

# Rigid Components Identification and Rigidity Enforcement in Bearing-Only Localization using the Graph Cycle Basis

Roberto Tron, Luca Carlone, Frank Dellaert, Kostas Daniilidis

**Abstract**—Bearing-only localization can be formulated in terms of *optimal graph embedding*: one has to assign a 2-D or 3-D position to each node in a graph while satisfying as close as possible all the bearing-only constraints on the edges. If the graph is *parallel rigid*, this can be done via spectral methods. When the graph is not rigid the reconstruction is ambiguous, as different subsets of vertices can be scaled differently. It is therefore important to first identify a partition of the problem into *maximal rigid components*. In this paper we show that the cycle basis matrix of the graph not only translates into an algorithm to identify *all* rigid sub-graphs, but also provides a more intuitive way to look at graph rigidity, showing, for instance, why triangulated graphs are rigid and why graphs with long cycles may lose this property. Furthermore, it provides practical tools to enforce rigidity by adding a minimal number of measurements.

## I. INTRODUCTION

The essence of the bearing-only (also known as angle-of-arrival) localization problem is to estimate the positions of points from linear, pair-wise direction constraints. The problem is most conveniently modeled by a graph, where the points are associated to the nodes, and the direction constraints to the edges. As such, this problem has appeared under various forms in different settings, such as (to cite a few) sensor network localization [1], [5], [15] and formation control [3], [6], [14], [18] in the controls and robotics communities, Structure from Motion in computer vision [4], [11], and graph drawings [13] in the discrete mathematics community. A schematic example is given in Figure 1.

In this paper we focus only on the *absolute* version of the problem, that is, we assume that all the bearing directions are expressed with respect to a common rotational reference frame. For the *relative* version of the problem, where the only information available is between pairs of bearings, we point the reader to [10] and the references therein.

A central notion for the bearing localization problem is the one of rigidity. Loosely speaking, a bearing-only problem is rigid when there are enough direction constraints such that all the feasible solutions are equivalent up to trivial transformations (global translations and scales). Then, the direction constraints can be used to build an homogeneous system of linear equations, and a solution can be found by looking at a one-dimensional nullspace of a constraint

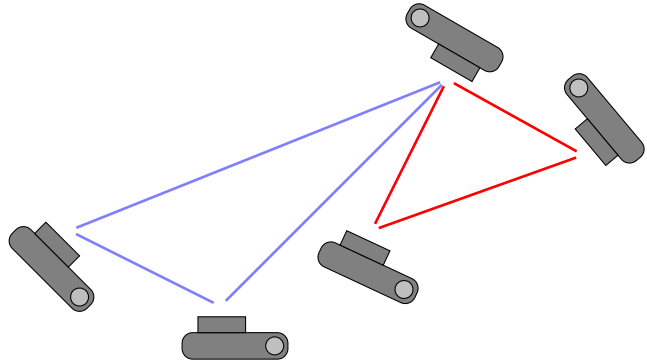


Fig. 1. A schematic example of bearing-only localization from computer vision. Using standard techniques, cameras with overlapping fields of view can measure their relative pose up to a scale for the translation. Each set of edges (blue and red) forms a rigid component where the position of the cameras can be found up to a common scale. However, the scale between the two components cannot be established from the available measurements.

matrix. On the other hand, when the direction constraints are not enough, the problem is *flexible* (i.e., non-rigid), and the above solution strategy fails because there are different solutions which are not trivially equivalent. Algebraically, this correspond to the fact that the dimension of the nullspace of the constraint matrix is greater than one. In this case, the best one can do is to segment the problem into the largest *rigid components* which can be solved independently.

To the best of our knowledge, the first algorithm for solving this segmentation task has been proposed in [9]. In that paper, the authors show how to extract the segmentation from the nullspace of the constraint matrix. The algorithm can be thought as *node-based*: the nullspace characterizes the space of solutions in terms of node positions, and the algorithm reasons on the clustering by comparing edges which have an endpoint in common.

In this paper, we propose a novel formulation and a novel solution which are *edge-based* and that can be considered “dual” to the ones of [9]. Instead of representing the space of feasible solutions using node positions, we use scales on the edges. The idea is simple: if two edges belong to the same rigid component they cannot be scaled independently, and fixing one edge uniquely determine the scale of the other edge in the same rigid component (although there are some additional subtle aspects, as we will explain in Section IV). This new representation not only provides an alternative way to identify the rigid components, but it also allows us to reason about the rigidity of the problem in terms of *cycles*

R. Tron and K. Daniilidis are with the Department of Computer and Information Science, University of Pennsylvania, USA, {tron,kostas}@seas.upenn.edu

L. Carlone and F. Dellaert are with the College of Computing, Georgia Institute of Technology, USA. luca.carlone@gatech.edu, frank@cc.gatech.edu

in the graph, thus providing insights into the reason why rigid graphs need to have small cycles, and on how one can add edges to make a problem rigid.

The rest of the paper is organized as follows. First, we review basic notions from graph theory (Section II) and the node-based localization problem for rigid problems (Section III). We then introduce our edge-based formulation (Sections IV and V), followed by our algorithm for identifying the rigid components of the problem (Section VI). Finally, we provide some insights on the relation between rigidity, long cycles, and the addition of edges (Section VII).

## II. ELEMENTS OF GRAPH THEORY

A directed graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is composed by a set of  $n$  vertexes  $\mathcal{V} = \{1, \dots, n\}$  and a set of  $m$  edges  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ . In this paper we assume that  $\mathcal{G}$  is *oriented*, that is, each edge appears only in one direction.

The *incidence matrix*  $\vec{B}$  of a directed graph is a  $n \times m$  matrix with elements in  $\{-1, 0, +1\}$  that describes the graph topology. Each column of  $\vec{B}$  corresponds to an edge and has exactly two non-zero elements. For the column corresponding to edge  $(i, j)$ , there is a  $-1$  on the  $i$ -th row and a  $+1$  on the  $j$ -th row.

A *walk* is an alternating sequence of vertices and edges, beginning and ending with a vertex, such that the vertices that follow and precede an edge are just the endpoint of that edge. A *cycle* is walk which starts and ends at the same vertex. A *circuit* is a cycle in which every node appears exactly once, except for the starting vertex which appears exactly twice. A circuit can be described by a vector in  $\{-1, 0, +1\}^m$ , in which an element is  $+1$  or  $-1$  if the corresponding edge is traversed respectively forwards (from tail to head) or backwards, and  $0$  if it does not appear.

A *cycle basis* of a graph is a minimal set of circuits such that any cycle in the graph can be written as a combination of the circuits in the basis. The number of independent circuits in the cycle basis is called *cyclomatic number* and, for a connected graph, it is equal to  $\ell = m - n + 1$ . A *cycle basis matrix* is a matrix  $C \in \{-1, 0, +1\}^{\ell \times m}$ , such that each row describes one of the circuits in the basis.

