)}$ we have

$$
e(\mathbf{u}^{(k+1)}) = \left\| \mathbf{y}^{(k+1)} - \overline{\mathbf{y}}^{(k+1)} \right\|_F^2 \le \left\| \mathbf{y}^{(k+1)} - \overline{\mathbf{y}}^{(k)} \right\|_F^2 = \overline{e}^{(k)} \tag{16}
$$

. Combining (15) and (16) we get,

$$
e(\mathbf{u}^{(k+1)}) \le \overline{e}^{(k)} \le e(\mathbf{u}^{(k)}) \tag{17}
$$

Thus, in every iteration the error is reduced or stays the same and convergence is guaranteed.

□

The procedure in (12) is known as the Gerchberg-Saxton procedure [12]. This procedure can also be viewed as Alternating Projection algorithm. The second line in (12) is a (nonlinear) projection on the space of vectors all having the same given modulus. The thirds line of (12) is a projection on the column space of $\mathbf{E}_1$. A relation to Gauss-Newton algorithm can also be established.

V Compact Cramér Rao Bound

In this section we derive a compact form of the Cramér Rao bound. The bound was previously derived in [3].

Based on (10) the log-likelihood is given (up to an additive constant) by

$$\ell(\mathbf{u}) = -\frac{1}{2\sigma^2}\sum_{j=1}^{L}(|y_j| - d_j)^2 = -\frac{1}{2\sigma^2}\left\||\mathbf{y}| - \mathbf{d}\right\|^2 \tag{18}$$

The derivatives of the log-likelihood function w.r.t. the real part and imaginary part of $\mathbf{u}$ are obtained next. The derivative w.r.t. $\mathbf{u}_r \triangleq \mathrm{Re}\{\mathbf{u}\}$ is given by,

$$\frac{\partial \ell(\mathbf{u})}{\partial \mathbf{u}_r} = -\frac{1}{\sigma^2}\sum_{j=1}^{L}\left(\left|\mathbf{q}_j^T\mathbf{u} + v_j\right| - d_j\right)\frac{\partial\left|\mathbf{q}_j^T\mathbf{u} + v_j\right|}{\partial \mathbf{u}_r} \tag{19}$$

Recalling that,

$$\left|\mathbf{q}_j^T\mathbf{u} + v_j\right| = \left[\mathbf{q}_j^T\mathbf{u}\mathbf{u}^H\mathbf{q}_j + \mathbf{q}_j^T\mathbf{u}v_j^* + \mathbf{q}_j^T\mathbf{u}^*v_j + \left|v_j\right|^2\right]^{1/2} \tag{20}$$

the derivative is given by

$$\begin{aligned}\frac{\partial\left|\mathbf{q}_j^T\mathbf{u} + v_j\right|}{\partial \mathbf{u}_r} &= \frac{1}{2\left|\mathbf{q}_j^T\mathbf{u} + v_j\right|}\left[\mathbf{u}^H\mathbf{q}_j\mathbf{q}_j^T + \mathbf{q}_j^T\mathbf{u}\mathbf{q}_j^T + v_j^*\mathbf{q}_j^T + v_j\mathbf{q}_j^T\right]^T \\ &= \frac{1}{\left|y_j\right|}\left[\mathbf{u}_r^T\mathbf{q}_j\mathbf{q}_j^T + \mathrm{Re}\{v_j\}\mathbf{q}_j^T\right]^T\end{aligned} \tag{21}$$

Substituting (21) in (19) we get,

$$\begin{aligned}\frac{\partial \ell(\mathbf{u})}{\partial \mathbf{u}_r} &= -\frac{1}{\sigma^2}\sum_{j=1}^{L}\left(\left|y_j\right|-d_j\right)\frac{1}{\left|y_j\right|}\left[\mathbf{u}_r^T\mathbf{q}_j\mathbf{q}_j^T+\mathrm{Re}\left\{v_j\right\}\mathbf{q}_j^T\right]^T \\ &= -\frac{1}{\sigma^2}\sum_{j=1}^{L}\left(\left|y_j\right|-d_j\right)\frac{1}{\left|y_j\right|}\mathbf{q}_j\left[\mathbf{q}_j^T\mathbf{u}_r+\mathrm{Re}\left\{v_j\right\}\right] \\ &= -\frac{1}{\sigma^2}\sum_{j=1}^{L}\left(\left|y_j\right|-d_j\right)\frac{\mathrm{Re}\left\{y_j\right\}}{\left|y_j\right|}\mathbf{q}_j = -\frac{1}{\sigma^2}\mathbf{E}_1^T\,\mathrm{Re}\{\mathbf{y}-\bar{\mathbf{y}}\}\end{aligned} \tag{22}$$

