



Article A Graph Localization Approach for Underwater Sensor Networks to Assist a Diver in Distress [†]

Roee Diamant ^{1,*} and Roberto Francescon ²

- ¹ Hatter Department of Marine Technologies, University of Haifa, Haifa 3498838, Israel
- ² Department of Information Engineering, University of Padova, Via Gradenigo 6/B, 35141 Padova, Italy; frances1@dei.unipd.it
- * Correspondence: roee.d@univ.haifa.ac.il
- + This paper is an extended version of our paper published in The concept for this work was published in the 14th Workshop on Positioning, Navigation and Communications (WPNC). This journal version includes a different formalization of the problem, consideration of packet collisions, and a demonstration of results from a sea experiment.

Abstract: In this paper, we focus on the problem of locating a scuba diver in distress using a sensor network. Without GPS reception, submerged divers in distress will transmit SOS messages using underwater acoustic communication. The study goal is to enable the quick and reliable location of a diver in distress by his fellow scuba divers. To this purpose, we propose a distributed scheme that relies on the propagation delay information of these acoustic SOS messages in the scuba divers' network to yield a range and bearing evaluation to the diver in distress by any neighboring diver. We formalize the task as a non-convex, multi-objective graph localization constraint optimization problem. The solution finds the best configuration of the nodes' graph under constraints in the form of upper and lower bounds derived from the inter-connections between the graph nodes/divers. Considering the need to rapidly propagate the SOS information, we flood the network with the SOS packet, while also using rateless coding to leverage information from colliding packets, and to utilize time instances when collisions occur for propagation delay evaluation. Numerical results show a localization accuracy on the order of a few meters, which contributes to quickly locating the diver in distress. Similar results were demonstrated in a controlled experiment in a water tank, and by playback data from a sea experiment for five network topologies.

Keywords: SOS transmission; underwater acoustic communication; underwater acoustic networks; graph localization; underwater localization

1. Introduction

In the last decade, ocean exploration has considerably increased, as has the involvement of human divers performing maintenance work and site surveys. In the harsh underwater environment, these activities pose a great risk to divers' safety. However, due to the constraints of their work, divers may sometimes find themselves in need of assistance when they are far away from their fellow divers. In such cases, modern diving systems include a way to acoustically broadcast an SOS message, e.g., [1]. These messages include the ID of the diver in distress and propagate well in the divers' network. However, without a way for fellow divers to identify the location of the diver in distress, these systems cannot provide immediate help.

In the absence of GPS reception, underwater localization mostly involves measuring the range and/or bearing to the acoustic source. While common solutions use trilateration from multiple anchor nodes or utilize the relative motion between the receiver and transmitter [2], localization of the source of an SOS packet requires a different approach. First, the need for a rapid response does not allow a centralized approach, such as trilateration. Second, due to the irregular motion of scuba divers, especially those working underwater



Citation: Diamant, R.; Francescon, R. A Graph Localization Approach for Underwater Sensor Network to Assist a Diver in Distress. *Sensors* 2021, *21*, 1306. https://doi.org/ 10.3390/s21041306

Academic Editor: Aboelmagd Noureldin

Received: 30 December 2020 Accepted: 9 February 2021 Published: 11 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). for long time periods of several hours, relying on self-navigation means for localizationsuch as inertial sensors, is not practical. Third, since the accurate pre-deployment of beacons is challenging in complicated environments, such as harbors where most underwater works occur, it is impractical to use anchor nodes in the localization solution. Instead, considering the likely sparsity of the divers' network, a graph localization (GL) approach is a better fit.

Solving through GL refers to the construction of a graph from links between sensor nodes and the positioning of the nodes within the graph, by turning and stretching the graph to best fit a utility function under constraints on the inter-connections between the nodes. Many formalized and heuristic solutions, including our preliminary work in [3], are offered for GL with the aim of encompassing ranges for all of the links in the graph. However, for underwater acoustic networks—where time-of-flight (ToF) measurements are often noisy with many outliers and where the network's graph is sparse—this objective may degrade the localization performance. Considering this problem, we propose a novel GL formalization, whose utility not only fits the structure of the formed graph to ToF measurements, but also aims to minimize the impact of outliers while setting constraints on the range between the graph's links. The result is a constraint multi-objective optimization problem that is solved by semi-definite programming. To further manage infeasibilities due to outliers, we present a relaxed convex optimization problem. Considering the need to quickly propagate the SOS message, we flood the SOS packet throughout the network. The expected collisions between the packets are handled through rateless coding. The capability to identify erasure symbol indicators in rateless coding also allows a ToF estimation for colliding packets. This way, no cooperation in scheduling is required between the network's nodes, and the delay from the time of the SOS transmission to the time of the localization decision reached at each node is obtained quickly. Numerical simulations show that the resulting localization accuracy is on the order of a few meters, which is acceptable for the purpose of approaching the diver in distress. The results were obtained by conducting a controlled experiment in a water tank, and using data from a sea experiment for five network topologies.

To summarize, compared to the state-of-the-art, our contribution and novelty are four-fold:

- A graph localization solution that includes upper and lower bounds—not just as regards the constraints of the graph optimization problem (as other methods propose) but also in its utility. As our results show, minimizing the difference between the upper and lower bounds leads to more accurate location identification.
- While previous methods only derive the graph localization problem, in cases where
 no feasible solution is found, our approach also offers a fallback in the form of a
 relaxed optimization problem. This is important since, due to the typical erroneous
 time-of-flight measurement and the sparse graph in underwater acoustic networks,
 rigid constraints may not allow a feasible solution.
- In contrast to current approaches that utilize time-of-flight information for the graph's link only for correctly received packets, our approach also offers a way to extract such information from colliding packets. This is important because, as a result of flooding, for the SOS packets across the network, many collisions are expected. We offer this capability using a novel scheme that derives the time-of-flight from the erasure indicators of a rateless code. Hence, the availability of the localization solution is improved, thereby decreasing the delay from the SOS transmission time to its source localization.
- To the best of our knowledge, ours is the first formalized graph localization approach for underwater acoustic networks. Since underwater acoustic communication has significant differences compared to radio frequency terrestrial networks, this contribution is not just in terms of the considered application, but also requires the modification of the localization solution. This is reflected by the choice of the upper and lower bounds

in the graph localization, in the SOS decoding protocol, and in the expected errors of the time-of-flight estimation. We have included sea trial results to support our claims.

The remainder of this paper is organized as follows. In Section 2, we present the state-of-the-art for underwater localization in the context of communication networks. In Section 3, we introduce our system's model and assumptions. Section 4 describes the details of our proposed SOS localization algorithm. In Section 5, we share results from numerical simulations and from real environments. Conclusions are drawn in Section 6. The main notations used in this paper are summarized in Table 1.

Table 1. List of major notations.

