This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

Underwater Localization in Sparse 3D Acoustic Sensor Networks

Wei Cheng[1,2], Amin Y. Teymorian, Liran Ma, Xiuzhen Cheng
[1]Department of Computer Science
The George Washington University
Washington, DC, USA
Email: {wcheng,amin,lrma,cheng}@gwu.edu

Xicheng Lu, Zexin Lu
[2]School of Computer Science
National University of Defense Technology
Changsha, Hunan, PRC
Email: {xclu,lzx}@nudt.edu.cn

***Abstract*—We study the localization problem in sparse 3D underwater sensor networks. Considering the fact that depth information is typically available for underwater sensors, we transform the 3D underwater positioning problem into its two-dimensional counterpart via a projection technique and prove that a non-degenerative projection preserves network localizability. We further prove that given a network and a constant k, all of the geometric k-lateration localization methods are equivalent. Based on these results, we design a purely distributed localization framework termed USP. This framework can be applied with any ranging method proposed for 2D terrestrial sensor networks. Through theoretical analysis and extensive simulation, we show that USP preserves the localizability of the original 3D network via a simple projection and improves localization capabilities when bilateration is employed. USP has low storage and computation requirements, and predictable and balanced communication overhead.**

***Index Terms*—3D underwater localization, acoustic sensor networks, network localization problem, localizability.**

I. Introduction

Underwater sensor networks (USNs) consist of a variable number of sensors designed to collaboratively monitor an oceanic environment. To achieve this objective, sensors self-organize into an autonomous network that can adapt to the characteristics of a given underwater area. The main motivations for USNs are their relative ease of deployment and lower costs, as they eliminate the need for underwater cabling and do not interfere with shipping activities.

USNs' unique properties have necessitated an innovative re-examination of problems related to localization [1]. Indeed, propagation delays, motion-induced Doppler shift, limited bandwidth, and multipath interference render many previously proposed solutions inaccurate or infeasible [2]. For example, commonly employed RSS-based localization techniques provide ambiguous results in underwater environments [3]. In addition, even the well-established Global Positioning System (GPS) does not work well underwater [4]. Furthermore, three-dimensional localization becomes even more challenging (e.g., being in range of a sufficient number of anchor nodes) due to the economically-driven sparseness of USN deployments [5]. These properties make the underwater localization a non-trivial task for which relatively very few options are available.

Our research is motivated by the following observations: (1) underwater sensors typically have depth information available through various techniques [6]; (2) it is not always feasible to deploy anchor nodes at the sea floor, especially for deep ocean environments; and (3) localization in terrestrial sensor networks has been extensively studied and many elegant ideas have been proposed. We seek to determine whether or not a localization framework for underwater sensor networks can be designed such that by employing sensor depth information, only anchor nodes on the sea surface (a horizontal plane) are required and existing 2D localization approaches can be easily adopted.

In this paper, we formally identify the conditions that make it possible to transform 3D underwater localization to 2D. In particular, we prove that each node preserves its localizability in the plane on which it is projected if the projection is non-degenerative. Under this condition, a node is localizable in the projection plane if and only if it is localizable in the original 3D underwater network. We then prove that a node can be localized by a geometric k-lateration localization method if and only if it can be localized by another k-lateration localization method.

We also design and extensively analyze a purely distributed underwater sensor positioning framework termed USP that employs our provably effective projection technique. The geometric k-lateration localization equivalence guarantees that USP preserves the capabilities of 2D localization methods and improves localization capabilities over existing 3D techniques. USP has low storage and computation requirements, incurs predictable and balanced communication overhead, and is robust to environment-induced errors.

The remaining portion of this paper is organized as follows. Section II provides a brief overview of related research. In Section III, we conduct a network localizability study that forms the foundation of our research. A detailed elaboration of the design of USP appears in Section IV, and an extensive analysis of USP's performance is provided in Section V. We conclude in Section VI with a discussion of future research directions.

II. Related Work

In this section, we briefly overview localization techniques proposed for USNs. For a more detailed literature survey, we refer the interested readers to [3], and the references therein.

Motivated by terrestrial GPS, "underwater GPS" schemes such as GIB (GPS Intelligent Buoys) [7] and PARADIGM [8] have been proposed. While GIB relies on a centralized server to compute location information for sensors, the autonomous underwater vehicles in PARADIGM are able to compute their locations on-board.

Additionally, Hahn *et al*. [9] propose a ping-pong style scheme to measure the round-trip delay for range estimation. This scheme requires a sensor to interrogate multiple surface buoys, an action that contributes to network throughput degradation because localization information and application communication share the same underwater channel. This contrasts with the distributed nature of USP, and its efficient and balanced communication overhead.

A silent positioning scheme is proposed in [1] where sensors discover their locations by passively listening to beacon messages. Despite providing receiver privacy, the assumption that four anchor nodes are able to cover the entire area of interest may not always hold. In USP, three dimensional localization is provided without the existence of a forth beacon node on the sea floor.

An area-based range-free underwater positioning scheme termed ALS [10] relies on anchor nodes that adjust power levels to partition a two-dimensional region into subareas, with each anchor node having its own non-overlapping partition. A sensor receives its position estimate from a central server after providing all of the areas (one for each anchor node) that it resides in. On the other hand, USP is a 3D localization scheme with finer position granularity than ALS (i.e., it computes the position of a node within a coordinate system as opposed to a position within a subarea).