Similarly for $\mathbf{u}_i \triangleq \mathrm{Im}\{\mathbf{u}\}$ we get

$$\frac{\partial \ell(\mathbf{u})}{\partial \mathbf{u}_i} = -\frac{1}{\sigma^2}\mathbf{E}_1^T\,\mathrm{Im}\{\mathbf{y}-\bar{\mathbf{y}}\} \tag{23}$$

More compactly these derivatives can be written as

$$\begin{aligned}\frac{\partial \ell(\mathbf{u})}{\partial \mathbf{u}_r} &= -\frac{1}{\sigma^2}\mathbf{E}_1^T\mathbf{D}_r(\left|\mathbf{y}\right|-\mathbf{d}) \\ \frac{\partial \ell(\mathbf{u})}{\partial \mathbf{u}_i} &= -\frac{1}{\sigma^2}\mathbf{E}_1^T\mathbf{D}_i(\left|\mathbf{y}\right|-\mathbf{d})\end{aligned} \tag{24}$$

Where

$$\begin{aligned}\mathbf{D}_r &\triangleq \mathrm{Re}\left\{\mathrm{diag}\left\{y_1/\left|y_1\right|,\cdots,y_L/\left|y_L\right|\right\}\right\} \\ \mathbf{D}_i &\triangleq \mathrm{Im}\left\{\mathrm{diag}\left\{y_1/\left|y_1\right|,\cdots,y_L/\left|y_L\right|\right\}\right\}\end{aligned} \tag{25}$$

Note that

$$E\left\{(\left|\mathbf{y}\right|-\mathbf{d})(\left|\mathbf{y}\right|-\mathbf{d})^T\right\} = \sigma^2\mathbf{I} \tag{26}$$

These expressions lead to,

$$\begin{aligned}E\left\{\frac{\partial \ell}{\partial \mathbf{u}_r}\left(\frac{\partial \ell}{\partial \mathbf{u}_r}\right)^T\right\} &= \frac{1}{\sigma^2}\mathbf{E}_1^T\mathbf{D}_r^2\mathbf{E}_1 \\ E\left\{\frac{\partial \ell}{\partial \mathbf{u}_i}\left(\frac{\partial \ell}{\partial \mathbf{u}_i}\right)^T\right\} &= \frac{1}{\sigma^2}\mathbf{E}_1^T\mathbf{D}_i^2\mathbf{E}_1 \\ E\left\{\frac{\partial \ell}{\partial \mathbf{u}_r}\left(\frac{\partial \ell}{\partial \mathbf{u}_i}\right)^T\right\} &= \frac{1}{\sigma^2}\mathbf{E}_1^T\mathbf{D}_r\mathbf{D}_i\mathbf{E}_1\end{aligned} \tag{27}$$

The Fisher Information Matrix is given by,

$$\mathbf{F}=\frac{1}{\sigma^2}\begin{bmatrix}\mathbf{E}_1^T\mathbf{D}_r^2\mathbf{E}_1 & \mathbf{E}_1^T\mathbf{D}_r\mathbf{D}_i\mathbf{E}_1\\ \mathbf{E}_1^T\mathbf{D}_r\mathbf{D}_i\mathbf{E}_1 & \mathbf{E}_1^T\mathbf{D}_i^2\mathbf{E}_1\end{bmatrix} \tag{28}$$

Thus, the CRB is given by the inverse of $\mathbf{F}$.

VI Small Error Analysis of the Maximum Likelihood Estimator

In this section we obtain expressions for the error covariance of the proposed algorithm.