Notations	Explanations			
\mathcal{N}	Set of nodes in the network			
S	ID of diver in distress			
\mathcal{A}	Nodes whose location relative to the assisting node is known			
$\boldsymbol{p}_{i}(i)$	Location of node <i>j</i> relative to that of node <i>i</i>			
$\dot{M}_{n,m}$	shortest pass route from node <i>n</i> to node <i>m</i>			
$U_{n,m}, L_{n,m}$	Upper and lower bound in link (n, m) , respectively.			
$T_{n,m}^{\mathrm{pd}}$	Propagation delay in link (n, m)			
\mathcal{R}_{i}	Route of a packet passed from a source to a node <i>j</i>			
Á	Maximum error of measured inter-node distances			
R	Number of allowed re-transmissions of the same packet			

2. State-of-the-Art for Underwater Network Localization

Underwater acoustic localization is a challenging task: Range estimations are affected by non-line-of-sight reflections, the transmitter–receiver pairs are often not timesynchronized, and the sound speed in water is not always known. Moreover, due to motion, a node hardly obtains range measurements to at least three anchor nodes while remaining at a fixed location. In [2,4,5], comprehensive surveys are presented for localization methods using underwater acoustics. The available methods can be categorized as passive localization, where the node-to-be-localized does not transmit signals, and active localization, where the node cooperates with other nodes via two-way signal transmissions to estimate its range and bearing to beacons through triangulation.

In the context of underwater sensor networks, a main challenge is the lack of a sufficient number of beacons to resolve localization ambiguities. This so-called under-ranked localization problem is solved by either considering already localized nodes as new, second degree, anchors [6]; or by utilizing the nodes' motion. In [7], localization using a single beacon is achieved by tracking the node's range over time. In [8], environmental information is utilized to estimate the location of the node within a map of likelihood values obtained by matching the measured channel to a propagation model. Similarly, reference [9] ranks localization ambiguities by their match to the signal's power spectra. However, performance quickly degrades when the environmental information is even slightly mismatched. For indoor localization, the authors of [10] used side information in the form of detection ranges to resolve the likelihood ambiguity problem, and template matching to a pre-measured signal strength is offered in [11]. However, more leverage can be gained by utilizing the possible relationships among the sensor nodes.

As evaluating the location of the diver in distress involves a sparse network, a solution may be found in GL tools, where nodes reflect vertices and ranges between nodes as non-directed edges. The localization solution is found by shifting and rotating the nodes within the graph so as to optimize a utility function, while setting the measured ranges between the nodes as problem constraints [12]. A complete analysis of the GL problem is given in [13], where the theory behind GL is expounded along with an optimal algorithm for solving the problem.

The most common GL approach is Multidimensional Scaling (MDS) [14], which finds a lower dimensional space that represents the graph by eigenvalue–decomposition to translate pairwise distances between nodes into an *n*-dimensional map. MDS variants allow for the setting of bounds on inter-link ranges and the filtering of solutions to mitigate error propagation from ranging inaccuracies [15–17]. Other approaches formalize the graph localization under proximity constraints in cases where the graph's connections are known [18], by manifold learning [19] and as a weighted least-square solution that ranks the feasibility of the node's relative locations [20]. However, the desire of the above methods to encompass ranges for all of the links in the graph degrades the performance for sparse networks, such as in underwater localization.

When the network graph is sparse and information on distances for some links is missing, the MDS algorithm can be modified to allow under-ranked graphs by applying constraint optimization on the Euclidean distance matrix [21]. The missing data for the distances can also be completed by applying a principle component analysis and decomposing the Euclidian matrix into observable and partly observable parts [22]. An alternative to MDS is the graph embedding approach [23], which makes use of a kernel function to map received signal strength information (RSSI) to the physical coordinates' space. The solution for the optimization problem is efficiently obtained in [18], using a convex simplification solved by Semi-Definite Programming (SDP) under constraints set by the graph connectivity information. To achieve higher accuracy, reference [19] uses manifold learning to calculate the mapping function between the signal strength measures' space and the coordinates' space, but relies on periodic updates from beacons. A modification proposed in [20] solves a regression problem after an optimization of the kernel function used for the initial mapping, and uses either RSSI or ToF. The proposed solutions mostly focus on the GL optimization conversion, and do not directly consider the problem of GL in a highly sparse network, where both beacon information and ranging measurements are scarce. However, regarding the problem of finding the location of the diver in distress in a multi-obstacle-commonly-shallow-water environment, this is the network's likely structure.

3. System Model

Our system includes a set N of N nodes, arranged as a non-directed graph. Due to the many obstacles expected in the diver's working area, the graph's connectivity is expected to be sparse. Each node is equipped with an underwater acoustic modem, which enables the scheduled transmission of packets within a network (using, e.g., the scheduling protocol in [24].

While a more accurate solution for underwater acoustic localization can be obtained by utilizing angle-of-arrival (AoA) measurements in the form of an ultra-short-baseline solution [25], this involves either a rigid hydrophone 3D array, which is hard to employ over the body of a scuba diver, or a 3D vector sensor, which is extremely expensive and is limited to low carrier frequencies. Instead, for a practical solution, we consider an omnidirectionalsingle-transceiver modem. We assume the modems are time-synchronized by an underlying process (e.g., [26]), their hardware/software delays in relaying a packet can be accurately estimated (e.g., [27]), and can estimate the range to the transmitting node by measuring the ToF of a packet and the effective sound speed in the link (using, e.g., [28]). We further assume access to the modem's physical layer, such that reliability measures of decoded symbols are available. These can be, for example, the log-likelihood ratio of a certain symbol to belong to the decoded hypothesis, the symbol's signal-to-noise ratio, or the normalized matched filter's output of the symbol. All of the above assumed features of the acoustic modem are available in the EvoLogics 18-34 software-defined modem [29], which we use in our experiments.

Our algorithm is performed distributively at all the network nodes. Without the loss of generality, at a certain time instance, node $s \in N$ issues an SOS message. This node is referred to as the *SOS source*, i.e., the diver in distress. Our solution works distributively at

each node $i \in \mathcal{N}$ to evaluate the position of node s, $p_s(i)$, within a Cartesian coordination system relative to node i, which, in turn, is set at (0,0). To simplify terms, in the following, we refer to node i as the *assisting node*, but recall that any node in the network that aims to assist the diver in distress would be called an assisting node. To evaluate $p_s(i)$, the assisting node draws from distance estimates between the graph's edges and from range and bearing estimates to its one-hop neighbors (using, e.g., the method in [30]).

Our algorithm avoids using anchor nodes or surface vessels, and the SOS source localization is performed distributively at each diver's modem. Our solution is thus practical for the considered applications of underwater construction works and underwater manned surveys where nodes are always submerged and cannot track their own position. As a result, since errors in time-of-flight measurement are proportional to the travel distance, as are the differences between the actual distance and upper and lower bounds for missing communication links, when the scuba divers are close to each other, the accuracy of the localization solution improves. Relying on depth information provided by the SOS source, we aim for two-dimensional space localization. An accurate and timely evaluation of the SOS source's location will allow the assisting node to promptly respond to the distress call.