When there is no direct communication between anchor nodes and sensors, network connectivity can be exploited for range estimation. In [11], three localization schemes (DV-hop, DV-distance, and Euclidean) based on network connectivity are proposed. Although Euclidean is shown to perform the best in anisotropic topologies, there is an expense of larger computation and communication overheads. As an extension to the Euclidean method, Zhou *et al*. [12] provide support for 3D USNs. This method relies on a relatively larger number of anchor nodes, which results in a higher deployment cost. Zhang *et al*. [13] proposes UR-PLACE, a protocol for underwater robot self-positioning that employs beacon flooding. Both the extensive local communication in [12] and the global flooding in UR-PLACE are bandwidth intensive and unavoidably degrade the throughput in USNs.

USP is a purely distributed localization framework that employs a provably effective projection technique, and can work with any ranging method. Additionally, it is especially tailored to the sparse deployments of USNs, and requires as few as three beacon nodes to bootstrap the scheme.

Before elaborating the design of USP, we study the issue of network localizability and develop the theoretical foundation on which USP is built.

III. Network Localizability Study

Since it may not be practical to place anchors on the sea floor in 3D USNs, they are usually deployed on the surface as buoys. However, a 3D position cannot be resolved if all of the reference nodes, no matter how many exist, reside on a single plane. What we need is a method to differentiate the real position of a sensor from the position of its image relative to the surface place. This problem may be solved if we employ the depth information that is typically available to underwater sensors. Specifically, given the depth of underwater sensors, we can map the positions of the anchor nodes to the plane containing the to-be-localized node. This mapping effectively transforms the problem of 3D underwater localization into a 2D positioning problem such that many of the elegant localization techniques for 2D terrestrial sensor networks become applicable.

In this section, we investigate whether or not a non-degenerative projection preserves *node localizability*. We prove that a node is localizable in the projection plane if and only if it is localizable in the original 3D network. We also prove that all of the geometric k-lateration localization methods are equivalent, which guarantees that our proposed USP preserves the capabilities of the 2D localization methods.

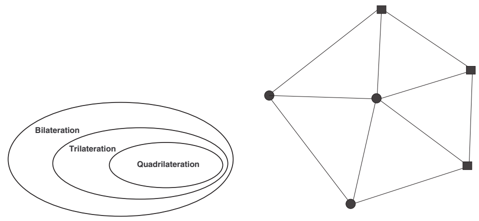
We begin with some basic definitions related to the network localization problem.

A. Background Information

The *network localization problem* is to determine a unique position for each node in a network given the positions of some nodes (termed "beacons" or "anchors") and the knowledge of some inter-node distances, which can be the real physical distances or some virtual distances such as the number of hops. A node is ***localizable*** if its location can be uniquely determined; otherwise, the node is ***unlocalizable***. Although a node is unlocalizable, it may still be possible to compute several candidate positions for it. These types of nodes are ***finitely localizable***. In most of the localization methods, a to-be-localized node localizes itself based on some ***reference nodes***. Here, the reference nodes are those that have obtained their location information before the to-be-localized node. The beacons are the ***initial reference nodes***. Let $G(V, E)$ be the graph representing this network where V is the set of nodes including the anchors and an edge $(u, v) \in E$ if and only if $\{u, v\} \subseteq V$ and the distance between u and v is available.

Several well known geometric localization methods such as bilateration, trilateration, and quadrilateration can be employed to localize the to-be-localized nodes. These methods differ in the number of reference nodes, denoted by k, a to-be-localized node must "reach" before it can start to compute its position. For simplicity, we use k-lateration to denote these localization methods, where $k = 2, 3, 4$ refers to bilateration, trilateration, and quadrilateration, respectively. The relationship among quadrilateration, trilateration, and bilateration localization methods is illustrated in Fig. 1(a).

In general, there should exist at least $d + 1$ anchors to uniquely localize a network in d-dimensional space. But as

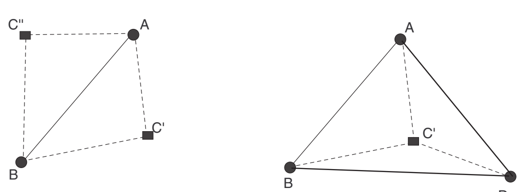


(a) (b)

Fig. 1. (a) The relationship between k-lateration localization methods. (b) A wheel network where circles representing the anchors, and rectangles representing the to-be-localized nodes.

pointed out in [14], given a 2D network as shown in Fig. 1(b), bilateration localization methods can localize the network, but trilateration or quadrilateration localization methods can not. The reason is that the number of reference nodes for a to-be-localized node to start the localization process is not large enough for trilateration or quadrilateration, although the network can be localized.

Although a bilateration localization method may not uniquely localize a node at the beginning, it can perform reductions as long as new reference nodes become available. For example as shown in Fig. 2 (a), to-be-localized node C has range information to two anchors A and B. Because it is essential to be adjacent to at least three anchors for computing a unique position in 2D, C can derive two candidate positions C' and C'', with each satisfying the distance constraints established by A and B. However, only one of them is node C's real position, thus making C a finitely localizable node with candidate position set $\{C', C''\}$. A ***reduction*** is a procedure that reduces the number of candidate positions. As shown in Fig.2 (b), node C can remove C'' from its candidate position set after the reference node D is introduced because C'' does not satisfy the distance constraint imposed by D.



(a) (b)

Fig. 2. Node C has two candidate positions in (a). C becomes localizable in (b) after reference node D is introduced.

Note that some nodes in the network can not be uniquely localized, but can be finitely localized. The ability to localize more nodes with a bilateration method does not come for free. In fact, for 2D localization, the computational complexity of a bilateration method is exponential [14], while trilateration is polynomial in the number of vertices.

B. Localizability Preservation Study

In this subsection, we prove that a non-degenerative projection preserves the localizability of the network.

