1 Introduction

During this lecture the professor discussed the papers: Mobile Assisted Localization in Wireless Sensor Networks by Priyantha et al 2005 and Theory of Network Localization by Aspnes et al. The following announcements for the class were given:

- The Professor updated the website and organized the topics and the corresponding papers.
- The papers that will be discussed in the next few weeks are already posted.
- The homeworks will essentially consists on paper summaries as in Homework 1.
- The next topic will be clock synchronization.

2 Mobile Assisted Localization in Wireless Sensor Networks (First Paper)

If nodes are sparsely distant then it is difficult to use the known strategies that we already studied like anchor nodes and beacon nodes. What if we don’t have enough nodes? Then it is difficult to produce a set of localization coordinated based on pair wise distances. To solve this problem the use of Mobile-Assisted Localization (MAL) is introduced. With this technique we use a handheld device which moves around the area searching for devices and is used to create beacon nodes and get distances from other nodes. These temporary “virtual nodes” should be located in strategic locations.

Because we are using a handheld device to find beacon nodes, there are no anchor nodes. Thus rotation, translation and reflections are possible and the local coordinates are unique with respect to these three. So the idea here is to locate these temporary nodes and this paper uses the theory of rigidity to accomplish this purpose.

Global Rigidity
Given \( n \) nodes \( \{1, 2, \ldots, n\} \) and \( \delta : (1, 2, \ldots, N)^2 \Rightarrow \mathbb{R} \) where \( \delta \) is a distance function, what can we say about the structures defined by these nodes and the set of distances? If the whole structure can be flipped or rotated then we can move it around looking the structure as one piece. The paper lists several kinds of rigidity where global is the strongest form of rigidity.

**Local Rigidity**

We encounter local rigidity when we don’t have global rigidity but there is some substructure which you can move around as a whole. In the picture for example, we notice that the triangle can be rotated, but the angle and slope changes with respect to the other nodes.

**Not Locally Rigid**

In this case we have a lack of relationship between any of the coordinates if they are moved around. This implies that the structure changes in a local sense and therefore has no local rigidity, even for a substructure.

Why do we care about rigidity? Rigidity has been studied by a lot of disciplines such as structural engineering, mathematics and molecular structures. There is already a lot of exist-
ing theory on this field and computer scientists are looking to see how it applies to network localization, especially since we are trying to figure out the position of nodes based on distances.

There was a question about How many structures you can come up with given a set of distances? The professor responded that this would depend on whether the set of distances defined a globally rigid structure. If so, there would be a unique structure defined, which could be translated rotated or reflected in space, still preserving the structure.

Another question was how to check the change of the angle after a change by rotation? The professor answered that this may be NP-hard in general, but that is not what we will do to figure out if a structure is rigid. We are trying to use the theory of rigidity to figure out if the network localization problem is solvable. Right now, we are defining things to set up the connection between the two.

3 A Little Rigidity Theory

The purpose of rigidity theory is guaranteeing uniqueness of a structure in 2D or 3D space.

Definition 1 An n-point formation in d-space consists of an assignment of coordinates to points $p_1, \ldots, p_n$ in d-space and a set of edges and length of edges, where d-space is the d-dimensional space.

Given a formation or a set of points in d-space, can we solve the network localization problem? We don’t know what the points really are. If we establish a relationship between rigidity and the localization problem then we can talk in terms of rigidity instead. We are going to try to bring more nodes in order to create a rigid structure.

Definition 2 An n-point formation $P$ in d-space is globally rigid if any other n-point formation $Q$ with the same edges and same length of those edges is the same as $P$ up to translation, rotation and reflection.

Globally rigid and not rigid are easy terms to define. Locally rigid is not as easy because there exists different kinds of it. For a graph to be globally rigid in d-dimensions we must have:

1. The removal of any $d$ vertices must leave the graph connected. Why this is true? Because the graph should be very connected to be able to rotate pieces of it.

2. The removal of any edge must leave the graph locally connected.

How does MAL work? A sub problem in this category is to determine the distances between existing nodes. Where should the virtual nodes be placed and distances be measured? In 3D each additional node gives 3 more unknowns but only 2 constraints. So, how would it help to add nodes?
As shown in the figure above, we can see that if we start with two nodes, we can move a mobile node along a line and take measurements at three distinct locations. The trick is to get distance measurements at 3 locations $m_0, m_1, m_2$ that are collinear. We can argue we would have the rigidity needed. With these three points we can get all the distances between these three anchors and the other two points.

**Fact 1** A point formation of 5 co-planar points $n_0, n_1, m_0, m_1, m_2$ where $m_0, m_1, m_2$ are collinear, together with the edges $(n_i, m_j) \forall i, j$ is globally rigid.

**Fact 2** A point formation of 9 points $n_0, n_1, n_2, m_0, m_1, \ldots, m_5$ where $n_0, n_1, n_2$ are not collinear and the $m'_j$s are co-planar is globally rigid.

**Fact 3** A point formation of 11 points $n_0, n_1, n_2, n_3, m_0, m_1, \ldots, m_6$ where $n_0, n_1, n_2, n_3$ are not coplanar and the $m'_j$s are co-planar is globally rigid.

We claim this to be true using rigidity.

We can keep extending the previous pattern and doing so will take polynomial time. MAL correctness is based on rigidity theory and it produces very accurate position estimates. The paper claim that MAL can be put in any localization algorithm like Anchor Free Localization (AFL) which is an alternative to the localization approach provided by Savvides that was studied in class.