4. Proposed SOS Localization Algorithm

4.1. Key Idea

Referring to the illustration in Figure 1, our solution to evaluate the location of the SOS source, relative to that of the assisting node, is based on propagating the SOS packet across the network while estimating the distances along the graph's edges. We then use these distances as constraints to project the distance matrix into a feasible relative geographical map. The distance within the communication link is obtained distributively by each node in the graph—either directly by detecting acoustic transmissions of one-hop neighbor nodes, or indirectly by piggy-backing such information on the relayed SOS packet.



Figure 1. Illustration of system setting.

In contrast to solutions such as [31], which utilize packets' known emission intervals to solve localization ambiguities, we look for the fastest way to propagate the SOS and thus choose to flood the messages in a non-organized fashion. During flooding, upon receiving a packet, each node immediately relays the SOS packet until a timeout occurs. To compensate for the expected packet collisions during this flooding, we use rateless channel coding [32], which also assists in the estimating of ToF for the interfering packets. Our algorithm works on several levels: SOS propagation, propagation delay estimation for incoming packets, and per-node localization of the SOS source. The details of each level are presented below, starting from the final location and down to the SOS propagation.

4.2. Position of the SOS Source

To evaluate $p_s(i)$, we use a matrix of evaluated distances, D(i), as input between the graph's edges as observed by node *i*. For a pair of nodes $n, m \in \mathcal{N}$, entry $D_{n,m}(i) = c \cdot T_{n,m}^{pd}(i)$, where *c* is the sound speed in water and $T_{n,m}^{pd}(i)$ is the ToF as evaluated directly or indirectly (details of how to obtain D(i) are provided below). If node *i* cannot evaluate the delay in a link (n, m), we set $D_{n,m}(i) = 0$.

To estimate $p_s(i)$, $j \in \mathcal{N}$, we offer two alternatives. The first, considered as a benchmark, is the multidimensional scaling (MDS) algorithm, which relies on an eigenvalue decomposition of D(i) [14]. Given D(i), MDS provides a map representation of the graph. To project this map representation into the UTM coordination system relative to the assisting node *i*, we shift the MDS solution to fit the node *i* in coordinates (0,0), and rotate the shifted map to fit the relative coordinates on the evaluated node *a*. This process is explained in detail in the following Section 4.2.2. While MDS provides a feasible solution, it does not allow an initial guess and may therefore fall into a local minimum of its loss function, which may lead to large errors. In addition, MDS does not allow for the utilization of prior information, such as upper and lower bounds on the information in D(i). We therefore offer a second alternative, referred to as *conditional GL* (CGL), whose output is the solution of an optimization problem that takes into account some upper and lower bounds on D(i).

4.2.1. Finding the Coordinate Representation of the Graph

For each node pair (n,m), $n, m \in \mathcal{N}$, the CGL algorithm sets an upper bound and a lower bound for the Euclidean distance, $U_{n,m}(i)$ and $L_{n,m}(i)$, respectively. The former uses the accumulated propagation delay between non-neighbor nodes along the packet's route until node *i*. In contrast, the latter is set according to the maximal delay in the packet's route. That is, $L_{n,m}(i) = c \cdot \max\{T_{p,q}^{pd}\} - \Lambda$, $(p,q) \in M_{n,m}$, where $M_{n,m}$ is the shortest pass route from node *n* to node *m*. Parameter Λ is the maximum error of the estimated distance, and is modeled as a function of the maximum expected delay spread in the channel, the ambiguity of the modem's immanent time delay, and the sound speed in water. Since the lower bound is loose, we introduce a scale parameter, ρ_L , which allows flexibility.

The CGL solution is found by matching the upper and lower bounds with the measured distances in D(i). Formally, let A be a set of nodes whose locations $p_a(i)$, $a \in A$ relative to node i are estimated by the assisting node via range and bearing estimations (Note that A can hold even one node), we obtain the estimated nodes' locations, $\hat{P}(i) = \{\hat{p}_s(i), \dots, \hat{p}_N(i)\}$, by solving the following optimization problem:

$$\hat{P}(i) = \underset{\tilde{P}(i)}{\operatorname{argmin}} \sum_{(n,m)|D_{n,m}>0} \{ [||\tilde{p}_n(i) - \tilde{p}_m(i)||_2 D_{n,m}(i)], |\rho_L \cdot L_{n,m}(i) - U_{n,m}(i)| \}$$
(1a)

s.t.
$$||\tilde{\boldsymbol{p}}_{n}(i) - \tilde{\boldsymbol{p}}_{m}(i)||_{2} \le U_{n,m}(i) \ \forall n, m | D_{n,m} > 0$$
, (1b)

$$||\tilde{\boldsymbol{p}}_{n}(i) - \tilde{\boldsymbol{p}}_{m}(i)||_{2} \ge \rho_{L} \cdot L_{n,m}(i) \ \forall n, m | D_{n,m} > 0.$$

$$(1c)$$

$$L_{n,m}(i) \le \rho_L \cdot L_{n,m}(i) \le U_{n,m}(i) \forall n, m.$$
(1d)

$$\boldsymbol{p}_n(i) - \Lambda \le \tilde{\boldsymbol{p}}_a(i) \le \boldsymbol{p}_a(i) + \Lambda, \ a \in \mathcal{A} \ . \tag{1e}$$

$$\tilde{\boldsymbol{p}}_i(i) = (0,0) \ . \tag{1f}$$

Problem (1) is multi-objective with two related objectives. The first aims to minimize the difference between the Euclidean distances of the evaluated locations and the measured distances, and the second guides the solution that minimizes the difference between the lower bound and upper bounds. As a result, the solution would not only seek for a minima in terms of the Euclidean difference between the measured delays and the SOS source location, but would also lean towards a solution with tight constraints, thereby reducing the impact of outliers in the time-of-flight measurements. Note that the objective function avoids the use of node pairs with no connecting route. The problem is constraint, such that the upper and lower bounds on the estimated coordinates fit the evaluated distance matrix D(i) (Constraint (1b) and (1c)), and to limit the range for the lower bound scale factor ρ_L (Constraint (1d)). Constraint (1e) fixes the estimated coordinates $\tilde{p}_a(i)$, $a \in A$ to their known relative locations, while providing leverage to the size of the error ambiguity Λ to allow a feasible solution. Finally, constraint (1f) fixes the coordinates of the assisting node to the location of the assisting node at (0,0).