Definition 3.1: Given a plane F in 3D , a ***projection*** is a function $P_F : R^3 \rightarrow R^3$, which projects a node v in the 3D space to a node v^F in the plane F, i.e. $P_F(v) = v^F$.
Note that P_F is a Euclidean transformation.

Definition 3.2: Given a 3D graph $G(V, E)$ and a plane F, the ***projection graph*** $G_F(V_F, E_F)$ is produced by the projection P_F, where $V_F = \{v^F | v \in V\}$ and $E_F = \{(v_i^F, v_j^F) | (v_i, v_j) \in E, i \neq j\}$.

Lemma 3.1: If P_F is non-degenerative, then $(v_i, v_j) \in E$ if and only if $(v_i^F, v_j^F) \in E_F$, where $i \neq j$.

Proof: P_F is a bijective function when there is no degeneration. Therefore the claim holds true according to Def. 3.1 and Def. 3.2. ■

Definition 3.3: Given a plane F and a node v, the ***relative distance*** D_v^F represents the distance from v to F, i.e., $D_v^F = v - v^F$. Note that D_v^F is a vector.

Definition 3.4: Given a plane F, the 3D coordinate system C_F derived from F is called a ***relative projection coordinate system***.

Theorem 3.1: Assume the projection P_F is non-degenerative. Given a 3D graph $G(V, E)$ and a plane F where D_v^F is known for $\forall v \in V$, then v is localizable in G if and only if v^F is localizable in G_F.

Proof: Since P_F is non-degenerative, it is bijective. For $\forall v \in V$, let (x_v^0, y_v^0, z_v^0) be the coordinates of v in 3D, and (x_v^F, y_v^F, z_v^F) be the coordinates of v^F in the relative projection coordinate system C_F. Since P_F is bijective, P_F^{-1} exists and it is bijective too. Therefore the Euclidean transformation between the two coordinate systems is bijective. Thus the mapping between (x_v^0, y_v^0, z_v^0) and (x_v^F, y_v^F, z_v^F) is unique. According to Lemma 3.1, P_F preserves the connectivity of G. Therefore v is localizable in G if and only if v^F is localizable in G_F. ■

Corollary 3.1: If P_F is bijective, then $G(V, E)$ is uniquely (finitely) localizable if and only if $G_F(V_F, E_F)$ is uniquely (finitely) localizable in the projection plane F.

Proof: Claims hold from Theorem 3.1. ■

Note that Theorem 3.1 and Corollary 3.1 indicate that a non-degenerative projection preserves the localizability of a network G. This observation motivates the design of our distributed underwater sensor positioning (USP) framework in the next section.

C. Localizability Equivalence Study

We have given an example that shows bilateration can localize a larger set of nodes than trilateration. In general, $(k-1)$-lateration is always better in localizability than k-lateration. We now investigate whether or not all the k-lateration methods are equivalent in localizability. A positive answer guarantees the generality of our proposed framework.

Intuitively, different k-lateration localization methods may localize different sets of localizable nodes. This could be caused by the fact that two k-lateration localization methods may differ in the way of using the initial reference nodes. In addition, a centralized k-lateration method may localize more number of nodes compared to its distributed/localized implementation. However, this intuition is not true for localizability.

We will next prove that a to-be-localized node can be localized by one k-lateration localization method if and only if it can be localized by another given the same initial reference nodes. This means that all of the k-lateration based localization methods are equivalent.

Let $G(V, E)$ be a d-dimension graph where $d > 1$. Our proof is based on the definitions that appear below.

Definition 3.5: A ***t-seed subgraph*** $G_{ts}(V_{ts}, E_{ts})$ of G is a graph induced by t vertices v_1, $v_2, \cdots$, v_t in G. In other words, $G_{ts}(V_{ts}, E_{ts})$ is a t-seed subgraph of G if $V_{ts} = \{v_1, v_2, \cdots, v_t\} \subseteq V$, and $(v_i, v_j) \in E_{ts}$ if and only if $(v_i, v_j) \in E$, where $i, j = 1, 2, \cdots, t$, and $i \neq j$.

For a to-be-localized network, all the initial reference nodes induce the t-seed subgraph, where t is the number of initial reference nodes. Note that, $t > d$ is necessary to uniquely localize a network in d-dimensional space.

Definition 3.6: A ***k-lateration extension*** of a subgraph $G_0(V_0, E_0)$ of G produces a new subgraph $G_1(V_1, E_1)$ of G where G_1 is an induced graph of $V_1 = V_0 \cup \{v | v \in V \setminus V_0, \exists v_1, v_2, \cdots, v_k \in V_0$, s.t. $(v, v_i) \in E$ for $i = 1, 2, \cdots, k$ $\}$.

Definition 3.7: Given a t-seed subgraph G_{ts} of G, a ***k-lateration extension subgraph*** $G_m(V_m, E_m)$, $m = 1, 2, \cdots$, is produced by k-lateration extensions starting from $G_0 = G_{ts}$.

Next, we will prove the main property of the k-lateration extension subgraph $G_m(V_m, E_m)$. We need the definition of k-credit node, which is introduced in [15].

Definition 3.8: Given a node T, if a node S has k vertex-disjoint paths to T, S is called a ***k-credit node***.

In [15], we proved the following theorem.

Theorem 3.2: A set of k vertex-disjoint paths from S to T can be found for a $(k-1)$-credit node S if there exists a k-credit node P and a path S_P between S and P such that S_P vertex-disjoints all the known $(k-1)$ paths from S to T and the k paths from P to T.

Lemma 3.2: A node v is a vertex in a k-lateration extension subgraph V_m, $m = 1, 2, \cdots$, if and only if v has at least k vertex disjoint paths to k distinct nodes in V_{ts}, and each of the nodes in v's paths also has at least k vertex disjoint paths to k distinct nodes in V_{ts}.

