Chapter VII
Sequential Localization with Inaccurate Measurements

Jia Fang
Yale University, USA

Dominique Duncan
Yale University, USA

A. Stephen Morse
Yale University, USA

ABSTRACT

The sensor network localization problem with distance information is to determine the positions of all sensors in a network given the positions of some sensors and the distances between some pairs of sensors. In this chapter the authors present a sequential algorithm for estimating sensor positions when only inaccurate distance measurements are available, and they use experimental evaluation to demonstrate network instances on which the algorithm is effective.

INTRODUCTION

In many situations where wireless sensor networks are used, only the positions of some of the sensors are known, and the positions of the remaining sensors must be inferred from the known locations and available inter-sensor distance measurements. More formally, consider n sensors in the plane labelled 1 through n, where the positions of some sensors are known, and the measured distances between some pairs of sensors are known. The sensors with known positions are called *anchors*. Since ranging devices are never exact, we consider the following model for the type of distance measurements obtained. For each inter-sensor distance measurement $\tilde{\tilde{d}}$, we assume that an *accuracy guarantee* denoted by $\varepsilon > 0$ of $\tilde{d}$ is given such that the actual inter-sensor distance is within ε of the measured distance $\tilde{d}$. Obviously,

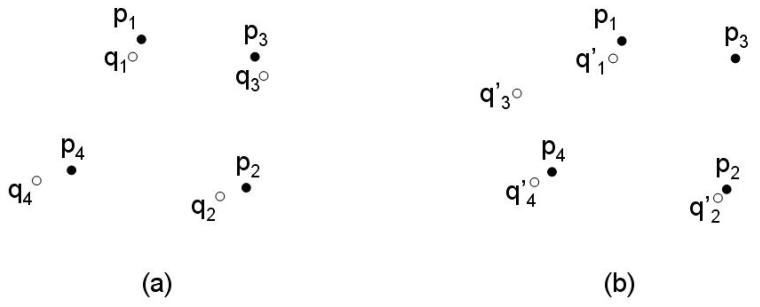
sensor positions are generally not uniquely determined by inaccurate distance measurements. In Moore et al.(2004) and Priyantha et al. (2003), it is pointed out that it is more important to obtain position estimates which reflect the general layout of the actual sensor positions, rather than simply position estimates which induce inter-sensor distances within some desired tolerance of the given inter-sensor distance measurements. In Priyantha et al. (2003), a modified spring based relaxation method is used to obtain sensor estimates, and it is shown via experimental evaluations that the estimated positions reflect the general layout of the actual sensor positions. Our work is most closely related to Moore et al. (2004) where the aim was to compute position estimates for subnetworks called "robust quadrilaterals" with correctness guarantees. More specifically, an algorithm is given which assigns position estimates to the sensors in a robust quadrilateral only if the position estimates can be guaranteed to be free of "flip ambiguities" with high probability Moore et al. (2004).

Roughly speaking, position estimates are desired so that the configuration of the estimated positions are approximately congruent to that of the actual positions. We capture this notion using the concept of "correctly oriented" position estimates which we define as follows. For two points p and q in R^2, let $l(p,q)$ denote the line segment with endpoints p and q. For $m \geq 4$ sensors labelled 1 through m and $i \in \{1,...,m\}$ let p_i denote the position of sensor i, and let q_i denote the estimated position of sensor i. The estimated positions $q_1,..., q_m$ are said to be *correctly oriented* if for all distinct $i,j,k,l \in \{1,...,m\}$, the line segments $l(p_i, p_j)$ and $l(p_k, p_l)$ intersect if and only if the line segments $l(q_i, q_j)$ and $l(q_k, q_l)$ intersect. As an illustration, suppose sensors 1, 2, 3 and 4 are positioned at p_1, p_2, p_3 and p_4 respectively as shown in Figure 1(a) and 1(b). For each sensor i, let q_i and q'_i denote two estimated positions of sensor i. Suppose q_i for i=1,2,3,4 are as shown in Figure 1(a), and q'_i are as shown in Figure 1(b). It is easy to see that $\{q_1, q_2, q_3, q_4\}$ are correctly oriented while $\{q'_1, q'_2, q'_3, q'_4\}$ are not correctly oriented.

In Section Correctly Oriented Position Estimates, we will demonstrate that correctly oriented position estimates can be used to deduce important geometric properties of the configuration of the actual sensor positions. The key difficulty however lies in determining if a set of position estimates are correctly oriented without knowing the corresponding actual sensor positions. In this work we will propose a position estimation algorithm for computing position estimates with *error bounds*, and give a sufficient condition on the position estimates and the corresponding error bounds for the position estimates to be correctly oriented.

In Anderson et al. (2007) a sequential localization algorithm for exact distance measurements was proposed in which the sensors of the network are processed one by one in a pre-determined order. That

Figure 1. (a) Correctly oriented position estimates, (b) Position estimates which are not correctly oriented



work was extended in Fang et al. (2006), Goldenberg et al. (2006) and Fang et al. (2008) to a sequential localization algorithm called Sweeps again for the case of *exact* distance measurements. In this chapter, we present an algorithm based on Sweeps, which we call modified Sweeps, for estimating sensor positions of a network when only inaccurate distance measurements can be obtained. In modified Sweeps, we aim to give correctness guarantees on estimated positions and characterize position estimates which are oriented correctly. More specifically, for each position estimate p computed by modified Sweeps, an error bound $e(p)$ is also computed such that the maximum distance between the position estimate p and the actual position is at most $e(p)$. The error bounds can be used by the final application to determine which sensor estimates are precise enough to be useful. More importantly, we will use the error bounds to give a sufficient condition for the position estimates to be correctly oriented, for as will be shown in Section Correctly Oriented Position Estimates, position estimates which are correctly oriented can be used to deduce geometric properties of the configuration of actual sensor positions. We note that however not all sensors are assigned position estimates even when the average degree of the network exceeds six. Like the algorithms proposed in Moore et al. (2004) and Anderson et al. (2007), modified Sweeps is a sequential algorithm in the sense that the sensors are processed one by one in some order. As was shown in Moore et al. (2004) and Anderson et al. (2007), the notion of processing sensors in a particular order can be used to gain important insight into the graphical characterization of networks whose positions or position estimates can be computed *efficiently*. In Section Efficiently Localizable Networks, we will give the graphical characterization of some networks for which modified Sweeps can be used to efficiently compute position estimates.

In Section Experimental Evaluations, we discuss the performance of modified Sweeps on a network of 100 sensors randomly deployed in a unit square. A noisy distance measurement is generated for each pair of sensors within a specified sensing radius. Roughly speaking, a distance measurement with a guaranteed accuracy of ε is "noisier" than a distance measurement with a guaranteed accuracy δ when $\delta < \varepsilon$. In general, we note that the number of sensors for which a position estimate is obtained increases with the average degree of the network, and as expected, the error bound associated with each position estimate increases as distance measurements become noisier. However, even for the case where the guaranteed accuracy of each distance measurement d is 8% of d, the respective error bounds of the position estimates is on average less than 1/6 of the sensing radius. The trade-off however is that only ~55% of the sensors are assigned position estimates.

HIGH LEVEL DESCRIPTION OF MODIFIED SWEEPS

In this section, we give a high level description of modified Sweeps and illustrate some key aspects of the algorithm via examples. A *candidate regions set* of a sensor is a set consisting of a finite number of regions in the plane with the property that the actual position of the sensor is in one of the regions, and the regions are either all polygons or all disks. We call each region in a candidate regions set of a sensor a *candidate region* of the sensor. We say that a candidate region of a sensor is *false* if the region does not contain the sensor's position. When the candidate regions set of a sensor consists of only one region, say with diameter d, then the centroid of the region must be within at most distance $d/2$ from the sensor's actual position. Obviously, it is most desirable to obtain for each sensor a candidate regions set which consists of just one candidate region with a "small" diameter.

Roughly speaking, modified Sweeps first computes a candidate regions set for each sensor by processing the sensors one by one in some order, and then refines each candidate regions set by process-

ing the sensors in an ordering different from the previous ordering. More specifically, to determine a candidate regions set for each sensor, an ordering of the sensors is first determined so that the anchors precede all other sensors in the ordering, and each non-anchor sensor has at least one "predecessor" in the ordering. A *predecessor* of a sensor is any other sensor preceding it in the ordering such that the measured distance between the two sensors is obtained. Assuming such an ordering exists, the algorithm "sweeps" through the network by processing the sensors sequentially according to the ordering, beginning with the first non-anchor sensor in the ordering. For each non-anchor sensor, a candidate regions set is computed for the sensor using the measured distances between the sensor and its predecessors, and the candidate regions sets, or known positions, of its predecessors. Once the last sensor in the ordering is processed, a candidate regions set will have been computed for each sensor. We call this first "sweep" a *candidate regions set generating sweep.* Once a candidate regions set has been computed for each sensor, subsequent "refining" sweeps are performed to remove, if possible, regions from each candidate regions set so as to obtain a candidate regions set of fewer elements. To perform a refining sweep, an ordering distinct from the one used to perform the previous sweep is determined, and the sensors are again processed sequentially according to the new ordering. In the following, we will use *singleton* to refer to a set consisting of exactly one element. For each non-anchor sensor v with a non-singleton candidate regions set, the candidate regions sets of v's predecessors, and the measured distances between v and its predecessors, are used to identify, if possible, those candidate regions of sensor v which do not contain v's actual position. More specifically, for each sensor, the minimum and maximum distances between the sensor's candidate regions and the candidate regions of its predecessors in the new ordering are used as a means of identifying false candidate regions of the sensor. The false candidate regions are then removed from the candidate regions set of the sensor to obtain a candidate regions set of fewer elements. Note that when two candidate regions are both disks or both polygons, the minimum and maximum distances between them are efficient to compute.

A sensor is localized if its candidate regions set consists of a point, i.e. a region with diameter zero. As noted previously, exact positions in general cannot be computed when distance measurements are inaccurate. Hence, the desired outcome for each sensor is that after a finite number of sweeps through the network, the candidate regions set of the sensor contains just one candidate region with a "small" diameter. Whether this will be the case will depend on the geometry of the configuration of the actual sensor positions. In Section Experimental Evaluations, we will show that for randomly deployed networks of 100 sensors, it is possible to obtain singleton candidate regions sets with error bounds less than 1/6 of the sensing range. In the following two sections, we will give examples of using modified Sweeps to estimate sensor positions in two simple networks. Although the networks are simple, they illustrate the two key aspects of modified Sweeps, namely generating and refining candidate regions sets.

