A Graph-based Approach to the Initial Guess of UWB Anchor Self-Calibration

Cedric Pradalier, Antoine Richard, Pete Schroepfer

A Graph-based Approach to the Initial Guess of UWB Anchor Self-Calibration

Cédric Pradalier[1,2], Pete Schroepfer[1,3], Antoine Richard[1,3]

***Abstract*— This article considers the challenge of recovering an initial topology of a mesh of devices based on noisy and incomplete measurements of their inter-distances. In comparison to earlier approaches, this paper provides a first guess of the topology, as opposed to recovering a precise geometry of each device. Using a graph-based model of the problem and properties of the 2D and 3D simplices, we show: (a) a topologically correct solution can be recovered when the solution is unique; (b) we can identify which vertices cannot be recovered and why, using a graph structure; and (c) the challenges due to local symmetries in making the solution non-unique. Results in simulation and with a mesh of UWB anchors demonstrate the applicability of the approach.**

I. INTRODUCTION

The advent of Ultra-Wide Band (UWB) ranging devices has led to a wealth of articles describing their use in the context of robotic localization, i.e. a *robot or mobile device* can measure its range to the anchors and infer its localization assuming the robot knows the 3D location of the *anchor* devices ("anchor(s)"). While determining the 3D location of the anchors is fairly simple when all inter-distances are known, it is significantly more complex with a partial set of measurements. With UWB it is often the case that only a portion of the inter distances are recovered when the anchors measure themselves the distance to their neighbours. This paper considers the challenge of automatically recovering the initial anchor locations, a problem named "self-calibration" with an incomplete inter-anchor measurement set. As will be discussed further, this self-calibration generally requires a good initial "guess" (often this must be acquired manually). After the initial guess, the self-calibration is typically solved using a non-linear gradient descent with knowledge of inter-anchor ranges to recover the anchor locations. This initial "guess" turns out to be non-trivial to recover automatically in practical situations, and the subject is poorly addressed in the current literature.

As a solution, this paper proposes a graph-based self-calibration approach to the extraction of this initial "guess" of the UWB anchor mesh topology from a set of inter-anchor measurements. By using graph data structures, we were able to robustly recover this initial guess of the topology despite noisy or missing measurements. Moreover, we were able to identify and qualify challenging parts of the mesh, allowing us to propose new anchor or tag placement so as to uniquely define the geometry. As an additional benefit, the approach proposed here does not make any assumption on the dimension of the problem, which can be in 2D or 3D depending on the practical setup. Once this initial guess is generated, it is assumed a non-linear optimization process can be instantiated from this initial guess to generate a more precise geometry, however, this point will not be discussed in this paper.

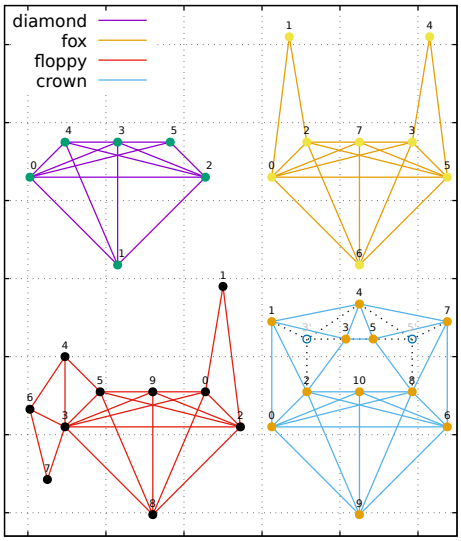

Fig. 1. Examples of mesh geometries used to validate the concept proposed in this paper. Circles are anchors, and edges are present when the range between the corresponding anchors is known. The dashed lines and empty anchors represent an alternative *crown* configuration.

II. DEFINITIONS

Before presenting the related work and methodology, we introduce the formal definitions of the concepts and variables used in this paper.