Consider the expression for Small Error Analysis [18]:

$$\operatorname{cov}\{\mathbf{u}\}=\left[\frac{\partial^2 e(\mathbf{u})}{\partial\mathbf{u}^2}\right]^{-1}\left[\frac{\partial^2 e(\mathbf{u})}{\partial\mathbf{u}\partial\mathbf{d}}\right]\operatorname{cov}\{\mathbf{d}\}\left[\frac{\partial^2 e(\mathbf{u})}{\partial\mathbf{u}\partial\mathbf{d}}\right]^{T}\left[\frac{\partial^2 e(\mathbf{u})}{\partial\mathbf{u}^2}\right]^{-T} \tag{29}$$

Note that each matrix has to be evaluated at the true value of $\mathbf{u}$ and $\mathbf{d}$.

The first derivatives w.r.t. the real and imaginary parts of $\mathbf{u}$ are given by

$$\begin{aligned}\left[\frac{\partial e(\mathbf{u})}{\partial\mathbf{u}_r}\right]_k&=\sum_{j=1}^{L}\underbrace{\left(\left|\mathbf{q}_j^T\mathbf{u}+v_j\right|-d_j\right)}_{\alpha_j}\underbrace{\left|\mathbf{q}_j^T\mathbf{u}+v_j\right|^{-1}}_{\beta_j}\underbrace{\operatorname{Re}\left\{\mathbf{q}_j^T\mathbf{u}+v_j\right\}\mathbf{q}_j^T\mathbf{e}_k}_{\gamma_j^R}\\ \left[\frac{\partial e(\mathbf{u})}{\partial\mathbf{u}_i}\right]_k&=\sum_{j=1}^{L}\underbrace{\left(\left|\mathbf{q}_j^T\mathbf{u}+v_j\right|-d_j\right)}_{\alpha_j}\underbrace{\left|\mathbf{q}_j^T\mathbf{u}+v_j\right|^{-1}}_{\beta_j}\underbrace{\operatorname{Im}\left\{\mathbf{q}_j^T\mathbf{u}+v_j\right\}\mathbf{q}_j^T\mathbf{e}_k}_{\gamma_j^I}\end{aligned} \tag{30}$$

The second derivatives w.r.t. the real and imaginary parts of $\mathbf{u}$ are given by

$$\begin{aligned}\frac{\partial^2 e(\mathbf{u})}{\partial\mathbf{u}_r(k)\partial\mathbf{u}_r(m)}&=\frac{\partial}{\partial\mathbf{u}_r(m)}\left[\sum_{j=1}^{L}\alpha_j\beta_j\gamma_j^R\mathbf{q}_j^T\right]\mathbf{e}_k\\ &=\sum_{j=1}^{L}\left[\frac{\partial\alpha_j}{\partial\mathbf{u}_r(m)}\beta_j\gamma_j^R+\alpha_j\frac{\partial\beta_j}{\partial\mathbf{u}_r(m)}\gamma_j^R+\alpha_j\beta_j\frac{\partial\gamma_j^R}{\partial\mathbf{u}_r(m)}\right]\mathbf{q}_j^T\mathbf{e}_k\end{aligned} \tag{31}$$

where

$$\begin{aligned} \frac{\partial \alpha_j}{\partial \mathbf{u}_r(m)} &= \frac{\partial}{\partial \mathbf{u}_r(m)} \left| \mathbf{q}_j^T \mathbf{u} + v_j \right| = \frac{\mathrm{Re}\{y_j\}}{|y_j|} \mathbf{q}_j^T \mathbf{e}_m \\ \frac{\partial \beta_j}{\partial \mathbf{u}_r(m)} &= \frac{\partial}{\partial \mathbf{u}_r(m)} \left| \mathbf{q}_j^T \mathbf{u} + v_j \right|^{-1} = \frac{\mathrm{Re}\{y_j\}}{|y_j|^3} \mathbf{q}_j^T \mathbf{e}_m \\ \frac{\partial \gamma_j^R}{\partial \mathbf{u}_r(m)} &= \frac{\partial}{\partial \mathbf{u}_r(m)} \mathbf{q}_j^T \mathbf{u}_r = \mathbf{q}_j^T \mathbf{e}_m \end{aligned} \tag{32}$$

Substituting (32) in (31) we get

$$\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r(k) \partial \mathbf{u}_r(m)} = \sum_{j=1}^{L} \left[\frac{\mathrm{Re}\{y_j\}}{|y_j|} \beta_j \gamma_j^R + \alpha_j \frac{\mathrm{Re}\{y_j\}}{|y_j|^3} \gamma_j^R + \alpha_j \beta_j \right] \mathbf{q}_j^T \mathbf{e}_k \mathbf{q}_j^T \mathbf{e}_m \tag{33}$$