Generating Candidate Regions Sets

For a pair of sensors i and j in a network, let d_{ij} denote the actual distance between the sensors, and if the measured distance between them is obtained, let $\widetilde{d}_{ij}$ denote the measured distance, and let ε_{ij} denote the guaranteed accuracy of $\widetilde{d}_{ij}$. By definition, if the measured distance $\widetilde{d}_{ij}$ is obtained for sensors i and j, then the actual distance d_{ij} between i and j must be within ε_{ij} of the measured distance:

$$d_{ij} \in [\widetilde{d}_{ij} - \varepsilon_{ij}, \widetilde{d}_{ij} + \varepsilon_{ij}] \quad (1)$$

Consider the network of four sensors labelled *a,b,c,v* and positioned at points π(a), π(b), π (c) and π (v) respectively, so that no three of the sensor positions are collinear. The sensors labelled *a*, *b* and *c* are anchors, and suppose distance measurements $\widetilde{d}_{av}$, $\widetilde{d}_{bv}$ and $\widetilde{d}_{cv}$ are obtained. Using the guaranteed accuracies of the distance measurements, we get the following:

$$d_{av} \in [\widetilde{d}_{av} - \varepsilon_{av}, \widetilde{d}_{av} + \varepsilon_{av}], d_{bv} \in [\widetilde{d}_{bv} - \varepsilon_{bv}, \widetilde{d}_{bv} + \varepsilon_{bv}], \; d_{cv} \in [\widetilde{d}_{cv} - \varepsilon_{cv}, \widetilde{d}_{cv} + \varepsilon_{cv}] \qquad (2)$$

The network is denoted by the graph shown in Figure 2 where each vertex corresponds to the sensor of the same label, and two vertices are adjacent if either the corresponding sensors are both anchors, or the distance measurement between the sensors is obtained.

An ordering of the sensors is first determined so that the anchors precede all other sensors, and each non-anchor sensor has at least one predecessor. One such ordering is *a,b,c,v*, and the predecessors of sensor *v* are *a*, *b* and *c* since the distance measurements between sensor *v* and anchors *a,b,c* are given. For $x \in \{a,b,c\}$, let A_x denote the ring centered at π(*x*) with inner radius $\widetilde{d}_{xv} - \varepsilon_{xv}$ and outer radius $\widetilde{d}_{xv} + \varepsilon_{xv}$. From (2) it follows that $\pi(v) \in A_a \cap A_b \cap A_c$. Suppose the three rings intersect as shown in Figure 3a.

The rings' intersection can be approximated by a disk D which contains $A_a \cap A_b \cap A_c$ as shown in Figure 3b. Clearly, $\pi(v) \in$ D, so the singleton set containing just D is a candidate regions set for sensor *v*. Ideally, D would be a disk with the least diameter that contains the common intersection of the three rings. However, modified Sweeps does not require this. In our implementation of modified Sweeps, we use an efficient algorithm to compute an approximating polygon which contains the ring intersection. Although the computed polygon is by no means the polygon of the least diameter which contains the ring intersection, experimental evaluations indicate that it is adequate in the sense that it yields position estimates for a non-trivial number of non-anchor sensors with reasonable error bounds. We discuss implementation details in Section Experimental Evaluations.

In the network above, it may have seemed redundant to approximate the common intersection of the three rings by a disk. In the next section we consider a network of five sensors, three of which are anchors, for which no ordering exists so that each of the non-anchor sensors has three anchors as predecessors. We use such a network to illustrate a refining sweep and in the process justify the assumption that candidate regions are constrained to be either disks or polygons.

Figure 2. Sensors a, b and c are anchors

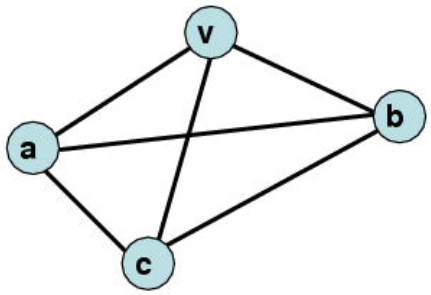
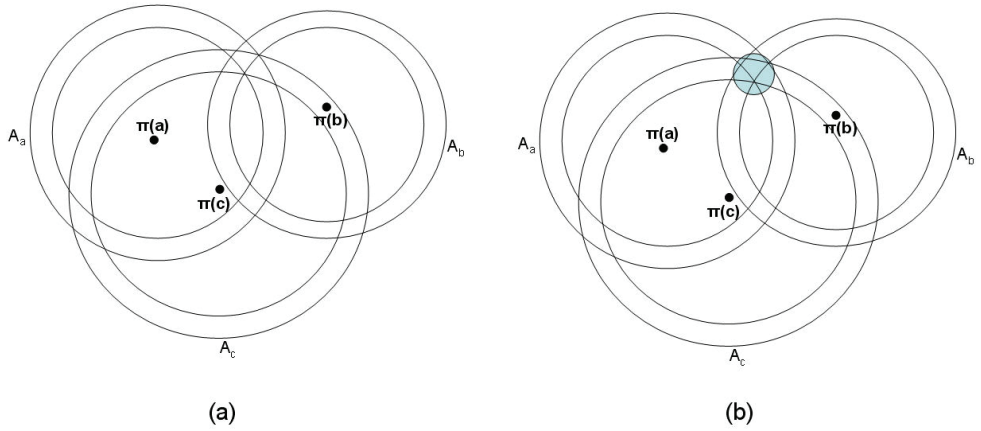


Figure 3. Rings intersection



Refining Sweep

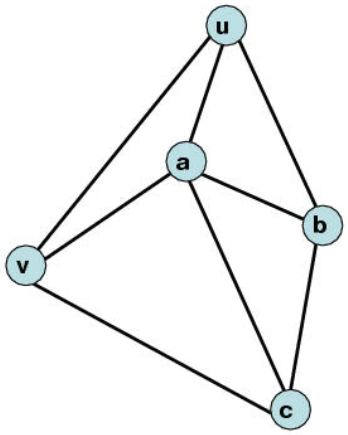
Consider the network which consists of five sensors labelled *a,b,c,u,v* and positioned at points *π(a)*, *π(b)*, *π(c)*, *π(u)* and *π(v)* respectively, so that no three of the sensor positions are collinear. Sensors labelled *a*, *b* and *c* are anchors, and measurements $\widetilde{d}_{au}$, $\widetilde{d}_{bu}$, $\widetilde{d}_{av}$, $\widetilde{d}_{cv}$ and $\widetilde{d}_{uv}$ are obtained. The network is denoted by the graph in Figure 4. Note that even when the distance measurements are exact, neither the positions of sensor *u* nor *v* are uniquely determined by their distances to the anchors alone. Let ε_{au}, ε_{bu}, ε_{av}, ε_{cv} and ε_{uv} denote the guaranteed accuracies of $\widetilde{d}_{au}$, $\widetilde{d}_{bu}$, $\widetilde{d}_{av}$, $\widetilde{d}_{cv}$ and $\widetilde{d}_{uv}$ respectively.

Using the measured distances and the corresponding guaranteed accuracies, we get:

$$d_{xu} \in [\widetilde{d}_{xu} - \varepsilon_{xu} \ , \ \widetilde{d}_{xu} + \varepsilon_{xu}], \text{ for all } x \in \{a, b\} \tag{3}$$

$$d_{xv} \in [\widetilde{d}_{xv} - \varepsilon_{xv} \ , \ \widetilde{d}_{xv} + \varepsilon_{xv}], \text{ for all } x \in \{a, c\} \tag{4}$$

Figure 4. Sensors a, b and c are anchors



$$d_{uv} \in [\tilde{d}_{uv} - \varepsilon_{uv}, \tilde{d}_{uv} + \varepsilon_{uv}] \quad (5)$$

An ordering of the sensors is first determined so that the anchors precede all other sensors, and each non-anchor sensor has at least one predecessor. One such ordering is *a,b,c,u,v*. Since *u* is the first non-anchor sensor, the first "sweep" begins by determining a candidate regions set for sensor *u*. Let A_a denote the ring centered at $\pi(a)$ with inner radius $\tilde{d}_{au} - \varepsilon_{au}$ and outer radius $\tilde{d}_{au} + \varepsilon_{au}$, and let A_b denote the ring centered at $\pi(b)$ with inner radius $\tilde{d}_{bu} - \varepsilon_{bu}$ and outer radius $\tilde{d}_{bu} + \varepsilon_{bu}$. It follows from (3) that $\pi(u) \in A_a \cap A_b$. Suppose the two rings intersect in two disjoint regions as shown in Figure 5a.

For reasons we will soon make clear, a disk approximation of the ring intersection is computed. The disk approximation is required to contain all the regions of intersection, and consists of two disjoint disks, each containing one of the contiguous regions of intersection. Ideally, the disk approximation would be two disks with the smallest diameters whose union contains the regions of intersection; however, the modified Sweeps algorithm does not require this. A valid disk approximation is shown in Figure 5b, and let D_u and D'_u denote the two disks. Note that D_u and D'_u are disjoint, and each of D_u and D'_u contains one of the contiguous regions of intersection. Obviously, $\pi(u) \in D_u \cup D'_u$, and the set consisting of D_u and D'_u is a candidate regions set for sensor *u*. In the limit case of exact distance measurements, the two rings will actually be circles and intersect in at most two points, in which case the disk approximation of the intersection region should simply be the set of intersection points.

The candidate regions set for sensor *v* is computed in the same way as that for sensor *u*. Let A'_a denote the ring centered at $\pi(a)$ with inner radius $\tilde{d}_{av} - \varepsilon_{av}$ and outer radius $\tilde{d}_{av} + \varepsilon_{av}$, and let A'_c denote the ring centered at $\pi(c)$ with inner radius $\tilde{d}_{cv} - \varepsilon_{cv}$ and outer radius $\tilde{d}_{cv} + \varepsilon_{cv}$. It follows from (4) that $\pi(v) \in A'_a \cap A'_c$. Suppose the two rings intersect at two disjoint regions as shown in Figure 6a. As shown in Figure 6b, let D_v and D'_v denote the two disks in the disk approximation. Clearly, the set consisting of D_v and D'_v is a candidate regions set for sensor *v*. In the actual modified Sweeps algorithm, the candidate regions set of sensor *u* is also used in the computation of the candidate regions set for sensor *v* since *u* is a predecessor of *v* in the chosen ordering. However, we skip this step in an effort to keep this example simple.

