3D Underwater Sensor Network Localization

Amin Y. Teymorian, *Student Member*, *IEEE*, Wei Cheng, *Student Member*, *IEEE*,
Liran Ma, *Member*, *IEEE*, Xiuzhen Cheng, *Member*, *IEEE*,
Xicheng Lu, *Member*, *IEEE*, and Zexin Lu, *Member*, *IEEE*

Abstract—We transform the 3D underwater sensor network (USN) localization problem into its 2D counterpart by employing sensor depth information and a simple projection technique. We first prove that a nondegenerative projection preserves network localizability. We then 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 bilateration localization scheme for 3D USNs termed as Underwater Sensor Positioning (USP). Through extensive simulations, we show that USP has the following nice features: 1) improved localization capabilities over existing 3D methods, 2) low storage and computation requirements, 3) predictable and balanced communication overhead, and 4) robustness to errors from the underwater environment.

Index Terms—Underwater sensor networks, projection, localizability study, 3D localization.

1 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.

Although USNs are envisioned to enable diverse applications such as aquatic resource monitoring, disaster prevention, and assisted navigation, their unique properties have necessitated an innovative reexamination of problems related to protocol layer design [1], [2], [3], [4], topology formation [5], [6], target tracking [7], and localization [8]. Indeed, propagation delays, motion-induced Doppler shift, limited bandwidth, and multipath interference render many previously proposed solutions inaccurate or infeasible [9].

This is especially relevant for sensor nodes location discovery [10] where, for example, commonly employed RSS-based localization techniques provide ambiguous results in underwater environments [9]. Even the well-established Global Positioning System (GPS) does not work well underwater [11]. Besides the unavailability of GPS, 3D localization becomes even more challenging due to the economically driven sparseness of USN deployments [12] and the unavailability of sufficient numbers of underwater beacons. These properties make USN localization a nontrivial task with relatively few options available.

In this paper, we study the problem of 3D underwater sensor network localization. The *network localization problem* seeks to determine a unique position for each node in a network given the positions of some nodes (termed as "beacons" or "anchors") and the knowledge of some internode distances (ranges). We formally identify the conditions that make it possible to transform the 3D underwater localization problem into its 2D counterpart, and design a projection-based distributed localization framework for underwater sensor positioning. Our research is motivated by the following observations:

- Localization in terrestrial sensor networks, which is known as 2D localization, has been extensively studied and many elegant ideas have been proposed. It is straightforward to directly adopt mature 2D localization schemes when the 3D localization problem is transformed into its 2D counterpart.
- Existing 3D underwater localization schemes requires noncoplanar anchor nodes to uniquely localize a sensor, which implies that at least one anchor with position information needs to be underwater. However, deploying anchor nodes underwater is either infeasible or cost-prohibitive, especially in deep oceanic environments. This is a "chicken or egg" problem: if there is an easy way to localize an anchor underwater, other sensors can be localized in a similar manner.
- Underwater sensors typically have depth information available through various techniques [13], which may be exploited such that only anchor nodes on the sea surface (a horizontal plane) are required. In this case, the surface anchors can be projected to the plane determined by the depth of the to-be-localized node, whose position can thereby be resolved.

Based on these observations, we intend to project the surface anchors to the plane where the to-be-localized sensor resides, by which the original 3D localization

- *A.Y. Teymorian, W. Cheng, and X. Cheng are with the Department of Computer Science, The George Washington University, 801 22nd Street NW, Suite 704, Washington, DC 20052.*
  E-mail: {amin, wcheng, cheng}@gwu.edu.
- *L. Ma is with the School of Technology, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931.*
  E-mail: lrma@mtu.edu.
- *X. Lu and Z. Lu are with the School of Computer Science, National University of Defense Technology, Changsha, Hunan 410073, PRC.*
  E-mail: {xclu, lzx}@nudt.edu.cn.

Manuscript received 16 Oct. 2007; revised 15 June 2008; accepted 10 Apr. 2009; published online 29 Apr. 2009.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2007-10-0314.
Digital Object Identifier no. 10.1109/TMC.2009.80.

1536-1233/09/$25.00 © 2009 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

problem is transformed to its 2D counterpart such that 2D localization techniques can be employed. We prove that each node preserves its localizability in the plane on which it is projected if the projection is nondegenerative. Under this condition, a node is localizable in the projection plane if and only if it is localizable in the original 3D underwater network. Equipped with this theory, we design a projection-based underwater network localization framework termed as **U**nderwater **S**ensor **P**ositioning (USP). The performance of USP is extensively investigated via simulation, which indicates that with only three surface anchors, USP improves localization capabilities over existing 3D techniques, has low storage and computation requirements, incurs predictable and balanced communication overhead, and is robustness to environment-induced errors.

USP is an iterative distributed localization framework that is designed especially for sparse 3D underwater sensor networks. It utilizes bilateration as the base localization method since bilateration works better than trilateration in 2D sparse networks [14]. Via projection, surface anchors and reference nodes[1] are projected to the horizontal plane on which the to-be-localized sensor resides. Via bilateration, the to-be-localized node starts its position computation when its distances to two anchor/reference nodes are available. Its unique position, if currently unavailable, can be iteratively resolved when more position information from neighboring anchors/reference nodes is obtained. To justify the localization capability of USP, we prove that a node can be localized by a geometric k-lateration localization method if and only if it can be localized by the other k-lateration localization methods. This result guarantees the equivalence of all geometric k-lateration localization methods, which facilitates the proof of our major result that USP is able to determine a unique position for a sensor if and only if the sensor is uniquely localizable in the original 3D underwater network.

The remaining portion of the paper is organized as follows: Section 2 provides a brief overview of related research. In Section 3, we conduct a network localizability study that forms the foundation of USP. A detailed elaboration on the design of USP appears in Section 4, and an extensive analysis of USP's performance is provided in Section 5. We conclude this paper in Section 6 with a discussion of future research directions.