Using matrix notations, we obtain:

$$\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r \partial \mathbf{u}_r} = \sum_{j=1}^{L} \left[\frac{\mathrm{Re}\{y_j\}}{|y_j|} \beta_j \gamma_j^R + \alpha_j \frac{\mathrm{Re}\{y_j\}}{|y_j|^3} \gamma_j^R + \alpha_j \beta_j \right] \mathbf{q}_j \mathbf{q}_j^T \tag{34}$$

For the true values of $\mathbf{u}$ and $\mathbf{d}$, we have: $\left| \mathbf{q}_j^T \mathbf{u} + v_j \right| = d_j$, thus $\alpha_j = \left| \mathbf{q}_j^T \mathbf{u} + v_j \right| - d_j = 0$, and the previous equation reduces to

$$\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r \partial \mathbf{u}_r} = \sum_{j=1}^{L} \left[\frac{\mathrm{Re}\{y_j\}}{|y_j|} \beta_j \gamma_j^R \right] \mathbf{q}_j \mathbf{q}_j^T = \sum_{j=1}^{L} \left[\frac{\mathrm{Re}\{y_j\}}{|y_j|} \frac{\mathrm{Re}\{y_j\}}{|y_j|} \right] \mathbf{q}_j \mathbf{q}_j^T \tag{35}$$

Using the diagonal matrices defined in (25) we obtain

$$\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r \partial \mathbf{u}_r} = \mathbf{E}_1^T \mathbf{D}_r^2 \mathbf{E}_1 \tag{36}$$

Proceeding similarly for the other derivatives yields:

$$\begin{aligned} \frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_i \partial \mathbf{u}_i} &= \mathbf{E}_1^T \mathbf{D}_i^2 \mathbf{E}_1 \\ \frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r \partial \mathbf{u}_i} &= \mathbf{E}_1^T \mathbf{D}_r \mathbf{D}_i \mathbf{E}_1 \end{aligned} \tag{37}$$

As a result:

$$\left[\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}^2} \right] = \begin{bmatrix} \mathbf{E}_1^T \mathbf{D}_r^2 \mathbf{E}_1 & \mathbf{E}_1^T \mathbf{D}_r \mathbf{D}_i \mathbf{E}_1 \\ \mathbf{E}_1^T \mathbf{D}_r \mathbf{D}_i \mathbf{E}_1 & \mathbf{E}_1^T \mathbf{D}_i^2 \mathbf{E}_1 \end{bmatrix} = \mathbf{F} \tag{38}$$

where $\mathbf{F}$ is the Fisher Information Matrix.

We now obtain the derivatives w.r.t. **d**.

Note that: $\frac{\partial \alpha_j}{\partial d_m} = -\delta_{m-j}$ $\quad \frac{\partial \beta_j}{\partial d_m} = 0$ $\quad \frac{\partial \gamma_j^R}{\partial d_m} = 0$ $\quad \frac{\partial \gamma_j^I}{\partial d_m} = 0$

We immediately get:

$$\begin{aligned} \frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r(k)\partial \mathbf{d}(m)} &= \frac{\partial}{\partial \mathbf{d}(m)}\left[\sum_{j=1}^{L} \alpha_j \beta_j \gamma_j^R \mathbf{q}_j^T\right]\mathbf{e}_k \\ &= \sum_{j=1}^{L}\left[\frac{\partial \alpha_j}{\partial \mathbf{d}(m)} \beta_j \gamma_j^R \mathbf{q}_j^T\right]\mathbf{e}_k = -\frac{\mathrm{Re}\{y_m\}}{|y_m|}\mathbf{q}_m^T \mathbf{e}_k \end{aligned} \tag{39}$$

Using matrix notations,

$$\begin{aligned} \frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_r \partial \mathbf{d}} &= \mathbf{D}_r \mathbf{E}_1 \\ \frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}_i \partial \mathbf{d}} &= \mathbf{D}_i \mathbf{E}_1 \end{aligned} \tag{40}$$

