A Computational Theory of Robust Localization Verifiability in the Presence of Pure Outlier Measurements

Mahroo Bahreinian, Roberto Tron

***Abstract*— The problem of localizing a set of nodes from relative pairwise measurements is at the core of many applications such as Structure from Motion (SfM), sensor networks, and Simultaneous Localization And Mapping (SLAM). In practical situations, the accuracy of the relative measurements is marred by noise and outliers; hence, we have the problem of quantifying how much we should trust the solution returned by some given localization solver. In this work, we focus on the question of whether an ℓ_1-norm robust optimization formulation can recover a solution that is identical to the ground truth, under the scenario of translation-only measurements corrupted exclusively by outliers and no noise; we call this concept *verifiability*. On the theoretical side, we prove that the verifiability of a problem depends only on the topology of the graph of measurements, the edge support of the outliers, and their signs, while it is independent of ground truth locations of the nodes, and of any positive scaling of the outliers. On the computational side, we present a novel approach based on the dual simplex algorithm that can check the verifiability of a problem, completely characterize the space of equivalent solutions if they exist, and identify subgraphs that are verifiable. As an application of our theory, we provide a procedure to compute *a priori* probability of recovering a solution congruent or equivalent to the ground truth given a measurement graph and the probabilities of each edge containing an outlier.**

I. Introduction

The problem of localizing a set of agents or nodes with pairwise relative measurements can be modeled as a *pose graph* [18], where the nodes are associated to vertices and pairwise relative measurements are associated to edges. Typical solutions are cast as maximizing the likelihood of the relative pairwise measurements given the estimated agent poses, possibly after choosing different statistical models that lead to different cost functions to be optimized; this approach has been referred to as Pose Graph Optimization (PGO) [10] Different versions of this problem have been of interest in a number of fields. In computer vision, the Structure from Motion (SfM) problem [16] aims to recover the location and orientation of cameras, and the location of 3-D points in the scene, given an unordered collection of 2D images. In sensor networks, the nodes need to be localized from relative translation or distance measurements [6], [9], [11]. In robotics, the Simultaneous Localization And Mapping (SLAM) [14], [24] problem aims to recover the pose trajectories of one or more mobile agents, while building a map of the environment, using multimodal measurements (extracted from images or inertial measurement units). In all these applications, pairwise measurements are generally corrupted by a combination of small-magnitude noise and large-magnitude outliers, due to hardware, environmental, and algorithmic factors [31].

The simplest and most common objective employed in PGO is the least square error [3], [13], which corresponds to the assumption that measurements are affected by Gaussian noise (typically having low variance). However, the solution of least square optimization can be greatly impacted by the presence of outliers (one or two isolated outliers can bias the solution for all the nodes). In [22], [23], the authors estimate the location of the nodes (with relative direction measurements) by minimizing a least square objective function with global scale constraints through a semi-definite relaxation (SDR), while [27], [28] solve a similar problem through constrained gradient descent; in both cases, although some theoretical analysis of the robustness of the method to noise is given, the resulting methods are not robust to outliers (due to the use of the least squares cost). To obtain robustness, a possible approach is to use a pre-processing stage (e.g., using Bayesian inference or other mechanisms) to pre-process the measurements and remove outliers, followed by PGO [19], [21], [30], [31], [33]. An alternative or complementary method is to optimize robust (ideally convex) cost functions, such as the Least Unsquared Deviation (LUD) [15], [34] or others [32]; in this case, the optimization can be carried out using re-weighting techniques (such as Iterative Reweighted Least Squares, IRLS [17] or others [1], [25]), or Alternate Direction Method of Multipliers (ADMM, [7], [12]). In all these robust approaches, it has been shown empirically that the results are close to the ground truth even in the presence of outliers; however, there have been no published attempts to characterize, in a precise way, what kind of situations can be tolerated by the solvers. The reader should contrast this, for example, to the simple case of the median in statistics, where it is well known that such estimator is robust up to 50 percent of outliers [8], [20], [29].

The goal of this paper is to obtain results for PGO that are similar in spirit to those available for the median in classical robust estimation theory. In order to obtain strong theoretical results on the effect of outliers alone, in this paper we focus on the case where we are interested in recovering only translations (not rotations), and there is no Gaussian noise (i.e., each measurement is either perfect, or corrupted by an outlier of arbitrarily high, but bounded, magnitude); we plan to extend our results to more realistic situations in our future work. As the objective function in the optimization, we use the least absolute value deviation (ℓ_1-norm), which is convex and

This work was supported by the National Science Foundation grant NSF NRI-1734454.

R. Tron is with Faculty of Mechanical Engineering and Systems Engineering, Boston University, Boston, MA 02215, USA tron @bu.edu

M. Bahreinian is with the Department of Systems Engineering, Boston University, Boston, MA 02215, USA mahroobh@bu.edu

allows us to bring the extensive tools from linear optimization to our disposal. Under these conditions, it can be empirically noticed that the robustness of the ℓ_1 cost function leads to three possible outcomes: the solution found by the solver and the ground truth are either congruent; different, but with the same value for the cost; or drastically different. Moreover, this categorization appear to depend on where the outliers are situated, but not on their absolute magnitude. We formalize this observation in the notion of *verifiability* for a graph. Given an hypothesis for the edge support of the outliers and their sign, we can use convex optimization theory to predict whether solving the ℓ_1 optimization problem can recover the ground truth solution, whether this can be done uniquely, and, if not, completely characterize the set of solutions, while identify which subsets of the graph can be exactly recovered. From this, and by knowing the probability of each edge to be an outlier with a given sign, we can then compute the probability that the recovered solution is completely or partially congruent to the ground truth embedding (without knowing the actual support of the outliers). Moreover, the procedure can be extended to identify subgraphs that can be uniquely localized with high probability.