2 Related Work

Typical localization techniques are either range-based or range-free. Range-based techniques rely on various mechanisms such as Time-of-Arrival (ToA) to estimate the distances (ranges) to anchor nodes, and then, convert these distance estimations to position information via trilateration (in 2D) or quadrilateration (in 3D). Compared to range-based methods, range-free methods usually provide coarser position information as only an area containing the to-be-localized node needs to be determined [15]. Since USP is a range-based underwater localization framework, which can employ with any ranging method, we will focus on the ranging techniques that are proposed especially for underwater sensor networks. For a comprehensive literature survey on underwater localization, we refer the interested reader to [9], and the references therein.

Ranging techniques are either communication-based or connectivity-based. In "underwater GPS" such as GPS Intelligent Buoys (GIBs) [16] and PARADIGM [17], active communication between a to-be-localized sensor and the surface buoys is needed, and therefore, time needs to be synchronized in order to convert time measurements into ranges. To avoid time synchronization, a ping-pong style scheme to measure the round-trip delay between the sensor and a buoy for range estimation is proposed by Hahn and Rice [18]. All these ranging methods require 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 [12]. To overcome this problem, a silent positioning scheme is proposed in [8], where sensors estimate the ranges by passively listening to the beacon messages exchanged among anchor nodes. This scheme requires no time synchronization and has low computation and communication overheads, but needs at least four noncoplanar anchors that can mutually hear each other.

When there is no direct communication between anchor nodes and sensors, network connectivity can be exploited for range estimation. In [19], DV-hop (range estimation is based on hop count), DV-distance (range estimation is based on cumulative range estimates obtained from Received Signal Strength Indicator (RSSI)), and euclidean (range estimation is based on solving euclidean equations formed with the distances to two neighbors, the distance between the two neighbors, and the distance from the neighbors to the base station) are proposed. Although euclidean method is shown to perform the best in anisotropic topologies, there is an expense of larger computation and communication overheads. The euclidean ranging method has been extended to support 3D underwater localization by Zhou et al. [20] when the communication range of the anchor nodes is short, and by Zhang and Cheng [21] when long-range anchor nodes are available. The extensive local flooding in [20] and the global beacon flooding in [21] are bandwidth-intensive, and therefore, unavoidably degrade the throughput in USNs. Additionally, a larger number of anchor nodes are required in [21], which results in a higher deployment cost.

A recent work termed as SLMP [22] employs GPS-enabled surface buoys and special-purpose anchor nodes. The anchors are deployed subsurface among other less powerful underwater sensors, but have the ability to self-localize via direct communication with four surface buoys. Temporal and spatial correlations in underwater group mobility patterns are leveraged to reduce the communication overhead incurred while localizing the sensors. Additionally, latency- and mobility-induced position errors are countered in [23] with probabilistic models of node position changes and maximum-a-posteriori estimates of a node's position at a particular time period. The work that is closest to USP is DNR [24]. The scheme requires at least

1. A *reference node* is a node whose position has been uniquely or partially determined, and therefore, it can serve as an anchor to other to-be-localized nodes.

three anchor nodes to dive to the depth of the to-be-localized sensor so that triangulation can be performed.

USP is fundamentally different from the aforementioned schemes. First, all of ranging techniques proposed in the literature mentioned above are applicable to USP as USP is a framework that does not rely on a dedicated ranging method. Second, USP has a stronger localization capability compared to existing 3D underwater localization methods. USP is a projection-based approach that is designed to iteratively localize all localizable sensors in a 3D underwater network via bilateration. Third, USP requires only three GPS-enabled buoys residing at the sea surface in order to localize a large percentage of nodes (even in sparse deployments, see Section 5).

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

3 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 anchor/reference nodes, no matter how many they are, 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 plane. This problem may be solved if we employ the depth information that is typically available to underwater sensors using various techniques [13]. Specifically, given the depth of underwater sensors, we can map the positions of the anchor nodes to the plane containing a 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 study whether or not a nondegenerative 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 USP preserves the capabilities of the 2D localization methods.

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

3.1 Background Information

Let $G(V, E)$ be the graph representing a 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.

A node is *localizable* if its location can be uniquely determined; otherwise, the node is *unlocalizable*. Although a node is unlocalizable, it is still possible to compute several candidate positions that form its *candidate position set*. These nodes are *finitely localizable*. For example, as shown in Fig. 1a, the 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 via trilateration, node C is unlocalizable. However, C can derive two candidate positions C' and C'', with each satisfying the

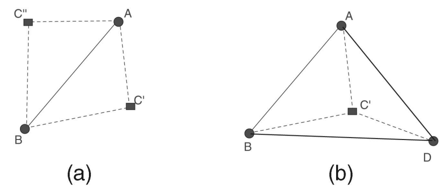

Fig. 1. (a) Node C has two candidate positions. (b) C becomes localizable after introducing anchor D and performing a reduction operation.

distance constraints established by A and B. Since only one of the positions is node C's real position, C is a finitely localizable node with candidate position set $\{C', C''\}$.

Note that if a finitely localizable node can get more positioning information from another anchor, it can perform a reduction operation on its candidate position set and become localizable. A *reduction* is a procedure that reduces the number of candidate positions. As shown in Fig. 1b, node C can delete C'' from its candidate position set after the internode distance to anchor node D is available because C'' does not satisfy the distance constraint imposed by D.[2]

If all of the nodes in G can be uniquely localized, G is *localizable*; if only a fraction of the nodes are uniquely localizable, G is *partially localizable*; if the number of possible positions for each node is finite, G is *finitely localizable*. A finitely localizable network is partially localizable.

Let us consider a graph G in d dimensions. G is a *trilateration network* if it has a graph representation with a trilateration ordering. A graph has a *trilateration ordering* if its vertices can be ordered as $v_1, v_2, v_3, \ldots, v_n$, where $v_1, v_2, \ldots, v_{d+1}$ are the "seeds," and each vertex v_i with $i > d+1$ is adjacent to at least three vertices v_j with $j < i$. Similarly, we can define *bilateration networks* and *quadrilateration networks*, which require each vertex v_i with $i > d+1$ to be adjacent to at least two or four vertices v_j with $j < i$, respectively.

