Autonomous Calibration Algorithms for Planar Networks of Cameras ⋆

Domenica Borra [b], Enrico Lovisari [a], Ruggero Carli [a], Fabio Fagnani [b], Sandro Zampieri [a]

[a] *Department of Information Engineering, University of Padova, Via Gradenigo 6/A, IT-35131 Padova, Italy*

[b] *Department of Mathematics, Polytechnic of Turin, Corso Duca degli Abruzzi, 24, IT-10192 Torino, Italy*

Abstract

This paper deals with the problem of the angular calibration for a network of cameras, namely the problem of estimating a common orientation reference frame. In the proposed set–up each camera obtains noisy measurements of its relative orientation with respect to some other cameras. The set of measurements can be described by a graph having the cameras as nodes and having an edge between two cameras if a relative orientation measurement is available. The paper proposes a novel two-step algorithm based on a choice of a basis for the set of graph cycles. The first step consists in computing a set of integer numbers, which provides a first rough estimate of the orientations. The second step exploits this information to build up a suitable quadratic minimization problem. Two actual implementations, corresponding to two different basis of cycles, are described and compared in terms of the worst–case scenario. Finally, through numerical simulations the algorithm is compared with another algorithm proposed in the literature for solving the same problem.

Key words: Distributed control, Cameras, Graph Theory, Communication protocols, Decentralized systems

1 Introduction

In the last years much effort has been devoted by the scientific community to develop leaderless distributed strategies for solving problems involving interacting agents which need to achieve a common goal. The advantages of these strategies, if compared with the centralized ones, are in terms of robustness of the resulting systems with respect to communication and/or agent failures, and in terms of adaptivity to environmental changes and simplicity in the system tuning. On the other hand, distributed strategies, based on local exchange of information among peer agents, can be more difficult to design and optimize and often exhibit slower convergence to the regime working conditions.

A large set of cameras communicating each other through a network is a widely used architecture in application areas like video surveillance and motion capture [1]. In a network of cameras one of the most crucial problems is calibration. For each camera this consists in understanding what is its position and orientation with respect to a global and common reference coordinate system. The importance of this information is clear in case the camera network is used for instance to track an external mobile object. Indeed, in this case, if the object is exiting from the sensing region of the i-th camera and it is entering in the sensing region of the j-th camera, then, in order to avoid to loose the target, the camera i has to communicate to the camera j where it has to move in order to see the object. It is manifest that both cameras must share the same coordinate system in order this to be possible.

⋆ The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreements n. 257462 HYCON2 Network of excellence and n. 223866 FeedNetBack.

Email addresses: borra@polito.it (Domenica Borra), lovisari@dei.unipd.it (Enrico Lovisari), carlirug@dei.unipd.it (Ruggero Carli), fagnani@polito.it (Fabio Fagnani), zampi@dei.unipd.it (Sandro Zampieri).

Usually, calibration is set off-line by human operators, or by a centralized unit. In any case it is usually a rather time consuming stage of the system setting. The algorithm we propose in this paper aims to complete this task autonomously, and requires no or very limited centralized coordination. This also allows the possibility to re–calibrate periodically and this is useful especially in case when some of the cameras are mobile.

The calibration problem over Euclidean spaces has recently been studied by Barooah and Hespanha in a great detail (see [2–4]). The problem considered there is the localization using noisy relative measurements, namely determining the coordinates of a set of vectors in a Euclidean space starting from the knowledge of some noisy differences of those vectors.

The cameras calibration problem is similar in the sense that each camera can be characterized by a position and an orientation in space. Some well–known methods in computer vision permit to obtain quite easily and efficiently relative positions and orientations of pairs of cameras whose sensing regions overlap. Then the problem that has to be solved is to determine, from these relative positions and orientations, the position and the orientation of the cameras with respect to a common reference coordinate system. Cameras calibration can be casted into an optimization problem (or a consensus) over the manifold $SE(3)$. Under certain assumptions one can show that this problem can decoupled into the estimation of the position and the estimation of the orientation. The latter can be seen as an optimization (or a consensus) over the manifold $SO(3)$. Moreover, under some further assumptions, the calibration over $SO(3)$ can be reduced to the calibration over the simpler manifold $SO(2)$.