II. Notation And Preliminaries

In this section we formally define our measurement model, the optimization problem for localizing nodes from relative measurements, and we define the notion of *verifiability*.

A. Graph Model

Definition 1: A sensor network is modeled as an oriented graph $G = (V, E)$, where $V = \{1, \dots, N\}$ represents the set of sensors, and $E \subset V \times V$ represents the pairwise relative measurements; we have $(i, j) \in E$ if and only if there is a measurement between node $i \in V$ and node $j \in V$. We assume that G is connected. We use $|V|$, $|E|$ to indicate the cardinality of the sets V and E, respectively.

Definition 2: An *embedding* of the graph associates each node i to a position $\mathbf{x}_i \in \mathbb{R}^d$. Mathematically, we identify an embedding with a matrix $\mathbf{X}_V = \begin{bmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_{|V|}\end{bmatrix} \in \mathbb{R}^{|V| \times d}$, with d being the ambient space dimension; we denote the ground truth embedding as $\mathbf{X}_V^*$.

Definition 3: A measurement between node i and j, $(i, j) \in E$, is modeled as

$$t_{ij} = \mathbf{x}_j^* - \mathbf{x}_i^* + \epsilon_{ij}, \tag{1}$$

where $\mathbf{x}_j^* - \mathbf{x}_i^*$ is the true translation between nodes i and j, and ϵ_{ij} is a random variable for outliers with distribution

$$\epsilon_{ij} = \begin{cases} 0, & \text{w.p. } 1 - p_{ij}^+ - p_{ij}^- \\ \mathcal{U}^-, & \text{w.p. } p_{ij}^- \\ \mathcal{U}^+, & \text{w.p. } p_{ij}^+ \end{cases}, \tag{2}$$

where $p_{ij}^-, p_{ij}^+ \in (0, 1)$ are a priori probabilities of having an outlier for the edge (i, j) with, respectively, negative or positive support, and $\mathcal{U}^-$,$\mathcal{U}^+$ are stochastic functions that returns a samples from a uniform distribution with arbitrary, but finite, non-zero support contained in, respectively, $\mathbb{R}_{<0}, \mathbb{R}_{>0}$. If $d > 1$, we assume that the entries of the vector ϵ_{ij} are i.i.d. with the same distribution (2).

We assume that the probabilities $p_E = \{p_{ij}\}_{(i,j)\in E}$ are known; as shown below in Theorem 3.3, our results are valid independently of the support for $\mathcal{U}^\pm$ (as long as it is finite).

From this point on, subscripts with V or E refer to the vector obtained by stacking the specified quantity considered for all nodes or edges (e.g., $p_E = \text{stack}(\{p_{ij}\}_{(i,j)\in E})$).

Definition 4: We define the *outlier support* $E_\epsilon \subset E$ such that $E_\epsilon = \{(i, j) \in E : \epsilon_{ij} \neq 0\}$.

B. Localization Through Robust Optimization

Given the relative pairwise measurements t_E in the graph G, we aim to find and characterize all the embeddings that minimize the sum of all absolute residuals, i.e.,

$$\min_{\mathbf{X}_V, \mathbf{x}_1 = \mathbf{0}} \sum_{(i,j)\in E} \|\mathbf{x}_j - \mathbf{x}_i - t_{ij}\|_1 \tag{3}$$

.

C. Global Translation Ambiguity

If we translate all the points in the embedding by a common translation, the cost (3) does not change, since the relative measurements also remain constant. Without loss of generality, we fix this translation ambiguity by choosing a global reference frame such that $\mathbf{x}_1^* = \mathbf{x}_1 = \mathbf{0}_d$. Since we assumed that the graph is connected (Definition 1), fixing $\mathbf{x}_1$ alone is sufficient to fix the global translation. For simplicity's sake, we keep $\mathbf{x}_1$ as a variable in the optimization problem (3) even though it is used to fix the global translational ambiguity.

D. Set of Global Optimizers $\mathcal{X}^{opt}$

We define as $\mathcal{X}^{\text{opt}}$ the set of local minimizers of (3). Since the objective function is convex (being the sum of convex functions), we have that $\mathcal{X}^{\text{opt}}$ is convex, and is exactly given by the set of global minimizers (see [4, Theorems 8.1, 8.3]). Moreover, using the fact that the value of $\mathbf{x}_1$ is fixed and that the graph is connected, it is possible to show that the objective function in (3) is radially unbounded, and therefore the set $\mathcal{X}^{\text{opt}}$ is compact. In fact, since (3) can be rewritten as a Linear Program (LP, see below), $\mathcal{X}^{\text{opt}}$ either reduces to a single point, or is a polyhedron with a finite number of *corners* (we use this term instead of *vertex* as a distinction from the individual elements of V).

E. Verifiability

If $E_\epsilon = \emptyset$, then t_E is identical to the true measurements, and the solution of (3) would be equal to the ground truth embedding $\mathbf{X}_V^*$. However, since (3) is a robust optimization problem, the optimum value could still correspond to $\mathbf{X}_V^*$ even in the presence of outliers ($E_\epsilon \neq \emptyset$). In the latter case, however, there could be multiple minimizers all giving the same value of the ℓ_1 objective. We start formalizing the situation with the following.

Definition 5: A *(localization) problem* is defined by a pair of a graph $G = (V, E)$ and a *signed outlier support* $E_\epsilon^\pm \subset E \times \{+, -\}$ (i.e., a subset of edges paired with signs). A problem is said to be *uniquely verifiable* if $\mathcal{X}^{\text{opt}} = \mathbf{X}_V^*$

(unique solution), *verifiable* if $\mathbf{X}_V^* \in \mathcal{X}^{\text{opt}}$ (possible multiple equivalent solutions), and *non-verifiable* otherwise.
Note that, according to the definitions, uniquely verifiable problems are also verifiable.