Since both computed candidate regions sets consist of more than one disjoint regions, a refining sweep will be performed to identify false candidate regions in each of the candidate region sets. We

Figure 5. Intersection of two rings, and its disk approximation

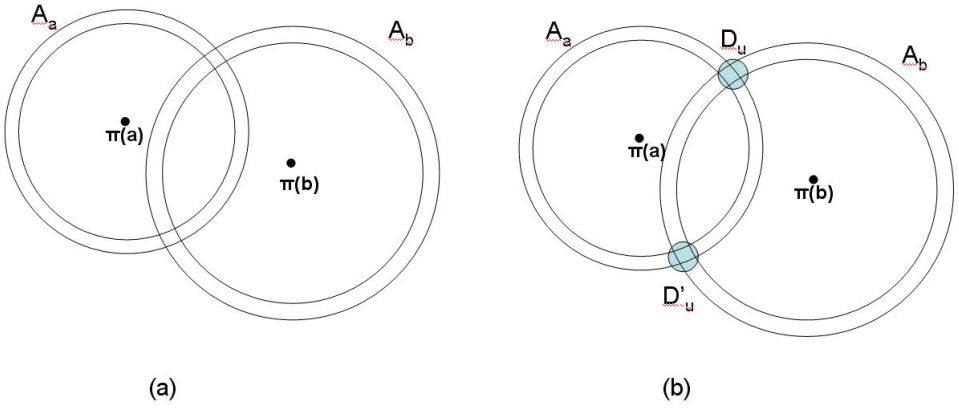
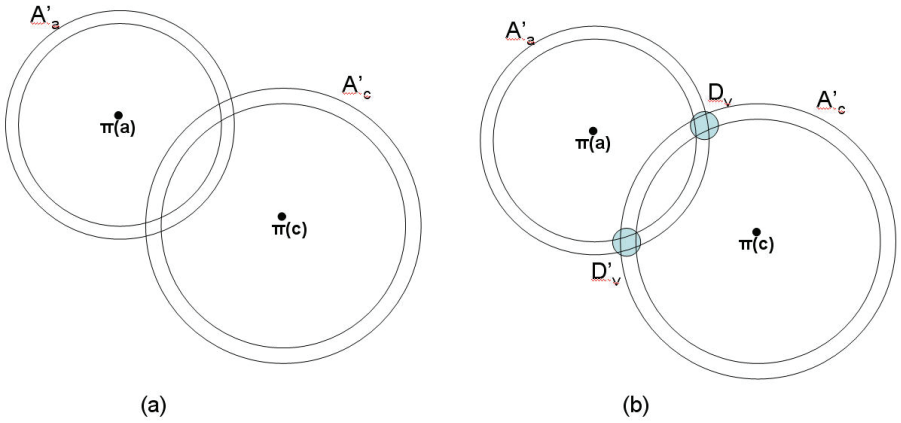


Figure 6. Intersection of two rings, and its disk approximation



first describe the underlying idea behind identifying false candidate regions of a sensor. Let D_1 and D_2 be either two disks or two polygons in the plane, and let $d_{min}(D_1, D_2)$ denote the minimum among all distances between a point in D_1 and a point in D_2:

$$d_{min}(D_1, D_2) = min_{p1 \in D1,\, p2 \in D2} \| p_1 - p_2 \| \quad (6)$$

Let $d_{max}(D_1, D_2)$ denote the maximum among all distances between a point in D_1 and a point in D_2:

$$d_{max}(D_1, D_2) = max_{p1 \in D1,\, p2 \in D2} \| p_1 - p_2 \| \quad (7)$$

Clearly, if $\pi(u) \in D_1$ and $\pi(v) \in D_2$, then it must be the case that $d_{uv} \in [d_{min}(D_1, D_2), d_{max}(D_1, D_2)]$. Moreover, from (5), we have that $d_{uv} \in [\widetilde{d}_{uv} - \varepsilon_{uv}, \widetilde{d}_{uv} + \varepsilon_{uv}]$, which implies $[d_{min}(D_1, D_2), d_{max}(D_1, D_2)]$ and $[\widetilde{d}_{uv} - \varepsilon_{uv}, \widetilde{d}_{uv} + \varepsilon_{uv}]$ cannot be disjoint when $\pi(u) \in D_1$ and $\pi(v) \in D_2$.

Without loss of generality, suppose $\pi(v) \in D_v$ and $\pi(u) \in D_u$. The crux of the modified Sweeps algorithm is based on the simple observation that if $\pi(v) \in D^*$, where D^* is a candidate region in the candidate regions set of sensor v, then there must be at least one candidate region D in the candidate regions set of sensor u, namely the candidate region which contains $\pi(u)$, such that:

$$[d_{min}(D^*, D), d_{max}(D^*, D)] \cap [\widetilde{d}_{uv} - \varepsilon_{uv}, \widetilde{d}_{uv} + \varepsilon_{uv}] \neq \emptyset \quad (8)$$

In other words, if for some candidate region D^* in the candidate regions set of v, we have that

$$[d_{min}(D^*, D), d_{max}(D^*, D)] \cap [\widetilde{d}_{uv} - \varepsilon_{uv}, \widetilde{d}_{uv} + \varepsilon_{uv}] = \emptyset \quad (9)$$

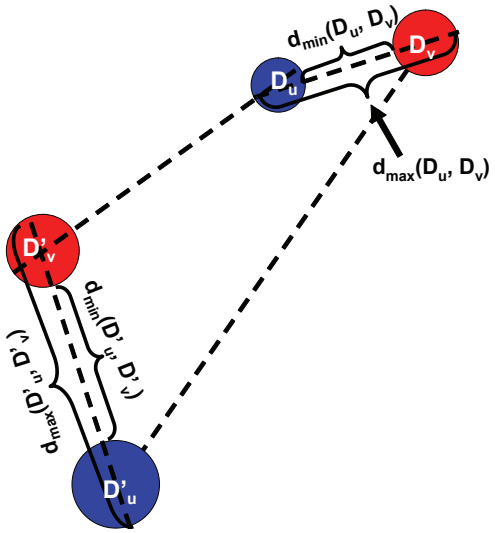
for *all* disks D in the candidate regions set of u, then it cannot be the case that $\pi(v) \in D^*$. In this case D^* can be removed from the candidate regions set of v, and the resulting set is again guaranteed to be a candidate regions set of sensor v. Note also that the minimum (maximum) among all distances

between a point in one region in the plane and another region in the plane is particularly easy to compute when the regions are either both disks or both polygons. In the case of two disjoint disks, the minimum and maximum distances between the disks can be obtained by first drawing the line containing the centers of both disks. The line intersects the boundaries of the two disks at four points, and the distance between the closest pair of those points is the minimum distance between the two disks, and the distance between the furthest pair of those points is the maximum distance between the two disks. In the case of two disjoint polygons, the two points, each belonging to one of the polygons, which induce the minimum (maximum) distance between the polygons must lie on the boundaries of the polygons which are straight line segments. Hence, the minimum and maximum distances between two polygons can be obtained by computing the minimum and maximum distances between pairs of straight line segments. When the regions are not confined to be disks or polygons, the computation is more complicated. Hence, in order to keep the computations simple and efficient, candidate regions sets are required to consist of either all disks or all polygons.

Recall that $\{D_u, D'_u\}$ and $\{D_v, D'_v\}$ are the candidate regions sets computed for u and v respectively in the first sweep. To refine the candidate regions set computed for sensor v, we process the sensors in the ordering u,v,a,b,c. Only sensor v is processed in this "refining sweep" since it is the only non-anchor sensor with a predecessor, namely sensor u, in the new ordering. Suppose the disks D_u, D'_u, D_v, and D'_v are positioned in the plane as shown in Figure 7.

If ε_{uv} is not too "large," then it is easy to see that $[d_{min}(D'_v, D_u), d_{max}(D'_v, D_u)]$ and $[d_{min}(D'_v, D'_u), d_{max}(D'_v, D'_u)]$ are both disjoint from the interval $[\tilde{d}_{uv} - \varepsilon_{uv}, \tilde{d}_{uv} + \varepsilon_{uv}]$. As shown above, the previous imply D'_v cannot contain $\pi(v)$ and so can be removed from the candidate regions set of v to obtain the new candidate regions set $\{ D_v \}$ of v. Removing D'_v from the candidate regions set of v provides a refinement of the candidate regions set of v in that the set of points in its candidate regions set is strictly reduced while still retaining the property that the remaining disks contain the sensor's actual position. A set consisting of a finite number of elements from R is said to be *algebraically independent over the rationals* if its elements do not satisfy a non-zero polynomial equation with rational coefficients. In other words, a set

Figure 7. Candidate regions sets of sensors u and v



consisting of *i* real numbers $x_1,...,x_i$ is algebraically independent over the rationals if there does not exist a non-zero polynomial equation (with rational coefficients) in *i* variables which is satisfied by $x_1,...,x_i$. The positions of a network's sensors are said to be *generic* if the set consisting of the position coordinates is algebraically independent over the rationals. When distance measurements are exact and the sensor positions are generic, it can be shown that $[d_{min}(D'_v, D_u), d_{max}(D'_v, D_u)]$ and $[d_{min}(D'_v, D'_u), d_{max}(D'_v, D'_u)]$ are always disjoint from the interval $[\widetilde{d}_{uv} - \varepsilon_{uv}, \widetilde{d}_{uv} + \varepsilon_{uv}]$, and so D'_v can be removed to obtain a smaller candidate regions set for *v* (Fang et al. (2006)).

After the refining sweep, $\{D_u, D'_u\}$ and $\{D_v\}$ are the new candidate regions sets of *u* and *v* respectively. A different ordering is now chosen so as to refine the candidate regions set of sensor *u*. Let the ordering be *v, u, a, b, c*. Again, if ε_{uv} is not too large, then it will be the case that $[d_{min}(D'_u, D_v), d_{max}(D'_u, D_v)]$ is disjoint from the interval $[\widetilde{d}_{uv} - \varepsilon_{uv}, \widetilde{d}_{uv} + \varepsilon_{uv}]$. If this holds, then D'_u cannot contain $\pi(u)$, so D'_u can be removed from the candidate regions set of *u*. The sum of the diameters of the disks in the resulting set is strictly reduced while retaining the property that at least one of the disks contains $\pi(u)$, thus refining the position estimate of sensor *u*.