Although the complexity of uniquely localizing a network is NP-hard [25], there exist classes of networks that can be efficiently localized. Examples of such classes include *bilateration networks* in one dimension, *trilateration networks* in two dimensions, and *quadrilateration networks* in three dimensions. In general, $(d+1)$-lateration networks can be uniquely localized with a complexity that is polynomial in the number of vertices in d dimensions.

The relationship between quadrilateration networks, trilateration networks, and bilateration networks is illustrated in Fig. 2. If the nodes localized by a localization method can construct a bilateration network, this localization method is a *bilateration localization method*. We can similarly define a *trilateration localization method* and a *quadrilateration localization method*. These methods differ in the number of reference nodes, denoted by k, that 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 $k = 2, 3$, and 4 refer to bilateration, trilateration, and quadrilateration, respectively.

2. In this simple example, when the distance to D is available, C can compute a unique position via trilateration when all range information has no error. The procedure fo solving the equations in trilateration involves the same reduction step.

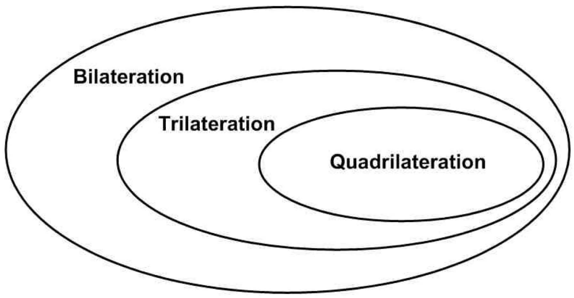

Fig. 2. The relationship between the classes of networks.

Based on the above definitions, it is easier to find a bilateration ordering than to find a trilateration ordering or quadrilateration ordering for a network. As pointed out in [14], given a 2D network, it is possible that a node can be uniquely localized by a bilateration localization method but cannot be localized by a trilateration localization method since the node may be excluded from the corresponding trilateration ordering. However, the ability to localize more nodes with a bilateration method does not come for free. Indeed, for 2D localization, the computational complexity of a bilateration localization method is exponential in the number of nodes [14], while that of a trilateration localization method is polynomial.

Therefore, it is possible that there exist some nodes that are localizable in a network but they are not included in any trilateration or quadrilateration ordering given the $d+1$ anchors. Furthermore, there exist localizable networks that cannot be localized by trilateration and quadrilateration methods but can be localized by a bilateration method. For example, a bilateration method can uniquely localize all nodes in a 2D wheel network given three nodes as anchors, as shown in Fig. 3, whereas trilateration and quadrilateration cannot.

3.2 Localizability Preservation Study

In this section, we prove that a nondegenerative projection preserves the localizability of the network.

Definition 3.1. *Given a plane F in the 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 an euclidean transformation. A projection is *nondegenerative* to a node set S if and only if $v_1^F \neq v_2^F$ when $v_1 \neq v_2$, where $v_1, v_2 \in S$.

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 nondegenerative, 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 Definitions 3.1 and 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.*

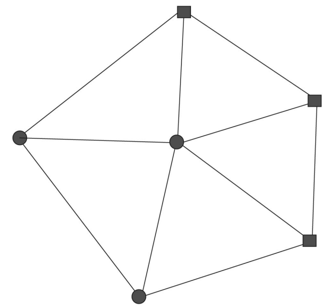

Fig. 3. A wheel network with circles representing the anchors and rectangles representing the to-be-localized nodes.

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 that the projection P_F is nondegenerative. 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 nondegenerative, 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 nondegenerative projection preserves the localizability of a network G. This observation motivates the design of our distributed USP scheme in Section 4.

3.3 Localizability Equivalence Study

There usually exist a number of anchors in a network. A k-lateration localization method may localize nodes either finitely or uniquely, in a step-by-step manner. Initially, the k-lateration localization method starts to compute a node's position if at least k of the node's neighbors are anchors. Any node can compute its position if it is adjacent to k or more anchors or reference nodes. The algorithm terminates when there does not exist any unlocalized node with k or more previously positioned neighbors. In general, a $(k-1)$-lateration localization method always performs better in terms of localizability that a k-lateration localization method.

Recall that a k-lateration localization method follows a k-lateration ordering. Intuitively, different k-lateration localization methods localize the nodes in different k-lateration orderings given the network and its anchors. In this section, we will prove that all of the k-lateration localization methods are equivalent in node localizability given the network and its initial reference nodes. This means that the nodes localized by all of the k-lateration localization methods will induce a unique network. In other words, a node can be localized by one k-lateration localization method if and only if it can be localized by the other k-lateration localization methods.

The proof is based on the definitions that appear below. Assume that $G(V, E)$ is a d-dimensional graph with $d > 1$.

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

For a to-be-localized network, all of 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 a 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* $V_1 = V_0 \cup \{v | v \in V \setminus V_0, \exists v_1, v_2, \ldots, v_k \in V_0$, *s.t.* $(v, v_i) \in E$ *for* $i = 1, 2, \ldots, k\}$.

Definition 3.7. *Given a t-seed subgraph* G_{ts} *of* G, *a* ***k-lateration extension subgraph*** $G_m(V_m, E_m), m = 1, 2, \ldots,$ *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 a k-credit node, which is introduced in [26].

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 [26], 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 *is vertex-disjoints with all of 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* $V_m, m = 1, 2, \ldots,$ *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 $\forall j = 1, 2, \ldots, i$.

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 $v_1, v_2, \ldots, v_j, \ldots, v_k \in V_i$, such that these k nodes joint v as shown in Fig. 4a. 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}.