In [31, Theorem 2], the authors also introduce the concept of verifiable edge and verifiable graph; however, that work considers only the case of a single outlier ($|E_\epsilon^\pm| = 1$). In this work, we generalize the same notion to more general cases.

III. Canonical LP Form And Verifiability

In this section we perform a series of transformations to the optimization problem (3) to reduce it to a canonical, one-dimensional LP (and its dual), allowing us to deduce that particular ground-truth embeddings X_V^* and outlier magnitudes ϵ_E do not affect the verifiability of a problem, thus ensuring that Definition 5, which depends only on the graph topology and the signed outlier support, is well posed.

A. Canonical Form

We first perform a change of variable so that the true embedding corresponds to the point at the origin. More in detail, we define a set of new variables $\mathbf{X}'_V$ such that

$$\mathbf{X}'_V = X_V - \mathbf{X}_V^*, \tag{4}$$

i.e., for each $i \in V$ we replace $\mathbf{x}_i$ by $\mathbf{x}'_i + \mathbf{x}_i^*$. If $\mathbf{X}^*$ is an optimal point for (3), then $\mathbf{X}' = \mathbf{0}_{|V|}$ is a minimizer for the following transformed problem:

$$\min_{\mathbf{x}'_V, \mathbf{x}'_1 = 0} \sum_{(i,j)\in E} \|(\mathbf{x}'_j + \mathbf{x}_j^*) - (\mathbf{x}'_i + \mathbf{x}_i^*) - (\mathbf{x}_j^* - \mathbf{x}_i^* + \epsilon_{ij})\|_1, \tag{5}$$

which reduces to

$$\min_{\mathbf{x}'_V, \mathbf{x}'_1 = 0} \sum_{(i,j)\in E} \|\mathbf{x}'_j - \mathbf{x}'_i - \epsilon_{ij}\|_1. \tag{6}$$

By inspecting (6), we can deduce the following:

Lemma 3.1: The canonical form of the optimization problem, and the definition of verifiability, do not depend on the specific value of $\mathbf{X}_V^*$.

Proof: Assume we have two problems with different true embeddings $\mathbf{X}_{V_1}^*, \mathbf{X}_{V_2}^*$, but with the same graph topology G, and the same outlier realization ε_E. The corresponding optimization problem in canonical form (6) are the same, hence, also their set of solutions (after the change of variable) is the same. The rest of the claim then follows from Definition 5. ∎

The practical implication of Lemma 3.1 is that we can reason about the verifiability of a problem independently from the specific true positions of nodes. To simplify our discussion, for the remainder of the paper and without loss of generality we use $\mathbf{x}$ instead of $\mathbf{x}'$.

B. Reduction to One-Dimensional Problems

The ℓ_1-norm $\|\cdot\|_1 \colon \mathbb{R}^d \to \mathbb{R}$ in the optimization objective can be decomposed into sums of absolute values across dimensions, i.e., (6) becomes

$$\min_{\mathbf{x}_V, [\mathbf{x}_1]_k = 0} \sum_{k=1}^{d} \sum_{(i.j)\in E} \big|[\mathbf{x}_j]_k - [\mathbf{x}_i]_k - [\epsilon_{ij}]_k\big|, \tag{7}$$

where $[v]_k$ denotes the k-th element of a vector $v \in \mathbb{R}^d$. The minimization problem (7) can then be decomposed into d separate optimization problems, each one with a solution set $[\mathcal{X}^{\text{opt}}]_k$, $k \in \{1, \dots, d\}$, and each one corresponding to a 1-D localization problem of the form

$$\min_{x_V, x_1 = 0} \sum_{(i.j)\in E} \mid x_j - x_i - \epsilon_{ij} \mid . \tag{8}$$

We postpone to Section IV-D the discussion of how to combine the results of our analysis from the different dimensions; until that section, we exclusively focus on the 1-D version of the problem.

C. Canonical Linear Program Form

In this section, we transform (8) into the equivalent standard Linear Program (LP) form, with a linear cost function subject to linear inequality constraints, and compute its dual. This will allow us to arrive to the conclusion that the exact magnitude of the outliers is not important in terms of verifiability, and only the signed outlier support matter.

We first introduce variables

$$Z_{ij} = |x_j - x_i - \epsilon_{ij}|, \ \ \forall (i,j) \in E, \tag{9}$$

to push the cost function into the constraints.

$$\min_{Z_E, x_V, x_1 = 0} \sum_{(i,j)\in E} Z_{ij} \tag{10a}$$

$$\text{subject to } x_j - x_i - \epsilon_{ij} \le Z_{ij}, \tag{10b}$$

$$-(x_j - x_i - \epsilon_{ij}) \le Z_{ij}, \tag{10c}$$

$$Z_{ij} \ge 0, \tag{10d}$$

$$\forall i \in V, (i,j) \in E.$$

Next, in order to obtain a standard LP form, all variables must be non-negative. We therefore split each variable x_i into the summation of two non-negative variables,

$$x_i = x_i^+ - x_i^-, \ x_i^+, x_i^- \ge 0. \tag{11}$$

Finally, we change the inequality constraints into equality constraints by introducing the slack variables S_E^+, S_E^-:

$$\min_{Z, x, x_1 = 0} \sum_{(i.j)\in E} Z_{ij}, \tag{12a}$$

$$\text{subject to } x_j^+ - x_j^- - (x_i^+ - x_i^-) - \epsilon_{ij} + S_{ij}^+ = Z_{ij}, \tag{12b}$$

$$-(x_j^+ - x_j^- - (x_i^+ - x_i^-) - \epsilon_{ij}) + S_{ij}^- = Z_{ij}, \tag{12c}$$

$$x_i^+, x_i^-, S_{ij}^+, S_{ij}^-, Z_{ij} \ge 0, \tag{12d}$$

$$\forall i \in V, \ (i,j) \in E.$$

Remark 1 (Value of S_E): If we add constraints (12b) and (12c), we obtain

$$S_{ij}^+ + S_{ij}^- = 2Z_{ij}. \tag{13}$$

Moreover, from (9) and (13),

$$(S_{ij}^+, S_{ij}^-) = \begin{cases} (2Z_{ij}, 0), & \text{if } Z_{ij} = -(x_j - x_i - \epsilon_{ij}) \\ (0, 2Z_{ij}), & \text{if } Z_{ij} = x_j - x_i - \epsilon_{ij}. \end{cases} \tag{14}$$