## III. PRELIMINARIES ON BEARING-ONLY LOCALIZATION

In this section we first formalize the bearing-only localization problem, and then review the standard definitions of rigidity together with the standard localization approach for rigid frameworks.

### A. Bearing-only Localization

We consider a formation of  $n$  robots at (unknown) positions  $\mathbf{x}_1, \dots, \mathbf{x}_n$ , with  $\mathbf{x}_i \in \mathbb{R}^d$ . The robots are able to acquire bearing measurements using on-board sensors (e.g., a camera). In particular, for some pairs  $(i, j)$ , robot  $i$  measures the bearing towards robot  $j$ :

$$\mathbf{u}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i + \epsilon_{ij}}{\|\mathbf{x}_j - \mathbf{x}_i + \epsilon_{ij}\|} \quad (1)$$

where  $\epsilon_{ij}$  models measurement noise. The measurement model (1) returns a unit vector, corresponding to a noisy

measurement of the translation (up to scale) between robots  $i$  and  $j$ . For the sake of analysis, we will often refer to the *noiseless* case, where  $\epsilon_{ij} = 0$  for all pairs  $(i, j)$ .

The objective of bearing-only localization is to estimate the positions  $\check{\mathbf{x}} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  from the measurements (1).

The problem can be conveniently summarized by a so-called *framework* or *design*  $\mathcal{F} = (\mathcal{G}, \mathcal{U})$  where

- $\mathcal{G}(\mathcal{V}, \mathcal{E})$  is a directed graph in which each node in  $\mathcal{V}$  is associated to a robot position  $\mathbf{x}_i$ , and the edge set  $\mathcal{E}$  contains the pairs  $(i, j)$  such that the relative measurement  $\mathbf{u}_{ij}$  is available.
- $\mathcal{U} = \{\mathbf{u}_{ij}\}_{(i,j) \in \mathcal{E}}$  is the set of bearing measurements.

A *subframework* of  $\mathcal{F}$  is a framework  $\mathcal{F}' = (\mathcal{G}', \mathcal{U}')$  where  $\mathcal{G}'$  is a subgraph of  $\mathcal{G}$  and  $\mathcal{U}' \subseteq \mathcal{U}$ .

A set of robot coordinates  $\check{\mathbf{x}} = \{\mathbf{x}_i\}_{i \in \mathcal{V}}$  is called an *embedding* for the framework  $\mathcal{F}$ . An *optimal embedding* or *drawing* of the framework is an embedding which minimizes a suitable error criterion with respect to the measurements.

### B. Rigid Components

The study of rigidity in bearing-only localization is connected to a practical question: do the available bearing measurements uniquely define robot positions?

Clearly we do not expect to be able to estimate absolute robot positions (as we only have relative measurements), and we cannot determine the scale (as we only have bearings).

Therefore, we need introduce the following notions.

*Definition 1 (Trivially parallel embeddings [5]):* We say that two embeddings are *trivially parallel* if they only differ by a global translation and scaling.

*Definition 2 (Rigidity [5]):* A framework is *parallel rigid* if all optimal embeddings are trivially parallel.

Intuitively, in a rigid framework, the measurements completely define robot positions up to scale and a global translation. A non-rigid framework is said to be *flexible*. In a flexible framework we can still identify rigid subframeworks.

*Definition 3 (Rigid components):* A *rigid component* of  $\mathcal{F}$  is a subframework  $\mathcal{F}' \subseteq \mathcal{F}$  such that  $\mathcal{F}'$  is rigid. A rigid component is said to be *maximal* if it is not a subframework of any other rigid component.

An intuitive way of thinking about rigid components is as follows: a subframework is rigid if after fixing the position of two nodes (which essentially fix the global frame and the scale of the embedding) all other nodes are uniquely identified by the bearing measurements.

We recall the following result from Kennedy *et al.* [9].

*Theorem 4:* The set of all maximally rigid components  $\mathcal{P} = \{\mathcal{G}'_i\}$  induces a *partition* of the original edge set  $\mathcal{E}$ .

Intuitively, Theorem 4 says that finding maximal rigid components is the same as finding suitable partitions of the edges, as each edge belongs to a single rigid component.

### C. Optimal Embedding in Rigid Frameworks

In the noiseless case, an optimal embedding  $\check{\mathbf{x}}$  must satisfy (1) for all the given measurements  $\mathcal{U}$ : i.e., for each edge  $(i, j)$ , the relative translation  $\mathbf{x}_j - \mathbf{x}_i$  induced by the

embedding should be collinear to the measured direction  $\mathbf{u}_{ij}$ :

$$\mathbf{u}_{ij} \times (\mathbf{x}_j - \mathbf{x}_i) = \mathcal{S}(\mathbf{u}_{ij})(\mathbf{x}_j - \mathbf{x}_i) = \mathbf{0} \quad \forall (i, j) \in \mathcal{E}, \quad (2)$$

where  $\times$  is the cross product (equal to zero when the vectors are parallel) and  $\mathcal{S}(\mathbf{u}_{ij})$  is a skew-symmetric matrix built from  $\mathbf{u}_{ij}$ . For instance, in a 3-D problem ( $d = 3$ ),  $\mathcal{S}(\mathbf{u}_{ij})$ :

$$\mathcal{S}(\mathbf{u}_{ij}) = \begin{bmatrix} 0 & -u_{ij}^z & u_{ij}^y \\ u_{ij}^z & 0 & -u_{ij}^x \\ -u_{ij}^y & u_{ij}^x & 0 \end{bmatrix} \quad (3)$$

with  $\mathbf{u}_{ij} = [u_{ij}^x \ u_{ij}^y \ u_{ij}^z]^\top$ .

In order to make the notation more compact, we let  $\mathbf{S} = \text{diag}(\{\mathcal{S}(\mathbf{u}_{ij})\}_{(i,j) \in \mathcal{E}}) \in \mathbb{R}^{3m \times 3m}$ , where  $m = |\mathcal{E}|$ , and introduce the *augmented incidence matrix*  $\check{\mathbf{B}}_d = \check{\mathbf{B}} \otimes \mathbf{I}_d$ , where  $\check{\mathbf{B}}$  is the incidence matrix of the graph  $\mathcal{G}$ , underlying the framework. With this notation, (2) becomes

$$\mathbf{S}\check{\mathbf{B}}_d\check{\mathbf{x}} = \mathbf{0}. \quad (4)$$

In a noiseless case, the set of solutions of (4) defines the optimal embeddings of the framework  $\mathcal{F}$ . Clearly, (4) will have infinite solutions, as the measurements define the embedding only up to scale and a global translation. In order to remove this ambiguity, the standard practice is to fix the scale of the reconstruction (e.g., by adding a constraint as  $\|\check{\mathbf{x}}\| = 1$ ), and set a node to the origin of the reference frame, i.e.,  $\mathbf{x}_1 = \mathbf{0}_3$ . This node is usually called an *anchor*. Fixing  $\mathbf{x}_1$  at the origin allows rewriting (4) as

$$\mathbf{S}\mathbf{B}_d\mathbf{x} = \mathbf{0}, \quad (5)$$

where the *reduced* incidence matrix  $\mathbf{B}_d$  is obtained by removing the first  $d$  columns of  $\check{\mathbf{B}}_d$  (corresponding to  $\mathbf{x}_1$ ). Similarly,  $\mathbf{x}$  is obtained by removing the anchor  $\mathbf{x}_1$  from  $\check{\mathbf{x}}$ .