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

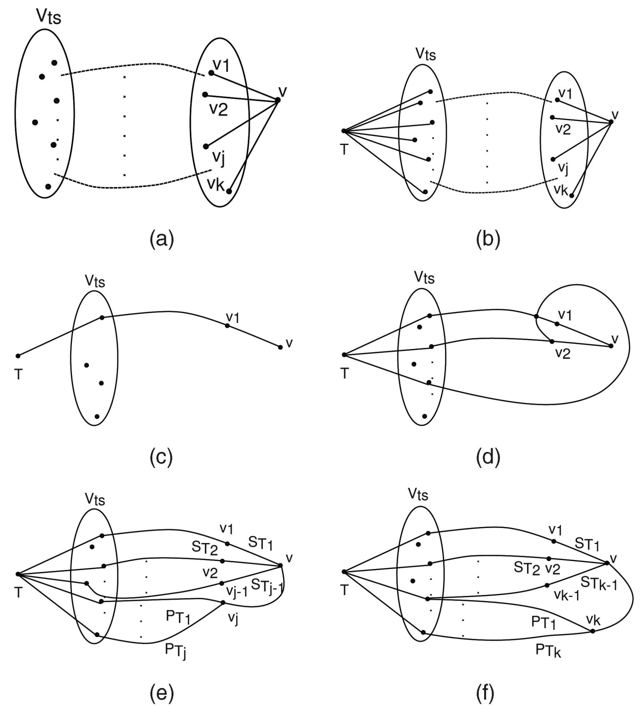

Fig. 4. The progress of constructing k vertex disjoint paths.

We assume that there is a virtual sink node T connecting all of the initial reference nodes, as shown in Fig. 4b. 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 $v_2, \ldots, v_j, \ldots, v_k$. 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. 4c. And this path does not pass any of $v_2, \ldots, v_j, \ldots, v_k$.
- Pick two of v_2's paths, which do not pass any of $v_3, \ldots, v_j, \ldots, v_k$, as shown in Fig. 4d. According to Theorem 3.2, there exist two vertex-disjoint paths between v and T.
- ...
- Pick j of v_j's paths, which do not pass any of $v_{j+1}, \ldots, v_k$, as shown in Fig. 4e. Based on Theorem 3.2, there exist j vertex-disjoint paths between v and T.
- ...
- Pick k paths from v_k's paths, as shown in Fig. 4f. 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 Definition 3.7. □

Definition 3.9. *Given a t-seed subgraph* G_{ts} *of* G, *a* ***maximum 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 the k-lateration ordering followed by a k-lateration localization methods starts with the graph induced by the initial reference nodes. Therefore, a new k-lateration extension subgraph is generated whenever a new node in the k-lateration ordering is localized. k-lateration localization method terminates, a maximal k-lateration extension subgraph is computed.

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 of 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 that there are two V_Ms, 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 of the k-lateration localization methods are equivalent, given the same set of initial reference nodes.*

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. □

4 USP Design

In this section, we present a distributed positioning scheme for 3D USNs termed as USP, deferring its analysis to the next section. USP is based on a novel projection-based localization technique that enables traditional 2D localization methods to be applicable to 3D environments.

The scheme is composed of two main phases: an offline predistribution phase and a distributed localization phase. The first phase consists of nodes being preloaded 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.

4.1 Network Model and Assumptions

We consider 3D USNs where relatively stationary nodes [12] are randomly distributed throughout an oceanic medium, with at least three anchor nodes included in the deployment. To simplify the process of endowing anchor nodes with their positions, they are placed on the surface as GPS-enabled buoys.

Practical issues such as economics suggest that the sensors will be sparsely deployed [12]. Furthermore, we require nodes to be capable of measuring distances (ranges) between themselves [9]. Note that USP is a localization framework that does not rely on any dedicated ranging method. Therefore, any of the ranging techniques surveyed in Section 2 is applicable.

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 [13].

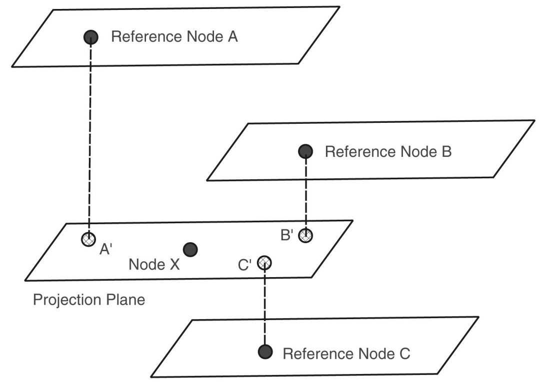

Fig. 5. The projection of three reference nodes A, B, and C to a plane containing the to-be-localized node X. The projected nodes are represented as A', B', and C', and their positions are used to localize X.

4.2 Projection Technique

Recall that traditional 3D underwater localization techniques (e.g., silent positioning [8]) require the existence of at least four noncoplanar anchors or 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 reference nodes from one plane to another. A simple projection is to map the reference nodes to the horizontal plane containing the to-be-localized node.

For example, consider an underwater sensor X that needs to compute its position within a 3D oceanic deployment area, as shown in Fig. 5. 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 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 nondegenerative if and only if no two nodes have the same x and y coordinates.

With a nongenerative projection, the task of localizing node X in a 3D space has been reduced to localizing X in a 2D space. Therefore, after three reference nodes A', B', and C' have been projected, elegant localization methods such as simple bilateration may be employed to localize node X.

Otherwise, if the projection is not nondegenerative, the to-be-localized node can easily detect this and respond accordingly. 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

TABLE 1
Preloaded System Parameters

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.

selects a different (not necessarily disjoint) set of reference nodes to project when available. Note that the sparse deployments of USNs make it unlikely that a degenerative projection will occur.

We also note that the postprojection distances used by the chosen localization technique are not the initial distances measured from the ranging method that is used. For example, consider to-be-localized node X and reference node B, as shown in Fig. 5. 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:

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

4.3 Predistribution

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 1, 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 5).

Therefore, the total time for an iteration i, with $1 \leq i \leq M$, can be expressed using (2):

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

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.

4.4 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 (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 from a reduction operation. 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 to 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 Section 4.2) 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.