This class of problems has already attracted much attention in the last years. In [5,6] a consensus algorithm on $SO(2)$ based on the gradient flow of a potential defined using the chordal distance is studied. In [7] a similar approach based on the geodesic distance is proposed in order to study the more general calibration problem on $SE(3)$. The drawback associated with both these approaches is that, in case we have many cameras, the proposed potentials exhibit a great number of local minima.

In [8] the problem of calibration on $SO(2)$ is considered when measurements of relative orientations are affected by additive noise. The authors propose a procedure to obtain a new set of relative orientations which is ensured to sum up to multiples of 2π over a chosen family of cycles, an idea which closely resembles what we propose in the present paper. The new set of relative orientations is then spread along a spanning tree to obtain an estimate of the orientations. Finally, in [9] an estimation algorithm is proposed for a model of the measurements in which some of them are ideal, while others are completely random. The estimate of the orientations is obtained via the computation of the eigenvector associated with the largest eigenvalue of a suitable Hermitian matrix.

The algorithm proposed in this paper is based on a non-convex optimization problem, as in [5,7]. We restrict ourselves to the simple case of calibration over $SO(2)$, and the cost to be minimized is based on the geodesic distance over this manifold. The set of available relative measurements is described by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V}$ represents the set of cameras and where the set of edges $\mathcal{E}$ represents the available relative orientation measurements between pairs of cameras. Our main idea is to break the estimation problem into two parts: first we estimate a combinatorial object, which is a set of integers, each associated with an edge in $\mathcal{E}$. Intuitively, these integers take care of the fact that measurement noises along the cycles in the graph do not sum up to 0, in general. Once this is done, the original optimization problem over a manifold can be reduced to a quadratic optimization problem, which can be easily solved using classic algorithms. The idea of using cycles has already been proposed in [10] in the context of localization over Euclidean spaces in order to improve the quality of the estimates. In fact, we propose two versions of the algorithm. One version is based on sets of cycles associated with spanning trees. Another version instead is based on sets of minimal cycles. Notice finally that the proposed method is consistent in the sense that, if there is no measurement noise, the solution given by the algorithm coincides with the true one.

This research has previously partially appeared in [11], and it is here enriched both in theoretical depth and as for simulative comparison with the existing literature.

The paper is organized as follows: in Section 2 we collect several useful graph theoretical definitions and results. In Section 3 and Section 4 we respectively formulate the optimization problem to be solved and we explain how it relates with the cameras calibration problem. Section 5 is devoted to the description of the proposed algorithm and to a worst–case analysis of its performance. In Section 6 it is shown how to distribute the algorithm over the network for the two particular choices of the set of cycles mentioned above, while in Section 7 we compare their performance. Finally, Section 8 proposes numerical simulations and a comparison with the algorithm proposed in [8]. Section 9 draws the conclusions.

2 Some tools from graph theory

In this Section we recall some known facts from algebraic graph theory which will be instrumental in the development of the paper.

An undirected graph is a couple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of nodes, and $\mathcal{E}$ is a subset of unordered pairs of elements of $\mathcal{V}$ called edges [1]. We let $M := |\mathcal{E}|$. An *orientation* on $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a pair of maps $s : \mathcal{E} \to \mathcal{V}$ and $t : \mathcal{E} \to \mathcal{V}$ such that $e = \{s(e), t(e)\}$ for every $e \in \mathcal{E}$. According to this definition, $s(e)$ and $t(e)$ are called the source and terminal node of the edge e, respectively. Assume from now on that we have fixed an orientation (s, t) on $\mathcal{G}$. The *incidence matrix* $A \in \{\pm 1, 0\}^{\mathcal{E}\times\mathcal{V}}$ of $\mathcal{G}$ is defined by putting $A_{e,s(e)} = 1$, $A_{e,t(e)} = -1$, and $A_{e,v} = 0$ if $v \neq s(e), t(e)$.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_\mathcal{T})$ of $\mathcal{G}$ is a connected subgraph of $\mathcal{G}$ which is a tree. Notice that $|\mathcal{E}_\mathcal{T}| = N - 1$.

A path h of length n is an ordered sequence of nodes $h = (v_1, v_2, \cdots, v_{n+1})$ such that $\{v_i, v_{i+1}\} \in \mathcal{E}$ for all $i = 1, \dots, n$. A path $h = (v_1, v_2, \cdots, v_{n+1})$ is said to be closed if $v_1 = v_{n+1}$. A closed path $h = (v_1, v_2, \cdots, v_n, v_1)$ is said to be a cycle if $n \geq 3$ and $v_i \neq v_j$ for all $i, j \in \{1, \dots, n\}$ with $i \neq j$. The support of a path is given by the set of its edges, namely, if $h = (v_1, v_2, \cdots, v_{n+1})$, then $\mathrm{supp}(h) := \{e \in \mathcal{E} \,|\, e = \{v_i, v_{i+1}\},\ \exists i = 1, \dots, n\}$.