In the noiseless case, the framework is rigid if and only if the matrix  $\mathbf{S}\mathbf{B}_d$  has exactly one singular value equal to zero, that is, its nullspace is one-dimensional. In this case, (5) admits a solution which is unique up to scale.

In the noisy case, eq. (5) admits no solution in general (except for the trivial solution  $\mathbf{x} = \mathbf{0}$ ), and one rather looks for a solution that minimizes the norm of the residual errors:

$$\min_{\mathbf{x} \in \mathbf{X}} \|\mathbf{S}\mathbf{B}_d\mathbf{x}\|^2 \quad (6)$$

where  $\mathbf{x}$  is restricted to a set  $\mathbf{X}$  to rule out the trivial solution  $\mathbf{x} = \mathbf{0}$ . In the existing literature, a common choice for  $\mathbf{X}$  is,

$$\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^{d(n-1)} : \|\mathbf{x}\| = 1\}. \quad (7)$$

This choice is sufficient for rigid frameworks (although it does not distinguish between  $\mathbf{x}$  and its opposite  $-\mathbf{x}$ ).

A more convenient choice, that we adopt in this paper, is

$$\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^{d(n-1)} : \mathbf{u}_{ij}^\top(\mathbf{x}_j - \mathbf{x}_i) \geq 1 \ \forall (i, j) \in \mathcal{E}\}. \quad (8)$$

where one enforces a minimum distance between the nodes along the measured direction. The set (8) removes the reflection ambiguity that would appear using (7). Moreover (8) defines a convex set, and this is advantageous from an

optimization point of view, compared with (7). For these reasons, variants of (8) are common in computer vision literature (see, e.g., [8], [16]), where they are often referred to as *chirality* constraints.

#### IV. A DIFFERENT VIEW ON RIGIDITY

Standard rigidity as discussed in Section III-B reasons in terms of nodes positions (the embedding) to define rigidity. In this paper we prefer to reason in terms of edges in the graph. We will do this using the notion of *interdependent* edges in Definition 6. Before doing that, we need to define the concept of *non-degenerate* embedding.

*Definition 5 (Non-degenerate embedding):* An embedding  $\check{\mathbf{x}} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  is said to be *non-degenerate* if for any pair  $(i, j) \in \mathcal{E}$ ,  $\mathbf{x}_i \neq \mathbf{x}_j$ .

Since two nodes can measure the relative bearing only if they are not collocated, it is natural to restrict the attention to non-degenerate embeddings in bearing-only localization. Now we are ready to define an *interdependent edge set*.

*Definition 6 (interdependent edge set):* A set of edges  $\mathcal{E}_R$  in a framework is said to be *interdependent* if, given a non-degenerate optimal embedding  $\check{\mathbf{x}}^a$  and a constant  $s$ , any other non-degenerate optimal embedding  $\check{\mathbf{x}}^b$ , is such that:

$$(\mathbf{x}_j^b - \mathbf{x}_i^b) = s(\mathbf{x}_j^a - \mathbf{x}_i^a) \quad \forall (i, j) \in \mathcal{E}_R \quad (9)$$

Definition 6 says that if a set of edges is interdependent we cannot change the corresponding inter-nodal distances independently, since, by fixing the scale of an edge, we also constrain the scale of the remaining edges in the set.

The concept of interdependent edge set is tightly coupled with the concept of rigidity in Definition 2 (at first sight the two may look identical). However, there is a subtle difference, shown in Figure 2: two sides of the trapezoid form a set of interdependent edges, while the corresponding set of nodes do not form a rigid component.

Fortunately, the two definitions are equivalent when the set of interdependent edges  $\mathcal{E}_R$  defines a connected subgraph.

*Proposition 7:* Given a framework  $\mathcal{F}(\mathcal{G}, \mathcal{U})$ , if a set of interdependent edges (Definition 6) forms a connected subgraph  $\mathcal{G}' \subseteq \mathcal{G}$ , then the subframework  $\mathcal{F}(\mathcal{G}', \mathcal{U}')$  is parallel rigid (Definition 2).

*Proof:* Let us prove the implication: rigidity  $\rightarrow$  interdependent edges. If the framework is rigid two optimal embeddings  $\check{\mathbf{x}}^a$  and  $\check{\mathbf{x}}^b$  can only differ by global translation and scaling, i.e., we can write  $\check{\mathbf{x}}^b = s\check{\mathbf{x}}^a + \mathbf{t}$ , for a suitable scaling  $s$  and translation  $\mathbf{t}$ . By inspection we easily see that this  $\check{\mathbf{x}}^b$  satisfies (9) hence the edges in the subframework are interdependent. The reverse implication can be proved in a

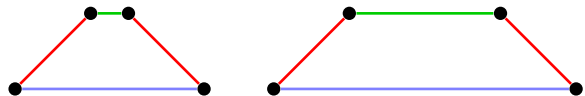


Fig. 2. Example of interdependent edges (edges of the same color are interdependent). While the red edges do not form a rigid component, they are interdependent: fixing the scale of the red edge on the left uniquely determines the scale of the red edge on the right.

similar way, observing that fixing the relative nodes positions as in (9) uniquely defines the position of the nodes as long as the graph is connected. ■

The idea is then to reason in terms of interdependent edges, rather than on the original definition of rigidity. This is convenient from a computation perspective and offers a more intuitive way to look at rigid components. Towards this goal, we first need to reformulate the bearing localization in terms of the scales for each edge.

## V. EDGE-BASED FORMULATION

In this section we propose a different formulation of the bearing-only localization problem. While later we prove that this formulation is equivalent to the standard of Section III-C, it is propaedeutic to our algorithm for the identification of the rigid components (Section VI).