Whether smaller candidate region sets of sensors *u* and *v* can be obtained depends on the configuration of the actual sensor positions, the distance measurements, and the guaranteed accuracies of the distance measurements. Intuitively, and as have been confirmed by experimental evaluations, the more accurate the distance measurements are, i.e. as $\varepsilon_{ij} \rightarrow 0$ for each measured distance $\widetilde{d}_{ij}$, the more likely it is that the candidate regions sets can be refined to be a singleton. In the limit case where distance measurements are exact, and the sensor positions are generic, it is easy to see that the candidate region sets computed for sensors *u* and *v* in the first sweep must each consist of two points. Furthermore, each of the candidate region sets can be refined to be a singleton candidate region set in the subsequent refining sweeps. In other words, if distance measurements are exact, then the two refining sweeps will remove all but the actual position from the candidate regions set of each sensor, and thus localize the network.

GLOBAL RIGIDITY AND LOCALIZABILITY

A *multi-point* $p=\{p_1,...,p_n\}$ in *d*-dimensional space is a set of *n* points in R^d labelled $p_1,...,p_n$. A multi-point is said to be *generic* if the set consisting of the coordinates of its points is algebraically independent over the rationals. Because we are only concerned with networks in the plane, we will henceforth restrict our attention to the case of *d=2*. A graph with vertex set *V* and edge set *E* is denoted by *(V, E)*. A *point formation* in R^2 of *n* points at a multi-point $p=\{p_1,...,p_n\}$ consists of *p* and a simple undirected graph G with vertex set $V=\{1,...,n\}$, and is denoted by (G, *p*). If *(i, j)* is an edge in G, then the *length of edge (i, j)* in the point formation (G, *p*) is the distance between p_i and p_j. Two point formations with the same graph have the same edge lengths if the length of each edge in the graph is the same in both point formations. Two point formations with the same graph are *congruent* if all inter-vertex distances are the same.

A point formation (G,*p*) in R^2 is *globally rigid* in R^2 if *every* other point formation in the plane with the same graph and edge lengths is congruent to (G,*p*). For any multi-point $p=\{p_1,...,p_n\}$ in R^2 and $\varepsilon > 0$, let $B_p(\varepsilon)$ denote the set of all multi-points $q=\{q_1,...,q_n\}$ in R^2 where $\| p_i - q_i \| < \varepsilon$ for all $i \in \{1,...,n\}$. A graph G is said to be *globally rigid* in R^2 if there exist multi-point *p* in R^2 and $\varepsilon > 0$ such that (G, *q*) is globally rigid in R^2 for all $q \in B_p(\varepsilon)$. It is known that if a multi-point *p* in R^2 is generic, then the point formation (G,*p*) is globally rigid in R^2 if and only if G is globally rigid in R^2 [6,7]. A number of efficient algorithms, such as Pebble Game, can be used for determining if a graph is globally rigid in R^2 [6,7,8].

A network with n sensors is modelled by a point formation (G,p) where each sensor corresponds to exactly one vertex of G, and vice versa, with *(i,j)* being an edge of G if the sensors corresponding to i and j are both anchors or if i and j are distinct and the distance measurement between the corresponding sensors is obtained, and $p=\{p_1,...,p_n\}$ where p_i is the position of the sensor corresponding to vertex i. We say that G is the graph of the network, and p is the multi-point of the network. We will restrict our attention to those networks with generic multi-points. In particular, this implies no two sensors occupy the same point and no three sensors are collinear in the networks we consider.

A network in which all distance measurements are exact is *localizable* if there corresponds exactly one position to each non-anchor sensor so that the given inter-sensor distances are satisfied. We consider the natural extension of this definition to networks with inaccurate distance measurements where computing exact sensor positions is (in general) impossible. We say that a network of n sensors positioned at $\pi(1),..., \pi(n)$ respectively is *localizable with precision* ρ if for all points $p_1,...,p_n \in R^2$ where $p_i = \pi(i)$ for all anchors i and $\| p_i - p_j \| \in (\tilde{d}_{ij} - \varepsilon_{ij}, \tilde{d}_{ij} + \varepsilon_{ij})$ for each distance measurement $\tilde{d}_{ij}$, we have that $\| p_i - \pi(i) \| \leq \rho$ for all $i \in \{1,...,n\}$. When distance measurements are exact, a network is localizable if and only if the network has three anchors and the graph of the network is globally rigid in R^2. When the distance measurements in a network are inaccurate, it is straightforward to show from the definitions that:

Lemma 1. If the graph of a network is not globally rigid in R^2, then there exists $\rho > 0$ such that the network is not localizable with precision ρ.

MODIFIED SWEEPS

We first give the terms and definitions to be used in defining the modified Sweeps algorithm. In the following, let N be a network of n sensors labelled 1 through n where each sensor i is positioned at $\pi(i)$. Let G=(V,E) denote the graph of N where $V=\{1,...,n\}$, and each vertex i corresponds to sensor i. For each vertex v, let $N(v)$ denote the set of vertices adjacent to v in G. We assume there are at least three anchors and that G is connected. For each $(i,j) \in E$ where at least one of i or j is a non-anchor sensor, let $\tilde{d}_{ij}$ denote the distance measurement obtained between sensors i and j, and let d_{ij} denote the actual distance between sensors i and j, i.e. $d_{ij} = \| \pi(i) - \pi(j) \|$. For each measured distance $\tilde{d}_{ij}$, let ε_{ij} denote the guaranteed accuracy of the measured distance. This implies that for each $(i,j) \in E$, $d_{ij} \in [\tilde{d}_{ij} - \varepsilon, \tilde{d}_{ij} + \varepsilon]$. To avoid degenerate cases, we will assume that $\varepsilon_{ij} < \tilde{d}_{ij}$ for all measured distance $\tilde{d}_{ij}$.

A *mapping* α has as its domain a non-empty subset U of V, and for each element u in U, $\alpha(u)$ is either a disk or polygon in the plane. Given any mapping α with domain U, α is called a *disk mapping* if $\alpha(u)$ is a disk for all $u \in U$, and a *polygon mapping* if $\alpha(u)$ is a polygon for all $u \in U$. For mapping α, let $\Delta(\alpha)$ denote the domain of α. Two mappings α and β are said to be *consistent with each other*, and we write $\alpha \sim \beta$, if for all $u \in \Delta(\alpha) \cap \Delta(\beta)$, α and β map u to the same region in the plane. Two mappings are always consistent if their domains are disjoint. For any positive integer k, consider a collection of k pairwise consistent mappings $\alpha_1,..., \alpha_k$ which are either all disk mappings or all polygon mappings. Let $u_k(\alpha_1,..., \alpha_k)$ denote the mapping with domain $\bigcup_{i \in \{1,...,k\}} \Delta(\alpha_i)$ whose restriction to $\Delta(\alpha_i)$ is equal to α_i for each $i \in \{1,...,k\}$. For a disk or polygon P in the plane, let *centroid(P)* denote the centroid of the convex hull of the points in P, and let *radius(P)* denote the maximum distance between *centroid(P)* and the boundary of P. Note that *centroid(P)* and *radius(P)* are easy to compute since P is constrained to be either a disk or polygon. For positive reals d and ε, let $A(P, d, \varepsilon)$ denote the ring centered at *centroid(P)* with outer radius $d+\varepsilon+$*radius(P)*, and inner radius $d-\varepsilon-$*radius(P)* if $d-\varepsilon-$*radius(P)*>0, and zero otherwise.

Consider sensor v and suppose that the measured distances between v and sensors $u_1,..., u_m$ are known. For $i\in\{1,...,m\}$, let d_i denote the measured distance between sensor v and sensor u_i, and let ε_i denote the accuracy guarantee of d_i. Now suppose $\alpha_1,..., \alpha_m$ are m pairwise consistent mappings such that $u_i\in\Delta(\alpha_i)$ and $v\notin \Delta(\alpha_i)$ for each $i\in\{1,...,m\}$. The mappings $\alpha_1,..., \alpha_m$ are also required to be either all disk mappings or all polygon mappings. Assuming $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)\neq \emptyset$, any collection of disks or polygons in the plane containing $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ can be considered a candidate regions set of sensor v *under the assumption* that region $\alpha_i(u_i)$ contains the position of u_i for each $i\in\{1,...,m\}$. We define the set $M(\alpha_1,..., \alpha_m, v, u_1,..., u_m)$ with the goal of keeping track of the candidate regions of sensor v assuming the position of each u_i is contained in the region $\alpha_i(u_i)$. If in fact each $\alpha_i(u_i)$ does contain the position of u_i, then $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ is non-empty. However, if some $\alpha_i(u_i)$, $i\in\{1,...,m\}$, does not contain the position of u_i, then $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ may be the empty set.

Clearly $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ consists of either zero, one or more contiguous regions. If $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i) = \emptyset$, then $M(\alpha_1,..., \alpha_m, v, u_1,..., u_m)$ is defined to be $\emptyset$:

$$M(\alpha_1,..., \alpha_m, v, u_1,..., u_m) = \emptyset \tag{10}$$

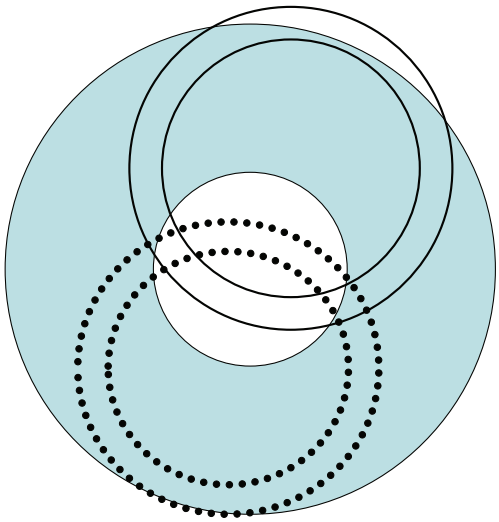
If $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ consists of one contiguous region, then let P_v be any region in the plane which contains $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$. If $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ is a point, then P_v is required to be that point. If $\alpha_1,..., \alpha_m$ are all disk (polygon) mappings, then we require that P_v be a disk (polygon). When $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ is not a point, then in keeping with the desire to compute "small" candidate regions for each sensor, P_v would ideally be the disk or polygon with the smallest diameter which contains $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$. However, the modified Sweeps algorithm does not require this to be so. In the instance of the modified Sweeps algorithm we implemented, we used a simple algorithm to determine a polygon containing $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ which has been shown to be both computationally efficient and adequate in our experimental evaluations on randomly deployed networks. Let β denote the mapping with domain $\{v\}\cup\bigcup_{i\in\{1,...,m\}} \Delta(\alpha_i)$. Let $\beta(v)$ be P_v, and for $i\in\{1,...,m\}$ and each $u\in \Delta(\alpha_i)$, define $\beta(u)= \alpha_i(u)$. Note that β is well defined since α_i, $i\in\{1,...,m\}$, are pairwise consistent, and $\beta, \alpha_1,..., \alpha_m$ are either all disk mappings or all polygon mappings. Let $M(\alpha_1,..., \alpha_m, v, u_1,..., u_m)$ be:

$$M(\alpha_1,..., \alpha_m, v, u_1,..., u_m)=\{\beta\} \tag{11}$$

Note that m being greater than or equal to three will not guarantee that $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ consists of one contiguous region. In Figure 8 below, the ring with the dotted boundary and the ring with the solid line boundary intersect in two disjoint regions, and the ring with the solid interior has a non-empty intersection with both of those regions. Hence, the three rings intersect in two disjoint contiguous regions.