Proof: If $v \in V_1$, then the claim holds trivially according to Definition 3.6. Now we assume that when $v \in V_j$ the claim is true for $\forall j = 1, 2, \cdots, i$.

Now consider the case when $v \in V_{i+1} \setminus V_i$. According to the definition of k-lateration extension, there exist at least k different nodes $v_1, v_2, \cdots, v_j, \cdots, v_k \in V_i$, such that these k nodes joint v as shown in Fig 3(a). Based on the assumption, each of these k nodes has at least k vertex disjoint paths to k distinct nodes that are in V_{ts}, and all of the nodes in its paths also have at least k vertex disjoint paths to k vertex distinct nodes that are in V_{ts}.

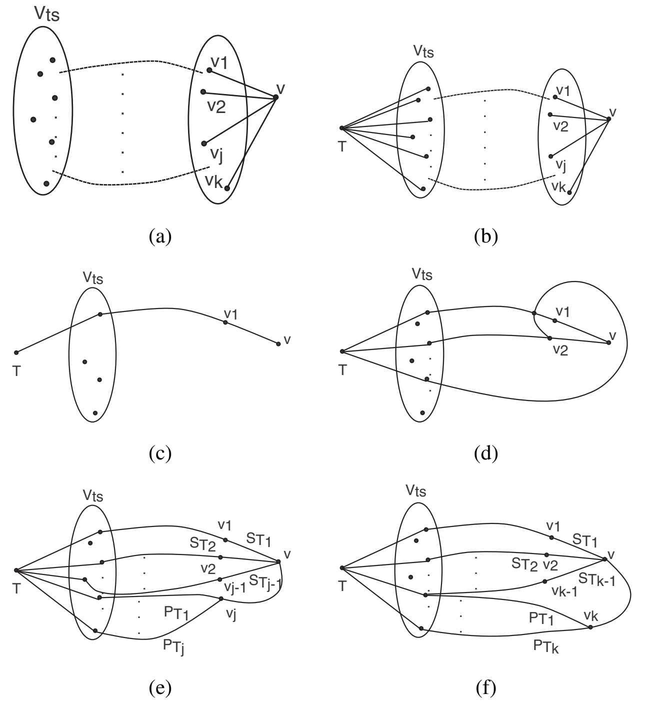


Fig. 3. The progress of constructing k vertex disjoint paths.

It is clear that the path/edge between v and any v_j, $1 \leq j \leq k$, vertex disjoints all the other already-known paths according to the Def. 3.7. In the following, we will construct k vertex disjunct paths from v to k distinct nodes in V_{ts}.

We assume that there is a virtual sink node T connecting all the initial reference nodes as shown in the Fig 3(b). We will construct k vertex-disjoint paths from v to T step by step as following.

- Pick one of v_1's paths, which does not pass any of v_2, $\cdots$, v_j, $\cdots$, v_k. There must exist such a path because it is impossible for k vertex-disjoint paths sharing $k-1$ nodes for v_1. The concatenation of this path and the edge between v and v_1 forms the first path for v, as shown in Fig. 3(c). And this path does not pass any of v_2, $\cdots$, v_j, $\cdots$, v_k.
- Pick two of v_2's paths, which do not pass any of $v_3, \cdots, v_j$, $\cdots, v_k$, as shown in Fig. 3(d). According to Theorem 3.2, there exist two vertex-disjoint paths between v and T .
- $\cdots$
- Pick j of v_j's paths, which do not pass any of $v_{j+1}, \cdots$, v_k, as shown in Fig. 3(e). Based on Theorem 3.2, there exist j vertex-disjoint paths between v and T.
- $\cdots$
- Pick k paths from v_k's paths, as shown in Fig. 3(f). Based on Theorem 3.2, there exist k vertex-disjoint paths between v and T.

Therefore when $v \in V_{i+1}$, the claim is true. The opposite direction is true according to Def. 3.7. ■

Definition 3.9: Given a t-seed subgraph G_{ts} of G, a ***maxi-***

mum k-lateration extension subgraph $G_M(V_M, E_M)$ of G is a k-lateration extension subgraph such that for $\forall v \in V \setminus V_M$, $|N(v) \cap V_M| < k$, where $N(v) = \{v_i | (v, v_i) \in E\}$, is the neighbor set of v in G.

Note that a maximum k-lateration extension graph is obtained when the k-lateration localization method terminates.

In the following we prove that given a t-seed subgraph G_{ts} of G, the maximum k-lateration extension subgraphs $G_M(V_M, E_M)$ computed from all the k-lateration localization methods are the same.

Theorem 3.3: Let $G_M(V_M, E_M)$ be any maximum k-lateration extension graph of the $G(V, E)$, which is derived from the same t-seed subgraph G_{ts}, then the V_M is unique.

Proof: Assume there are two V_M's, V_{M_1} and V_{M_2}, which means that G has two different maximum k-lateration extension graphs from G_{ts}. For $\forall v \in V_{M_1}$, it can be concluded that $v \in V_{M_2}$ according to Lemma 3.2. Therefore $V_{M_1} \subseteq V_{M_2}$. Similarly, it can be concluded that $V_{M_2} \subseteq V_{M_1}$. Thus, $V_{M_1} = V_{M_2}$. ■

Corollary 3.2: All the k-lateration localization methods are equivalent, given the same initial reference set.

Proof: Nodes that are localized by a k-lateration based localization method are the elements of V_M. Since V_M is unique, the proposition is true. ■

Therefore, we can conclude that all the k-lateration localization methods are equivalent given the same initial reference node set.