Note that, in the case of $L_{n,m}(i) = 0$, problem (1) is convex and can be solved via a combination of semi-definite programming (cf. [18]) and multi-objective optimization (e.g., [33]). However, for a non-zero lower bound, (1) is non-convex and its convergence to a global minimum is not guaranteed. Considering this problem, we use the solution of the benchmark MDS algorithm as an initial guess, and solve (1) via quadratic programming. However, in cases where the MDS provides a bad initial estimate or when the estimated distance matrix D(i) includes a large error, the algorithm may not find a feasible solution. We therefore also consider a more relaxed version of (1) that avoids imposing the locations of the nodes in A:

$$\hat{\boldsymbol{P}}(i) = \underset{\tilde{\boldsymbol{P}}(i)}{\operatorname{argmin}} \sum_{(n,m)|D_{n,m}>0} [||\tilde{\boldsymbol{p}}_{n}(i) - \tilde{\boldsymbol{p}}_{m}(i)||_{2} D_{n,m}(i)] + |\rho_{L} \cdot L_{n,m}(i) - U_{n,m}(i)|$$
(2a)

s.t.
$$||\tilde{p}_{n}(i) - \tilde{p}_{m}(i)||_{2} \le U_{n,m}(i) \ \forall n, m | D_{n,m} > 0$$
, (2b)

$$||\tilde{\boldsymbol{p}}_{n}(i) - \tilde{\boldsymbol{p}}_{m}(i)||_{2} \ge \rho_{L} \cdot L_{n,m}(i) \ \forall n, m | D_{n,m} > 0 \ . \tag{2c}$$

$$L_{n,m}(i) \le \rho_L \cdot L_{n,m}(i) \le U_{n,m}(i) \forall n, m,$$
(2d)

and directly uses the MDS output as an initial solution, i.e., its non-shifted map representation. Problem (2) is distinctively easier than (1), and is a uni-objective constraint convex optimization problem that holds a global minima. We obtain its solution by the simplex method. We distinguish problem (1) from problem (2) by referring to the former as *direct optimization* and to the latter as *relaxed optimization*.

4.2.2. Matching of Coordinates

Both the initial MDS solution for direct optimization (1) and the solution for relaxed optimization (2) require matching the resulting map representation to the coordination system of the assisting node. We now discuss the process of this correction.

To project the solution of the MDS or of the relaxed optimization into a coordination system relative to the assisting node *i*, we perform shifting through

$$\boldsymbol{\beta} = \boldsymbol{p}_i(i) - \boldsymbol{\hat{p}}_i(i) , \qquad (3)$$

and the estimated coordinates of a node $j \in \mathcal{N}$ are shifted by

$$\hat{\boldsymbol{p}}_{j}^{\mathrm{s}}(i) = \boldsymbol{p}^{\mathrm{init}}{}_{j}(i) - \boldsymbol{\beta} , \qquad (4)$$

where $p^{\text{init}}_{j}(i)$ is either the output of the MDS (to solve (1)) or $\hat{p}_{j}(i)$ from (2). Next, we rotate the shifted coordinates to fit the position of the nodes in A. In order to achieve this, we estimate a rotation angle, θ , which minimizes the errors between the evaluated locations of the nodes in A and their relative locations, as estimated directly by the assisting node. The evaluation of θ is performed by

$$\theta^* = \mathcal{F}\{\varepsilon(a)\}\tag{5a}$$

s.t.
$$\varepsilon(a) = \left\{ \left\| \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \hat{p}_{a}^{s,x}(i) \\ \hat{p}_{a}^{s,y}(i) \end{bmatrix} - \begin{bmatrix} p_{a}^{x}(i) \\ p_{a}^{y}(i) \end{bmatrix} \right\|_{2} \right\},$$
 (5b)

where $(\hat{p}_a^{s,x}(i), \hat{p}_a^{s,y}(i))$, and $(p_a^x(i), p_a^y(i))$ are the *x* and *y* coordinates of the shifted evaluated solution (4), and the *x* and *y* coordinates known relative positions of the nodes in \mathcal{A} , respectively.

We consider several options for function \mathcal{F} :

- Summation: $\mathcal{F} = \operatorname{argmin} \sum_{a \in \mathcal{A}} \varepsilon(a)$.
- $\operatorname{Minimize:}_{\theta} \mathcal{F} = \operatorname{argmin}_{\theta} \varepsilon(a), \ \forall a \in \mathcal{A}.$
- Average: replace (5a) with $\theta^*(a) = \arg_{\theta} \min\{\varepsilon(a)\}$, and set $\theta^* = \sum \theta^*(a) / |\mathcal{A}|, \forall a \in \mathcal{A}$.

The 'summation' approach gives equal weight to all nodes in \mathcal{A} , while the 'minimize' approach performs rotation considering only the node in \mathcal{A} that leads to the minimal error. The 'average' approach simply averages all the rotation angles that best fit each of the nodes in A. Finally, each node $j \in N$ is rotated by

$$\begin{bmatrix} \hat{\boldsymbol{p}}_{j}^{\mathbf{r},\mathsf{x}}(i)\\ \hat{\boldsymbol{p}}_{j}^{\mathbf{r},\mathsf{y}}(i) \end{bmatrix} = \begin{bmatrix} \cos\theta^{*} & -\sin\theta^{*}\\ \sin\theta^{*} & \cos\theta^{*} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{p}}_{j}^{\mathbf{s},\mathsf{x}}(i)\\ \hat{\boldsymbol{p}}_{j}^{\mathbf{s},\mathsf{y}}(i) \end{bmatrix}.$$
(6)

When solving through direct optimization, we plug the solution $\left[\hat{p}_{i}^{r,x}(i), \hat{p}_{i}^{r,y}(i)\right]^{T}$ at the input to (1). Note that, if solving through the relaxed optimization approach, the solution of (6) is the final estimate.

4.3. SOS Propagation

Our process starts with the transmission of the SOS messages by the SOS source. Considering packet failures, we let the SOS source transmit *R* SOS packets. Since the network's graph is expected to be sparse, there is a need to relay the SOS message between the divers/nodes. Respecting the need for a prompt response to the SOS call, we adopt the flooding approach, so that a node successfully receiving the SOS packet immediately relays it further. Similar to the SOS transmission, each relayed packet is transmitted R times. However, to limit packet collisions, a node continues flooding an incoming packet only if the packet has not already visited all of its one-hop neighbor nodes. The latter requires piggy-backing the list of receiving nodes on top of each forwarded packet. To further mitigate packet collisions, the *R* re-transmissions from each node *i* are spaced by a predetermined backoff Δ_i . These backoff values are known to all nodes and can be chosen to reduce collisions as discussed in [34].