We can also form the dual optimization problem of (12),

$$\max_{P_{ij}^+, P_{ij}^-} \quad \sum_{(i,j)\in E} \epsilon_{ij}(P_{ij}^+ - P_{ij}^-), \tag{15a}$$

$$\text{subject to} \sum_{j,(j,i)\in E} (P_{ji}^+ - P_{ji}^-) - \sum_{j,(i,j)\in E} (P_{ij}^+ - P_{ij}^-) = 0, \tag{15b}$$

$$-P_{ij}^+ - P_{ij}^- \leq 1, \tag{15c}$$

$$P_{ij}^+, P_{ij}^- \leq 0, \tag{15d}$$

$$\forall i \in V, (i,j) \in E,$$

where P_{ij}^+ is the dual variable associated to constraint (12b), and P_{ij}^- is the dual variable associated to constraint (12c).

Remark 2 (Strong duality and verifiability): Assume that the localization problem (G, E_ϵ) is verifiable or uniquely verifiable. Then, the origin is primal optimal, i.e., $\mathbf{0}_{|V|} \in \mathcal{X}^{\text{opt}}$, and from (9), we have that, at the primal optimal solution $(X^* = 0, Z_E^*, S_E^{+*}, S_E^{-*})$:

$$\sum_{(i,j)\in E} Z_{ij}^* = \sum_{(i,j)\in E} |\epsilon_{ij}| = \sum_{(i,j)\in E_\epsilon^\pm} |\epsilon_{ij}|; \tag{16}$$

note that, in the last equality, the sum is only over edges in the outlier support.

If a linear programming problem has an optimal solution, so does its dual, and the respective optimal costs are equal; this is known as the strong duality property [5, Theorem 4.4]. Combining this observation with (16), we have that, for a dual optimal solution (P_E^{+*}, P_E^{-*}),

$$\sum_{(i,j)\in E} Z_{ij}^* = \sum_{(i,j)\in E_\epsilon^\pm} \epsilon_{ij}(P_{ij}^{+*} - P_{ij}^{-*}) = \sum_{(i,j)\in E_\epsilon^\pm} |\epsilon_{ij}|. \tag{17}$$

Remark 3 (Discrete optimal solution for dual variables): Note that constraints (15c) and (15d), together with (17) imply that the dual optimal solution is given by $(P_{ij}^{+*}, P_{ij}^{-*}) \in \{(-1,0),(0,-1)\}$, for all $(i,j) \in E_\epsilon^\pm$ (i.e., there are two discrete cases for each edge with outliers, and the selection depends on the sign of ϵ_{ij}), and $-1 \leq P_{ij}^{+*}, P_{ij}^{-*} \leq 0$ for the remaining edges.

These remarks allow us to prove the following.

Lemma 3.2: For a fixed outlier support E_ϵ, if we change the scale of the outliers by positive factor, the verifiability of the graph does not change.

Proof: Assume that the localization problem $(G, E_\epsilon^\pm)$ is verifiable or uniquely verifiable, and that $(X_V^* = 0, Z_E^*, S_E^*)$ is a primal optimal solution, while (P_E^{*+}, P_E^{*-}) is a dual optimal solution. If we replace each outlier ϵ_{ij} with a positively scaled version $u_{ij}\epsilon_{ij}$, $u_{ij} > 0$, $(i,j) \in E$ (the case $u_{ij} = 0$ is excluded, otherwise the outlier support would change), the cost function in (15) changes, but not the constraints, so (P_E^{*+}, P_E^{*-}) is still a dual feasible solution. Considering the second equality in (17) from Remark 2 together with Remark 3, we have that the new dual cost after rescaling is

$$\sum_{(i,j)\in E} u_{ij}\epsilon_{ij}(P_{ij}^{+*} - P_{ij}^{-*}) = \sum_{(i,j)\in E_\epsilon^\pm} u_{ij}|\epsilon_{ij}|. \tag{18}$$

At the same time, the solution $(X_V^* = 0, \{u_{ij}Z_{ij}^*\}_{(i,j)\in E}, \{u_{ij}S_{ij}^*\}_{(i,j)\in E})$ is primal feasible, and the corresponding cost is

$$\sum_{(i,j)\in E} u_{ij}Z_{ij}^* = \sum_{(i,j)\in E} u_{ij}|\epsilon_{ij}|. \tag{19}$$

From (18) and (19) together with strong duality, we can therefore conclude that $(X_V^* = 0, \{u_{ij}Z_{ij}^*\}_{(i,j)\in E}, \{u_{ij}S_{ij}^*\}_{(i,j)\in E})$ (respectively, (P_E^{*+}, P_E^{*-})) is primal (respectively, dual) optimal. This shows that $X_V^* = 0$ is an optimal solution, and the rescaled problem is again verifiable; hence, one problem is verifiable if and only if all the positive scaled versions are also verifiable. ■

Combining lemmata 3.1 and 3.2 we have the following:

Theorem 3.3: The notion of verifiability depends only on the graph topology G, the support of the outliers E_ε , and the sign of the outliers.

Technically speaking, the proof above does not cover the case of unique verifiability, in the sense that the they do not exclude the case where a verifiable problem might become uniquely verifiable after rescaling (or viceversa). We are investigating this issue in our current work.