We assume the problem at hand requires recovery of the 2D or 3D locations of N anchors, denoted as (x_i, y_i) or (x_i, y_i, z_i), from a set of M inter-anchor measurements $d_{i,j,k}$, with $i, j \in [1, N]$, $i < j$ and $k \in [1, M]$. Importantly, we do not assume a range has been observed between every pair of anchors. In cases where the range between a pair of anchors has been measured at least once, we denote the range as $\bar{d}_{i,j}$ the mean range measurement between these two anchors.

The graph structure we will define relies on the geometric notion of simplex. In practical terms, a simplex is the smallest shape in a given space, i.e. a segment in 1D, a

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 871260. Corresponding author: cedric.pradalier@georgiatech-metz.fr
[1]CNRS IRL2958 GT-CNRS, Metz, France
[2]GeorgiaTech Lorraine, Metz, France
[3]Georgia Institute of Technology, Atlanta, USA

triangle in 2D or a tetrahedron in 3D. In order to be generic, we will refrain from using the term triangle or tetrahedron in the following, but instead refer to Simplex$[D]$ for a simplex in dimension D. A Simplex$[D]$ is defined by $D+1$ vertices, and possesses $D+1$ "faces" which are Simplex$[D-1]$ objects.

In the context of a localization problem, a vertex V of a Simplex$[D]$ can be estimated from the knowledge of the location of the vertices of the opposite Simplex$[D-1]$ face, and D range measurements from V to the D known vertices. This is typically referred to as the trilateration problem. Similarly, the shape of Simplex$[D]$ can be recovered from $D(D-1)/2$ inter-vertex ranges. In this case, the location of the vertices is defined up to an unknown isometry (i.e. translation, rotation or symmetry). In particular, the symmetries lead to 2 equivalent solutions in 2D, and 4 equivalent solutions in 3D. The challenge we address in this paper is to build a globally consistent geometry from the reconstruction of a set of simplices assuming we have more than $D+1$ anchors.

If the ranges are noisy, the two trilateration problems can be solved using a least square minimization with its own initial guess. In 2D, one vertex is set at the origin, the second at the observed range on the x axis, the third just needs to be off the x axis, so we start from an equilateral triangle. The sign of the y coordinate of the third vertex is not decidable from the ranges. In 3D, the 3 first vertices are set in the (x, y)-plane, as in 2D. The 4th vertex is set to obtain a regular tetrahedron. The sign of its z coordinate is not decidable either.

Finally, let us now consider two simplices S_1 and S_2 sharing a face F, and let V_1 and V_2 the respective vertices of S_1 and S_2 opposite to F. If the locations of the vertices of S_1 are known, and the distance between V_1 and V_2 is known, then V_2 can be uniquely estimated. In other words, the knowledge of this distance lifts the reconstruction ambiguity stemming from the symmetries. We will denote V_2 as the validation vertex of V_1 over the face F.

III. Related Work

The related work on self-calibration of UWB networks has been thoroughly surveyed in [1]. Among the various issues considered, this survey lists 36 reference articles dealing specifically with the issue of self-calibration. For space sake, only a limited number of references are listed here.

In this list, many papers focus on distributed estimation [2], where every anchor tries to obtain an estimate of its own location from the surrounding anchors. By contrast, we are considering a global estimation of geometry, thus the challenge of distributed estimation will not be discussed further. A subset of articles solving the global self-calibration problem [3] consider iterative estimations where a group of N landmarks is already known. When a new one is observed, an iterative estimator refines its position, taking into account the movement of the observer. The iterative nature of the algorithms and the reliance on some estimate of the observer leads to a different set of hypotheses than those we are considering.

For the papers solving the global self-calibration problem without the iterative approach or observer, a two-stage approach can be seen in all of them: first an initial geometry is obtained in various ways, then the final geometry is computed through a refinement process, often deploying non-linear optimization or Bayesian inference. However, unlike the article presented here, the initial stage of the problem has yet to be solved explicitly, verifiable, or practically (e.g. with missing measurements). Further, the methods deployed do not provide any feedback when reconstruction is not possible.