As illustrated in Figure 2, the relayed SOS packet includes information on the ID of the SOS source and the transmission time of the first SOS packet; the propagation time of the packet, considering the hardware/software delays in the relay node; and the packet's route. While the latter is used to control the flooding procedure, together with the propagation time, it also gives an estimate of the time delay from the receiving node to the SOS source. This route information includes the ID of the relaying node as well as the number of the transmitted packets. For example, the route of a packet that passed through nodes *s*, *n*, *p* and *j* (in that order) is coded as $\mathcal{R}_j = \{[s, r_s]^T, [n, r_n]^T, [p, r_p]^T, [j, r_j]^T\}$ with two rows: $\mathcal{R}_{j,1}$ and $\mathcal{R}_{j,2}$, where, for a node $m \in \mathcal{R}_{j,1}$, $r_m \in \mathcal{R}_{j,2}$ is a scalar within the range $\{1, \ldots, R\}$ (recall R is the number of re-transmissions) determined by the number of re-transmissions of the same packet already made by node *m*. As we will show further on, the time delay and route information are also used to evaluate the time delay between the graph's vertexes. To allow a separate time delay evaluation from each packet, the packets are numbered. Finally, considering the expected packet collisions, each packet is channel-coded and includes a cyclic redundancy check (CRC) to determine if it has been successfully received.



Figure 2. Structure of transmitted packets.

4.3.1. Estimation of the Propagation Delay

The propagation delay in the link of an incoming SOS packet from a node j is determined by the packet's transmission time and the measured ToF. To significantly reduce the packet's size, rather than including the transmission time as part of the packet, we obtain the information on the propagation time in the link from node j based on the SOS transmission time, the accumulated time delay, and the route passed by the packet as explained in the following.

Let T_0 and T_i be the SOS transmission time stamp (piggy-backed over the incoming packet, see Figure 2) and the time node *i* received the packet from *j*, respectively. In addition, denote T_j^d as the accumulated delay up to node *j*, and let $T^{hardware}$ be the known hardware time delay. The propagation delay between *i* and *j* is estimated by

$$T_{i,j}^{\text{pd}} = T_i - T_j^{\text{d}} - \sum_{n \in \mathcal{R}_{j,1}} \left(r_n \cdot \Delta_n + T^{\text{hardware}} \right) - T_0 .$$
⁽⁷⁾

The time delay from node *i* to the SOS source *s* is upper bounded by

$$T_{i,s}^{\rm pd} = T_j^{\rm d} + T_{i,j}^{\rm pd} ,$$
 (8)

which is also coded in the outgoing packet from *i* as the new time delay value. That is, $T_i^d = T_{is}^{pd}$.

Using (7), node *i* can form a graph of propagation delay information between any node pairs that appear in packets received by *i*. However, due to the many packet collisions, not all the packets may be received. As a result, some propagation delay information may be missing, and the delay graph would not be full. To complete the missing information, following [14], we run a shortest path algorithm (e.g., the Dijkstra algorithm) and obtain an upper bound for the missing delay by accumulating the delay along the shortest route. Let $M_{j,i}$ be the shortest pass route from node *j* to node *i*, arranged by connecting node pairs. The missing delay information is obtained by

$$T_{i,j}^{\rm pd} = \sum_{(p,q)\in M_{j,i}} T_{p,q}^{\rm pd} .$$
(9)

Note that, in some cases, route $M_{j,i}$ cannot be found, for example, if no packet that passed through *j* was received by *i*. In this case, we eliminate node *j* from the graph constructed by *i*.

4.3.2. SOS Decoding

Due to the flooding of the SOS packets, we expect many packet collisions. We observe that, although the flooded packets carry different information of the accumulated delay and routing path, the SOS information part is always the same. Considering this, we protect the SOS part of the packet by means of rateless coding [35]. In this way, although the decoding of single packets may fail, each incoming packet brings new information that

is accumulated until the packet has been successfully decoded. To this end, we form a large coding matrix and allocate each node with columns to perform the rateless coding. To support the R re-transmissions made by each node, this list of columns allows each node to perform R rateless coding. That is, upon transmitting a packet, the transmitter codes by using its allocated columns from the coding matrix for the transmission of the r out of R similar packets.

By setting the above list of columns prior to the network's deployment, we make sure it is known to all nodes, and the receiving node can add the received symbols to that already received from each transmitting node. Then, by identifying the IDs of the nodes transmitting the received packets, the receiver tries to decode the accumulated symbols using the columns of the coding matrix allocated to these transmitting nodes. As illustrated in Figure 2, to determine the ID of the transmitting nodes, and to identify the transmission number from the *R* possible ones, we include a signature sequence at the head and tail of each packet. This sequence is unique for each transmitter and for each of its *R* re-transmissions. Assuming only two packets collide at any given moment, at least one of the packets' heads and tails is properly received, thereby allowing the identification of the transmission properties.

4.3.3. ToA Estimation

Measuring the ToA, T_i , is trivial when the packet is properly decoded. We can also match T_i and the transmitting node and set it according to the arrival of the head sequence, even when the packet fails. However, using rateless coding, we can also identify T_i when the beginning of the packet collides with another packet. This is possible when the tail sequence has been properly decoded, such that the origin of the packet is identified. If the decoding fails and the beginning of the packet is corrupted, we identify the ToA based on the erasure symbols. In particular, we determine the arrival time as the beginning of a sequence of such erasure symbols.

Erasures are symbols identified by the process of rateless decoding as erroneous [35] for those symbols whose log-likelihood ratio (LLR) is low. Identifying the time instance when the two packets collide is performed in three steps:

- Smoothing: Removing erasure symbols which do not appear as a sequence. In the absence of information about the distribution function of the symbols' LLR, the smoothing operation is performed using a median filter of degree 3.
- 2. **Thresholding:** In the case where after the smoothing of individual erasures there exists only one sequence of erasures, we set the arrival time according to the sequence's beginning time before smoothing. Otherwise, we further remove sequences of small numbers of erasures by setting a threshold on the number of erasure symbols. The threshold is set according to the expected symbol error rate (SER) and the measured signal-to-noise ratio (SNR). If an SNR measurement is not available (for example, when a commercial underwater modem is used), the SER is set empirically while removing the number of erasure symbols due to packet collisions. That is,

$$SER = \frac{S^{\text{erasure,smooth}}}{2S^{\text{packet}} - S^{\text{erasure}}} , \qquad (10)$$

where *S*^{erasure,smooth} is the number of smoothed erasures, *S*^{erasure} is the number of erasure symbols after the smoothing procedure, and *S*^{packet} is the number of transmitted symbols in a packet. We then set a threshold, Th, to detect a sequence such that

$$(SER)^{1h} < \beta , \qquad (11)$$

where β is a user-defined probability to receive consecutive Th erasure symbols. We then identify the beginning of the packet collisions and the time instance when the colliding packet arrived as the start of the first sequence of at least Th consecutive erasure symbols.

5. Performance Evaluation

We will now discuss the performance of our SOS localization algorithm. We measure the performance in terms of the algorithm's convergence time, ρ_{converge} , and the Euclidean error of the evaluated SOS source location, ρ_{error} . The convergence time is defined as the time elapsed between the transmission of the first SOS packet and the time until the last assisting node is able to estimate the location of the SOS source. To evaluate the performance of our algorithm, we conducted both numerical simulations and tests using underwater modems. The simulations explore the performance for various graph constellations, while tests in a water tank and in the field demonstrate the feasibility of the algorithm for use with real hardware and under harsh conditions.