IV. Verifiability computation

A. Linear Programming

In this section, we discuss how the dual simplex algorithm can be used to compute the verifiability of a given problem. As a result of the previous section, for our analysis, the values of ϵ_E can be choosen randomly, as long as they have the correct edge support $E_\varepsilon^\pm$. We start by rewriting the LP (12) in matrix form:

$$\begin{aligned} \min_q \quad & c^{\mathrm{T}} q \\ \text{subject to} \quad & Aq = b \\ & q \geq 0. \end{aligned} \tag{20}$$

The vector $c = \text{stack}(\mathbf{0}_{2|V|}, \mathbf{1}_{|E|}, \mathbf{0}_{2|E|})$ contains the set coefficients in the cost function, while $A \in \{0,1,-1\}^{2|E|\times(2|V|+3|E|)}$, and $b = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right] \otimes \epsilon_E$ defines the constraints (where $\otimes$ denotes the Kronecker's product). Finally, the vector $q = \text{stack}(x_V^+, x_V^-, Z_E, S_E^+, S_E^-) \in \mathbb{R}^{2|V|+3|E|}$ contains the decision variables.

Given the standard form of the optimization problem (20), we can use the dual simplex algorithm [5] to find all the corners of the set of minimizers $\mathcal{X}^{\text{opt}}$. Next, we present the algorithm and application to our problem.

	0-th col.	x_V^+	x_V^-	Z_E	S_E^+	S_E^-
0-th rowx	0	$\mathbf{0}_V$	$\mathbf{0}_V$	$\mathbf{1}_E$	$\mathbf{0}_E$	$\mathbf{0}_E$
$q_{B(1)}$	$b(1)$	|	|	|	|	|
$q_{B(2)}$	$b(2)$	$a_{x_V^+}$	$a_{x_V^-}$	a_{Z_E}	$a_{S_E^+}$	$a_{S_E^-}$
$\vdots$	$\vdots$	|	|	|	|	|
$q_{B(2\|E\|)}$	$b(2\|E\|)$					

Fig. 1: Initial simplex tableau, with labeled rows and columns

B. Localization Via the Dual Simplex Method

The dual simplex method is based on the following concepts:

1) Basic variables (BVs): a subset of variables (q_B), that, together with the constraints, defines the current candidate solution in the algorithm. Non-basic variables (NBV) are always zero.
2) Simplex tableau: a $(2|E|+1) \times (2|V|+3|E|-1)$ array where
   - The zeroth column represents the value of the set of basic variables (q_B). It is initialized with the vector b.
   - The zeroth row contains the *reduced costs*, which are defined as the penalty cost for introducing one unit of the variable q_i to the cost. These are initialized with the vector c.
   - Columns one to $2(|V|-1)+3|E|$ are each one associated with one variable, where we excluded the columns corresponding to x_1^+, x_1^-, since x_1 is fixed in the optimization. These columns are initialized with the matrix A.

For our initial estimated solution, we set all variables to zero except the slack variables; as a result, our initial BVs correspond to the set of slack variables, while the rest are NBVs. See Fig. 1 for an illustration of the initial tableau.

A typical iteration starts with some basic variables containing negative elements, and all reduced costs non-negative. For instance, in Fig. 1, the initial BVs are selected to be slack variables where $S_{ij}^+ = \epsilon_{ij}$ and $S_{ij}^- = -\epsilon_{ij}$, hence, there are some negative initial BVs, while all reduced costs are non-negative (as all elements of vector c are non-negative). These two properties are always maintained by the algorithm from one iteration to the next.

The iterations of the algorithm then follow these steps:

1) *Check for termination due to optimality:* Examine the elements of zeroth column (which constitutes the basic set). If all of them are non-negative, we have an optimal basic solution and the algorithm terminates.
2) *Choose pivot row:* Find some ν such that $[q_B]_\nu < 0$.
3) *Check for termination due to unbounded solution:* Considering the ν-th row of the tableau, with elements $r_1, \ldots, r_{2(|V|-1)+3|E|}$, if all the elements of the row are non-negative, the optimal dual cost is $+\infty$ and algorithm terminates. Since the set of minimizers $\mathcal{X}^{\text{opt}}$ in our problem is bounded (see Section II-D), this condition is never encountered in our application.
4) *Choose pivot column:* For each i such that $r_i < 0$, compute the ratio $\bar{c}_i/|r_i|$ where $\bar{c}_i$ is the reduced cost of variable q_i and let j be the index of a column that correspond to the smallest ratio.
5) *Pivoting:* Remove the variable $[q_B]_\nu$ from the basis, and have variable q_j take its place. Add to each row of the tableau a multiple of the ν-th row (pivot row) so that r_j (the pivot element) becomes 1 and all other entries of the pivot column become 0. As a result, the total cost is reduced by the reduced cost $\bar{c}_j$.
6) Repeat the algorithm from step 2 until all elements of q_B are non-negative or the algorithm otherwise terminates.

After solving the simplex tableau, we get the basic optimal solution, which contains non-negative elements, together with non-negative reduced costs. The solution of the dual simplex algorithm is an optimal solution for (20), and is a corner point of the feasible region (Theorem 2.3, [5]). If we have multiple optimal solutions (i.e., $\mathcal{X}^{\text{opt}}$ is not a singleton), there will be multiple other corners with the same cost.

Hence, it is of interest to computationally enumerate all the corners of $\mathcal{X}^{\text{opt}}$, as discussed next.

C. Characterizing $\mathcal{X}^{opt}$ And Verifiability

The LP problem 20 can have multiple optimal solutions only when two conditions are met [2]:

1) There exists a non-basic variable with zero reduced cost. Pivoting this variable into the basis would not change the value for the cost function.
2) There exists a degenerate basic solution, i.e. some basic variables are equal to zero.