For example, some papers solving the global self-calibration problem using a set of inter-anchor measurements, assumes the initial guess of the geometry can be recovered but do not explicitly state how [4]. Other papers, use a first guess resulting from a set of linear equations[5]. However, this method assumes either all the inter-anchor distances are measured or that at least the distance from one anchor to all the others is already known. Just considering the examples from fig. 1, one can see these hypotheses cannot be verified in practical cases. More importantly, the linear approach does not provide an explainable analysis of the distribution when the mesh geometry cannot be recovered.

The closest paper that comes to solving the initial guess is [6], which solves a form of global self-calibration using only set-constraints (anchor A is within communication range of anchor B, anchor C is east of anchor A). The global optimization resulting from these set-constraints is similar in spirit to our approach. However, using only the set representation leads to a coarser reconstruction than what can be achieved taking into account the range measurements, the simplex geometry, and the graph structure we are proposing.

Another very close work is presented in [7] which also use a graph theoretic approach to present the problem mathematically as a multidimensional scaling with missing data, but also makes the link to the very significant amount of work related to the rigidity of truss structure in mechanics. Although the mathematical framework in [7] is more formal, it is only evaluated in 2D. Our approach is evaluated both in 2D and 3D and the simplex graph structure provides a level of explainability not present in [7].

In comparison to the related work, this paper focuses on solving the global self-calibration problem when there is an incomplete set of inter-anchor measurements. Its main advantages being it does not depend on a complete set of inter-anchor measurements and its ability to provide an initial guess of the mesh geometry in 2D or 3D, with an explainable analysis explicitly highlighting the part of the mesh that cannot be recovered correctly. This allows us to also propose a strategy for adding new anchors or moving an anchor to specific locations to disambiguate the geometry.

IV. Methodology

Our approach to the UWB mesh geometry recovery is based on the construction of multiple graph data structures

from which a globally consistent reconstruction can be achieved.

A. Simplex identification

From the set of averaged range measurements $\bar{d}_{i,j}$, our first task is to identify potentially recoverable simplices. To this end, we build a graph of vertices where every node is an anchor, and edges exists where range measurements exists. In such a graph, recoverable simplices$[D]$ are cliques of $D+1$ nodes. Finding cliques in a graph is a costly enumeration problem for large cliques and dense graphs, but a fairly innocuous problem in the graphs we are considering here. From the exhaustive list of $(D+1)$-cliques in the graph, we now assume that we have recovered a list of simplices. An example of such a graph is visible in fig. 2 (left), and the resulting simplices are visible as nodes on the right-hand-side graph in the same figure.

Note that any node with a degree lower than D will not be localizable. We flag all such vertices at this stage and remove them from the graph. This may reduce the degree of other nodes, which are in turn removed from the graph, until all the anchors that cannot be localized have been identified and excluded.

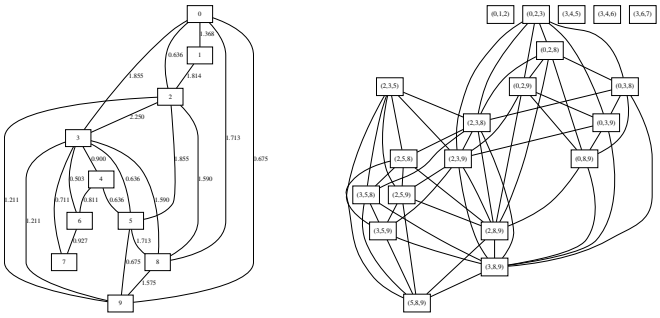

Fig. 2. Point graph and resulting simplex graph from the *floppy* example in fig. 1.

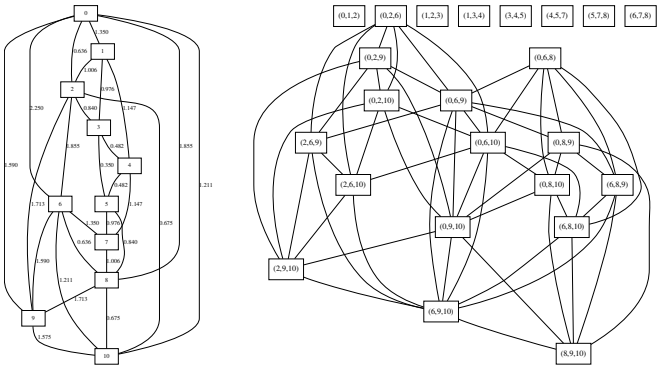

Fig. 3. Point graph and resulting simplex graph from the *crown* example in fig. 1.