If $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ consists of two or more disjoint regions, then let P_{v1} and P_{v2} be any two regions whose union contains $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$. If $\alpha_1,..., \alpha_m$ are all disk (polygon) mappings, then we require that P_{v1} and P_{v2} both be disks (polygons). In keeping with our desire to obtain "small" candidate regions for each sensor, P_{v1} and P_{v2} would ideally be disjoint regions with the smallest possible diameters whose union contains the regions of intersection. However, the modified Sweeps algorithm does not require this to be so. If $\cap_{i\in\{1,...,m\}} A(\alpha_i(u_i), d_i, \varepsilon_i)$ consists of two points, then we require that P_{v1} and P_{v2} be the two points. Let β_1 and β_2 denote mappings both with domain $\{v\}\cup\bigcup_{i\in\{1,...,m\}} \Delta(\alpha_i)$ defined as follows. Let $\beta_1(v)$ be P_{v1}, $\beta_2(v)$ be P_{v2}, and for $i\in\{1,...,m\}$ and each $u\in \Delta(\alpha_i)$, let $\beta_1(u)= \alpha_i(u)$ and β_2

Figure 8. Three rings intersecting in two disjoint regions



(u)= α_i (u). Note that β_1 and β_2 are well defined since α_i, $i \in \{1,...,m\}$, are pairwise consistent, and β_1, β_2, α_1,..., α_m are either all disk mappings or all polygon mappings. Let $M(\alpha_1,..., \alpha_m, v, u_1,..., u_m)$ be:

$$M(\alpha_1,..., \alpha_m, v, u_1,..., u_m) = \{ \beta_1, \beta_2 \} \quad (12)$$

Modified Sweeps will use M to compute candidate regions sets of sensors.

Let P_1 and P_2 be either two disks or two polygons in the plane. Let $min(P_1, P_2)$ and $max(P_1, P_2)$ denote the minimum and maximum distance, respectively, between P_1 and P_2. Note that it is computationally efficient to determine $min(P_1, P_2)$ and $max(P_1, P_2)$ since P_1 and P_2 are either both disks or both polygons. Let $I(P_1, P_2)$ denote the real line interval with left endpoint $min(P_1, P_2)$ and right endpoint $max(P_1, P_2)$.

Algorithm

In the following, we give an algorithm which computes for each sensor v a sequence of sets $S(v, i)$, $i = 1, 2, ...$ such that each $S(v,i)$ consists of a finite number of mappings and $\{\alpha(v) \mid \alpha \in S(v, i)\}$ is a candidate regions set for sensor v.

Let $[v] = v_1, v_2, v_3,..., v_n$ be an ordering of the vertices of G where v_1, v_2, v_3 are anchors. For v_i, $i \in \{1,2,3\}$, let α_i be the mapping with domain $\{v_i\}$ where $\alpha_i(v_i)$ is the known position of v_i. We require that α_i, $i \in \{1,2,3\}$, be either all disk mappings, or all polygon mappings. A point in the plane is considered a degenerate disk (polygon) with diameter zero. For $i \in \{1,2,3\}$, let $S(v_i, 1)$ be defined as:

$$S(v_i, 1) = \{ \alpha_i \}, \quad i \in \{1,2,3\} \quad (13)$$

The sets $S(v_i, 1)$, $i > 3$, are computed iteratively as follows. For v_i, $i > 3$,, let $u_1,...,u_m$ denote the vertices adjacent to v_i in G and which precedes v_i in the ordering $v_1,..., v_n$: $N(v_i) \cap \{ v_1,..., v_{i-1} \} = \{ u_1,...,u_m \}$. Define $S(v_i, 1)$ using M as :

$$S(v_i,1) = \bigcup_{\alpha_j \in S(u_j,1) \quad j\in\{1,...,m\}, \quad and\, \alpha_j \sim \alpha_k \;\forall\, j,k \in \{1,...,m\}} M(\alpha_1,\alpha_2,...,\alpha_m,v_i,u_1,u_2,...,u_m) \tag{14}$$

Roughly speaking, $S(v_i, 1)$ is the set of mappings "storing" the candidate regions of sensor v_i corresponding to each combination of candidate regions of v_i's predecessors in the ordering. It is easy to see that each $S(v, 1)$ consists of a finite number of disk (polygon) mappings, and for each sensor v, the set $\{\alpha(v) \mid \alpha \in S(v, 1)\}$ is a candidate regions set for sensor v. We call $S(v, 1)$ a *candidate mapping set* of sensor v and we call $\{\alpha(v) \mid \alpha \in S(v, 1)\}$ the candidate regions set of sensor v *obtained by the first sweep*.

Suppose for some $k \geq 1$ that $S(v, k)$, $v \in V$, have been computed, and that for each sensor v, the set $\{\alpha(v) \mid \alpha \in S(v, k)\}$ is a finite candidate regions set for sensor v. Let $u_1,..., u_n$ be any ordering of the vertices of G such that at least one vertex u_i is adjacent to some vertex u_j where $j<i$, and all vertices v where $S(v, k)$ is a singleton precede all vertices u where $S(u, k)$ is not a singleton. For each vertex u_i, let $P(u_i)$ denote the set of vertices adjacent to u_i in G and which precede u_i in the ordering $u_1,..., u_n$: $P(u_i) = N(u_i) \cap \{u_1,..., u_{i-1}\}$. Let s denote the number of vertices v for which $S(v, k)$ is a singleton. For $i \in \{1,...,s\}$, define:

$$S(u_i, k+1) = S(u_i, k) \tag{15}$$

For $i \in \{s+1,...,n\}$, if $P(u_i)=\emptyset$, then define:

$$S(u_i, k+1) = S(u_i, k) \tag{16}$$

Now suppose $P(u_i)$ is non-empty. Recall that for each $w \in P(u_i)$, $\tilde{d}_{wui}$ is the measured distance between sensors u_i and w, and ε_{wui} is the guaranteed accuracy of $\tilde{d}_{wui}$. The underlying idea behind obtaining $S(u_i, k+1)$ from $S(u_i, k)$ and $S(w, k+1)$, $w \in P(u_i)$, is as follows. By assumption, the set $\{\alpha(u_i) \mid \alpha \in S(u_i, k)\}$ is a candidate regions set of sensor u_i. Let P be any candidate region of sensor u_i from the set. Suppose that for all $w \in P(u_i)$, $\{\alpha(w) \mid \alpha \in S(w,k+1)\}$, is a candidate regions set of w. This implies that for all $w \in P(u_i)$, there is a region P_w* in $\{\alpha(w) \mid \alpha \in S(w, k+1)\}$ which contains the position of w. Suppose that for some sensor $w \in P(u_i)$ that $I(P,P')$ is disjoint from $[\tilde{d}_{wui} - \varepsilon_{wui}, \tilde{d}_{wui} + \varepsilon_{wui}]$ for all $P' \in \{\alpha(w) \mid \alpha \in S(w,k+1)\}$. This implies P cannot contain the position of sensor u_i, for if P did contain the position of sensor u_i, then I(P, P_w*) is not disjoint from $[\tilde{d}_{wui} - \varepsilon_{wui}, \tilde{d}_{wui} + \varepsilon_{wui}]$. In this case we say that P is an *identified* false candidate region of sensor u_i. To obtain the set $S(u_i, k+1)$, we remove all mappings β from $S(u_i, k)$ where $\beta(u_i)$ is an identified false candidate region of sensor u_i. Since only mappings β where $\beta(u_i)$ is a false candidate region of u_i is removed from $S(u_i, k)$ to obtain $S(u_i, k+1)$, it follows that $\{\alpha(u_i) \mid \alpha \in S(u_i, k+1)\}$ must still be a candidate regions set for sensor u_i. In the following, for notational convenience, let $w_1,...,w_m$ be the elements of $P(u_i)$, and define $S(u_i, k+1)$ as:

$$S(u_i,k+1) = \{u_{m+1}(\alpha,\alpha_1,...,\alpha_m) \mid \alpha \in S(u_i,k), \alpha_j \in S(w_j,k+1)\ \forall j \in \{1,...,m\},$$

$$\beta \sim \gamma\ \ \forall \beta,\gamma \in \{\alpha,\alpha_1,...,\alpha_m\},$$

$$I(\alpha(u_i),\alpha_j(w_j)) \bigcap [\tilde{d}_{uiwj} - \varepsilon_{uiwj}, \tilde{d}_{uiwj} + \varepsilon_{uiwj}] \neq \emptyset\ \ \forall j \in \{1,...,m\}\} \tag{17}$$

Since each $S(v, k)$, $v \in V$, consists of a finite number of elements, it follows from equation 17 that $S(v, k+1)$ must also consist of a finite number of elements. Furthermore, for each sensor v the set $\{\alpha(v) \mid \alpha \in$

$S(v, k+1)\}$ is a candidate regions set for sensor v. We call $\{\alpha(v) \mid \alpha \in S(v, k+1)\}$ the candidate regions set of sensor v obtained by the $(k+1)$th sweep, and we call $S(v, k+1)$ a *candidate mapping set* of sensor v. It is easy to see that for each sensor v, the candidate regions set of sensor v obtained by the $(k+1)$th sweep is a subset, not necessarily proper, of the candidate regions set of sensor v obtained by the kth sweep.

Suppose that for sensor v and some $k \geq 1$ that $S(v, k)$ has been computed, and $S(v, k)$ is a singleton. Let α be the mapping in $S(v, k)$. Furthermore, suppose there exist $u,w \in \Delta(\alpha)$ where distance measurements $\tilde{d}_{uv}$ and $\tilde{d}_{wv}$ are obtained, i.e. $u,w \in N(v)$, and $A(\alpha(u), \tilde{d}_{uv}, \varepsilon_{uv}) \cap A(\alpha(w), \tilde{d}_{wv}, \varepsilon_{wv})$ consists of two disjoint regions such that $\alpha(v)$ contains just one of those regions. When the previous hold, the position estimate of sensor v is taken to be the centroid of $\alpha(v)$ and the error bound is taken to be radius($\alpha(v)$).