If the two conditions above are met, the corners in $\mathcal{X}^{\text{opt}}$ can be enumerated using a depth first search [26]:

1) Prepare a queue Q of corners to visit, with the corresponding tableau, and initialize it with the current solution found by the dual simplex algorithm,
2) For each corner in Q and its associated tableau,
   a) Choose C_{col} as the set of columns associated to non-basic variables with zero reduced cost, for all $j \in C_{col}$,
      i) Choose C_{row} as the set of elements of the j-th pivot column which are positive,
      ii) For $i \in C_{row}$, we perform the pivoting, so that the pivot element in i-th row and j-th column becomes 1 and all other entries of the pivot column become 0,
      iii) Add the current corner to the queue Q, if is not in it already,
3) Go to step 2 until the queue Q is empty.

Remark 4: In terms of our localization problems, the pivoting variables and the motion from one corner of $\mathcal{X}^{\text{opt}}$ to another can be given a physical interpretation. We defined as Z_{ij} the cost of edge (i,j). Assuming we have a verifiable graph, from (16), the cost of edge (i,j) is equal to $|\epsilon_{ij}|$. When we move (pivot) to another corner with the same cost, the set of basic variables changes, but the value of all the other variables remains the same. So, if a non-basic variable takes the place of basic variables from the set x_V^+ or x_V^-, it does not

produce a new optimal embedding (because such variables where already equal to zero). If a pivoting variable takes the place of non-zero basic variable Z_{ij}, then Z_{ij} becomes zero, which means the cost of edge (i,j) changes to zero, and if $\epsilon_{ij} \neq 0$ then from (16), x_i and x_j are not equal to zero anymore. As the value of cost function remains the same, the loss of cost of edge (i,j) must be compensated with the costs of the rest of the edges. If we pivot a non-basic variable to the non-zero basic variable S^+_{ij} or S^-_{ij}, from (13), it implies the value of Z_{ij} becomes zero which means the cost of edge (i,j) changes to zero. So, pivoting non-basic variable in order to find alternative solutions means shifting the cost of outliers from one edge to the others.

There are three cases for the set of optimal solutions, $\mathcal{X}^{\text{opt}}$:

1) *Uniquely verifiable solution:* Pivoting new variables to the basis does not result in new corner point; we therefore have a unique optimal solution $\mathcal{X}^{\text{opt}} = \{\mathbf{0}_V\}$, and from (4) we conclude that the resulting embedding is congruent to the ground truth.
2) *Verifiable (non-unique) solution:* We have multiple optimal solutions, including the origin ($\mathbf{0}_V \in \mathcal{X}^{\text{opt}}$); hence, there are multiple optimal embeddings, with one of them being congruent to the ground truth.
3) *Non-verifiable:* In this case, $\mathbf{0}_V \notin \mathcal{X}^{\text{opt}}$, and the ground truth embedding is not an optimal solution.

D. Combining Solutions From Multiple Dimensions

In Section III-B, we reduced one d-dimensional optimization problem of the form (6) to d 1-D optimization problems of the form (8). Now, we need to combine the optimal solutions of all dimensions to characterize the d-dimensional optimal solution. Let $[\mathcal{X}^{\text{opt}}]_k$ represents the set of optimal solutions for the LP (10) of dimension k. The value of the cost function (10) is the same for all corner points in $[\mathcal{X}^{\text{opt}}]_k$. Due to this fact, we can pick a 1-D corner point from each set $[\mathcal{X}^{\text{opt}}]_k$, $k \in \{1,\dots,d\}$, and combine them to build a d-dimensional corner point:

$$x^{opt} = \text{stack}(X_1^{opt},\dots,X_d^{opt}),\ X_k^{opt} \in [\mathcal{X}^{\text{opt}}]_k. \tag{21}$$

Let $|[\mathcal{X}^{\text{opt}}]_k|$ represents the cardinality of the set $[\mathcal{X}^{\text{opt}}]_k$; then, we have $N = \prod_{k=1}^{d} \left|[\mathcal{X}^{\text{opt}}]_k\right|$ d-dimensional corner points. To have a unique verifiable graph, we therefore need all the individual 1-D problems to be also unique verifiable, i.e. $|[\mathcal{X}^{\text{opt}}]_k| = 1$ for all $k \in \{1,\dots,d\}$.

E. Maximal verifiable components

If for all corners a subset of components V' in the solution are always zero (i.e., $[X_k^{opt}]'_V = 0$ for all k), then the position of those particular nodes, and all their relative positions, are congruent to the true embedding. As a consequence, also all their relative costs are the same. Hence, while the entire problem $G, E_\epsilon^\pm$ is not verifiable, the sub-problem $G', E_\epsilon'^\pm$, where $G' = (V', E')$, $E' = \{(i,j) \in E : i,j \in V'\}$ is verifiable. We call the maximal connected components of G' defined in this way the *maximal verifiable components* of G.

V. Verifiability Probability

Given a tuple $(G, E_\epsilon^\pm)$ of a graph and a signed outlier support, we can define a function that indicate if the associated localization problem is verifiable,

$$\text{Ver}(G, E_\epsilon^\pm) = \begin{cases} 1 & \text{if } 0 \in X^{\text{opt}} \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

This function can be implemented by using the dual simplex algorithm discussed above.

Given the edge outlier probabilities xx defined in (2), we can take the expectation of $\text{Ver}(G,\cdot)$ over different outlier realizations, and hence characterize the a priori probability of recovering a localization that is cost-equivalent to the true one, without knowing the exact value or support of the outliers.

Definition 6: We define the *verifiability probability* p_{Ver} as the probability of recovering a solution whose cost is the same as the ground truth, i.e., $p_{\text{Ver}} = \mathbb{E}_\epsilon[\text{Ver}(G, E_\epsilon^\pm)]$, where $\mathbb{E}_\epsilon[\cdot]$ is the expectation over all the realizations of outliers.