Thus,

$$\left[\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}\partial \mathbf{d}}\right]^T = \left[\mathbf{D}_r \mathbf{E}_1 \quad \mathbf{D}_i \mathbf{E}_1\right] \tag{41}$$

Using the assumption that $\mathrm{cov}\{\mathbf{d}\} = \sigma^2 \mathbf{I}$ we get from (29),

$$\begin{aligned} \mathrm{cov}\{\mathbf{u}\} &= \sigma^2 \left[\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}^2}\right]^{-1} \left[\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}\partial \mathbf{d}}\right] \left[\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}\partial \mathbf{d}}\right]^T \left[\frac{\partial^2 e(\mathbf{u})}{\partial \mathbf{u}^2}\right]^{-T} \\ &= \sigma^2 \mathbf{F}^{-1} \left[\mathbf{D}_r \mathbf{E}_1 \quad \mathbf{D}_i \mathbf{E}_1\right]^{\mathrm{T}} \left[\mathbf{D}_r \mathbf{E}_1 \quad \mathbf{D}_i \mathbf{E}_1\right] \mathbf{F}^{-1} \\ &= \mathbf{F}^{-1} \end{aligned} \tag{42}$$

Thus, as expected, the Maximum Likelihood estimation covariance is equal to the Cramér Rao bound for small errors.

VII Computational load

In this section we evaluate the computational load of the proposed algorithm. The algorithm consists of two distinct phases. In phase one the nodes location is estimated by solving linear equations. In phase two we use the Gerchberg-Saxton iterations.

Phase one

Phase one consists of a few iterations. In each iteration we find the location of locatable nodes that become anchors in the following iteration.

We use the following symbols:

$M \geq 3$ number of anchors in the current iteration

N number of locatable nodes (nodes connected to 3 anchors or more)

$J \geq 3N$ number of connections between locatable nodes and anchors

Task	Number of Multiplications	Comments
$\mathbf{v} = \mathbf{E}_2 \mathbf{z}$	J	only single 1 per line of $\mathbf{E}_2$
$\mathbf{G} = \mathrm{diag}\{\mathbf{v}\}\mathbf{E}_1$	J	only single 1 per line of $\mathbf{E}_2$
$\mathbf{Q}$	J	
$\left\lvert\mathbf{E}_1\right\rvert^{\dagger}$	$3J$	J for $\left\lvert\mathbf{E}_1^T\right\rvert\left\lvert\mathbf{E}_1\right\rvert$; J for $\left(\left\lvert\mathbf{E}_1^T\right\rvert\left\lvert\mathbf{E}_1\right\rvert\right)^{-1}$ J for $\left(\left\lvert\mathbf{E}_1^T\right\rvert\left\lvert\mathbf{E}_1\right\rvert\right)^{-1}\left\lvert\mathbf{E}_1\right\rvert$
$\mathbf{P} = \mathbf{I} - \left\lvert\mathbf{E}_1\right\rvert\left\lvert\mathbf{E}_1\right\rvert^{\dagger}$	J	
$\mathbf{Pb}$	$\leq J^2$	
$\mathbf{PG}$	$\leq JN$	
$\left[\mathbf{PG} \quad \mathbf{PG}^*\right]^H \left[\mathbf{PG} \quad \mathbf{PG}^*\right]$	$\leq 4N^2 J$	
$\left(\left[\mathbf{PG} \quad \mathbf{PG}^*\right]^H \left[\mathbf{PG} \quad \mathbf{PG}^*\right]\right)^{-1}$	$\leq (2N)^3 = 8N^3$	
$\left(\left[\mathbf{PG} \quad \mathbf{PG}^*\right]^H \left[\mathbf{PG} \quad \mathbf{PG}^*\right]\right)^{-1}\left[\mathbf{PG} \quad \mathbf{PG}^*\right]^H$	$\leq 4N^2 J$	
$\left(\left[\mathbf{PG} \quad \mathbf{PG}^*\right]^H \left[\mathbf{PG} \quad \mathbf{PG}^*\right]\right)^{-1}\left[\mathbf{PG} \quad \mathbf{PG}^*\right]^H \mathbf{Pb}$	$\leq 2NJ$	

Thus the total number of multiplications can be approximated by $8N^2 J$.