Instead of writing the parallelism constraint as in (2), we note that the parallelism between  $\mathbf{x}_j - \mathbf{x}_i$  and  $\mathbf{u}_{ij}$  requires that, for some unknown scale  $\lambda_{ij}$ , it holds:

$$(\mathbf{x}_j - \mathbf{x}_i) - \lambda_{ij}\mathbf{u}_{ij} = \mathbf{0}, \quad \forall (i, j) \in \mathcal{E} \quad (10)$$

In matrix form, this becomes

$$\check{\mathbf{B}}_d^\top \check{\mathbf{x}} - \mathbf{U}\boldsymbol{\lambda} = \mathbf{0}, \quad (11)$$

where,  $\mathbf{U} = \text{diag}(\{\mathbf{u}_{ij}\}_{(i,j) \in \mathcal{E}}) \in \mathbb{R}^{dm \times m}$  is a sparse matrix with diagonal blocks equal to the vectors  $\mathbf{u}_{ij}$ . Note that since the measurements have unit norm,  $\mathbf{U}$  has orthonormal columns, i.e.,  $\mathbf{U}^\top \mathbf{U} = \mathbf{I}_m$ . In this formulation, we compute the scale vector  $\boldsymbol{\lambda} \doteq \{\lambda_{ij}\}_{(i,j) \in \mathcal{E}}$  in addition to  $\check{\mathbf{x}}$ .

As in the standard formulation of Section III-C, we remove the translational ambiguity by fixing  $\mathbf{x}_1 = \mathbf{0}_d$ , which amounts to substituting  $\check{\mathbf{B}}$  and  $\check{\mathbf{x}}$  with their reduced versions:

$$\mathbf{B}_d^\top \mathbf{x} - \mathbf{U}\boldsymbol{\lambda} = \mathbf{0}. \quad (12)$$

As before, in the noisy case, we replace the set of linear constraints with the least-squares minimization

$$\min_{\mathbf{x}, \boldsymbol{\lambda} \in \Lambda} \|\mathbf{B}_d^\top \mathbf{x} - \mathbf{U}\boldsymbol{\lambda}\|^2, \quad (13)$$

where, to rule out trivial solutions, we restrict  $\boldsymbol{\lambda}$  in

$$\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^m : \lambda_{ij} \geq 1 \forall (i, j) \in \mathcal{E}\}. \quad (14)$$

Compared to (6), in our formulation we add a nondegeneracy condition on the scale vector  $\boldsymbol{\lambda}$  rather than on the embedding  $\mathbf{x}$ .

If the scale factors  $\boldsymbol{\lambda}$  were known, (13) would have been a localization problem from relative position measurements, that admits a unique solution for connected graphs and can be readily solved via linear least squares [2].

### A. Relation with the Standard Formulation

Before moving on with our analysis, let us clarify the relation between (13) and the standard formulation of (6).

*Proposition 8 (Equivalence with standard formulation):*

There exists a bijective mapping between the solution set of Problem (13) and the solution set of Problem (6).

*Proof:* Let us rewrite Problem (13) as:

$$\min_{\mathbf{x}} \left( \min_{\boldsymbol{\lambda} \in \Lambda} \|\mathbf{B}_d^\top \mathbf{x} - \mathbf{U}\boldsymbol{\lambda}\|^2 \right) \quad (15)$$

where we split the minimization w.r.t.  $\mathbf{x}$  and  $\boldsymbol{\lambda}$ . Since  $\boldsymbol{\lambda}$  appears quadratically in (15), its optimal value, for every choice of  $\mathbf{x}$ , is:

$$\boldsymbol{\lambda}^*(\mathbf{x}) = (\mathbf{U}^\top \mathbf{U})^{-1} \mathbf{U} \mathbf{B}_d^\top \mathbf{x} = \mathbf{U} \mathbf{B}_d^\top \mathbf{x}, \quad (16)$$

which, component-wise, becomes

$$\lambda_{ij}^*(\mathbf{x}) = \mathbf{u}_{ij}^\top (\mathbf{x}_j - \mathbf{x}_i). \quad (17)$$

Note that we have to impose  $\boldsymbol{\lambda} \in \Lambda$ ; however, comparing (17) with (8), we can substitute the condition  $\boldsymbol{\lambda} \in \Lambda$  with  $\mathbf{x} \in \mathbf{X}$ . Hence (13) becomes

$$\min_{\mathbf{x} \in \mathbf{X}} \|(\mathbf{I}_{dm} - \mathbf{U}\mathbf{U}^\top) \mathbf{B}_d^\top \mathbf{x}\|^2. \quad (18)$$

Since  $\mathbf{I}_{dm} - \mathbf{U}\mathbf{U}^\top$  is block diagonal (with  $d \times d$  blocks), we can develop the objective function into single terms:

$$\begin{aligned} \|(\mathbf{I}_{dm} - \mathbf{U}\mathbf{U}^\top) \mathbf{B}_d^\top \mathbf{x}\|^2 &= \sum_{(i,j) \in \mathcal{E}} \|(\mathbf{I}_d - \mathbf{u}_{ij}\mathbf{u}_{ij}^\top)(\mathbf{x}_j - \mathbf{x}_i)\|^2 \\ &= \sum_{(i,j) \in \mathcal{E}} (\mathbf{x}_j - \mathbf{x}_i)^\top (\mathbf{I}_d - \mathbf{u}_{ij}\mathbf{u}_{ij}^\top) (\mathbf{x}_j - \mathbf{x}_i) \\ &= \sum_{(i,j) \in \mathcal{E}} (\mathbf{x}_j - \mathbf{x}_i)^\top \mathcal{S}(\mathbf{u}_{ij})^\top \mathcal{S}(\mathbf{u}_{ij}) (\mathbf{x}_j - \mathbf{x}_i) \\ &= \sum_{(i,j) \in \mathcal{E}} \|\mathcal{S}(\mathbf{u}_{ij})(\mathbf{x}_j - \mathbf{x}_i)\|^2 \end{aligned} \quad (19)$$

where we used the fact that the matrix  $(\mathbf{I}_d - \mathbf{u}_{ij}\mathbf{u}_{ij}^\top)^2 = \mathbf{I}_d - \mathbf{u}_{ij}\mathbf{u}_{ij}^\top$ , and the fact that (as one can verify by direct computation)  $\mathcal{S}(\mathbf{u}_{ij})^\top \mathcal{S}(\mathbf{u}_{ij}) = \mathbf{I}_d - \mathbf{u}_{ij}\mathbf{u}_{ij}^\top$ .

In matrix form, (18) becomes  $\min_{\mathbf{x} \in \mathbf{X}} \|\mathbf{S}\mathbf{B}_d \mathbf{x}\|^2$ , which is identical (6), hence proving that solving Problem (13) is the same as solving Problem (6), and there is a bijective mapping between the corresponding solutions via (16). ■

Proposition 8 tells us every conclusion we draw on (13) can be readily applied to the standard formulation (6).

### B. Cycle Basis Matrix and Scale Estimation

In this section we show that Problem (13) can be rewritten in terms of the sole scale vector  $\boldsymbol{\lambda}$ , using a cycle basis matrix of the graph underlying the localization problem.