### 4 Theory of Network Localization by Aspnes et al

Consider a connected network $N$ in $d$-space with $n$-nodes $1, 2, \ldots, n$ located at fixed points $p_1, p_2, \ldots, p_n$ in $\mathbb{R}^d$. Each node has neighbors and this neighbor relationship is assumed to be
symmetric (i.e. if \(a\) is a neighbor of \(b\) then \(b\) is a neighbor of \(a\)). The first \(m\) nodes are special beacon nodes and we know the position of those under the assumption that beacon nodes are fixed. What we are trying to formalize here is that given the definition of the network, the set of distances and positions, can we answer the localization question? The answer is yes only if the network is rigid. We represent this network as an undirected graph \(G_N = (V, E_N)\) with vertices 1, 2, \ldots, \(N\) and edges between vertices that are the neighbors in the network. Now we can define the network localization problem \((nl)\).

**Definition 3** The network localization problem \((nl)\) takes as input \(G_N, \delta_n,\) and \(\{P_1, \ldots, P_m\}\) and produces the output \((P_{m+1}, \ldots, P_n)\). This definition of \(nl_N\) says that given the network graph, the positions of \(m\) beacons and the distances between neighbors \((\delta_N(i, j)\) for \((i, j) \in E_N)\), we want to determine the position of all the remaining nodes in the network consistent with distance constraints.

The Network Localization Solvability Problem \((nls)\) asks if, given a network graph of \(n\) nodes, the positions of \(m\) beacons and the distances between the neighbors, there is a unique solution for the positions of the remaining nodes. More formally, given a graph \(G_N\), the distance function \(\delta_n\), and the beacons \(P_1, \ldots, P_m\), is there exactly one set of positions \(P_{m+1}, \ldots, P_n\) that satisfy these constraints?

We define the Generic Network Localization Solvability Problem \((gnls)\) which is a more restrictive version of \(nls\). Given a network graph, a distance function and the position of the beacons, the solution should not only be unique at this set of points, but also unique in an open neighborhood of these points. In other words, there should be a unique solution not only for the given input, but even for lightly perturbed versions of this input.

**Definition 4** A Grounded Network Graph is a Network Graph where each beacon node has an edge to every other beacon node.

This implies that we can calculate the exact distance between beacon nodes. We will be considering only grounded network graphs from this point on.

Model a network with a point formation \(F_p = (p_1, p_2, \ldots, p_n, L)\) \(\forall i, p_i \in \mathbb{R}^d\)

\[
\begin{pmatrix}
p_1 \\
p_2 \\
\vdots \\
p_n
\end{pmatrix}
\]

\(L\) is a set of \(k\) links \(\forall (i, j) \in L, \text{length}(i, j)\) is the Euclidean distance between \(p_i\) and \(p_j\).

A Formation \(F_p = (p_1, \ldots, p_n, L)\) uniquely determines:

1. An undirected graph \(G_{F_p} = (V, L)\)
2. A distance function \(\delta_{F_p} : L \to \mathbb{R} \forall (i, j) \in L, \delta_{F_p}(i, j)\) is the length of the edge.
Definition 5  A map \( T : \mathbb{R}^d \Rightarrow \mathbb{R}^d \) is a distance preserving map if \( \forall p, q \in \mathbb{R}^d, |T(p) - T(q)| = |p - q| \) (i.e. distance between \( p \) and \( q \) is the same as the distance between \( T(p) \) and \( T(q) \)).

Question raised in class: How do we compute the function algorithmically?
The professor answered that computing this function seems to be hard but we don’t have to explicitly find it. These definitions are just to help us understand rigid structures and argue properties about them. We will come up with the algorithm later.

Definition 6  Two \( n \)-point columns \( p \) and \( q \) in \( \mathbb{R}^d \) are congruent if \( \exists T : \mathbb{R}^d \Rightarrow \mathbb{R}^d \) (i.e. there exists a distance preserving map) s.t. \( T \) is distance preserving and \( \forall i \in \{1, 2, \ldots, n\}, T(q_i) = p_i \).

We are mapping \( q_i \)'s \Rightarrow p_i \)'s and distances are preserved. Not only the points \( q_1, q_2, \ldots, q_n \) are mapped but all points in \( \mathbb{R}^d \) are.

Definition 7  Two point formation \( F_p \) and \( F_q \) are congruent if

1. \( G_{F_p} = G_{F_q} \)
2. Point columns \( p \) and \( q \) are congruent.

Given \( F_p \) and \( F_q \) that are congruent, one can be translated into another. If one is globally rigid, the other is too.

\( F_p \) and \( F_q \) are congruent \( \Rightarrow \delta_{F_p} = \delta_{F_q} \)

Does \( G_{F_p} \) and \( \delta_{F_p} \) uniquely determine the formation \( F_p \)? No, clearly any formation \( F_q \) that is congruent to \( F_p \) also has the same graph and distance set. Also, there may be other formations that are not congruent to \( F_p \) but still have the same graph and distance set.

If \( F_p \) is globally rigid, \( G_{F_p} \) and \( \delta_{F_p} \) uniquely determines \( F_p \) up to congruence. This means that the graph and distance set determine only the formations that are congruent to \( F_p \). A formation that is not congruent to \( F_p \) cannot have the same graph and distance set, if \( F_p \) is globally rigid.

Fact 4  \( F_p \) is globally rigid if \( \forall F_q (G_{F_q} = G_{F_p} \text{ and } \delta_{F_p} = \delta_{F_q}) \Rightarrow F_p \text{ is congruent to } F_q \).

There was a question whether angles are preserved between two congruent structures. When using distance preserving mapping we are keeping the structure, since every pair of points preserve their distances. Thus, angles are also preserved.

The discussion of this paper will continue in the next lecture and be followed by Aditya Dhananjay and Puviarasan Pandian’s presentations on Time Synchronization.