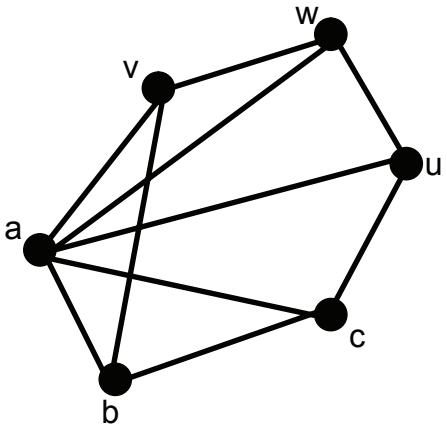
The order in which the sensors are processed in the first sweep can be a determining factor in the diameter of the candidate regions in each of the generated candidate region sets, and the number of regions in the set. This is because each sensor's candidate region sets is computed using the candidate regions sets of its neighbors preceding it in the orderings, and the distance measurements between itself and those neighbors. Intuitively, candidate regions with small diameters can be obtained for a sensor if the sensor has at least two neighbors preceding it in the ordering such that the distance measurements between the sensor and those neighbors have small accuracy guarantees and the candidate regions of those neighbors are also small. An ordering can also be chosen "on the go." For example, suppose we have sensors *a,b,c,u,v,w* where *a,b,c* are anchors. The anchors always comprise the first three sensors in the ordering. Suppose the graph of the network is as shown in Figure 9 below. So anchors *a* and *c* are neighbors of sensor *u*, anchors *a* and *b* are neighbors of sensor *v*, and anchor *a* and sensors *u* and *v* are neighbors of sensor *w*. Furthermore, suppose ε_{av}, ε_{bv}, ε_{aw} and ε_{vw} are "small" as compared to ε_{uc}. One way to order the sensors on the go is as follows. Since sensor v has distance measurements with small accuracy guarantees to two sensors with known positions, sensor v can be ordered as the first sensor to follow the anchors. If the candidate regions set computed for sensor v consists of candidate regions with "small" diameters, then sensor w has two distance measurements with small accuracy guarantees to two sensors with either exact positions or candidate regions with small diameters. Hence, let sensor w be the next sensor in the ordering following sensor v. Furthermore, if ε_{uw} is small, then there is a clear advantage in placing w in front of u in the first ordering.

In the following, we give a heuristic for choosing an ordering for generating candidate regions sets based on the factors discussed above. Given the distance measurement $\tilde{d}_{ij}$ between sensors i and j with accuracy guarantee ε_{ij}, let $p_{ij} = \varepsilon_{ij}/\tilde{d}_{ij}$. Let the anchors be the sensors which precede all others in the ordering. Suppose the first ith sensors in the ordering has already been chosen, where $i \geq 3$, and denote the ordering thus far by $v_1,....,v_i$. Let W denote the set of all sensors which are not in the ordering thus far. To choose the $i+1$th sensor, let U denote the set of all sensors in W which are adjacent to two or more sensors in $\{ v_1,....,v_i\}$. If U is empty, then let v_{i+1} be the sensor $v \in W$ such that v is a neighbor of some v_j, $j \leq i$, and p_{vvj} is less than or equal to all other p_{xy} where $x \in W$ and $y = v_k$ for some $k \leq i$. Otherwise, if U is not empty, then let v_{i+1} be a sensor $u \in U$ such that the average of p_{uvj}, $v_j \in \{ v_1,....,v_i\} \cap N(u)$, is less than or equal to the average of p_{wvj}, $v_j \in \{ v_1,....,v_i\} \cap N(w)$, for all $w \in U$.

CORRECTLY ORIENTED POSITION ESTIMATES

In this section, we introduce the concept of "correctly oriented" position estimates, and demonstrate that a set of correctly oriented position estimates can be used to deduce geometric properties of the configu-

Figure 9. a, b and c are anchors



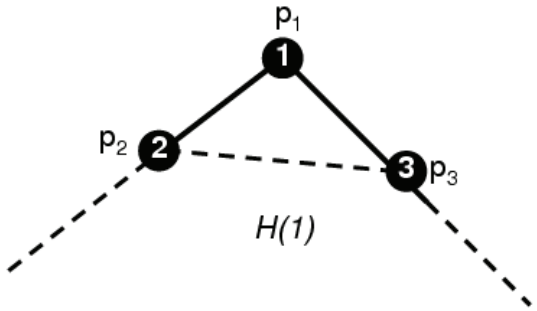
ration of the corresponding actual sensor positions. We then give a sufficient condition on the position estimates and their corresponding error bounds for the position estimates to be correctly oriented.

Consider in the plane a set of four sensors labelled 1,2,3,4 such that no three sensors are collinear. For $i \in \{1,2,3,4\}$, let p_i and q_i denote the actual and estimated position of sensor *i* respectively, and suppose the estimated positions are correctly oriented and no three of the estimated positions are collinear. Let *H(1)* denote the region of the plane which does not contain p_1 and is bounded by the line segment $l(p_2, p_3)$ and the two half-lines both with origin at p_1, and containing the points p_2 and p_3, respectively. The regions *H(2)*, *H(3)* are defined analogously. See Figure 10 for an illustration of *H(1)* which is the region of the plane bounded by the dotted lines.

Suppose the actual position of sensor 4 lies in *H(1)*. Clearly, p_4 lies in *H(1)* if and only if the line segments $l(p_1, p_4)$ and $l(p_2, p_3)$ intersect. Since q_1, q_2, q_3, q_4 are correctly oriented, it follows that $l(p_1, p_4)$ intersects $l(p_2, p_3)$ if and only if $l(q_1, q_4)$ intersects $l(q_2, q_3)$. Therefore, using the estimated positions, it can be determined if the actual position of sensor 4 lies in *H(1)*. By similar reasoning, the estimated positions can be used to determine if p_4 lies in *H(2)* and *H(3)*. If the actual position of sensor 4 does not lie in any of the *H(i)*, $i \in \{1,2,3\}$, then the actual position of one of the sensors must lie in the convex hull determined by the actual positions of the other three sensors. Hence, a set of correctly oriented sensor position estimates can be used to deduce certain geometric properties of the configuration of actual sensor positions. However, the difficulty lies in determining if a set of position estimates are correctly oriented without knowing the corresponding actual sensor positions.

In the following, we will give a sufficient condition for a set of position estimates to be correctly oriented. We begin by defining a quadrilateral given two circles in the plane. Let C_i and C_j be two circles in the plane centered at q_i and q_j respectively, and let $Q(C_i, C_j)$ denote the quadrilateral defined as follows. Let *l* denote the line segment obtained by extending the line segment $l(q_i, q_j)$ to C_i and C_j as shown in Figure 11a. So *l* is the line passing through q_i and q_j and whose endpoints are on C_i and C_j. Let T_i denote the line tangent to C_i at the endpoint of *l* on C_i, and define T_j similarly. Lines T_i and T_j are denoted in bold in Figure 11a. Let l_i denote the line passing through q_i and which is also perpendicular to *l*. Clearly, l_i intersects C_i at exactly two points, which we denote by s_{i1} and s_{i2}. Similarly, if l_j denotes the line passing through q_j which is perpendicular to *l*, then l_j must intersect C_j at exactly two points, which we denote by s_{j1} and s_{j2}. Without loss of generality, suppose s_{i1} and s_{j1} lie on the same side of line

Figure 10. The region H(1)



segment l. Let T_L denote the line passing through s_{i1} and s_{j1}, and let T_R denote the line passing through s_{i2} and s_{j2}. Lines T_L and T_R are denoted in bold in Figure 11a. It is easy to see that T_i and T_j are parallel, but that neither T_i and T_j are parallel to the two lines T_L and T_R. So let c_1 and c_2 be the two points where T_L intersects with T_i and T_j, and let c_3 and c_4 be the two points where T_R intersects with T_i and T_j. Let $Q(C_i, C_j)$ denote the convex hull of the four intersection points c_1, c_2, c_3, c_4 as shown in Figure 11b.

Consider $t > 3$ sensors $u_1,..., u_t$ with estimated positions $q_1,..., q_t$ respectively. For $i \in \{1,...,t\}$, let e_i denote the error bound of q_i, and let D_i and C_i denote respectively the disk and circle in the plane centered at q_i with radius e_i. Consider the following condition on the geometry of $C_1,..., C_t$:

Condition 1. For all distinct $i, j, k, l \in \{1,...,t\}$, the quadrilaterals $Q(C_i, C_j)$ and $Q(C_k, C_l)$ are either disjoint, or $Q(C_i, C_j)$ and $Q(C_k, C_l)$ intersect in a quadrilateral Q such that Q is disjoint from each of the circles C_i, C_j, C_k, C_l, and if e and e' are opposite edges of Q, then both edges are contained in the interior of $Q(C_i, C_j)$ or $Q(C_k, C_l)$.

Figure 12 shows the relative positions of four circles C_i, C_j, C_k and C_l in the plane which satisfy Condition 1. If Condition 1 is satisfied by C_i, $i \in \{1,...,t\}$, then the position estimates $q_1,..., q_t$ must be correctly oriented. The proof of this is straightforward and relies upon the observation that if quadrilaterals $Q(C_i, C_j)$ and $Q(C_k, C_l)$ are disjoint, then no line segment with endpoints in C_i and C_j can intersect any line segment with endpoints in C_k and C_l. If $Q(C_i, C_j)$ and $Q(C_k, C_l)$ are not disjoint, and they satisfy Condition 1, then the opposite is true, i.e. given a line segment with endpoints in C_k and C_l, and a line segment with endpoints in C_i and C_j, the two line segments must intersect. Assuming Condition 1 holds, then for all distinct $i, j, k, l \in \{1,...,t\}$, if $q'_i \in D_i$, $q'_j \in D_j$, $q'_k \in D_k$ and $q'_l \in D_l$, then $l(q'_i, q'_j)$ intersects $l(q'_k, q'_l)$ if and only if $l(q_i, q_j)$ intersects $l(q_k, q_l)$. The previous implies $q_1,..., q_t$ must be correctly oriented since for each $i \in \{1,...,t\}$, the actual position of sensor u_i is contained in D_i.