First, we note that, for any given scale vector  $\boldsymbol{\lambda}$ , we can compute the optimal position estimate:

$$\mathbf{x}^*(\boldsymbol{\lambda}) = (\mathbf{B}_d \mathbf{B}_d^\top)^{-1} \mathbf{B}_d \mathbf{U} \boldsymbol{\lambda} \quad (20)$$

To make the notation more compact we define the matrix

$$\mathbf{P}_B = \mathbf{I}_{dm} - \mathbf{B}_d^\top (\mathbf{B}_d \mathbf{B}_d^\top)^{-1} \mathbf{B}_d \quad (21)$$

Plugging  $\mathbf{x}^*$  (as a function of  $\boldsymbol{\lambda}$ ) back into (13), we get:

$$\min_{\boldsymbol{\lambda} \in \Lambda} \|\mathbf{P}_B \mathbf{U} \boldsymbol{\lambda}\|^2 \quad (22)$$

Now we note that it holds (see, e.g., [12]):

$$P_B = C_d^\top (C_d C_d^\top)^{-1} C_d \quad (23)$$

where  $C_d = C \otimes I_d$  and  $C \in \{-1, 0, +1\}^{\ell \times m}$  is a *cycle basis matrix* of the graph underlying the problem.

Developing the squared norm in (22) and using (23):

$$\arg \min_{\lambda \in \Lambda} \lambda^\top U^\top C_d^\top (C_d C_d^\top)^{-1} C_d U \lambda \quad (24)$$

Defining  $D = (C_d C_d^\top)^{-\frac{1}{2}}$  (the cycle matrix is full rank, hence the matrix  $D$  is always invertible), we get our final formulation

$$\arg \min_{\lambda \in \Lambda} \|D C_d U \lambda\|^2 = \arg \min_{\lambda \in \Lambda} \|M \lambda\|^2 \quad (25)$$

where we defined the *constraint matrix of the framework* as  $M = D C_d U \in \mathbb{R}^{d\ell \times m}$ . Essentially, we reformulated Problem (6), which looks for a suitable embedding  $x$ , and transformed it into Problem (25), which looks for the scale factors  $\lambda$ . From Proposition 8, we know that the two problems are equivalent. In the noiseless case, the framework is rigid if and only if the matrix  $M$  has exactly one singular value equal to zero, that is, its nullspace is one-dimensional.

Note that, since  $D$  is an invertible transformation, the nullspace of the constraint matrix  $M = D C_d U$  is the same as the null space  $C_d U$ . As pointed out in the following, this matrix has an intuitive interpretation that combines both geometric and topological aspects of bearing-only localization.

*Remark 9 (Structure of  $C_d U$ ):* The matrix  $C_d U$  has the same structure of the cycle basis matrix  $C$ , but while  $C$  has scalar entries  $-1, 0, +1$  for edge  $k$ , the matrix  $C_d U$  includes  $d$ -vectors  $-u_k, \mathbf{0}, +u_k$ , respectively (see Figure 3 for a simple example). Therefore,  $C_d U$  captures at the same time the topology of the graph (via  $C_d$ ) and the geometric aspects (via  $U$ ), and block rows of the matrix include the bearing measurements collected along each cycle in the graph.

We will now use our formulation in (25) to identify all the rigid components in a flexible framework.

## VI. EDGE-BASED RIGID COMPONENTS IDENTIFICATION

For the sake of analysis, we first consider the noiseless case, for which the unknown scales satisfy  $M \lambda = \mathbf{0}$  (compare with (25)).

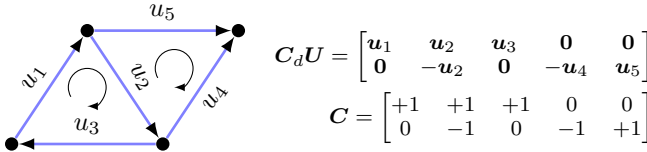


Fig. 3. Example of structure of the matrix  $C_d U$  for a small problem with five bearing measurements. Each block row of  $C_d U$  can be simply built by traversing the first (left) or second (right) cycle of the graph in arbitrary direction (clockwise in the example), and including the encountered bearing measurements in the corresponding columns (with a negative sign if the edge does not agree with the direction of travel along the cycle, positive otherwise).

In the following we write  $v^a \sim v^b$  to denote equality up-to-scale between two vectors  $v^a$  and  $v^b$ . Moreover, we denote with  $(A)_k$  the  $k$ -th row of a matrix  $A$ . We state the following result, which is the first main contribution of the paper.

*Theorem 10:* Let us consider a noiseless framework  $\mathcal{F}$  with edges numbered from 1 to  $m$ . Consider the constraint matrix  $M$  and define  $L \in \mathbb{R}^{m \times k_L}$  to be a basis for the nullspace of  $M$ . Then two edges  $k = (i, j)$  and  $\tilde{k} = (\tilde{i}, \tilde{j})$  are interdependent if and only if  $(L)_k \sim (L)_{\tilde{k}}$ .

*Proof:* The unknown scale vector  $\lambda$  satisfies  $M \lambda = \mathbf{0}$ , hence  $\lambda \in \text{null}(M)$ . Then, we can write

$$\lambda = L a, \quad (26)$$

where  $a \in \mathbb{R}^{k_L}$ . Then we want to prove the result by showing that fixing the scale for an edge  $k$  uniquely fixes the scale for all edges  $\tilde{k}$  such that  $(L)_k \sim (L)_{\tilde{k}}$ .

Let us fix the scale for an arbitrary edge  $k$  to be 1:

$$\lambda_k = e_k^\top \lambda = 1, \quad (27)$$

where we used  $e_k$  to indicate the vector in the standard basis of  $\mathbb{R}^m$  corresponding to the edge  $k = (i, j)$ .

Therefore, we want to explore the set of vector  $\lambda$  in the form (26) (i.e., that satisfy  $M \lambda = \mathbf{0}$ ), and that keep fixed the  $k$ -th scale as in (27):

$$1 = e_k^\top L a = (L)_k a. \quad (28)$$

Note that  $(L)_k \neq \mathbf{0}$ , otherwise any optimal embedding would be degenerate. Equation (28) is an underdetermined linear system with a single equation. One can easily verify that

$$\bar{a} = (L)_k^\top / \|(L)_k^\top\|^2 \quad (29)$$

is a particular solution to this equation and, from linear algebra [7], all the solutions of the linear system (28) can be written as

$$a^b = \bar{a} + N_k b, \quad (30)$$

where  $N_k \in \mathbb{R}^{k_L \times k_L - 1}$  is a basis for the nullspace of the vector  $(L)_k$  and  $b \in \mathbb{R}^{k_L - 1}$ .