IV. USP Design

In this section, we present a distributed positioning framework for three dimensional USNs termed USP. USP is based on a novel projection-based localization technique that enables traditional 2D localization methods to be applicable to 3D environments.

The framework is composed of two main phases: an offline pre-distribution phase and a distributed localization phase. The first phase consists of nodes being pre-loaded with initial configuration information (e.g., the amount of time allocated to each iteration), while the latter iteratively executes the distributed localization technique. Before presenting USP, we discuss its network model and underlying assumptions, and elucidate the projection technique that it employs.

A. Network Model and Assumptions

We consider three dimensional USNs where relatively stationary nodes [5] are randomly distributed throughout an oceanic medium, with at least three anchor nodes being included in the deployment. To simplify the process of endowing these nodes with their positions, they are placed on the surface as GPS-enabled bouys.

Practical issues such as economics suggest that the sensors will be sparsely deployed [5]. Additionally, as pointed out in [6], the propagation characteristics of radio waves in water dictate that sensors will employ acoustic waves for communication. Therefore, we require sensors to be capable of using ToA to measure distances between themselves [3]. A simple ToA-based ranging method that does not rely on time synchronization is proposed in [16]

Each sensor also employs its depth information. This information is typically computed with a pressure sensor and knowledge of the pressure-depth relationship that is associated with the medium of interest. Other techniques for obtaining this information include having sensors adjust self-regulated wires attached to a seabed anchor [6].

B. Projection Technique

Traditional 3D underwater localization techniques (e.g., silent positioning [1]) require the existence of at least 4 non-coplanar anchors or other previously localized nodes termed "reference nodes" to be within communication range of the to-be-localized node. However, in USP, this requirement is obviated through the use of sensor depth information and a location projection technique that maps the positions of neighboring reference nodes from one plane to another. A simple projection is to map the referenc nodes to the horizontal plane containing the to-be-localized node.

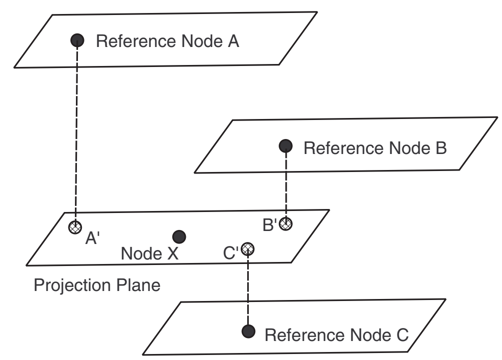


Fig. 4. The projection of three reference nodes A, B, and C to a plane containing the to-be-localized node X. The projected nodes are represented as A', B', and C', and their positions are used to localize X.

For example, consider an underwater sensor X that needs to compute its position within a 3D oceanic deployment area as shown in Fig. 4. In this scenario, node X is within communication range of three reference nodes A, B, and C located at known positions (x_A, y_A, z_A), (x_B, y_B, z_B), and (x_C, y_C, z_C), respectively.

Given X's measure of its depth as z_X, and the successfully received broadcasts of the locations of A, B, and C, node X can compute a projection of each node onto its plane P_X (i.e., the horizontal plane containing node X). Specifically, node A is projected onto P_X as node A' located at position (x_A, y_A, z_X), and nodes B and C are projected analogously as nodes B' and C', with the first located at position (x_B, y_B, z_X) and the second at position (x_C, y_C, z_X). Note that this projection is non-degenerative if and only if no two nodes have the same x and y coordinates.

If the projection is non-generative, the task of localizing node X in a three-dimensional space has been reduced to localizing a node in a two-dimensional space. Therefore, after three reference nodes A', B', and C' have been projected,

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2008 proceedings.

elegant localization methods such as trilateration may be employed to localize node X.

Otherwise, if the projection is not non-degenerative, the to-be-localized node can easily detect and respond. Since the positions of A', B', and C' are known, the to-be-localized node can simply check to see if any of the two reference nodes have the same position in the projection plane. Similarly, if a line computed between a pair of reference nodes is equal to a line computed between a different pair of reference nodes, a degenerative projection is detected. In either case, the to-be-localized node simply selects a different (not necessarily disjoint) set of reference nodes to project when available. Note that the sparse deployments of USNs make it almost unlikely that a degenerative projection will occur.

We also note that the post-projection distances used by the chosen localization technique are not the initial distances measured from the ranging method that is used. For example, consider the to-be-localized node X and the reference node B as shown in Fig. 4. The position of B is known by X to be (x_B, y_B, z_B) because B is a reference node for X. Additionally, X has the ability to measure its depth as z_X. Therefore, if the distance between X and B that is calculated by ranging is d_r, the distance between X and B that is used by the localization technique, ${d_r}'$, can be computed as:

$$ {d_r}' = \sqrt{{d_r}^2 - (z_X - z_B)^2}. \quad (1) $$

C. Pre-Distribution

Prior to deployment, each sensor is preloaded with a unique ID. Each node also maintains candidate position sets PS and NS, which will store the position information of themselves and their neighbors, respectively. Additionally, three nodes are selected at random to be anchors. These nodes bootstrap the localization procedure by announcing their positions once deployed.

System parameters M, Δ_B, Δ_C, and Δ_S, as listed in Table I, are also initialized during this phase. While Δ_B and Δ_C support fundamental USP operations, Δ_S helps mitigate the effect that error sources such as receiver system delay and underwater multipath fading have on the performance of USP. Note that these parameters can be estimated before deployment through analysis or simulation (as shown in Section V). Therefore, the total time for an iteration i, with $1 \leq i \leq M$, can be expressed using Eq. (2).