B. Simplex graph definition

The goal of this stage is to build a graph of simplices. Let us define $\mathcal{G}_S$ the graph where each node is a Simplex$[D]$, and there is an edge between two nodes if the corresponding simplices share a face and the distance between vertices opposite to this face is known. The resulting graph can be seen in fig. 2 (right). In fig. 2, simplices $(0,1,2)$ and $(3,4,5)$ (the ears) are not connected, so they cannot be uniquely reconstructed.

Extracting the connected components of $\mathcal{G}_S$ is easily done. This provides sets of simplices for which symmetry ambiguities can all be lifted. If there are multiple connected components, this may be an indication the mesh geometry cannot be uniquely recovered. We will discuss special cases in IV-E and propose remediation action in IV-D.

To further exploit the graph structure, we build a spanning tree on every connected component. A spanning tree is a subset of the graph that contains all the nodes, but only enough edges to create a tree and maintain connectivity. This can be simply implemented with a depth-first search through the graph because we don't need a minimum spanning tree. A spanning tree must be built from a given root, which can be any node of the connected component. In practice, we select the largest Simplex$[D]$ because its reconstruction may be less sensitive to the measurement noise.

C. Geometry estimation

At this stage, we assume that $\mathcal{G}_S$ contains a single connected component and a spanning tree $\mathcal{T}_S$ has been constructed over $\mathcal{G}_S$. From this, we can iteratively reconstruct a unique mesh geometry up to an unobservable global isometry (translation, rotation and flip). We start with the root of $\mathcal{T}_S$. Unless we have different prior knowledge, we set this first simplex on the origin and align with the axis, as mentioned in II. Then we recursively iterate through the nodes of $\mathcal{T}_S$. For a node S, we assume the vertices of the parent S_p are known. Because S and S_p share a face, there is only one vertex V to recover in S. As mentioned in II, this vertex can be estimated by trilateration using the D known ranges. Any reconstruction ambiguity can then be lifted by using the distance between V and its validation vertex in S_p, known from the construction of $\mathcal{G}_S$. Hence, V is uniquely defined as long as S_p is not a degenerated simplex (flat triangle or flat tetrahedron). Note that, because a vertex may belong to many simplices, its location may be estimated in different ways in different simplices. An example of the resulting reconstruction can be seen in fig. 4. As mentioned before, the reconstruction is up to an unknown rotation and translation, and only the anchors that can be uniquely reconstructed are displayed in these graphs.

At the end of this process, all the nodes of $\mathcal{G}_S$ have been processed and as a result all the anchor location have been estimated. Importantly, beyond the specific trilateration algorithm, which differs in 2D or 3D, this process does not make any assumption on the dimension of the reconstruction problem.

D. Identifying ambiguous simplices

When $\mathcal{G}_S$ is made of more than one connected component, this often means the mesh geometry cannot be recovered uniquely from the set of ranges. In this section, we consider

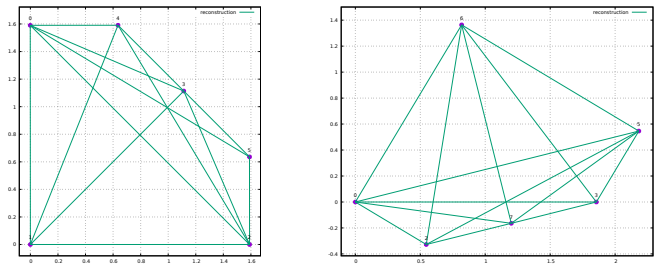

Fig. 4. Reconstructed geometry from the *diamond* and *fox* example in fig. 1. The *fox* ears on the right are not reconstructed because they cannot uniquely be identified from the ranges.

different configurations that can lead to this situation and detail how they can be addressed.