Going back to the original problem (27), all the solutions that keep fixed the  $k$ -th scale are given by

$$\lambda^b = L a^b = L \bar{a} + L N_k b, \quad b \in \mathbb{R}^{k_L - 1}. \quad (31)$$

To prove the first part of the claim, let us consider an edge  $\tilde{k}$  that is in the same interdependent edge set of  $k$ . By definition of interdependent edge set, fixing the scale of  $k$  uniquely determines the scale of  $\tilde{k}$ , therefore, for any choice of  $b, \tilde{b} \in \mathbb{R}^{k_L - 1}$  it has to hold:

$$e_{\tilde{k}}^\top (\lambda^b - \lambda^{\tilde{b}}) = 0. \quad (32)$$

Substituting (31), this implies that

$$(L)_{\tilde{k}} N_k (b - \tilde{b}) = 0. \quad (33)$$

Since  $b$  and  $\tilde{b}$  are arbitrary, the only way to satisfy this equality is that  $(L)_{\tilde{k}}$  is in the nullspace of  $N_k$ . Recalling that  $N_k$  spans the nullspace of the vector  $(L)_k$ , and using the

relation between the four fundamental spaces of a matrix<sup>1</sup>, we deduce that  $(\mathbf{L})_{\tilde{k}}$  belongs to  $\text{span}((\mathbf{L})_k)$ . However, since these are vectors, we finally have that

$$(\mathbf{L})_k \sim (\mathbf{L})_{\tilde{k}}, \quad (34)$$

thus proving the first part of the claim.

To show the second part of the claim, assume that  $(\mathbf{L})_k \sim (\mathbf{L})_{\tilde{k}}$ . This implies that  $\text{null}((\mathbf{L})_{\tilde{k}}) = \text{null}((\mathbf{L})_k)$ . Using the same arguments as the above, (31) implies that fixing the scale of edge  $k$  also fixes the scale of  $\tilde{k}$ . ■

Theorem 10 effectively transforms the problem of identifying interdependent set of edges in a framework into the (noiseless) problem of clustering lines in  $\mathbb{R}^{k_L}$ . The matrix  $\mathbf{L}$  can be easily computed from SVD of  $\mathbf{M}$ . Then, the number of interdependent sets is given by the number of distinct directions appearing in the rows of  $\mathbf{L}$ , and the rigid components can be simply found by grouping together edges whose rows in  $\mathbf{L}$  have the same direction. Notice that the directions for different interdependent sets need to be only distinct, and not linearly independent. Therefore, the number of interdependent sets can be, in general, larger than the dimension of the nullspace of  $\mathbf{M}$ .

Theorem 10 provides a way to partition the original set of edges  $\mathcal{E}$  into interdependent sets. From the interdependent sets it is easy to find the maximal rigid components, as established by the following corollary.

*Corollary 11 (Rigid components, interdependent edges):* Consider a collection of interdependent sets of edges in a framework, as given by Theorem 10. Then, for each interdependent set  $\mathcal{E}_R$ , each subset  $\mathcal{E}_c \subseteq \mathcal{E}_R$  that induces a connected graph defines a rigid subframework.

The proof is trivial and leverages the result of Proposition 7 that establishes the equivalence between interdependent edge set and rigidity for connected graphs.

The corollary shows that after computing the sets of interdependent edges, we can easily find rigid components of the original framework by simply looking for connected components in the set of edges.

#### A. Extension to Noisy and Almost Degenerate Cases

In the general noisy case ( $\epsilon_{ij} \neq 0$ ), there is no embedding  $\mathbf{x}$  which can satisfy all the constraints at the same time. As a consequence, we will have that  $\text{null}(\mathbf{M}) = \{\mathbf{0}\}$ , and we lose the structure given by the matrix  $\mathbf{L}$ . In practice, however, for reasonable levels of noise the last  $k_L$  singular values of  $\mathbf{M}$  will still be close, although not equal, to zero (Figure 4c).

A similar situation can also appear in the noiseless case for almost-degenerate frameworks. Consider two maximally rigid components connected with three edges (Figure 4). Intuitively, if these edges are exactly parallel, then a non-trivial partition appears (Figure 4a). On the other hand, if the edges are almost, but not exactly, parallel, then the entire framework is rigid (Figure 4b). In the latter situation, however, the relation between the scales of the two initial

<sup>1</sup>The four fundamental spaces of a matrix  $A$  are  $\text{span}(A)$ ,  $\text{null}(A)$ ,  $\text{span}(A^T)$  and  $\text{null}(A^T)$ .

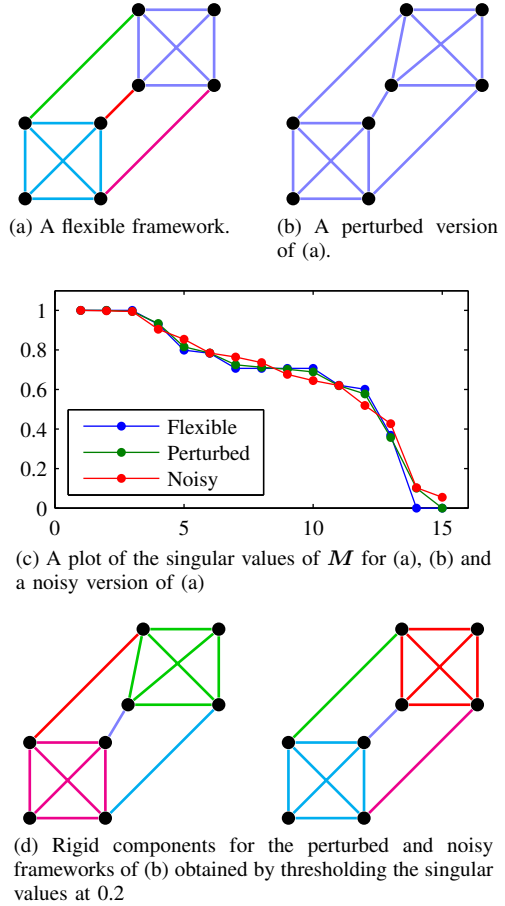


Fig. 4. Example of a flexible (a), almost flexible (b) and flexible but noisy frameworks. Different colors indicate different rigid components. For the last almost flexible and noisy frameworks, the smallest singular values of  $\mathbf{M}$  are close to zero (c). If we threshold them, we can still obtain a useful segmentation into rigid components (d) which is consistent with the underlying flexible structure (a).

components would be quite sensitive to any noise in the measurements of the two connecting edges, and, from a practical standpoint, it might be better to consider the two components as separate.

In both scenarios, we can fix a threshold under which all the singular values of  $\mathbf{M}$  are considered to be zero, thus still obtaining a matrix  $\mathbf{L} \in \mathbb{R}^{m \times k_L}$ . The threshold intuitively indicates the level of noise or the deviation from a parallel condition that we are willing to tolerate.