The interpretation of this number is the a priori probability that the ground truth embedding X_V^* belongs to $\mathcal{X}^{\text{opt}}$, the set of minimizers of (3). For instance, if we assume the edge positive outlier probability is p^+, and the edge negative outlier probability is p^-, then we can define $p(\epsilon_E) = (p^+)^{|E_\epsilon^+|}(p^-)^{|E_\epsilon^-|}(1-p^+-p^-)^{(|E|-|E_\epsilon^-|-|E_\epsilon^+|)}$ and $p_{\text{Ver}} = \mathbb{E}_\epsilon[\text{Ver}(E, E_\epsilon^\pm)]$.

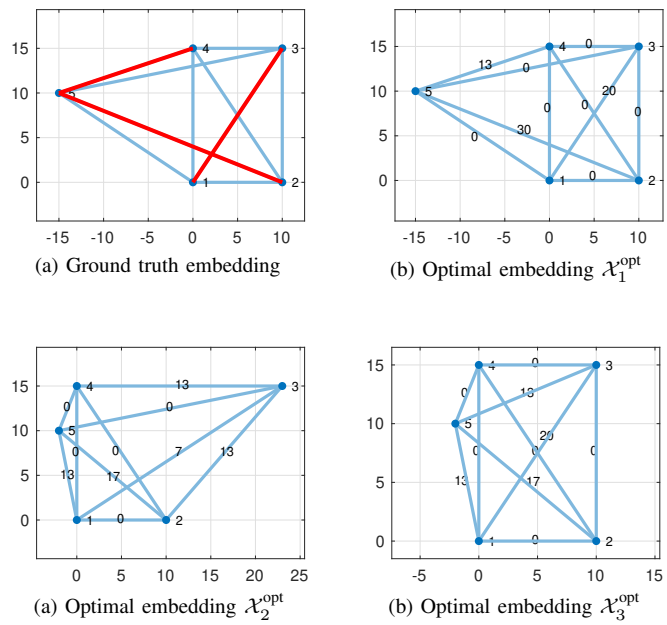
Note that an analogous quantity could be computed for unique verifiability, although we would need to expand our results to make this rigorous (see comments immediately after Theorem 3.3). Moreover, a similar concept could be extended to each individual edge, or any arbitrary subset of edges, by asking whether they are part of a maximal verifiable component (Section IV-E). Nonetheless, a formal exploration of these concepts is out of the scope of the present paper.

VI. Numerical Examples

In this section we apply our theory and algorithm[1] to a simple graph with 5 nodes and 10 edges, $\mathbf{X}_V \in \mathbb{R}^{5\times 2}$ (Fig. 2). We start with the case where three relative measurements in first coordinate are outliers and all other measurements (Fig. 2a) are accurate. In this example, positive and negative outlier have the same probability $p^+ = p^- = \frac{1}{2}p$. After solving the optimization problem associated to this graph, we find three different embeddings that represent the corners of $\mathcal{X}^{\text{opt}}$; these are shown in Fig. 2b, 2a and 2b.

In Fig. 2b, the resulted embedding is identical to the ground truth embedding, which means that $\mathbf{x}_V \in \mathcal{X}^{\text{opt}}$, and the graph is verifiable. However, since we have multiple solution, the graph is not uniquely verifiable. In the figures, the cost of associated to each edge is shown; it can be seen that different corners shift the cost to different edges, although their sum remains the same. The locations of nodes $V' = \{1,2,4\}$ are identical to their ground truth locations, and the costs of edges $E' = \{(4,1),(2,1),(2,4)\}$ remain the same in all

[1]The algorithm is implemented in MATLAB at thttps://github.com/Mahrooo/Robust-Localization-Verifiability.git



(a) Ground truth embedding (b) Optimal embedding $\mathcal{X}_1^{\text{opt}}$

(a) Optimal embedding $\mathcal{X}_2^{\text{opt}}$ (b) Optimal embedding $\mathcal{X}_3^{\text{opt}}$

Fig. 2: Verifiable graph with 5 nodes and 10 edges, 3 edges are outliers are shown by red color in Fig. 2a, the cost of each edge is shown on each and the cost of optimal solution for all embeddings are equal to the cost of the ground truth embedding which is 63

embeddings, so the subgraph $G = (V', E')$ is a maximal verifiable component.

Assuming that the edge outlier probability p_{ij} is $\frac{1}{2}p$ for all edges $(i, j) \in E$, then for our graph in this example the verifiability probability for this graph can be evaluated as

$$\begin{aligned} p_{\text{Ver}} &= (1-p)^{10} + 20(\frac{p}{2})(1-p)^9 + 180(\frac{p}{2})^2(1-p)^8 \\ &+ 920(\frac{p}{2})^3(1-p)^7 + 2680(\frac{p}{2})^4(1-p)^6 + 4524(\frac{p}{2})^5(1-p)^5 \\ &+ 4560(\frac{p}{2})^6(1-p)^4 + 2820(\frac{p}{2})^7(1-p)^3 \\ &+ 1080(\frac{p}{2})^8(1-p)^2 + 240(\frac{p}{2})^9(1-p) + 24(\frac{p}{2})^{10}, \end{aligned} \tag{23}$$

where the coefficients come from Table I. As shown in Fig. 3, if $p = 0$ we have a verifiable graph with probability $p_{\text{Ver}} = 1$;

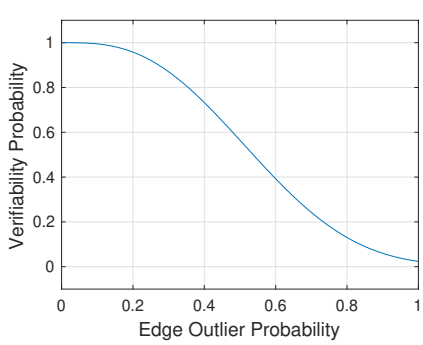


Fig. 3: Verifiability probability for the graph in Fig 2

TABLE I: Verifiability analysis for all possible cases of outlier supports $E_\varepsilon^\pm$

#outliers, $\lvert E_\epsilon^\pm \rvert$	#possible combinations, $\binom{\lvert E \rvert}{\lvert E_\epsilon^\pm \rvert}$	#verifiable combinations
0	1	1
1	20	20
2	180	180
3	960	920
4	3360	2680
5	8064	4524
6	13440	4560
7	15360	2820
8	11520	1080
9	5120	240
10	1024	24

as we increase the probability of more edges to be outliers, the probability of having access to the verifiable graph decreases.