4.5 USP Localization Capability

USP employs both projection and bilateration to achieve 3D underwater localization. On one hand, the multi-iteration execution procedure of USP computes a maximal bilateration extension subgraph of the original network in a step-by-step manner. At any iteration, each node executes lines 13-17 once. Starting from the t-seed subgraph induced by all anchor nodes, each iteration produces a binary extension to the previous subgraph, and USP terminates when no new position information is broadcasted. Therefore, based on Theorem 3.3 and Corollary 3.2, USP is able to localize any node that could be localized by bilateration. On the other hand, Corollary 3.1 indicates that USP is able to localize any node that is localizable in the original 3D network since only a nondegenerative projection is utilized (see Section 4.2). As bilateration has the potential to localize more nodes than trilateration and quadrilateration given a network and a set of anchor nodes, we claim that USP provides an optimal solution, i.e., USP can localize all sensors that could be localized by any possible bilateration, trilateration, and quadrilateration-based localization method. This elaboration leads to the following corollary:

Corollary 4.1. *USP is able to determine a unique position for a sensor if and only if the sensor is uniquely localizable in the original 3D underwater network.*

Generally speaking, a bilateration localization method cannot localize all of the localizable nodes in the original 3D network without projection. However, without employing bilateration, a localization method may not work well in a sparse 3D network. Furthermore, without a distributed implementation, a bilateration localization method cannot work well for large-scale networks. USP is a distributed localization framework that seamlessly unifies projection and bilateration, the two techniques that jointly guarantees its optimal localization capability.

5 Evaluation

In this section, we analyze the performance of USP through extensive MATLAB [27] simulations. Relevant simulation parameters are outlined below.

- The network consists of 1,000 nodes (including three anchor nodes) randomly deployed in a 3D 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.

The reported results include localization efficiency, storage and computation overheads, energy consumption, and robustness to errors.

5.1 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 these nodes. Recall that in Section 3.3, 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 3D localization problem into its 2D counterpart.

This transformation allows USP to possess significantly improved localization capabilities over traditional 3D 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.

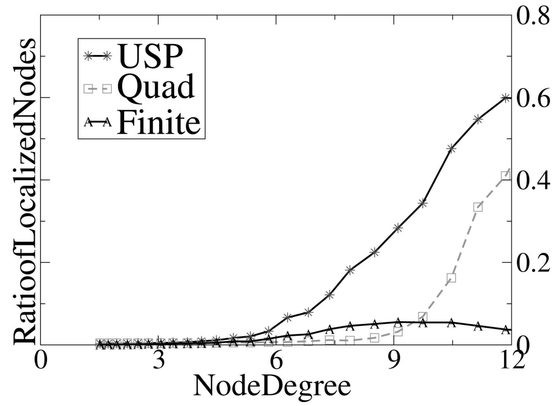

Fig. 6. The ratio of nodes localized by USP in comparison with that of quadrilateration. The ratio of nodes finitely localized by USP is also shown. All computations are made with respect to average node degree.

Indeed, as illustrated in Fig. 6, the ratio of nodes localized by USP reaches about 47 percent, while that of quadrilateration is near 15 percent 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 [12]. 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. 6 is the number of nodes that USP finitely localizes. This value represents an increase in the percentage of localizable nodes by about 5 percent 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 [28].

Note that these 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. 7, 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.

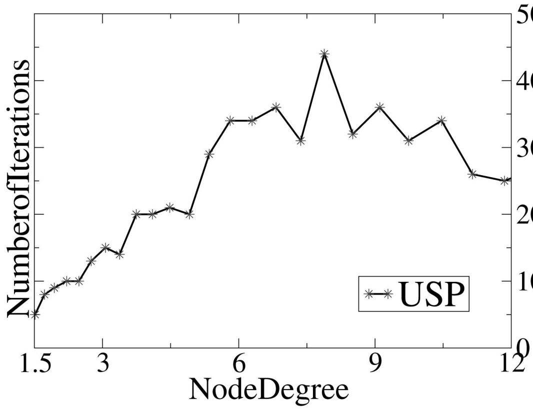

Fig. 7. The number of iterations required by USP to localize a network with respect to the average node degree.

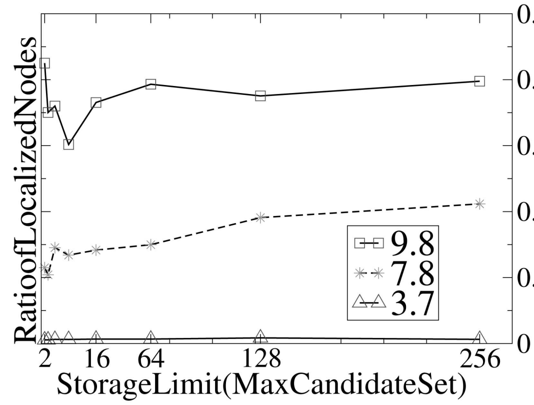

Fig. 9. The ratio of the number of localized nodes to the overall network size with respect to different candidate position size restrictions for three different average node degrees.

5.2 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. 8, 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. 8. 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.

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 this storage restriction has on the localization capabilities must be investigated. Insight into this relationship is provided in Fig. 9.

The curves in Fig. 9 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.

5.3 Energy Consumption

The energy supply and available bandwidth are two closely related and severely limited resources in USNs [13], 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 has been completed. The number of listening nodes during each iteration of USP is illustrated in Fig. 10 with respect to the average node degree.

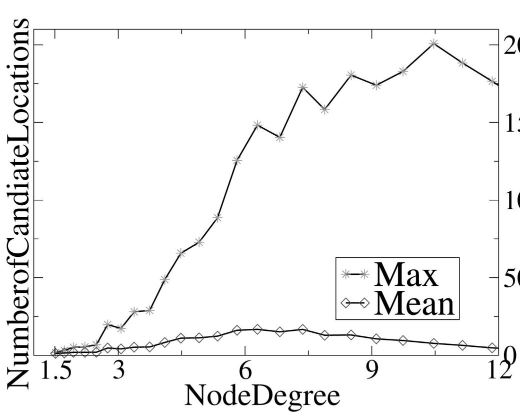

Fig. 8. The average size of a candidate position set with respect to the average node degree. The maximum size of a candidate position set is plotted as well. These curves also reflect the computation overhead associated with USP.