In order to formally describe the different situations, we first define a new graph structure, a bipartite graph $\mathcal{G}_B$, where nodes are the simplices and their faces. Edges in $\mathcal{G}_B$ connect the simplices with their faces, and the faces with the simplices containing them. Fig. 5 provides an example of such graphs for the *crown* example.

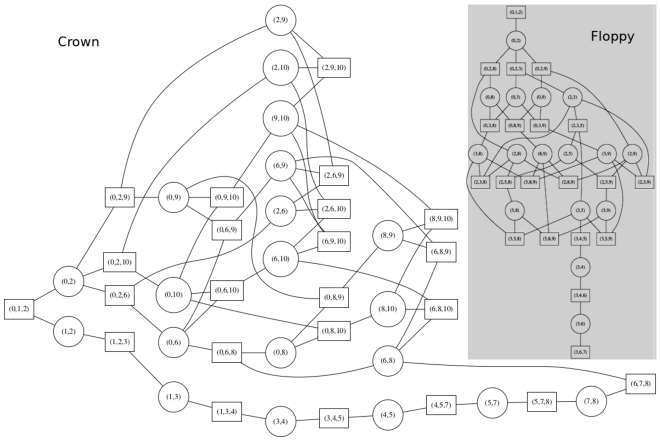

Fig. 5. Bipartite simplex graph from the *crown* and *floppy* (insert) examples in fig. 1. Simplex nodes are represented with boxes and face nodes with circles. The truss on top of the crown leads to the clear cycle on the right of the graph.

In $\mathcal{G}_B$, it can be shown that an edge between simplices S_1 and S_2 of $\mathcal{G}_S$ is equivalent to the existence of a cycle[1] containing S_1 and S_2 in $\mathcal{G}_B$. By extension, it can be seen that a simplex truss structure (as seen in the top part of the *crown* example, fig. 5), even if it does not lead to edges in $\mathcal{G}_S$, leads to a cycle in $\mathcal{G}_B$. In practice, the presence of a cycle usually defines a unique geometry, but it is possible to produce pathological cases where symmetries lead to ambiguities despite a cycle.

We can now categorize the situations resulting from a disconnected $\mathcal{G}_S$ graph. Let us first consider that $\mathcal{G}_S$ and $\mathcal{G}_B$ are both made of several connected components. In this case, the geometry cannot be recovered completely, so we typically only consider the biggest connected component of $\mathcal{G}_B$ and assume the other parts of $\mathcal{G}_B$ cannot be recovered.

From now on, we consider that $\mathcal{G}_S$ is disconnected and $\mathcal{G}_B$ is connected. Furthermore, we assume that the simplices belonging to the largest connected component in $\mathcal{G}_S$ have been identified and labelled as "known". If we consider an "unknown" simplex S made of $D+1$ faces, then, by construction, there is at least one face that starts a path in $\mathcal{G}_B$ leading to the "known" nodes. If only one face leads to the "known" simplices, then the geometry of the current simplex cannot be recovered uniquely.

This can be seen, in fig. 5, in *floppy* with simplices $(1,2,3)$, $(3,4,5)$, $(3,4,6)$ and $(3,6,7)$.

Note, however, the length of the shortest path can indicate an ambiguity of a simplex reconstruction. For instance, with a length of 1, the considered simplex has one face made of known vertices, and only one unknown vertex, which can be on either side of the known face. This is the case with the "ears" of the *fox*. With a length of 2, there are 2 unknown vertices and 4 possible configurations. This level of ambiguity happens in the "floppy" ear of the *floppy example*, with a sequence of 3 unrecoverable nodes. This information may be useful when considering where to move an active agent to collect new ranges and lift ambiguities. Simplices must be solved in order of their graph distance to the known vertices to maximize the confidence in the recovered geometry.

E. Handling long cycles

When a cycle in a graph is long, in most cases, you can still recover the geometry, but as it lengthens the chances diminish. To combat this, we can use the properties of the two different graphs to help with reconstruction. Let us now consider a truss structure as seen in the top of the *crown* example. In this case, any simplex belonging to the truss has two different paths reaching the "known" nodes through different faces (these paths are clearly visible in the string of nodes on the bottom of fig. 5), and this long path is a part of a cycle in $\mathcal{G}_B$.