Unfortunately, since we have truncated the singular values, the property mentioned in Theorem 10 will hold only approximately, i.e.,  $(\mathbf{L})_k$  and  $(\mathbf{L})_{\tilde{k}}$  will have similar, but not exactly equal, directions when the edges  $k$  and  $\tilde{k}$  are interdependent. We therefore need to resort to heuristics to solve the (noisy) problem of clustering lines in  $\mathbb{R}^{k_L}$ , and thus find the grouping of the edges. In our implementation, we compute a matrix  $\mathbf{A} \in \mathbb{R}^{m \times m}$  of pairwise angles between the rows of  $\mathbf{L}$ , and then use Quickshift [17] to obtain the final clustering. An example of the result is given in Figure 4d.

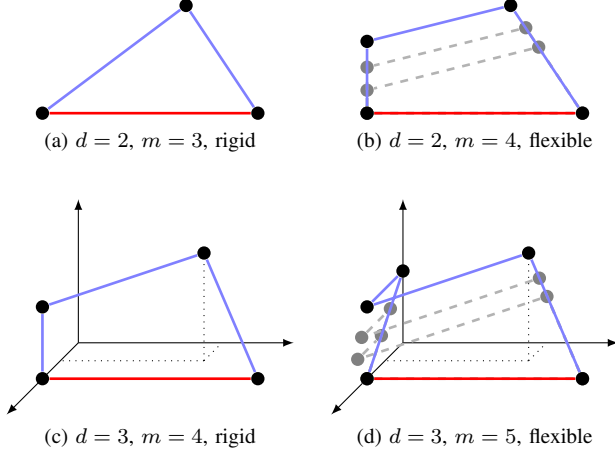


Fig. 5. Illustrative examples for the rigidity of cycles in  $d$  dimensions. For flexible graphs, the dashed lines represent alternative optimal embeddings which keep fixed the edge marked in red.

## VII. CYCLES AND RIGIDITY ENFORCEMENT

In this section, we give results that show the relation between rigidity, long cycles, and addition of edges.

We start with the following corollary, which is a direct result of our formulation.

*Corollary 12 (Single-Cycle Frameworks):* A framework in dimension  $d$  with  $m > d + 1$  edges arranged in a single cycle cannot be rigid.

*Proof:* In the presence of a single cycle, the matrix  $M$  has dimensions  $d \times m$ . In order for the framework to be rigid,  $\text{null}(M)$  must have dimension equal to one. This is equivalent to saying that  $M$  must contain exactly  $m - 1$  linearly independent columns. However, this condition cannot be satisfied when  $m - 1 > d$ . ■

Illustrative examples of the result in Corollary 12 are given in Figure 5 for 2-D and 3-D setups.

Next, we consider the question of how the rigidity of a framework changes with the addition of new edges. The practical value of this question is that, in many situations, the measured bearing are obtained through algorithms that use thresholds to reject estimates that are too noisy. However, it might be advantageous to selectively lower this threshold in order to accept measurements that might be noisier, but that also make the problem rigid. In other situations, one might be able to control the acquisition of new measurements (e.g., through the motion of a robot). Therefore, it is of interest to determine which measurements have a larger potential “payoff” in terms of rigidity.

For the sake of analysis, we only consider the noiseless case. Consider a framework  $\mathcal{F}$  whose underlying graph is connected. For our purposes, we can consider the matrix  $M = C_d U$  instead of  $M = DC_d U$  (as we mentioned, the two matrices have the same nullspace). Now, let  $\mathcal{F}'$  be a framework obtained from  $\mathcal{F}$  by adding an edge between two existing nodes. We denote the corresponding measurement as  $u_{add}$ . Since we are assuming that initially  $\mathcal{F}$  is connected,

the addition of one edge implies the creation of a new cycle. Therefore, the cycle basis matrix of  $\mathcal{F}'$  will have an additional row and an additional column:

$$C' = \begin{bmatrix} C & \mathbf{0} \\ C_{add} & 1 \end{bmatrix}, \quad (35)$$

where  $C_{add} \in \mathbb{R}^{1 \times m}$ . Note that the upper-right block of  $C'$  is zero because the original cycles in  $C$  cannot include the edge that we just added. We can then state the following.

*Proposition 13:* Studying the rigidity of the framework  $\mathcal{F}'$  is equivalent to studying the rigidity of a single cycle with  $k_L + 1$  edges and a constraint matrix of the form

$$M_{cycle} = [M_{add} L \quad u_{add}], \quad (36)$$

where  $M_{add} = (C_{add} \otimes I_d)U \in \mathbb{R}^{d \times m}$  and  $L \in \mathbb{R}^{m \times k_L}$  is a basis for  $\text{null}(M)$ .

*Proof:* Following the definitions above, the constraint matrix for  $\mathcal{F}'$  is given by

$$M' = \begin{bmatrix} M & \mathbf{0} \\ M_{add} & u_{add} \end{bmatrix}. \quad (37)$$

Denote the vector of scales for  $\mathcal{F}'$  as

$$\lambda' = \begin{bmatrix} \lambda \\ \lambda_{add} \end{bmatrix} \in \mathbb{R}^{m+1}. \quad (38)$$

We need to consider the nullspace of  $\mathcal{F}'$ , that is, the space of solutions of the equation

$$M' \lambda' = \begin{bmatrix} M \lambda \\ M_{add} \lambda + u_{add} \lambda_{add} \end{bmatrix} = \mathbf{0}. \quad (39)$$

Note that the first block-row of (39) implies that  $\lambda$  must be in the nullspace of  $M$ . We can therefore perform the substitution  $\lambda = Lb$ ,  $b \in \mathbb{R}^{k_L}$ , and rewrite (39) as

$$\mathbf{0} = \begin{bmatrix} M L b \\ M_{add} L b + u_{add} \lambda_{add} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ M_{add} L & u_{add} \end{bmatrix} \begin{bmatrix} b \\ \lambda_{add} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ M_{cycle} \end{bmatrix} \begin{bmatrix} b \\ \lambda_{add} \end{bmatrix}. \quad (40)$$

Since the first block-row in (40) is zero, the dimension of  $\text{null}(M')$  is the same as the one of  $\text{null}(M_{cycle})$ . Since  $M_{cycle}$  has  $d$  rows, we can interpret it as the constraint matrix of a framework whose cycle basis  $C_{cycle}$  is a single row, and the columns of  $M_{cycle}$  represent bearing vectors. ■

Proposition 13 has a couple of important implications. First, from Corollary 12, if the dimension of the nullspace of the original  $M$  (that is,  $k_L$ ) is larger than  $d$ , then  $\mathcal{F}'$  cannot be rigid.<sup>2</sup> Second, the rigidity of  $\mathcal{F}'$  depends on both the topology of  $\mathcal{F}'$  and the specific values of the bearings. This is captured by Proposition 13 even in the case of *non-generic* embeddings, that is, embedding where the coordinates of the nodes are algebraically dependent. This is in contrast

<sup>2</sup>This does not imply that the frameworks with more than two rigid components cannot be made rigid with the addition of a single cycle ( $k_L$  is the dimension of the null space of  $M$ , and, as we mentioned, it does not generally coincide with the number of rigid components). Think, for instance, of the trapezoid in Figure 2 with the addition of a diagonal.