EFFICIENTLY LOCALIZABLE NETWORKS

As we have noted previously, whether a position estimate of some desired precision can be computed by modified Sweeps for a sensor depends on the geometry of the configuration of actual sensor positions, the accuracy of the distance measurements, and the graph of the network. This observation applies to any

Figure 11. $Q(C_i, C_j)$

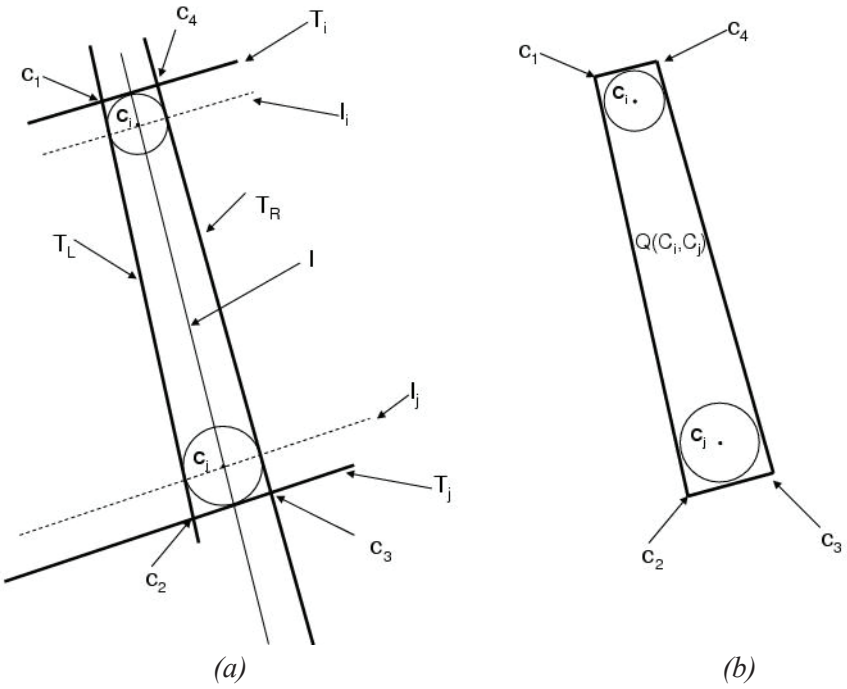
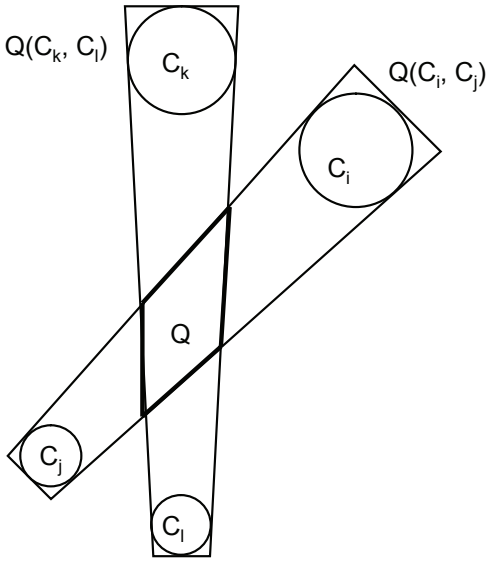


Figure 12. Four circles satisfying Condition 1



localization algorithm when distance measurements are inaccurate. For a general network, it is difficult to characterize the conditions under which a sufficiently accurate position estimate can be computed for each sensor. Furthermore, the characterization of efficiently localizable networks is far from complete even for the case of exact distance measurements. Hence, we will give the graphical characterization of some networks whose positions can be determined exactly and efficiently for the limit case of exact distance measurements. For such networks, it follows by a simple continuity argument that there are sufficiently small guaranteed accuracies $\varepsilon_{ij} > 0$ for each measured distance $\widetilde{d}_{ij}$ such that modified Sweeps can be used to compute a position estimate for each sensor which has a "small" error bound. A key component of the modified Sweeps algorithm is its sequential nature whereby sensors are processed in some order. As will be illustrated below, the notion of processing sensors in some order enables the graphical characterization of networks for which modified Sweeps can be used to *efficiently* compute position estimates in the limit as $\varepsilon_{ij} \to 0$ for each measured distance $\widetilde{d}_{ij}$.

A graph is said to have a *trilateration ordering* if its vertices can be ordered as $u_1, u_2, u_3,..., u_n$ so that u_1, u_2, u_3 induce a complete subgraph, and each u_i, $i>3$, is adjacent to at least three vertices u_j where $j<i$. In Anderson et al. (2007) it was shown that localizable networks whose graphs have trilateration orderings can be efficiently localized, i.e. in a number of operations that is polynomial in the number of sensors. It is straightforward to show that such networks can also be efficiently localized by modified Sweeps in one sweep. In the following, we give a graphical characterization of a class of networks which can be efficiently localized but whose graphs do not necessarily have trilateration orderings. Suppose the network N has $n>4$ sensors and that the vertices of its graph G can be ordered as $v_1, v_2, v_3,..., v_n$ so that v_1, v_2, v_3 correspond to anchors, and that each v_i, $i>3$, is adjacent to vertices v_{i-1} and v_{i-2}. Furthermore, for each v_i, where either $i=n$ or $3<i<n$ and i is odd, v_i is also adjacent to some v_j where $j \neq i-3$ and j is either odd or $j \leq 3$. We call such an ordering an *augmented triangulation* ordering. If G has an augmented triangulation ordering, then G must be globally rigid. This implies any network with a generic multi-point, three anchors, and whose graph has an augmented triangulation ordering must be localizable. Furthermore, G can have an augmented triangulation ordering without possessing any trilateration orderings. To see this note that if G has a trilateration ordering then G must have at least $3(n-3)+3$ edges since each vertex following the first three vertices in the ordering must be adjacent to at least three vertices preceding it. Suppose G has an augmented triangulation ordering where each v_i, where $i<n$ and i is even, is adjacent to *exactly* two vertices preceding it, and each v_j where j is either equal to n or j is odd, is adjacent to *exactly* three vertices preceding it. In this case G has $(5(n-3)/2)+3$ edges when n is odd and $(5(n-4)/2)+6$ when n is even. In either case, G would have less than the minimum number of edges required for a trilateration ordering.

Suppose an augmented triangulation ordering $v_1, v_2, v_3,..., v_n$ of G is chosen to compute the first sweep in modified Sweeps. In other words, $v_1, v_2, v_3,..., v_n$ is the ordering used in computing $S(v,1)$, $v \in V$. In this case, each $S(v,1)$, $v \in V$, contains at most two mappings, and $S(v_i,1)$ where i is odd must be a singleton. This implies N can be efficiently localized by modified Sweeps since the computational complexity of modified Sweeps is entirely dependent on the size of the sets generated during each sweep. Since an augmented triangulation ordering is not necessarily a trilateration ordering, the previous also implies that N can be efficiently localized by modified Sweeps even if its graph G does not have any trilateration orderings.

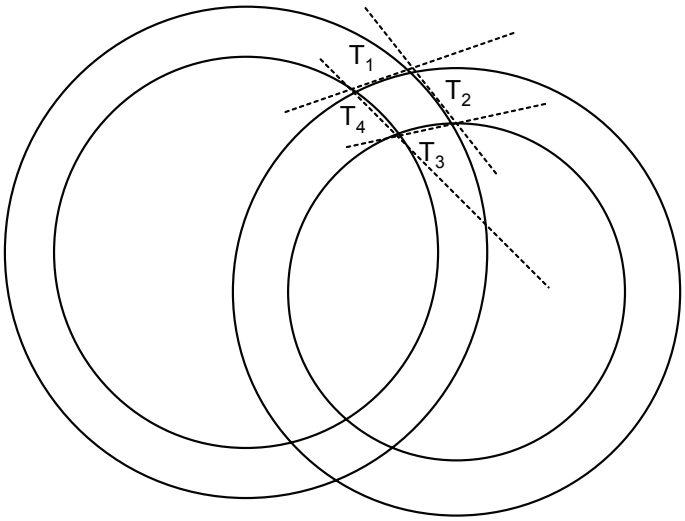
EXPERIMENTAL EVALUATIONS

In our experimental evaluations, we used Matlab to first generate a random network of 100 nodes using three input parameters R, m and η, where R is the sensing range, m is the number of anchors, and η is the noise factor. Sensor positions are randomly generated from the distribution that is uniformly distributed on the 1-by-1 two dimensional space, and m of those sensors are randomly chosen to be anchors. For each pair of sensors within sensing range R a noisy distance measurement is generated using the input parameter noise factor η. More specifically, η is specified to be between zero and one, and for each pair of sensors within sensing range, a distance measurement $\tilde{d}$ is generated such that the actual inter-sensor distance d is within $\eta \cdot \tilde{d}$ of the generated distance measurement, i.e. $|\ \tilde{d} - d\ | \leq \eta \cdot \tilde{d}$. The limit case of exact distance measurements corresponds to when noise factor η is zero: $|\ \tilde{d} - d\ | = 0$. In other words, the guaranteed accuracy of each generated distance measurement is $\eta \cdot 100$ percent of the distance measurement. Roughly speaking this corresponds to the notion that the distance measurement between two sensors become less accurate as the distance between the two sensors increase. For reasons we will specify below, we will also consider a class of networks such that distance measurements between particular pairs of sensors within sensing range are *not* generated.

For ease of implementation, we used convex polygons, as opposed to just general polygons, to approximate ring intersection regions since convex polygons are particularly easy to manipulate. We compute the convex polygon approximation of ring intersections by means of a simple algorithm using tangent lines and convex hulls. More specifically, consider the intersection region of two rings that intersect in two disjoint regions as shown in Figure 13. Each intersection region is comprised of four arcs which we denote by a_1, a_2, a_3, and a_4. Without loss of generality, suppose a_i is disjoint from a_{i+2} for $i=1,2$. Note that each arc lies on either the inner or outer circle of one of the two rings. If arc a_i lies on the outer circle of one of the rings, then let T_i denote a line that is tangent to the arc at roughly the midpoint of the arc. If arc a_i lies on the inner circle of one of the rings, then let T_i denote the line passing through the endpoints of a_i. See Figure 13 below for an illustration. For $i=1,2,3$, let t_i denote the point where T_i and T_{i+1} intersects, and let t_4 denote the point where T_4 and T_1 intersects. We obtain a convex polygon which contains one of the contiguous regions of the ring intersection by taking the convex hull of t_1, t_2, t_3, and t_4.