With respect to the detection, we are looking for a simplex which is not connected in $\mathcal{G}_S$ and connected in $\mathcal{G}_B$. We then conduct a breadth-first search from every face of such a simplex until we can find a "known" node. If two such searches find a half-path to the set of known simplices, this simplex belongs to a truss structure and the two half-paths can be combined in a single path. It is critical, however, that the half-paths do not share either a simplex or even a face: their destinations can be the same known simplex, but they cannot enter it via the same face, otherwise the reconstruction will be ambiguous. One must also avoid that the faces from which the path leaves the first known simplex and enters the last known simplex ($(0,2)$ and $(8,6)$ in the *crown* example) are not collinear/coplanar, as this would lead to another ambiguity.

As discussed in the *floppy* example, every new simplex in a string of $\mathcal{G}_B$-connected simplices multiplies the number of possible configurations of the string by two, due to

[1] In graph theory, a cycle is a sequence of node starting and finishing at the same vertex, with no other repetition.

symmetries (this is valid in 2D and in 3D). Because we are considering a string of simplices starting from known vertices and finishing on known vertices, the solution we propose is then to recursively trilaterate every simplex on the string, starting from the known vertices on one end. At the other end, the correct configuration will be the one predicting correctly the position of a known vertex. In most practical cases, this approach will recover uniquely an estimate of the complete geometry, as can be seen for the *crown* example in fig. 6. Modifying this example by moving anchors 3 and 5 to position 3' and 5' creates an ambiguous geometry that cannot be recovered uniquely, as seen on the right of the figure. This situation may be detected when multiple configurations predict correctly the position of the reference known vertex. However, this detection requires adding many manual thresholds and our statistical analysis showed there are always ambiguous situations, especially in presence of measurement noise (table I).

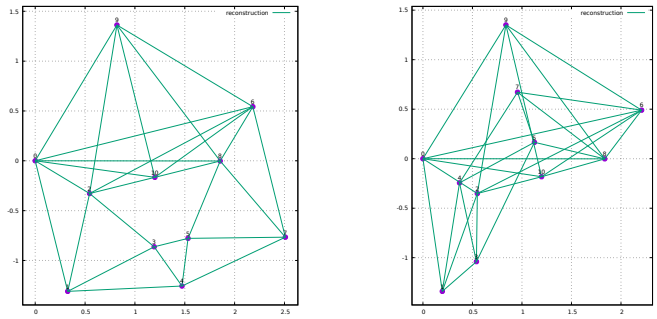

Fig. 6. Left: reconstruction of the *crown* example using the data shown in fig. 1. Right: reconstruction failure in an ambiguous case.

F. Proposed algorithm

Building on the sections above, the proposed algorithm can be summarized as follows:

1) Build a graph of points and use a clique enumeration algorithm to identify the simplices, but reject the simplices which cannot be trilaterated.
2) Build a graph of vertices with validation constraints ($\mathcal{G}_S$), extract its connected components and spanning forest.
3) Initialize the root simplex from the largest tree in the forest to an arbitrary position and orientation, then iteratively follow the tree, trilaterating one vertex at a time.
4) Identify truss structure linking a known simplex to another one, and recursively reconstruct the truss structure hypotheses to select the most likely.
5) Mark any remaining vertex as unrecoverable.

V. Evaluation

A. Simulations

The first evaluation of the proposed algorithm is through a set of randomized simulations. We consider the random placement of 10 anchors in 2D in $[-5,5]^2$ or in 3D in $[-5,5]^3$ (at least $1m$ apart), and 100 anchor-to-anchor measurements. The measurements are affected by a variable noise in $\{0.05, 0.10, 0.15\}$ and a variable maximum range in $\{5,6,7,8,9,10\}$. For each combination, 1000 random distributions and measurements are generated and evaluated. For every configuration, we compute the maximum reconstruction error with respect to the ground truth distribution, and we issue a warning if the reconstruction is ambiguous. For every dimension and noise level, we report, in tables I, the statistics of this error over all the trials (for all the max ranges) and the number of ambiguous configuration detected (see fig. 6). In some extreme cases, large reconstruction errors occur for ambiguous geometries that cannot be decided. These have been flagged by hand, and the number of the manual flags is also reported. Finally, we also compute the reconstruction statistics excluding the configuration deemed ambiguous, in the 'G' lines in the tables.