Phase two

For I iterations we need $J + (3J + N)I$ multiplications.

VIII Numerical Examples

In this section we show the effectiveness of the proposed algorithm via a numerical example. Over an area of 10×10 distance unit we placed 25 stations, three of which are anchors, as shown in figure 1.

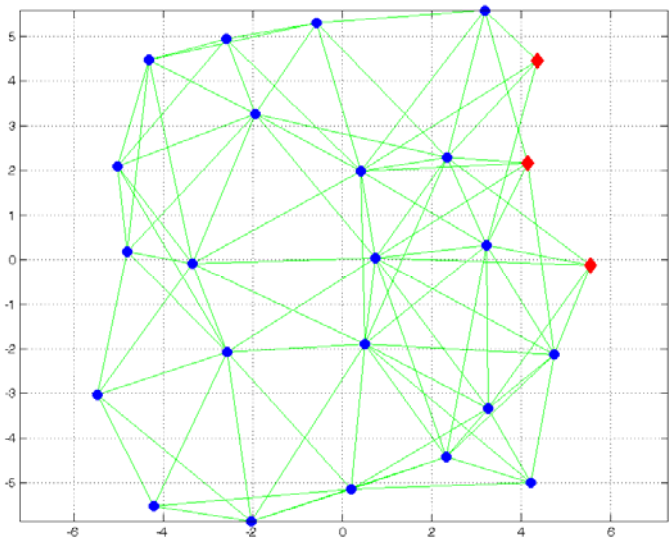

Figure 1: Twenty two nodes + three anchors with limited connectivity

The algorithm received as input the anchors positions and the marked distances with random errors. We changed the standard deviation of the distance errors from 1% to 9% of the measured distances mean. We performed 200 experiments for each distance error variance. The average (over all nodes) of the location estimate standard deviation is plotted against the distance standard deviation in Figure 2. Also plotted is the error lower bound (CRLB). As can be seen, as the distance errors decrease the location errors approach the lower bound.

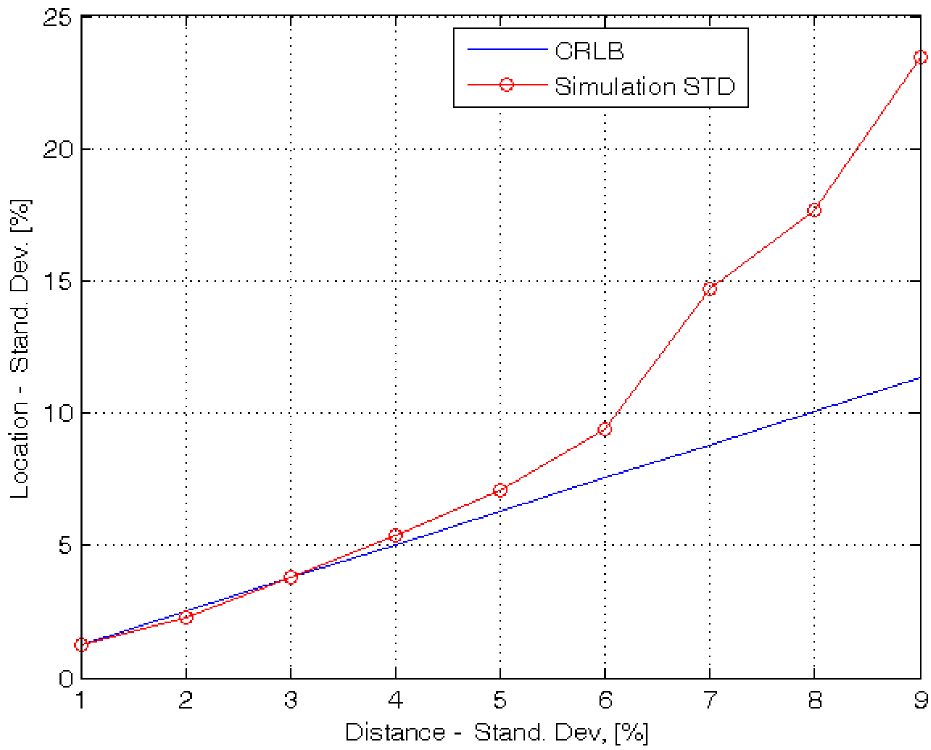

Figure 2: Location standard deviation versus distance standard deviation measure as a percentage of the average distance between nodes.