5.1. Numerical Simulations

Our simulations involve five nodes, which is an appropriate number for divers performing maintenance work. Results are expected to improve as the number of nodes increases. We perform a Monte Carlo set of 1000 random placements of nodes and obstacles. In each run, the nodes are randomly placed in a square area of $2000 \times 2000 \text{ m}^2$. The communication range is set for 1500 m such that the graph is not fully connected, but no node is entirely disconnected. To further increase the graph's sparsity, the square area includes horizontal and vertical obstacles, uniformly randomly placed in each simulation run with lengths uniformly distributed at [50, 100] m, such that communication is possible only if an acoustic line-of-sight exists between the transmitter and the receiver. We consider the case of one and two nodes in A, whose relative locations are evaluated by the assisting node through range and bearing measurements. In each simulation run, we choose the identity of the nodes in A uniformly randomly.

We compare the performance of five methods: (1) our solution to (1) referred to as *ULB*; (2) our solution to (2) referred to as *UB*; (3) the non-classical MDS algorithm that allows missing links in the graph by an iterative solution starting from a random location; (4) a scheme presented in [21] termed *RBC-MDS*; and (5) a scheme presented in [22] termed *RPCA-MDS*. We choose the RBC-MDS as a benchmark since, like our approach, it handles missing links in a GL solution. Complementing this is the RPCA-MDS approach, which also utilizes partial information in the graph. Naturally, all schemes receive the same information regarding the links' delays. The simulation results are analyzed by a cumulative density function (CDF). To obtain sufficient statistics, the 1000 simulated placements are divided into 10 points in the CDF.

Each simulation run starts with the transmission of an SOS packet of total duration 100 ms, including a synchronization signal of duration 50 ms. The SOS packet includes 32 bits and is protected by a (4,7) Reed–Solomon code, while relayed packets include 40 bits, further coded using a (5,7) Reed–Solomon code. Both packets include a CRC code of 9 bits. We consider a worst case scenario, where the corresponding symbols of all colliding packets are corrupted.

We start by exploring the delay in propagating the SOS packet across the divers' network. In Figure 3, we show the CDF of the convergence time ρ_{converge} . The results obtained are less than 10 s in 95% of the cases, and are quite robust, regardless of the graph topology. These results demonstrate the efficiency of our algorithm in rapidly flooding the SOS packet across the divers' network, even in the presence of strong interference. Figure 3 also includes results when no symbol erasure information is used. The results, termed *No-Erasures*, show a significant degradation in performance compared to using erasures in the process of estimating the time-of-flight. This is because many packets collide when the SOS packet is flooded; however, using erasure indicators, we can still also evaluate the time-of-flight for these colliding packets.



Figure 3. Empirical CDF of convergence time ρ_{converge} . The average result is 8.5 s and 10.4 s for using and not using erasure information, respectively.

Next, we consider the performance in terms of the Euclidean error, ρ_{error} . The CDF of ρ_{error} is shown in Figure 4 when set A holds only a single node. We report that the average result for the use of both upper and lower bounds is 36.4 m; the average result for the use of only upper bounds is 160.9 m; the average result for MDS is 542.5 m; the average result for RBC-MDS is 442.6; and the average result for RPCA-MDS is 155.9. We observe that, due to its use of both direct and partial information, the results of RPCA-MDS are better than those of MDS and RBC-MDS, and are somewhat similar to UB. However, the improved results of ULB demonstrate the clear advantage of using conditional localization for the task of SOS localization. Observing the small variation of the results observed for conditional localization, we conclude that this method is robust in terms of the graph topology. Comparing the results of both types of conditional localization, a clear advantage of using both the upper and lower bounds is observed, at the cost of complexity.



Figure 4. Empirical CDF of ρ_{error} . Set \mathcal{A} holds only one node.

To comment on the proper way to localize the SOS source when set A holds two nodes of estimated relative locations, in Figure 5, we show the CDF of ρ_{error} for the solution of the direct optimization problem (1) (*Direct*), and for the solution of the relaxed optimization problem (2) (*Relaxed*) using the three rotation options described in Section 4.2.2. We conclude that the Relaxed solution is always better than the Direct one. This is because the performance of the Direct solution relies on the performance of the rotated MDS solution, which-as observed by the results of Figure 4—is, in fact, inaccurate. Comparing the three options to define (5a), we observe that the performance of the 'Summation' approach is far better than that of the 'Average' approach, and slightly better than the 'Minimize' approach. This is mainly because the 'Summation' approach accounts for all nodes in A, and we used accurate relative locations of the nodes in A.



Figure 5. Empirical CDF of ρ_{error} for the solution of (1), and for the solution of (2) using the three rotation options for setting \mathcal{F} in (5a). Set \mathcal{A} includes two nodes.

We now compare the performance of the ULB conditional localization method when set A holds one or two nodes. The results are presented in Figure 6 in terms of the CDF of ρ_{error} . As expected, the results improve when set A includes two nodes, with an average gain of 11.3 m. However, this requires the assisting node to know its location, relative to two nodes, which is not always guaranteed. Still, observing the absolute Euclidean error, we conclude that the ULB method performs well and provides the localization capability of a diver in distress with sufficient accuracy when set A includes only one node.



Figure 6. Empirical CDF of ρ_{error} . Set A holds one and two nodes. \mathcal{F} in (5a) is defined according to the 'Summation' approach.

5.2. Tests in Water Tank

To test the feasibility of our algorithm with real hardware, we set up a proof-of-concept test in a water tank of size 60×70 cm. To reduce reverberations, the walls and floor of the tank were covered with sound-absorbing material. Using the T2Mo tool [36], we accessed the physical layer of four EvoLogics modems to employ our algorithm. As is depicted in Figure 7a, the black modem was set as the SOS source. While the modems were stationed very close to each other, due to the strong interference in the tank, the formed network graph was sparse. In particular, as illustrated in Figure 7b, two topologies were tested, and five runs were performed for each topology.



(a) Picture of the setting of the water tank demonstration



(**b**) Topologies tested in the water tank. Node 1 is the SOS source. Dashed lines represent active links.

Figure 7. Tank experiment.

Since the tank dimensions are small, performance is only measured in terms of the delay until the SOS information is received (SOS detection), and in terms of the delay until

a node receives enough information to perform localization of the SOS source, i.e., the algorithm convergence time (*Data collection*). The results are presented in Table 2. We observe quite small delays. In fact, the main delays are due to the modems' processing time, and to packet collisions. The mean delay for having the SOS information is 3.25 s, and the average convergence time is 24.53 s. Due to the long propagation delay in the acoustic link, these delays are expected to grow for real underwater channels; at the same time, less collisions are expected. This was demonstrated using data from a sea experiment, as discussed in the following.