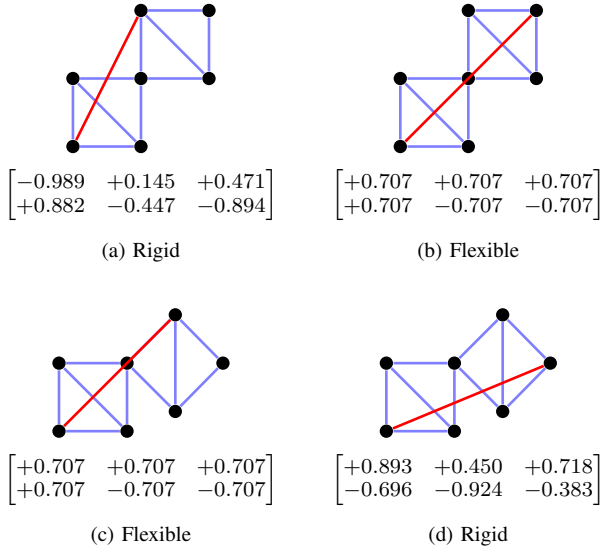


Fig. 6. Example where topological information alone cannot be used to predict rigidity. Blue: existing (flexible) framework. Red: additional edge. Matrix: normalized columns of  $M_{cycle}$ . Note that the topology of the graph is the same in each of the two pairs (a),(c) and (b),(d). However, depending on the position of the nodes, frameworks with the same graph topology can have different rigidity. The equivalent cycle framework correctly predicts the rigidity: the columns of  $M_{cycle}$  contain linearly dependent (in fact, identical) columns for (b), (c), and linearly independent columns for (a), (d).

with existing results which considers only the topology of the graph (through combinatorial conditions) and assume *generic* embeddings [5]. A detailed example of this is given in Figure 6.

## VIII. SIMULATIONS AND CONCLUSION

To conclude, we validate our theoretical results by testing our algorithm for the noiseless case on the same configurations as [9, Figure 5]. Results are shown in Figure 7. One can easily verify, by visual inspection, that the algorithm performed correctly, and that each component that has been found is rigid.

We would like to stress that our results, although equivalent to those obtained in [9], have been obtained by using an *edge-based* method instead of a *node-based* method, thus empirically showing that both methods are viable in practice. The real difference between the two is that our formulation has a simple geometric interpretation (Remark 9), provides additional insights on the relation between rigidity and the topology of the graph (expressed in terms of cycles), and enables to design strategies to enforce rigidity by adding edges to an existing graph (as shown in Section VII).

As future work, we plan to more rigorously characterize and compare the robustness of the node-based and edge-based methods in the presence of noisy measurements.

## REFERENCES

[1] J. Aspnes, W. Whiteley, and Y. R. Yang. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006.

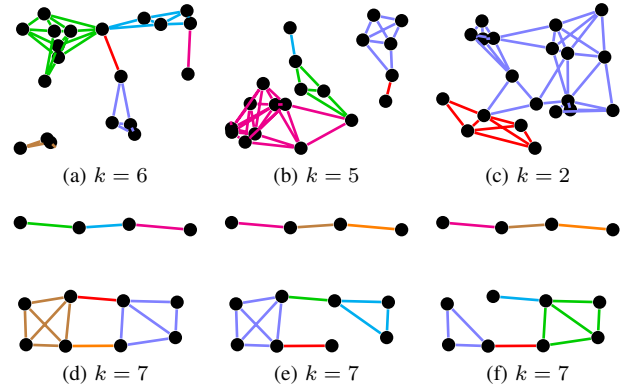


Fig. 7. Results of our algorithm for identifying rigid components on six different sample configurations. Different colors indicate different rigid components. The sub-captions indicates the number of components found.

[2] P. Barooah and J.P. Hespanha. Estimation on graphs from relative measurements. *Control System Magazine*, 27(4):57–74, 2007.

[3] A. N. Bishop, I. Shames, and B. Anderson. Stabilization of rigid formations with direction-only constraints. In *IEEE Conference on Decision and Control*, pages 746–752, 2011.

[4] M. Brand, M. Antone, and S. Teller. Spectral solution of large-scale extrinsic camera calibration as a graph embedding problem. In *European Conf. on Computer Vision (ECCV)*, pages 262–273, 2004.

[5] T. Eren, W. Whiteley, and P.N. Belhumeur. Using angle of arrival (bearing) information in network localization. In *IEEE Conference on Decision and Control*, pages 4676–4681, 2006.

[6] A. Franchi and P. R. Giordano. Decentralized control of parallel rigid formations with direction constraints and bearing measurements. In *IEEE Conference on Decision and Control*, pages 5310–5317, 2012.

[7] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[8] F. Kahl and R. Hartley. Multiple view geometry under the l-infinity norm. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(9):1603–1617, 2008.

[9] R. Kennedy, K. Daniilidis, O. Naroditsky, and C. J. Taylor. Identifying maximal rigid components in bearing-based localization. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 194–201, 2012.

[10] R. Kennedy and C. J. Taylor. Network localization from relative bearing measurements. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014.

[11] Hongdong Li. Multi-view structure computation without explicitly estimating motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2777–2784, 2010.

[12] W.J. Russell, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. 59(6):2834–2846, 2011.

[13] B. Servatius and W. Whiteley. Constraining plane configurations in computer-aided design: Combinatorics of directions and lengths. *SIAM J. on Discr. Mat.*, 12(1):136–153, 1999.

[14] G. Stacey and R. Mahony. A port-hamiltonian approach to formation control using bearing measurements and range observers. In *IEEE Conference on Decision and Control*, 2013.

[15] C. J. Taylor and J. R. Spletzer. A bounded uncertainty approach to cooperative localization using relative bearing constraints. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2500–2506, 2007.

[16] R. Tron and R. Vidal. Distributed 3-D localization of camera sensor networks from 2-D image measurements. In *IEEE Trans. Automat. Contr.*, 2014.

[17] A. Vedaldi and S. Soatto. Quick shift and kernel methods for mode seeking, 2008.

[18] S. Zhao, F. Lin, K. Peng, B. M. Chen, and T. H. Lee. Distributed control of angle-constrained cyclic formations using bearing-only measurements. *Systems and Control Letters*, 63:12–24, 2014.