D	Noise	mean err(m)	median err(m)	3rd quart.(m)	max err(m)	#warn #manual
2	0.05 A	0.07	0.04	0.06	11.66	25
	G	0.05	0.04	0.06	0.52	0
	0.10 A	0.14	0.09	0.12	12.16	42
	G	0.1	0.09	0.12	0.63	1
	0.15 A	0.21	0.13	0.18	11.72	67
	G	0.15	0.13	0.17	1.07	5
3	0.05 A	0.19	0.08	0.12	18.63	78
	G	0.09	0.07	0.11	1.19	6
	0.10 A	0.37	0.16	0.25	18.68	173
	G	0.18	0.15	0.22	1.46	25
	0.15 A	0.56	0.24	0.37	15.37	298
	G	0.27	0.22	0.33	1.49	80

Reconstruction error (m) as a function of dimension D, measurement noise (m), averaged over 6 maximum ranges and 1000 trials. The first line show statistics over all the trials, and the second one considers only trials for which no ambiguity warning has been raised.

TABLE I

Simulation results for 2D and 3D reconstruction

These results show that the reconstruction error of the proposed method stays below $1m$ in average over the range of conditions, except when the geometry is ambiguous. Every line of the tables summarizes 6000 trials. In 2D, the detection of ambiguous cases is relatively rare, and the need to manually flag extreme cases only occurs in the presence of noise. In 3D, the problem is much harder and ambiguous situations occurs much more often, which results also in much more manual flags.

B. Real experiments

Our second set of experiments involves 11 UWB anchors (MDEK1001) with anchor names taken from a popular book. With the proper driver, these anchors can be requested to establish a mesh network and estimate the ranges to their neighbours. A calibration of the error distribution showed a standard deviation of $0.15m$ for the ranges considered here. With this setup, 4 experiments in 2D are reported here (printed figures do not lend themselves well to 3D plots). Here we show the reconstructed geometry in comparison

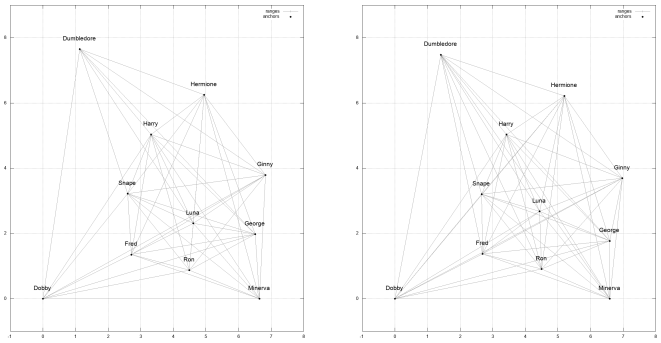

Fig. 7. Comparison of geometry recovery indoor, from laser (left) and UWB (right) ranges, experiment 1.

to the real one, as we are only considering the problem of obtaining the initial guess.

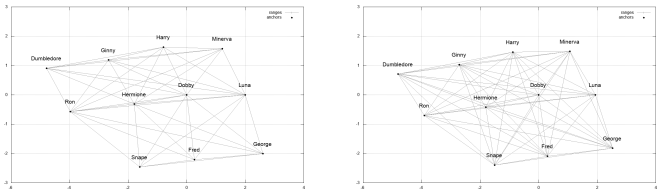

Fig. 8. Comparison of geometry recovery indoor, from laser (left) and UWB (right) ranges, experiment 2.

In the first two experiments, with an indoor setup, we measured the inter-anchor distances with a high-precision laser range finder, reconstructed the geometry, and manually validated it. We then compare this with the automatically reconstructed geometry obtained from UWB ranges in figures 7 and 8. In both cases, the topology is well recovered. Note that the geometry is defined up to an isometry, so we manually selected the appropriate translation and rotation to simplify the comparison.