	Node 2	Node 3	Node 4
SOS detection (s)	2.147	2.356	0.6
Data collection (s)	26.38	30.08	25.76
SOS detection (s)	0.35	5.24	9.76
Data collection (s)	25.16	19.91	19.89
	SOS detection (s) Data collection (s) SOS detection (s) Data collection (s)	Node 2SOS detection (s)2.147Data collection (s)26.38SOS detection (s)0.35Data collection (s)25.16	Node 2 Node 3 SOS detection (s) 2.147 2.356 Data collection (s) 26.38 30.08 SOS detection (s) 0.35 5.24 Data collection (s) 25.16 19.91

Table 2. Tank test results.

5.3. Data from a Sea Experiment

To demonstrate the performance of our approach in realistic scenarios, we examined playback results from a sea experiment performed in May 2009 in the Haifa harbor, Israel, for an underwater acoustic network formed by four boats creating five different sparse network topologies, as illustrated in Figure 8. As described in [37], the playback is performed by sending the SOS packet and its relays through the channels recorded in the experiment, and by considering the recorded delays in the nodes and in the communication links. Using this type of data-augmented approach, we take into account both real delays in the channel and distortions caused by the ambient noise and the multipath channel. The data from this experiment are shared in [38].



Figure 8. Map of the sea experiment (picture taken from Google Maps).

We test performance in terms of the delay from an SOS transmission until convergence of the assisting nodes, and by the range and bearing accuracy averaged for the choice of the SOS source. The results of ULB are presented in Table 3 and are compared against the MDS, RBC-MDS, and RPCA-MDS benchmarks. We observe that, mostly because no limitations are set for MDS, MDS has a slight advantage over the other schemes in terms of the convergence time, with an average convergence time of 11.6 s vs. 12.5 s for our ULB scheme. In addition, for the fully connected network in Topology 1, MDS obtains the best performance. However, similar to the simulation results, when the topology is sparse, MDS performance degrades significantly compared to the other approaches, while the performance of RPCA-MDS far exceeds it. Still, in terms of localization accuracy, for all sparse topologies, ULB achieves the best performance, with an average localization error across topologies of 44.8 m. We note that these results of CGL are of the same order as those in our simulations, which reflects CGL's ability to also perform well in realistic channels.

Topology	Metric	CGL (ULB)	MDS	RBC-MDS	RPCA-MDS
Topology 1	Delay (s)	7.8	7.5	7.7	7.9
	Accuracy [m]	16	13	15	16
Topology 2	Delay (s)	11.3	10.8	11	11.5
	Accuracy [m]	22	98	45	32
Topology 3	Delay (s)	9.2	8.6	8.9	9.2
	Accuracy [m]	45	173	61	56
Topology 4	Delay (s)	16.1	14.6	15.2	16.3
	Accuracy [m]	63	221	87	73
Topology 5	Delay (s)	18.2	16.5	17.5	18.2
	Accuracy [m]	78	215	117	101

Table 3. Sea experiment results for playback data from [38]. Five topologies are analyzed.

6. Conclusions

In this paper, we suggested an efficient distributed GL algorithm for estimating the relative location of a diver in distress, to enable timely assistance provided by fellow divers. For each possible assisting diver, our GL uses propagation delay measurements from the received or forwarded SOS packets, and solves an optimization problem for the projection of inter-divers' distances into locations, while constraining the solution by upper and lower bounding the diver-to-diver distances. Using the location of the assisting diver as a pivot, we shift and rotate the resulting position of the SOS source to find its location, relative to that of the assisting node. Our approach does not require the use of any anchor nodes of known locations, and requires the direct measurement of only bearing and range to one single node in the network. Considering the expected sparsity of the divers' network graph, we flood the SOS packet to quickly propagate the SOS information, and use rateless coding to combat packet collisions. These collisions are, in fact, utilized to determine the timeof-flight of packets according to the start of a sequence of erroneous symbols. Numerical simulations show that, without using any anchor nodes, our method evaluates the location of the SOS source with an accuracy of the order of a few meters. Results from a controlled experiment in a water tank, and data used from a sea experiment demonstrate the system's capability to localize a diver in distress in a real environment using actual underwater modems. Future research will include the investigation of further GL algorithms.

Author Contributions: Conceptualization, R.D.; methodology, R.D.; software, R.D. and R.F.; validation, R.D. and R.F.; formal analysis, R.D. and R.F.; data curation, R.D.; writing, R.D.; visualization, R.D. and R.F.; project administration, R.D.; funding acquisition, R.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the MOST-BMBF German-Israeli Cooperation in Marine Sciences 2018-2020 (Grant # 3-16573), and by the MOST action for Agriculture, Environment, and Water for the year 2019 (Grant # 3-16728).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