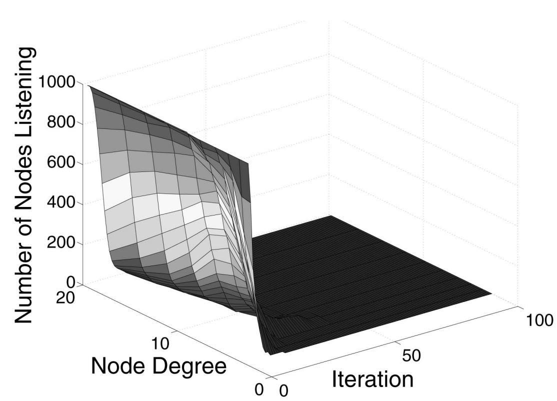

Fig. 10. The number of listening nodes during each iteration of USP with respect to the average node degree.

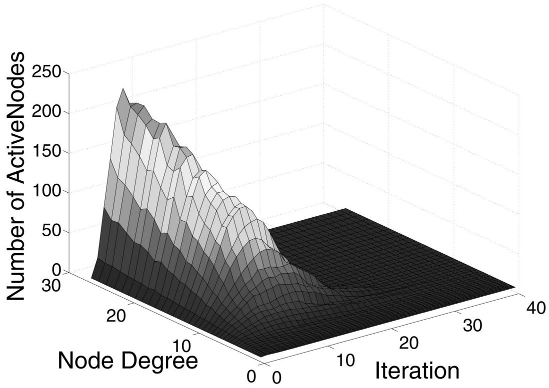

Fig. 11. The number of active nodes during each iteration as it relates to the average node degree.

The graph shows that the number of listening nodes 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. 10 is the effect that average node degree has on the number of listening nodes. We observe that the number of listening nodes 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.

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 [12]. As shown in Fig. 11, USP has a relatively predictable number of active nodes during each iteration. 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.

We also notice from Fig. 11 that at higher node degree, the number of nodes climbs up faster and reaches its peak earlier, and drops down to the ground faster and earlier. This phenomenon is caused by the fact that a denser deployment results in a larger number nodes updating their positions and a smaller candidate position set at each iteration. Roughly speaking, the rising edge of the curve represents the process of candidate position discovery, and the falling edge illustrates the procedure of reduction.

Contrasting with the number of nodes active per iteration (as shown in the previous figure), we illustrate the average number of iterations that each node is active with respect to node degree in Fig. 12. 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.

5.4 Robustness

It is obvious that both ranging errors and depth errors negatively affect the accuracy of USP. In this section, we

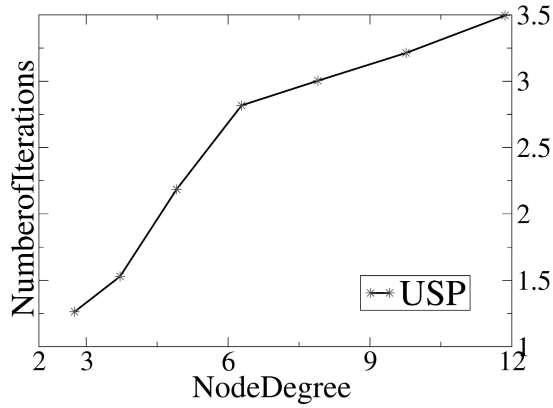

Fig. 12. The average number of iterations that each node is active with respect to the average node degree.

examine their effect on the positioning error of USP by simulation.

As mentioned earlier, USP is a projection-based 3D underwater localization framework that does not define a specific ranging method. Therefore, when considering positioning error, we simply assume that ranging errors are Gaussian distributed. To be specific, we assume that ranging errors satisfy $\mathcal{N}(0, \sigma^2)$, where σ^2 is set to be 2, 4, 6, 8, 10, 12, 14, 16, and 20 percent in our simulation. Underwater ranging errors result from the variations in water temperature, salinity, and the overall clarity [29]. For the same reason, we assume that depth errors are Gaussian-distributed with a mean of 0 and a variance of 1. At underwater environment, the depth information is usually obtained by measuring water pressure. Current water pressure sensor technologies could provide very accurate underwater depth measurement [30]. We first investigate the effect of ranging errors when perfect depth information (no depth error) is available. As illustrated in Fig. 13, the location error increases with the increase of node degree. This is because at a higher node degree, a larger number of iterations are needed to uniquely localize a node. We also notice that the increase of the positioning error is much slower than that of the ranging error. This indicates that USP is robust to ranging errors. When the ranging error is above 10 percent, the two lower curves approach to each other because the

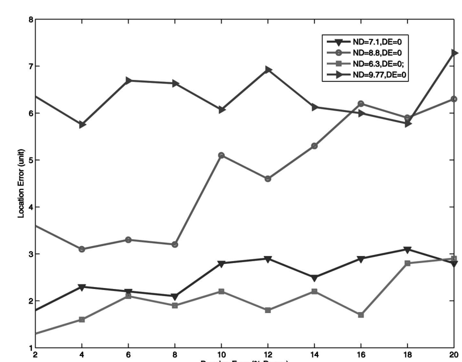

Fig. 13. The cumulative localization error with respect to the ranging error for different average node degrees, where ND and DE represent node degree and depth error, respectively.

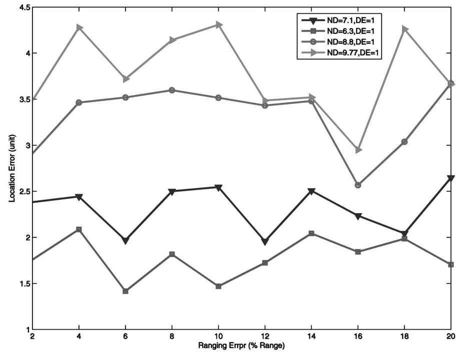

Fig. 14. The cumulative localization error with respect to the ranging error for different average node degrees when depth errors satisfy $\mathcal{N}(0,1)$, where ND and DE represent node degree and depth error, respectively.

transmission ranges corresponding to these two curves are close as the node degrees differ by only about 1. For the same reason, the two upper curves approach to each other when ranging errors are larger than 10 percent.