M	Number of iterations USP will be executed
Δ_B	Time sending/receiving broadcasts per iteration.
Δ_C	Time updating a node's PS per iteration.
Δ_S	Per iteration silence period.

TABLE I
PRELOADED SYSTEM PARAMETERS.

$$ \Delta_{T_i} = \Delta_{B_i} + \Delta_{C_i} + \Delta_{S_i} \quad (2) $$

This definition allows for variance among the minimum lengths (i.e., the minimum amount of time to compute a given iteration) of each Δ_{T_i}. An example of its usefulness can be seen in that while broadcasts are made during each iteration, the number of broadcasts made differs from iteration to iteration. However, for succinctness of notations, we consider each iteration to take the same amount of time, which is denoted as Δ_T.

D. Distributed Localization

USP is executed for a maximum number of iterations M by each of the deployed nodes in a distributed manner. Its psuedocode appears below.

Algorithm USP

```
1:  during(Δ_B)
2:  if new_pos_info then
3:      broacast(pos_info)
4:      new_pos_info ← false
5:  end if
6:  if receive(neighbor_pos_info) then
7:      update(NS, neighbor_pos_info)
8:      recv_info ← true
9:  end if
10:
11: during(Δ_C)
12: if |PS| = 0 then
13:     if |NS| ≥ 2 and recv_info is true then
14:         new_pos ← project_location(NS)
15:         update(PS, new_pos)
16:         new_pos_info ← true
17:     end if
18: end if
19: if |PS| > 1 and recv_info is true then
20:     PS' ← reduction(PS, NS)
21:     if |PS \ PS'| > 0 then
22:         PS ← PS'
23:         new_pos_info ← true
24:     end if
25: end if
26: recv_info ← false
27:
28: during(Δ_S)
29: sleep(Δ_S)
```

As shown in Eq. (2), the total time for each iteration is composed of three main time periods. During Δ_B, the first time period, each sensor performs a local broadcast of any new position information that it has (line 3, USP). This information is available when a node is just deployed (when a node is an anchor) or when a node's location information is updated. A sensor also updates the position information of any neighbor from which it receives position information broadcasts (line 7, USP) during this period.

Next, the second time period Δ_C has sensors compute their position information (using bilateration operations) when new position information broadcast is received. If a sensor has no previous position information (line 12, USP), it attempts to compute its position via the projection technique (described in Subsection IV-B) with the position information of its neighbors (line 14, USP). Alternatively, if a sensor already has position information, it attempts to reduce its set of candidate positions (line 20, USP).

Lastly, all sensors sleep for a period of Δ_S. After completion of this step (line 29, USP), an iteration of total length Δ_T has finished, and the subsequent iteration of USP begins.

V. Evaluation

In this section, we provide a comprehensive analysis of the performance of USP through extensive MATLAB simulations. Relevant simulation parameters are outlined below, and the reported results reflect properties that include localization efficiency, storage and computation overheads, energy consumption, and robustness to errors.

- The network consists of 1000 nodes (including 3 anchor nodes) randomly deployed in a 3-dimensional cubic region with a size of $100 \times 100 \times 100$ units.
- Sensing range varies to control the density and connectivity of the network. Since underwater sensor networks are sparse, the highest node degree considered in this paper is approximately 10.
- The outcomes of all simulations are averaged over 100 network instances.

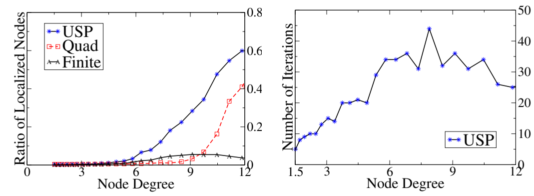
A. Localization Capability

The localization capability of USP is evaluated by analyzing both its ability to localize nodes and the number of iterations required to localize those nodes. Recall that in Section III-C, USP is formally shown to be able to localize all nodes that are capable of being localized by any bilateration method (e.g., Sweeps [14]), thereby transforming the three-dimensional localization problem into its two-dimensional counterpart.

This transformation allows USP to possess significantly improved localization capabilities over traditional three-dimensional localization techniques such as quadrilateration. More specifically, the ability of nodes to compute location information (i.e., a unique or ambiguous position using a trilateration or bilateration method, respectively) with as few of two reference nodes in USP as opposed to the four reference nodes required for quadrilateration provides a significant performance increase.

Indeed, as illustrated in Fig. 5(a), the ratio of nodes localized by USP reaches about 47% while that of quadrilateration is near 15% when the average node degree is 10. The relative performance increase is even higher for smaller node degrees. This is an important characteristic given the sparse nature of USN deployments [5]. As average node degree increases, a network is more easily localized because a greater percentage of nodes can be covered by each anchor node.

Also reflected in Fig. 5(a) is the number of nodes that USP finitely localizes. This value represents an increase in the percentage of localizable nodes by about 5% of the overall network size, and is a nice additional feature of USP because for many applications (e.g., target tracking) partial location information for a known set of nodes is preferable to incorrect or missing information for an unknown set of nodes [17].



(a) (b)

Fig. 5. (a) The ratio of nodes localized by USP in comparison with that of quadrilateration. The ratio of nodes finitely localized by USP is also shown. All computations are made with respect to average node degree. (b) The number of iterations required by USP to localize a network with respect to the average node degree.

Note that these aforementioned results are obtained when including only three anchor nodes in the initial deployment. As can be expected, the number of localized nodes increases with the number of anchor nodes. This can be intuitively explained by the greater likelihood of having neighbors with known position information.

Complementing USP's ability to increase the number of localized nodes is the number of iterations that are required to actually localize the network. As indicted by Fig. 5(b), the distributed nature of USP is suited particularly well for sparse networks; only about 20 iterations are needed for USNs with an average node degree that is ≤ 6.