- 1. DiveAlert System. Available online: http://www.divealert.com (accessed on 18 November 2016).
- Tuna, G.; Gungor, V. A survey on deployment techniques, localization algorithms, and research challenges for underwater acoustic sensor networks. *Int. J. Commun. Syst.* 2017, 30, e3350. [CrossRef]
- 3. Diamant, R.; Francesco, R.; Zorzi, M. A graph localization approach to assist a diver-in-distress. In Proceedings of the Workshop on Positioning, Navigation and Communications (WPNC), Bremen, Germany, 25–26 October 2017; pp. 1–6. [CrossRef]
- Tan, H.; Diamant, R.; Seah, W.; Waldmeyer, M. A survey of techniques and challenges in underwater localization. *Ocean Eng.* 2011, *38*, 1663–1676. [CrossRef]
- Islam, T.; Park, S.H. A Comprehensive Survey of the Recently Proposed Localization Protocols for Underwater Sensor Networks. *IEEE Access* 2020, *8*, 179224–179243. [CrossRef]
- 6. Han, G.; Zhang, C.; Liu, T.; Shu, L. MANCL: A multi-anchor nodes collaborative localization algorithm for underwater acoustic sensor networks. *Wirel. Commun. Mob. Comput.* **2016**, *16*, 682–702. [CrossRef]
- Lin, Y.; Tao, H.; Tu, Y.; Liu, T. A node self-localization algorithm with a mobile anchor node in underwater acoustic sensor networks. *IEEE Access* 2019, 7, 43773–43780. [CrossRef]
- 8. Dubrovinskaya, E.; Casari, P.; Diamant, R. Bathymetry-aided underwater acoustic localization using a single passive receiver. *J. Acoust. Soc. Am.* **2019**, *146*, 4774–4789. [CrossRef]
- 9. Gemba, K.L.; Hodgkiss, W.S.; Gerstoft, P. Adaptive and compressive matched field processing. J. Acoust. Soc. Am. 2017, 141, 92–103. [CrossRef]
- 10. Nevat, I.; Peters, G.W.; Avnit, K.; Septier, F.; Clavier, L. Location of things: Geospatial tagging for IoT using time-of-arrival. *IEEE Trans. Signal Inf. Process. Netw.* **2016**, *2*, 174–185. [CrossRef]
- 11. Mokhtari, G.; Anvari-Moghaddam, A.; Zhang, Q.; Karunanithi, M. Multi-residential activity labelling in smart homes with wearable tags using BLE technology. *Sensors* **2018**, *18*, 908. [CrossRef]
- 12. Rai, S.; Varma, S. Localization in wireless sensor networks using rigid graphs: A review. *Wirel. Pers. Commun.* 2017, *96*, 4467–4484. [CrossRef]
- 13. Aspnes, J.; Eren, T.; Goldenberg, D.K.; Morse, A.S.; Whiteley, W.; Yang, Y.R.; Anderson, B.D.O.; Belhumeur, P.N. A theory of network localization. *IEEE Trans. Mob. Comput.* 2006, *5*, 1663–1678. [CrossRef]
- Shang, Y.; Ruml, W.; Zhang, Y.; Fromherz, M.P.J. Localization from Mere Connectivity. In Proceedings of the 4th ACM International Symposium on Mobile Ad-Hoc Networking & Computing, Annapolis, MD, USA, 1–3 June 2003; ACM: New York, NY, USA, 2003; pp. 201–212.
- 15. Shang, Y.; Ruml, W. Improved MDS-Based Localization. In Proceedings of the Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM), Hong Kong, China, 7–11 March 2004; Volume 4, pp. 2640–2651.
- 16. Vivekanandan, V.; Wong, V.W. Ordinal MDS-Based Localization for Wireless Sensor Networks. *Int. J. Sens. Netw.* 2006, 1, 169–178. [CrossRef]
- 17. Wang, C.; Jiming, C.; Youxian, S.; Shen, X. Wireless Sensor Network Localization with Isomap. In Proceedings of the International Conference on Communications (ICC), Dresden, Germany, 14–18 June 2009; pp. 1–5.
- Doherty, L.; El Ghaoui, L. Convex Position Estimation in Wireless Sensor Networks. In Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOMM), Anchorage, AK, USA, 22–26 April 2001; IEEE: Las Vegas, NV, USA, 2001; pp. 1655–1663.
- 19. Yang, B.; Xu, J.; Yang, J.; Li, M. Localization algorithm in wireless sensor networks based on semi-supervised manifold learning and its application. *Clust. Comput.* **2010**, *13*, 435–446. [CrossRef]
- 20. Chen, J.; Wang, C.; Sun, Y.; Shen, X.S. Semi-supervised Laplacian regularized least squares algorithm for localization in wireless sensor networks. *Comput. Netw.* 2011, 55, 2481–2491. [CrossRef]
- 21. Dokmanic, I.; Parhizkar, R.; Ranieri, J.; Vetterli, M. Euclidean distance matrices: Essential theory, algorithms, and applications. *IEEE Signal Process. Mag.* 2015, *32*, 12–30. [CrossRef]
- Liu, D.; Mansour, H.; Boufounos, P.; Kamilov, U.S. Robust Sensor Localization Based on Euclidean Distance Matrix. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Valencia, Spain, 22–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 7998–8001.
- Wang, C.; Chen, J.; Sun, Y.; Shen, X. A graph embedding method for wireless sensor networks localization. In Proceedings of the Global Telecommunications Conference (GLOBECOMM), Honolulu, HI, USA, 30 November–4 December 2009; IEEE: Taipei, Taiwan, 2009; pp. 1–6.
- 24. Diamant, R.; Shirazi, G.N.; Lampe, L. Robust Spatial Reuse Scheduling in Underwater Acoustic Communication Networks. *IEEE J. Ocean. Eng.* 2014, 39, 32–46. [CrossRef]
- Rypkema, N.R.; Fischell, E.M.; Schmidt, H. One-way travel-time inverted ultra-short baseline localization for low-cost autonomous underwater vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4920–4926. [CrossRef]

- Gong, Z.; Li, C.; Jiang, F. AUV-aided joint localization and time synchronization for underwater acoustic sensor networks. *IEEE Signal Process. Lett.* 2018, 25, 477–481. [CrossRef]
- Kebkal, O.; Komar, M.; Kebkal, K. D-MAC: Hybrid media access control for underwater acoustic sensor networks. In Proceedings of the 2010 IEEE International Conference on Communications Workshops (ICC), Capetown, South Africa, 23–27 May 2010; pp. 1–5.
- 28. Diamant, R.; Tan, H.P.; Lampe, L. LOS and NLOS classification for underwater acoustic localization. *IEEE Trans. Mob. Comput.* **2014**, *13*, 311–323. [CrossRef]
- Dol, H.; Casari, P.; Zwan, T.V.D.; Otnes, R. Software-defined underwater acoustic modems: Historical review and the NILUS approach. *IEEE J. Ocean. Eng.* 2016, 42, 722–737. [CrossRef]
- Zhou, M.; Zhong, Z.; Fang, X. Sensor-Target Geometry for Hybrid Bearing/Range Underwater Localization. IFAC Proc. Vol. 2013, 46, 724–729. [CrossRef]
- Schott, D.J.; Faisal, M.; Hoeflinger, F.; Reindl, L.M.; Andreú, J.B.; Schindelhauer, C. Underwater localization utilizing a modified acoustic indoor tracking system. In Proceedings of the IEEE International Conference on Underwater System Technology: Theory and Applications (USYS), Kuala Lumpur, Malaysia, 18–20 December 2017; pp. 1–5. [CrossRef]
- Luby, M. LT codes. In Proceedings of the Symposium on Foundations of Computer Science, Vancouver, BC, Canada, 19 November 2002; pp. 271–280.
- Miettinen, K. Nonlinear Multiobjective Optimization; International Series in Operations Research and Management Science; Springer: Berlin/Heidelberg, Germany, 1999; Volume 12.
- Diamant, R.; Francescon, R.; Zorzi, M. Topology-Efficient Discovery: A Topology Discovery Algorithm for Underwater Acoustic Networks. *IEEE J. Ocean. Eng.* 2018, 43, 1200–1214. [CrossRef]
- 35. Diamant, R.; Lampe, L. Adaptive error-correction coding scheme for underwater acoustic communication networks. *IEEE J. Ocean. Eng.* **2015**, *40*, 104–114. [CrossRef]
- Campagnaro, F.; Francescon, R.; Guerra, F.; Favaro, F.; Casari, P.; Diamant, R.; Zorzi, M. The DESERT underwater framework v2: Improved capabilities and extension tools. In Proceedings of the Underwater Communications and Networking Conference (UComms), La Spezia, Italy, 30 August–1 September 2016; pp. 1–5.
- 37. Casari, P.; Campagnaro, F.; Dubrovinskaya, E.; Francescon, R.; Dagan, A.; Dahan, S.; Zorzi, M.; Diamant, R. ASUNA: A Topology Data Set for Underwater Network Emulation. *IEEE J. Ocean. Eng.* **2020**, *46*, 307–318. [CrossRef]
- 38. Diamant, R. Data from the Sea Experiment. Available online: https://sites.google.com/marsci.haifa.ac.il/asuna/ (accessed on 26 October 2020).