When depth errors satisfy $\mathcal{N}(0,1)$, we obtain very similar results, as reported in Fig. 14. Comparing to Fig. 13, we notice that the location error is smaller when the depth information is not perfect. This demonstrates another advantage of our projection technique, which makes the two different types of errors (depth and ranging) try to cancel each other's effect. Equation (1) better explains this situation, as ranging errors contribute to the positive term and depth errors contribute to the negative. An example illustrating this situation is given in Fig. 15.

6 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 nondegenerative projection preserves the network localizability and that all of the k-lateration localization methods are equivalent. Then, a novel distributed localization scheme termed as USP for sparse 3D sensor networks is proposed. USP employs a distributed nondegenerative 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:

1. improve localization capabilities over existing 3D methods;
2. maintain consistently low storage overhead and computation overhead;
3. display predictable and balanced communication overhead;
4. 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 postdeployment endowment of

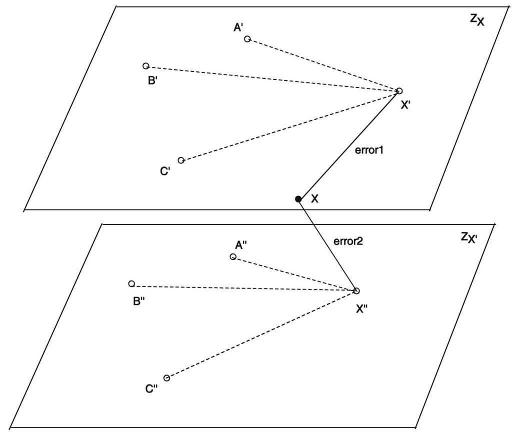

Fig. 15. An example to illustrate the situation when the effect of depth errors and that of the ranging errors try to cancel each other. In this example, $A(0,0,0)$, $B(-3,-1,0)$, and $C(-2,-4,0)$ are three reference nodes and X is the to-be-localized node. When only ranging errors exist, the measured distances from X to A, B, and C are $AX = 10.57$, $BX = 11.05$, and $CX = 10.07$. After projection (projected to the plane z_X), we have $A'X' = 3.43$, $B'X' = 4.71$, and $C'X' = 1.17$. Then, the computed position for X is $X'(1.14, -3.23, 10)$. In this case, the location error ($error1$) is 1.96. When depth errors do exist, the projection plane becomes $z_{X'}$. Now, we have $A''X'' = 5.51$, $B''X'' = 6.21$, and $C''X'' = 4.22$ and the computed position for X is $X''(1.74, -5.02, -9.14)$. In this case, the location error ($error2$) is 0.90, which is smaller than the case with a perfect depth information.

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 accumulated errors.

We also propose to investigate the applicability of geometric constraint solvers [31] for sensor network localization. Generally speaking, the localization problem can be viewed as an equation-solving problem. Existing localization methods differ in the way the constraints are constructed and utilized. Therefore, in theory, geometric constraint solvers can be employed to localize the rigid parts of the network via seeking the unique subsolution among all the possible solutions.

Acknowledgments

The research was partially supported by the US National Science Foundation under grants CNS-0347674 and CNS-0721669, the National Basic Research Program of China (973) under Grant No. 2005CB321801, and the National Natural Science Foundation of China under Grant No. 60673167. An earlier version of this work appeared in the Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08) [32].

References

[1] P. Xie and J.-H. Cui, "R-MAC: An Energy-Efficient MAC Protocol for Underwater Sensor Networks," *Proc. Second Ann. Int'l Conf. Wireless Algorithms, Systems and Applications (WASA '07)*, pp. 187-198, 2007.

[2] T.M.D. Pompili and I.F. Akyildiz, "Routing Algorithms for Delay-Insensitive and Delay-Sensitive Applications in Underwater Sensor Networks," *Proc. ACM MobiCom*, pp. 298-309, 2006.

[3] P. Xie and J.-H. Cui, "An FEC-Based Reliable Data Transport Protocol for Underwater Sensor Networks," *Proc. 16th Int'l Conf. Computer Comm. and Networks (ICCC '07),* pp. 747-753, 2007.

[4] Q. Ren, Q. Liang, and X. Cheng, "Latency Optimized MAC Protocol for Underwater Acoustic Sensor Networks: Cross-Layer Approach," *Elsevier Ad Hoc Networks,* to appear.

[5] S. Ibrahim, J.-H. Cui, and R. Ammar, "Surface Gateway Deployment for Underwater Sensor Networks," *Proc. IEEE Military Comm. Conf. (Milcom '07),* pp. 1-7, 2007.

[6] L. Zhao, Q. Liang, and X. Cheng, "Optimal Cluster Size for Underwater Acoustic Sensor Networks," *J. Ocean Eng.,* to appear.

[7] Q. Liang and X. Cheng, "Underwater Acoustic Sensor Networks: Target Size Detection and Performance Analysis," *Proc. IEEE Int'l Conf. Comm. (ICC '08),* pp. 3151-3155, 2008.

[8] X. Cheng, H. Shu, Q. Liang, and D.H.-C. Du, "Silent Positioning in Underwater Acoustic Sensor Networks," *IEEE Trans. Vehicular Technology,* vol. 57, no. 3, pp. 1756-1766, May 2008.

[9] V. Chandrasekhar, W.K. Seah, Y.S. Choo, and H.V. Ee, "Localization in Underwater Sensor Networks: Survey and Challenges," *Proc. First ACM Int'l Workshop Underwater Networks (WUWNet '06),* pp. 33-40, 2006.

[10] M.P.F. Koushanfar, S. Slijepcevic, and A. Sangiovanni-Vincentelli, "Location Discovery in Ad-Hoc Wireless Sensor Networks," *Ad Hoc Wireless Networking,* X. Cheng, X. Huang, and D.-Z. Du, eds., Kluwer, 2003.