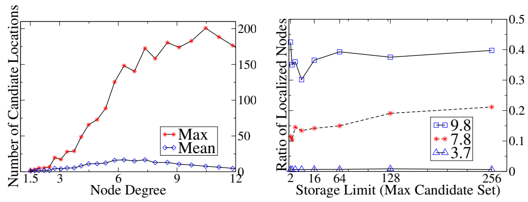
Additionally, a reasonable maximum number of iterations of about 45 occurs when the average node degree is between 7 and 8. This can be expected as around this level of connectivity many small disconnected network clusters begin to be assimilated into a larger single component. Consequently, the number of localizable nodes increases faster than the number of nodes that can be localized per iteration at the beginning of the simulation.

B. Storage and Computation Overhead

The storage overhead imposed by USP is also relevant as a node's candidate position set may store multiple ambiguous positions prior to obtaining a unique position via, for example, some reduction operation. As indicated by Fig. 6(a), the average number of candidate positions by each node is about 16 regardless of the average node degree.

Note that this value is simultaneously a metric for the computation overhead associated with USP. Specifically, the average size of a node's candidate position set represents the average number of reduction operations that should be performed by a node in order to uniquely localize itself.

The maximum size of a node's candidate position set is also plotted in Fig. 6(a). Based on the significant difference between this curve and the mean curve, we conclude that although the possibility of greater storage and computation overheads exists, the proposed algorithm does not require much storage space or reduction of very large candidate position sets on average.



(a) (b)

Fig. 6. (a) The average size of a candidate position set with respect to the average node degree. The maximum size of a candidate position set is plotted as well. These curves also reflect the computation overhead associated with USP. (b) The ratio of the number of localized nodes to the overall network size with respect to different candidate position size restrictions for three different average node degrees.

Despite the low average storage and computation overheads of USP, it may be argued that the occasional maximum candidate position set size creates too much of a resource burden for each underwater sensor. Therefore, a way to further relax this constraint is desirable. One intuitive solution is to limit the allocated memory budget to a more suitable amount. However, the effect of this storage restriction has on the localization capabilities must be investigated. Insight into this relationship is provided by Fig. 6(b).

The curves in Fig. 6(b) correspond to average node degrees of 3.7, 7.8, and 9.8, and each begins to flatten out at a storage limit of about 16 candidate positions. This value corresponds nicely with the previously discussed average candidate position set size. Furthermore, the relatively constant percentage of localized nodes after this size restriction indicates that USP has the ability to localize most localizable nodes with reasonable storage and computation overheads.

C. Energy Consumption

The energy supply and available bandwidth are two closely related and severely limited resources in USNs [6], with communication decreasing both the available battery power and bandwidth. Therefore, we evaluate USP in terms of its two most energy intensive activities: receiving messages and broadcasting messages.

Recall that in USP each node receives messages (or "listens") for position updates until it has a unique location or the algorithm stops after the maximum number of iterations M have been completed. The number of nodes listening during each iteration of USP is illustrated in Fig. 7 with respect to the average node degree.

The graph shows that the number of nodes listening decreases as the number of iterations increases. This behavior is attributed to the number of localized nodes (hence no longer listening) increasing with the number of iterations. Also shown in Fig. 7 is the effect that average node degree has on the number of listening nodes. We observe that the number of nodes listening is quite balanced with respect to node degree. For example, the number of listening nodes steadily increases until an average node degree of about 9 is reached, at which point the value begins to steadily decrease.

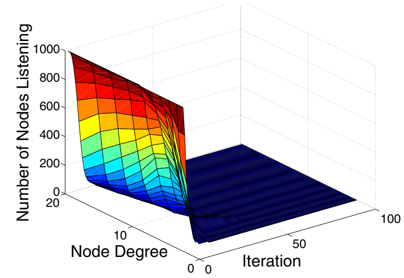


Fig. 7. The number of nodes listening during each iteration of USP with respect to the average node degree.

Of particular importance to the energy consumption in USP is the number of nodes broadcasting messages (or "active nodes") per iteration. Indeed, a typical acoustic modem uses about 50 J/s when transmitting, while only 0.2 J/s when receiving [5]. As shown in Fig. 8, USP has a relatively small and predictable number of active nodes during each iteration.

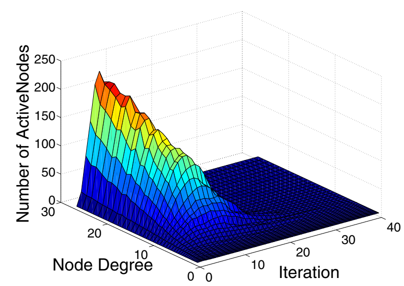
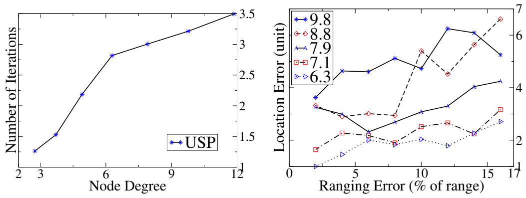


Fig. 8. The number of active nodes during each iteration as it relates to the average node degree.

Specifically, for sparse node deployments, the number of active nodes steadily increases to about 25 at iteration number 9, and then gradually decreases until about iteration number 20 when there are no active nodes remaining. The symmetry of this curve can be attributed to the relatively large number of nodes that have updated position information, and hence messages to broadcast, near the midpoint of the duration of USP.