We evaluated modified Sweeps on networks whose graphs have "augmented bilateration" orderings. A graph is said to have a *bilateration ordering* if its vertices can be ordered so that v_1, v_2, v_3 induce a complete subgraph, and each v_i, $i>3$, is adjacent to at least two vertices v_j where $j<i$. A network's graph is said to have an *augmented* bilateration ordering if it has a bilateration ordering $v_1,\ldots, v_n$ where $v_1,\ldots, v_m$ are the $m \geq 3$ anchors and if v_i, $i>m$, is only adjacent to two vertices v_j where $j<i$, then v_i must also be adjacent to v_{i+1} where v_{i+1} is adjacent to at least two vertices v_j where $j<i$ and at least one of which is not adjacent to v_i. If a graph has at least one augmented bilateration ordering, but no such ordering is also a trilateration ordering, then we say that the graph's augmented bilateration orderings are *untrilaterable*. In our evaluations, we consider both networks whose graphs have augmented bilateration orderings, and untrilaterable augmented bilateration orderings. The ordering chosen for the first sweep is an augmented bilateration ordering of the network, and we use a simple connectivity based procedure for obtaining such an ordering. We begin by labelling the $m \geq 3$ anchors to be $v_1,\ldots, v_m$. Assuming the ordering determined thus far is $v_1,\ldots, v_i$ where $i \geq m$, the $i+1$th sensor in the ordering is determined as follows. If v is any sensor that is not in the ordering and v is the neighbor of three or more sensors v_j where $j< i+1$, then we let v be the $i+1$th sensor in the ordering. Otherwise, if there is a pair of sensors u and v

Figure 13. Convex polygon approximation of ring intersection



such that *u* and *v* are neighbors and each of u and v is adjacent to two sensors v_j where j <i+1, and v is adjacent to at least one sensor *v_j*, *j* < *i*+1, which is not adjacent to *v_i*, then we let *u* and *v* be the *i+1*th and *i+2*th sensors in the ordering. Let *P(v,1)*, *v*∈ *V*, denote the set of sensors adjacent to *v* and preceding *v* in the ordering of the first sweep. We implemented a slightly altered version of modified Sweeps in that if *S(v,1)* is a singleton for some *v*∈ *V*, then each *S(u,1)*, *u* ∈ *P*(*v*,1), is refined using *S*(*v*,1) before proceeding with the first sweep. More specifically, if α is the mapping in the singleton *S(v,1)*, then for each *u* ∈ *P*(*v*,1), *S(u,1)* is refined using *S(v,1)* by removing all mappings β from *S*(*u*,1) where $\beta(u) \neq \alpha(u)$. After *S(u,1)*, *u* ∈ *P*(*v*,1), have each been refined by *S*(*v*,1), each $\{\gamma(u) \mid \gamma \in S(u,1)\}$ remains a candidate region set for sensor *u*. Using this slightly altered version of modified Sweeps, we sweep through the network only once, i.e. by only computing *S*(*v*,1), *v* ∈ *V*. Theoretically, similar results can be obtained by sweeping through the network multiple times using the original modified Sweeps algorithm. However, for evaluation purposes, we have found the altered implementation to be more computationally efficient and the computed error bounds on the position estimates were reasonable.

We now discuss two scenarios in detail. First, networks of 100 sensors are generated whose graphs have an augmented bilateration ordering for which the input parameters are as follows: sensing range *R*=0.2, number of anchors *m*=15, and noise factor η = 0.08. We averaged the results of modified Sweeps over 100 randomly deployed instances of such networks with the aforementioned parameters. Since there are 15 anchors, there are a total of 85 non-anchor sensors. We found that on average 47 of the 85 non-anchor sensors are assigned a position estimate, and the average error bound was 0.0303, which is less than 1/6 of the sensing range. Hence, on average, the actual position of a sensor can be guaranteed to be within 0.0303 of its estimated position. The average of the distance between each position estimate and the actual sensor position, did not exceed 0.02, and was in general far less than the average error bound. As expected, when the noise factor is decreased to 0.05, more sensors on average are assigned position estimates, and the corresponding error bounds are also lower.

We next considered networks of 100 sensors whose graphs have only untrilaterable augmented bilateration orderings. The input parameters used to generate the networks are identical as those used to generate networks in the previous scenario: sensing range R=0.2, number of anchors m=15, and a noise factor of $\eta = 0.08$. However, after the network is generated, the sensors which are adjacent to more than two anchors are identified, and for each such sensor, distance measurements are generated only between the sensor and two of the anchors the sensor is adjacent to. If the graph of the resulting network contains an augmented bilateration ordering, then the ordering can be guaranteed to be untrilaterable. The results are averaged over 100 instances of randomly deployed networks with the aforementioned parameters and whose graphs have only untrilaterable augmented bilateration orderings. For these networks, less sensors on total are assigned position estimates by Sweeps as compared to the previous scenario. More specifically, 40 of the sensors, as opposed to the previous 47, are assigned position estimates. The average error bound of the position estimates was 0.0366, which is just slightly higher than the previous scenario.

Generally speaking, we found that as the sensing range or the number of anchors of the network increased, the number of sensors for which a position estimate was computed increased. To illustrate this trend, we considered networks of 100 sensors whose graphs have an augmented bilateration ordering for which the input parameters are: sensing range R=0.2, and noise factor $\eta = 0.05$. By varying the number of anchors, we see what effect this has on the number of sensors for which a position estimate was computed and on the average error bound of the position estimates. We found the most meaningful results to occur when 10-25 anchors were used. There is a significant amount of variation in the case when there are less than 10 anchors, due to the fact that the sample size is too small. Twenty simulations were run for each of the following cases: 10 anchors, 15 anchors, 20 anchors and 25 anchors, and averaged in the Monte Carlo method. In Figure 14, we see an increase in the number of sensors for which a position estimate is computed as the number of anchors increases from 10 to 25. In Figure 15, we see that the mean error decreases from 0.0054 to 0.004 as the number of anchors increases from 10 to 25, which is what we expect.

CONCLUSION

In this chapter we presented a sequential algorithm called modified Sweeps for estimating sensor positions of a network when only inaccurate distance measurements and some anchor positions are available. If a position estimate p is computed for a sensor, then an error bound $e(p)$ is also computed so that the actual position of the sensor can be guaranteed to be within distance $e(p)$ of the estimated sensor position. We define the concept of correctly oriented position estimates, and show by example that a set of correctly oriented estimated sensor positions can be used to deduce certain geometric properties of the configuration of actual sensor positions. We also give a sufficient condition on the estimated positions and the corresponding error bounds in order to guarantee that the estimated positions are correctly oriented. We show by experimental evaluations that for randomly deployed networks whose graphs have an augmented bilateration ordering, modified Sweeps is able to assign positions to more than half of the non-anchor sensors, and furthermore, the error bounds of the estimated positions are small as compared to the sensing range. As we noted in the experimental evaluations section, the orderings in which sensors are processed are determined by a simple method using only connectivity. For future work, we will evaluate the algorithm using different techniques for determining orderings. In particular, we are

Figure 14. Anchors vs. number of sensors with estimated positions

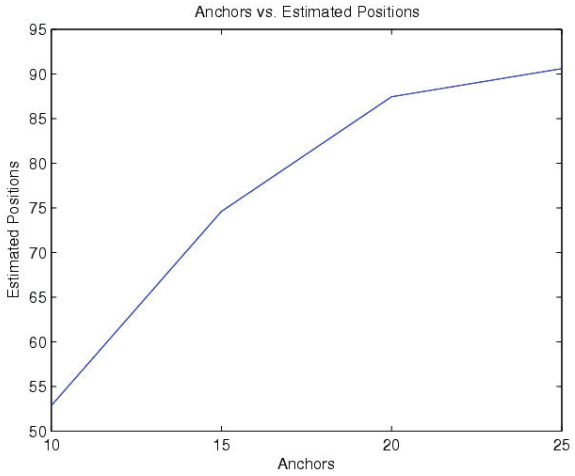
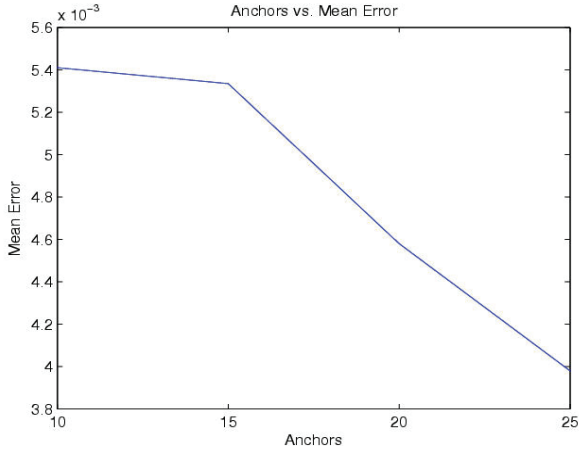


Figure 15. Anchors vs. mean error bound



interested in those methods that take into account the guaranteed accuracies of distance measurements. We also aim to extend the proposed algorithm to a decentralized setting and carry out more extensive experimental evaluations using actual sensor data.

REFERENCES

Anderson, B. D. O., Belhumeur, P. N., Eren, T., Goldenberg, D. K., Morse, A. S., Whiteley, W., & Yang, Y. R. (2007). Graphical properties of easily localizable sensor networks. *Wireless Networks*. Thre Netherlands: Springer.

Fang, J., Cao, M., Morse, A. S., & Anderson, B. D. O. (2006). Localization of sensor networks using sweeps. In *Proceedings of CDC*, (pp. 4645–4650), San Diego, CA.

Fang, J., Cao, M., Morse, A. S., & Anderson. B. D. O. (2008). Sequential Localization of Sensor Networks. *SIAM J. on Control and Optimization*, to appear.

Goldenberg, D. K., Bihler, P., Cao, M., Fang, J., Anderson, B. D. O., Morse, A. S., & Yang, Y. R. (2006). Localization in sparse networks using sweeps. *Proceedings of Mobicom* (pp. 110–121).

Hendrickson, B. (1992). Conditions for unique graph realizations. *SIAM J. Comput.*, *21*(1), 65–84.

Jackson, B., & Jordan, T. (2005). Connected rigidity matroids and unique realizations of graphs. *J. Comb. Theory Ser. B*, *94*(1), 1–29.

Jacobs, D., & Hendrickson, B. (1997). An algorithm for two-dimensional rigidity percolation: the pebble game. *J. Comput. Phys.*, *137*(2), 346–365.

Moore, D., Leonard, J., Rus, D., & Teller, S. (2004). Robust distributed network localization with noisy range measurements. In *Proc. 2nd ACM SenSys*, (pp. 50–61), Baltimore, MD.

Priyantha, N. B., Demaine, E., & Teller S. (2003) Poster abstract: anchor-free distributed localization in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, (pp. 340–341,) New York, NY, USA. ACM.