IX Summary and Conclusions

In this work we show that given incomplete but precise distance measurements the nodes of an ad-hoc network can be located precisely by solving linear equations. If the available distance measurements are noisy the proposed algorithm provides a good initial location estimate for the iterative solution of the Maximum Likelihood estimator. Using complex numbers we show that the problem can be viewed as a phase-retrieval problem and therefore the celebrated Gerchberg Saxton algorithm (GSA) can be applied to obtain the Maximum Likelihood estimates. The advantage of the GSA is that it is guaranteed to converge. Although the Cramer Rao bound was presented in earlier publications we provide a compact close form expression.

References

1. S. Čapkun, M. Hamdi, J-P. Hubaux, "GPS-free Positioning in Mobile Ad-Hoc Networks," *Proceedings of the 34th Hawaii International Conference on System Sciences*, Jan. 2001.

2. D. Moore, J. Leonard, D. Rus, S. Teller. "Robust distributed network localization with noisy range measurements." *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04)*, Baltimore, MD., pp. 50–61, November, 2004

3. N. Patwari, A.O. Hero, M. Perkins, N.S. Correal, R.J. O'Dea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. on Signal Processing*, Vol. 51, No. 8, pp. 2137-2148, Aug. 2003.

4. L. Doherty, K.S.J. Pister, L. E. Ghaoui, "Convex Position Estimation in Wireless Sensor Networks" *Proceedings IEEE INFOCOM*, Vol. 3, pp. 1655-1663, 2001.

5. P. Biswas, Y. Ye, "Semidefinite Programming for Ad Hoc Wireless Sensor Network Localization," *Proceedings of the 3rd Information Processing for Sensor Networks conference (IPSN)*, pp. 46-54, April, 2004.

6. D.K. Goldenberg, A. Krishnamurthy, W.C. Maness, Y.R. Yang, A. Young, A.S. Morse, A. Savvides, B.D.O. Anderson "Network Localization in Partially Localizable Networks" *Proceedings of IEEE INFOCOM 2005*, Miami, FL, March, 2005.

7. C. Savarese, J.M. Rabaey, K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*, Pages: 317 – 327, 2002.

8. K. Langendoen, N. Reijers, "Distributed Localization in Wireless Sensor Networks: A quantitative Comparison," *Computer Networks* 43 (2003) pp. 499-518.

9. J. Albowicz, A. Chen, L. Zhang, "Recursive Position Estimation in Sensor Networks" *IEEE International Conference on Network Protocols*, pages 35—41, Nov., 2001.

10. N. Bulusu, J. Heidemann, D. Estrin, "GPS-less Low-Cost Outdoor Localization for Very Small Devices," *IEEE Personal Communications*, pp. 28-34, October 2000.

11. Y.M. Kwon, K. Mechitov, S. Sundresh, W. Kim, G. Agha, "Resilient Localization for Sensor Networks in Outdoor Environments," *Proceedings of the 25th IEEE inter. Conf. on Distributed Computing Systems (ICSCS'05)*, 2005.

12. R.W. Gerchberg, W.O. Saxton, "A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures," *Optik*, vol. 35, pp. 237-246, 1972.

13. J.R. Fienup, "Phase Retrieval Algorithms: A Comparison," Applied Optics, Vol. 21, No. 15, pp. 2758-2769, Aug. 1982.

14. A-J. van der Veen, A. Paulraj, "An Analytic Constant Modulus Algorithm," *IEEE Trans. on Signal Processing*, Vol. 44, No. 5, May 1996.

15. B. Friedlander, "A Passive Localization Algorithm and its Accuracy Analysis," *IEEE Journal of Oceanic Engineering*, Vol. OE-12, No. 1, pp. 234-245, Jan. 1987.

16. J.O. Smith and J. Abel, "The Spherical Interpolation Method of Source Localization," *IEEE Journal of Oceanic Engineering*, Vol. OE-12, No. 1, pp. 246-252, Jan. 1987.

17. H.L. Van Trees, *Detection, Estimation, and Modulation Theory, Part 1*, John Wiley and Sons, 2001.

18. A.J. Weiss, "On the Accuracy of a Cellular Location System Based on RSS Measurements," *IEEE Transactions on Vehicular Technology*, Vol. 52, No. 6, November, 2003.