[11] I.F. Akyildiz, D. Pompili, and T. Melodia, "Underwater Acoustic Sensor Networks: Research Challenges," *Elsevier's J. Ad Hoc Networks,* vol. 3, no. 3, pp. 257-279, 2005.

[12] J. Partan, J. Kurose, and B.N. Levine, "A Survey of Practical Issues in Underwater Networks," *Proc. First ACM Int'l Workshop Underwater Networks (WUWNet '06),* pp. 17-24, 2006.

[13] I.F. Akyildiz, D. Pompili, and T. Melodia, "Challenges for Efficient Communication in Underwater Acoustic Sensor Networks," *SIGBED Rev.,* vol. 1, no. 2, pp. 3-8, 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," *Proc. ACM MobiCom,* pp. 110-121, 2006.

[15] V. Chandrasekhar and W.K.G. Seah, "Area Localization Scheme for Underwater Sensor Networks," *Proc. IEEE OCEANS Asia Pacific Conf.,* pp. 1-8, 2006.

[16] C. Bechaz and H. Thomas, "GIB System: The Underwater GPS Solution," *Proc. Fifth Europe Conf. Underwater Acoustics,* 2000.

[17] T.C. Austin, R.P. Stokey, and K.M. Sharp, "Paradigm: A Buoy-Based System for auv Navigation and Tracking," *Proc. MTS/IEEE OCEANS Conf. and Exhibition,* vol. 2, pp. 935-938, 2000.

[18] M. Hahn and J. Rice, "Undersea Navigation via a Distributed Acoustic Communication Network," *Proc. Turkish Int'l Conf. Acoustics,* 2005.

[19] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS)," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '01),* vol. 5, pp. 2926-2931, 2001.

[20] Z. Zhou, J.-H. Cui, and S. Zhou, "Localization for Large-Scale Underwater Sensor Networks," UCONN CSE Technical Report UbiNet-TR06-04, 2006.

[21] Y. Zhang and L. Cheng, "A Distributed Protocol for Multi-Hop Underwater Robot Positioning," *Proc. IEEE Int'l Conf. Robotics and Biomimetics (ROBIO '04),* pp. 480-484, 2004.

[22] Z. Zhou, J.-H. Cui, and A. Bagtzoglou, "Scalable Localization with Mobility Prediction for Underwater Sensor Networks," *Proc. IEEE INFOCOM,* pp. 2198-2206, 2008.

[23] D. Mirza and C. Schurgers, "Motion-Aware Self-Localization for Underwater Networks," *Proc. Third ACM Int'l Workshop Wireless Network Testbeds, Experimental Evaluation and Characterization (WuWNeT '08),* pp. 51-58, 2008.

[24] M. Erol, L.F. Vieira, and M. Gerla, "Localization with Dive'n'Rise (DNR) Beacons for Underwater Acoustic Sensor Networks," *Proc. ACM Int'l Workshop Underwater Networks (WUWNet '07),* pp. 97-100, 2007.

[25] J. Aspnes, W. Whiteley, and Y.R. Yang, "A Theory of Network Localization," *IEEE Trans. Mobile Computing,* vol. 5, no. 12, pp. 1663-1678, Dec. 2006.

[26] W. Cheng, X. Cheng, X. Lu, Z. Lu, J. Su, B. Wang, and Y. Liu, "Route Recovery in Vertex-Disjoint Multipath Routing for Many-to-One Sensor Networks," *Proc. ACM MobiHoc,* pp. 209-220, 2008.

[27] "Matlab—High-Level Technical Computing Language," http://www.mathworks.com/products/matlab, 2009.

[28] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," *Proc. Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* pp. 50-61, 2004.

[29] "Underwater Acoustics Technical Guides—Speed of Sound in Seawater," http://www.npl.co.uk/accoustics/techguides/soundseawater/speedsw.pdf, 2009.

[30] http://www.sensorsone.co.uk, 2009.

[31] W. Bouma, I. Fudos, C. Hoffmann, J. Cai, and R. Paige, "Geometric Constraint Solver," *Computer-Aided Design,* vol. 27, no. 6, pp. 487-501, 1995, citeseer.ist.psu.edu/article/bouma95geometric.html.

[32] W. Cheng, A.Y. Teymorian, L. Ma, X. Cheng, X. Lu, and Z. Lu, "Underwater Localization in Sparse 3d Acoustic Sensor Networks," *Proc. IEEE INFOCOM,* pp. 236-240, 2008.

Amin Y. Teymorian is currently working toward the PhD degree in the Department of Computer Science, The George Washington University, Washington DC. His research interests include wireless networks, network security, and algorithm design and analysis. He is a student member of the IEEE.

Wei Cheng is currently working toward the PhD degree at the George Washington University, Washington DC. His research interests include channel scheduling in wireless mesh networks, localization in wireless sensor networks, and polynomial-time algorithm design and analysis. He is a student member of the IEEE.

Liran Ma received the doctoral degree in computer science from the George Washington University, Washington DC, in 2008. He is an assistant professor in the School of Technology at Michigan Technological University. His current research interests include wireless networking, network and system security, and mobile computing. He is a member of the IEEE.

Xiuzhen Cheng received the MS and PhD degrees in computer science from the University of Minnesota—Twin Cities, in 2000 and 2002, respectively. She is an associate professor in the Department of Computer Science, The George Washington University, Washington DC. Her current research interests include cyberphysical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She is a program cochair or vice cochair for several conferences such as WASA 2009, MSN 2009, ICPP 2009, and MASS 2009. She worked as a program director for the US National Science Foundation (NSF) for six months in 2006 and joined the NSF again as a part-time program director in April 2008. She received the NSF CAREER Award in 2004. She is a member of the IEEE.

Xicheng Lu is a professor at the National University of Defense Technology, Hunan, China. His research interests include computer networks, high-performance computing, and wireless networks. He is a member of the IEEE.

Zexin Lu is a professor at the National University of Defense Technology, Hunan, China. His research interests include high-performance computing, wireless networks, and network systems. He is a member of the IEEE.