The number of these nodes then begins to decrease as the nodes belonging to each node's neighborhood begin to become localized. Additionally, because denser deployment results in more nodes receiving updated position information with each broadcast, we observe that the number of active nodes steadily increases with respect to node degree.

Contrasting with the number of nodes active per iteration, we illustrate the average number of iterations that each node is active with respect to node degree in Fig. 9(a). The results indicate that the purely distributed nature of USP enables to localize themselves quickly, with an average of about three position update messages sent by each node. This also suggests that USP makes very efficient use of the limited bandwidth available in USNs.



(a) (b)

Fig. 9. (a) The average number of iterations that each node is active with respect to the average node degree. (b) The cumulative localization error with respect to ranging error for different average node degrees.

D. Robustness

The underwater environments present unique challenges to accurate localization. Indeed, variations in water temperature, salinity, and overall clarity adversely affect the accuracy of ranging methods. The combined influence that the time of day, weather, and depth have on these variations require any effective localization methods to tolerate effectively random ranging errors.

We therefore evaluate the ability of USP to localize sensors in the presence of noisy distance measurements. Specially, all range measurements in USP include different Gaussian noise at different average node degrees. The noise is zero-mean with a standard deviations of 2%, 4%, 6%, 8%, 10%, 12%, 14%, and 16% of the range applied to all distance measurements. As indicated by Fig. 9(b), the location error increases with the node degree. It is because the number of iterations reaches the top when the node degree is around 9 as shown in Fig 5(b), and more iterations introduces more accumulative error. Therefore, we can also conclude that Fig. 9(b) shows the worst cases with USP.

VI. Summary and Future Work

In this paper, we have studied the localization problem in 3D underwater acoustic sensor networks. To employ the depth information available to an underwater sensor, projection is introduced to transform the 3D localization system to 2D such that popular terrestrial positioning techniques can be easily applied. We prove that a non-degenerative projection preserves the network localizability and all of the k-lateration localization methods are equivalent. Then a novel distributed localization framework termed USP for sparse 3D sensor networks is proposed. USP employs a distributed non-degenerative projection technique where reference nodes are projected to the plane that contains the to-be-localized sensor. Through extensive simulation, we show that USP is able to i) improve localization capabilities over existing 3D methods; ii) maintain consistently low storage overhead and computation overhead; iii) display predictable and balanced communication overhead; iv) perform localization that is robust to underwater acoustic channel errors.

Additionally, the design of USP is general enough to support relative sensor positioning. We will therefore explore the feasibility of post-deployment endowment of anchor nodes with position information so that a transformation of the relative coordinate system may be computed. We also plan to incorporate a network partitioning and joining strategy so as to reduce the total number of iterations and the accumulative errors.

Acknowledgment

This work was completed when Wei Cheng was visiting the George Washington University. The research was partially supported by the US National Science Foundation under grants CNS-0347674 and CNS-0721669, and partially supported by the National Basic Research Program of China (973) under Grant No. 2005CB321801 and the National Natural Science Foundation of China under Grant No. 60673167.

References

[1] X. Cheng, H. Shu, Q. Liang, and D. H.-C. Du, "Silent positioning in underwater acoustic sensor networks," in *IEEE Transactions on Vehicular Technology*, (to appear).

[2] M. P. F. Koushanfar, S. Slijepcevic and A. Sangiovanni-Vincentelli, "Location discovery in ad-hocwireless sensor networks," *Ad Hoc Wireless Networking*, (editors X. Cheng, X. Huang and D.-Z. Du).

[3] V. Chandrasekhar, W. K. Seah, Y. S. Choo, and H. V. Ee, "Localization in underwater sensor networks: survey and challenges," in *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 33–40.

[4] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges," *Elsevier's Journal of Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[5] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," in *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks*, 2006, pp. 17–24.

[6] I. F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for efficient communication in underwater acoustic sensor networks," *SIGBED Rev.*, vol. 1, no. 2, 2004.

[7] C. Bechaz and H. Thomas, "Gib system: The underwater gps solution," in *Proceedings of 5th Europe Conference on Underwater Acoustics*, 2000.

[8] T. C. Austin, R. P. Stokey, and K. M. Sharp, "A buoy-based system for auv navigation and tracking," in *OCEANS 2000 MTS/IEEE Conference and Exhibition*, 2000.

[9] M. Hahn and J. Rice, "Undersea navigation via a distributed acoustic communication network," in *Proceedings of Turkish International Conference on Acoustics*, 2005.

[10] V. Chandrasekhar and W. K. G. Seah, "Area localization scheme for underwater sensor networks," in *Proceedings of the IEEE OCEANS Asia Pacific Conference*, 2006.

[11] D. Niculescu and B. Nath, "Ad hoc positioning system (aps)," in *GLOBECOM*, 2001.

[12] Z. Zhou, J.-H. Cui, and S. Zhou, "Localization for large-scale underwater sensor networks," in *UCONN CSE Technical Report: UbiNet-TR06-04*, 2006.

[13] Y. Zhang and L. Cheng, "A distributed protocol for multi-hop underwater robot positioning," in *Proceedings of IEEE International Conference on Robotics and Biomimetics*, 2004.

[14] D. K. Goldenberg, P. Bihler, Y. R. Yang, M. Cao, J. Fang, A. S. Morse, and B. D. O. Anderson, "Localization in sparse networks using sweeps," in *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2006, pp. 110–121.

[15] W. Cheng, X. Cheng, X. Lu, and Z. Lu, "Route recovery in vertex-disjoint multipath routing for many-to-one sensor networks," in *Mobihoc '08 (Submitted)*.

[16] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "Tps: A time-based positioning scheme for outdoor wireless sensor networks," in *INFOCOM*, 2004.

[17] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2004, pp. 50–61.