VII. Conclusions And Future Works

In this work, we consider the estimation of an embedding for nodes with relative translation measurements affected by outliers (but no noise) through the minimization of an ℓ_1-norm cost function. We introduce the notion of verifiability, which characterizes when we can expect to recover a solution with cost equal to the true one; we show that the concept of verifiability depends only on the topology of the network and where the outliers are placed, and we also provide a way to compute it using the dual simplex method. From a more practical standpoint, we define the verifiability probability, which characterizes the a priori reliability that can be expected from a given measurement graph (given a priori probabilities of outliers for each edge). There are many possible directions for our future work. First, we plan to include the effects of amplitude-limited noise to our measurement models, and study its effect of noise on our results; concurrently, we will study different cost functions, such as the Huber-loss function and piece-wise linear loss functions.

References

[1] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Robust map optimization using dynamic covariance scaling. In *IEEE International Conference on Robotics and Automation*, pages 62–69, May 2013.

[2] G. Appa. On the uniqueness of solutions to linear programs. *Journal of the Operational Research Society*, 53(10):1127–1132, 2002.

[3] M. Arie-Nachimson, S. Z. Kovalsky, I. Kemelmacher-Shlizerman, A. Singer, and R. Basri. Global motion estimation from point matches. In *Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 81–88. IEEE, 2012.

[4] A. Beck. *Introduction to nonlinear optimization: theory, algorithms, and applications with MATLAB*, volume 19. Siam, 2014.

[5] D. Bertsimas and J. N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[6] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye. Semidefinite programming based algorithms for sensor network localization. *ACM Transactions on Sensor Networks (TOSN)*, 2(2):188–220, 2006.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[8] F. Crosilla, A. Beinat, A. Fusiello, E. Maset, and D. Visintini. Advanced procrustes analysis models in photogrammetric computer vision.

[9] M. Cucuringu, Y. Lipman, and A. Singer. Sensor network localization by eigenvector synchronization over the euclidean group. *ACM Transactions on Sensor Networks (TOSN)*, 8(3):19, 2012.

[10] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the rgb-d SLAMsystem. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 1691–1696, 2012.

[11] T. Eren, W. Whiteley, A. S. Morse, P. N. Belhumeur, and B. D. Anderson. Sensor and network topologies of formations with direction, bearing, and angle information between agents. In *IEEE International Conference on Decision and Control*, volume 3, pages 3064–3069. IEEE, 2003.

[12] T. Erseghe. A distributed and scalable processing method based upon admm. *IEEE Signal Processing Letters*, 19(9):563–566, 2012.

[13] V. M. Govindu. Combining two-view constraints for motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–II, Dec 2001.

[14] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, winter 2010.

[15] P. Hand, C. Lee, and V. Voroninski. Shapefit: Exact location recovery from corrupted pairwise directions. *Communications on Pure and Applied Mathematics*, 71, 06 2015.

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[17] P. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics-theory and Methods - COMMUN STATIST-THEOR METHOD*, 6:813–827, 01 1977.

[18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. G2o: A general framework for graph optimization. In *IEEE International Conference on Robotics and Automation*, pages 3607–3613, May 2011.

[19] G. Lerman and Y. Shi. Estimation of camera locations in highly corrupted scenarios: All about that base, no shape trouble. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2868–2876, 06 2018.

[20] A. Mitiche. *Computational analysis of visual motion*. Springer Science & Business Media, 2013.

[21] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *IEEE International Conference on Computer Vision*, pages 3248–3255, 2013.

[22] O. Ozyesil, A. Singer, and R. Basri. Stable camera motion estimation using convex programming. *SIAM Journal on Imaging Sciences*, 8, 07 2014.

[23] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer. A survey on structure from motion. *Acta Numerica*, 26, 01 2017.

[24] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph slam. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1879–1884, Oct 2012.

[25] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph slam. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1879–1884, Oct 2012.

[26] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[27] R. Tron and R. Vidal. Distributed image-based 3-d localization of camera sensor networks. In *IEEE International Conference on Decision and Control*, pages 901–908, Dec 2009.

[28] R. Tron and R. Vidal. Distributed 3-d localization of camera sensor networks from 2-d image measurements. *IEEE Transactions on Automatic Control*, 59(12):3325–3340, Dec 2014.

[29] W. N. Venables and B. D. Ripley. *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.

[30] K. Wilson and N. Snavely. Robust global translations with 1DSfM. In *IEEE European Conference on Computer Vision*, volume 8691, pages 61–75, 09 2014.

[31] Z. Yang, C. Wu, T. Chen, Y. Zhao, W. Gong, and Y. Liu. Detecting outlier measurements based on graph rigidity for wireless sensor network localization. *IEEE Transactions on Vehicular Technology*, 62(1):374–383, Jan 2013.

[32] C. Zach. Robust bundle adjustment revisited. In *IEEE European Conference on Computer Vision*, pages 772–787, 09 2014.

[33] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433. IEEE, 2010.

[34] O. Özyeşil and A. Singer. Robust camera location estimation by convex programming. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2674–2683, June 2015.