Finally, two experiments were conducted outdoor. In this case, the reference position of the anchors was measured using an RTK GPS and overlayed on Google Earth. The resulting reconstructions can be seen in figures 9 and 10. In both cases, the topology is accurately recovered. In the second case, however, it was incomplete as anchor 'Ron' had noisy range measurements leading to a missing simplex in $\mathcal{G}_S$, thereby resulting in an ambiguous geometry. Also note that the pose of 'Dumbledore' in fig. 10 is probably incorrect due to overestimated ranges by the UWB anchors. Nonetheless, refining the geometry over time with more measurements would likely make 'Ron' recoverable and 'Dumbledore's position more precise.

VI. CONCLUSIONS

This paper proposes an algorithm to estimate an initial guess of the geometry of a mesh of devices, knowing a subset of inter-device distances. This has been developed for and tested on UWB anchors, but the approach is potentially more generic. In comparison with earlier approaches, the graph-based approach provides an explainable solution, is robust

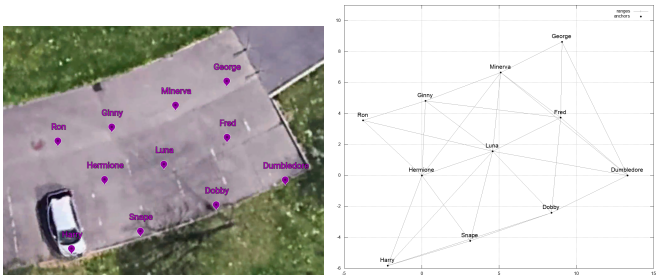

Fig. 9. Comparison of geometry recovery outdoor, from RTK GPS (left) and UWB (right) ranges, experiment 1.

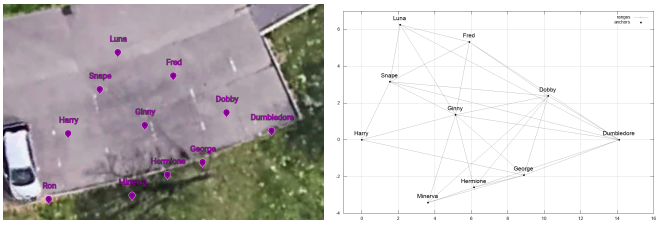

Fig. 10. Comparison of geometry recovery outdoor, from RTK GPS (left) and UWB (right) ranges, experiment 2.

to missing measurements and capable of providing identified guesses when appropriate. Statistical tests on random configurations show the sensitivity of the geometry reconstruction to local symmetries, in particular in presence of measurement noise. Tests on real configurations with UWB beacons show the applicability of the approach, but also highlight the challenges of recovering a unique geometry in the presence of symmetries.

REFERENCES

[1] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, "Self-calibration and collaborative localization for uwb positioning systems: A survey and future research directions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–27, 2021.

[2] M. R. Gholami, L. Tetruashvili, E. G. Ström, and Y. Censor, "Cooperative wireless sensor network positioning via implicit convex feasibility," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5830–5840, 2013.

[3] B. Li, N. Wu, H. Wang, D. Shi, W. Yuan, and J. Kuang, "Particle swarm optimization-based particle filter for cooperative localization in wireless networks," in *2013 International Conference on Wireless Communications and Signal Processing*. IEEE, 2013, pp. 1–6.

[4] M. Hamer and R. D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *IEEE Access*, vol. 6, pp. 22 292–22 304, 2018.

[5] K. Batstone, M. Oskarsson, and K. Åström, "Towards real-time time-of-arrival self-calibration using ultra-wideband anchors," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2017, pp. 1–8.

[6] L. Doherty, L. El Ghaoui *et al.*, "Convex position estimation in wireless sensor networks," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, vol. 3. IEEE, 2001, pp. 1655–1663.

[7] M. Oskarsson, K. Åström, and A. Torstensson, "Prime rigid graphs and multidimensional scaling with missing data," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 750–755.