Appendix: Real Numbers Formulation for Three Dimensions

The algorithm can be easily extended to 3 dimensions by replacing the complex vectors with matrices. Denote the Cartesian coordinates of the *j*-th vector by x_j, y_j, z_j. Define the matrix of sensor coordinates,

$$\begin{aligned} \mathbf{X} &\triangleq [\mathbf{p}_1, \mathbf{p}_2, \cdots \mathbf{p}_{n+m}]^T \\ \mathbf{p}_j &\triangleq [x_j, y_j, z_j]^T \end{aligned} \tag{43}$$

Equation (2) becomes,

$$\mathbf{EX} = \mathbf{Y}; \quad \mathbf{E} \in \mathbb{R}^{L \times n+m}, \ \ \mathbf{X} \in \mathbb{R}^{n+m \times 3}, \ \ \mathbf{Y} \in \mathbb{R}^{L \times 3} \tag{44}$$

where **E** remains the same as in equation (2) and **Y** is a three column real matrix. Equations (3) and (4) become

$$\mathbf{E}_1\mathbf{U} + \underbrace{\mathbf{E}_2\mathbf{Z}}_{\mathbf{V}} = \mathbf{Y}, \quad \left\| Y(j,:) \right\|^2 = d_j^2 \tag{45}$$

where **U** represents the coordinates of the unknown stations and **Z** represents the coordinates of the anchors. We use $Y(j,:)$ to represent the *j*-th row of **Y**.

The *j*-th row of (45) is given by,

$$\left\| \mathbf{q}_j^T \mathbf{U} + V(j,:) \right\|^2 = \left\| Y(j,:) \right\|^2 = d_j^2 \tag{46}$$

which is equivalent to,

$$\mathbf{q}_j^T \mathbf{U}\mathbf{U}^T \mathbf{q}_j + \mathbf{q}_j^T \mathbf{U} V^T(j,:) + V(j,:)\mathbf{U}^T \mathbf{q}_j = d_j^2 - \left\| V(j,:) \right\|^2 \tag{47}$$

Using the Kronecker product we get,

$$\left(\mathbf{q}_j^T \otimes \mathbf{q}_j^T \right) \sum_{k=1}^{3} \mathbf{u}_k \otimes \mathbf{u}_k + 2\mathbf{q}_j^T \sum_{k=1}^{3} V_{jk} \mathbf{u}_k = d_j^2 - \left\| V(j,:) \right\|^2 \tag{48}$$

where $\mathbf{u}_k$ is the k-th column vector of $\mathbf{U}$.

Collecting all rows we get,

$$\begin{aligned} &\mathbf{Q} \sum_{k=1}^{3} \mathbf{u}_k \otimes \mathbf{u}_k + \sum_{k=1}^{3} \mathbf{G}_k \mathbf{u}_k = \mathbf{b} \\ &\mathbf{G}_k(j,:) \triangleq 2\mathbf{q}_j^T V_{jk} \end{aligned} \tag{49}$$

We now find the projection matrix, $\mathbf{P}$, for the space orthogonal to the column space of $\mathbf{Q}$. Left multiplying (49) by $\mathbf{P}$ we get,

$$\sum_{k=1}^{3} \mathbf{P}\mathbf{G}_k \mathbf{u}_k = \mathbf{P}\mathbf{b} \tag{50}$$

This is a linear set of equations similar to the set of equations corresponding to planar geometry.

The Gerchberg Saxton iterations become,

$$\begin{aligned} &i) \quad Y^{(k)}(j,:) = \mathbf{q}_j^T \mathbf{U} + V(j,:) \\ &ii) \quad \overline{Y}^{(k)}(j,:) = \frac{d_j Y^{(k)}(j,:)}{\left\| Y^{(k)}(j,:) \right\|} \\ &iii) \quad \mathbf{U}^{(k+1)} = (\mathbf{E}_1^T \mathbf{E}_1)^{-1} \mathbf{E}_1^T (\overline{\mathbf{Y}}^{(k)} - \mathbf{V}) \end{aligned} \tag{51}$$

Proof of convergence goes along the same lines as for the two-dimensional case.