Consider now $\mathbb{Z}^\mathcal{E}$, the $\mathbb{Z}$-module of $\mathbb{Z}$-valued row vectors whose components are labelled by $\mathcal{E}$. Given $\boldsymbol{r} \in \mathbb{Z}^\mathcal{E}$, we define its support as

$$\mathrm{supp}(\boldsymbol{r}) := \{e \in \mathcal{E} \,|\, \boldsymbol{r}(e) \neq 0\}$$

We now associate with every path h, an element $\boldsymbol{r}_h \in \mathbb{Z}^\mathcal{E}$ as follows. First, if $h = (v_1, v_2)$, we put

$$\boldsymbol{r}_h(e) = \begin{cases} A_{ev_1}, & \text{if } e = \{v_1, v_2\} \\ 0, & \text{otherwise.} \end{cases}$$

Then, for a generic path $h = (v_1, v_2, \cdots, v_{n+1})$ we define $\boldsymbol{r}_h(e)$ as $\boldsymbol{r}_h(e) = \sum_{i=1}^n \boldsymbol{r}_{(v_i,v_{i+1})}(e)$. In particular, for paths h with non-repeating edges, $\boldsymbol{r}_h$ is built by assigning $\boldsymbol{r}_h(e) = 1$ if the edge e appears in h and it is crossed with the same orientation of $\mathcal{G}$, assigning $\boldsymbol{r}_h(e) = -1$ if the edge e appears in h and it is crossed with the reverse orientation of $\mathcal{G}$ and finally assigning $\boldsymbol{r}_h(e) = 0$ if the edge e does not appear in h.

Denote now by Γ the $\mathbb{Z}$-submodule of $\mathbb{Z}^\mathcal{E}$ generated by all the vectors $\boldsymbol{r}_h$ as h varies in the set of closed paths. It holds true that Γ has dimension equal to $M - N + 1$, i.e. there exist $\boldsymbol{r}_{h_1}, \dots, \boldsymbol{r}_{h_{M-N+1}}$ forming a $\mathbb{Z}$-basis of Γ. This fact is well known in the slight different context where no orientation is considered and where vector spaces are used [12]. The following proposition, whose proof is given in the Appendix, clarifies the situation in our setting.

Proposition 2.1 *We have that*

(1) $\Gamma = \{\boldsymbol{r} \in \mathbb{Z}^\mathcal{E} \,|\, \boldsymbol{r}A = 0\}$*;*
(2) Γ *has dimension equal to* $M - N + 1$*.*

The following proposition is also proved in the appendix.

Proposition 2.2 *Let* $\boldsymbol{r}_{h_1}, \dots \boldsymbol{r}_{h_{M-N+1}}$ *be a* $\mathbb{Z}$*-basis of* Γ*. Defined moreover* $R \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times\mathcal{E}}$ *to be the matrix having* $\boldsymbol{r}_{h_1}, \dots, \boldsymbol{r}_{h_{M-N+1}}$ *as rows. Then*

(1) $\ker R = \mathrm{Im} A$*, where* $\ker R := \{\boldsymbol{K} \in \mathbb{Z}^\mathcal{E} | R\boldsymbol{K} = 0\}$ *and* $\mathrm{Im} A := \{\boldsymbol{K} \in \mathbb{Z}^\mathcal{E} | \boldsymbol{K} = A\boldsymbol{h},\ \exists \boldsymbol{h} \in \mathbb{Z}^\mathcal{V}\}$*.*
(2) there exists $X \in \mathbb{Z}^{\mathcal{E}\times(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})}$ *such that* $RX = I$ *where* I *is the identity matrix.*

[1] More precisely an edge is a subset of $\mathcal{V}$ with two elements.

3 Problem Formulation

The estimation problem over $SO(2)$ we want to study can be described as follows. Assume we have a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an orientation $s : \mathcal{E} \to \mathcal{V}$ and $t : \mathcal{E} \to \mathcal{V}$. With each node $v \in \mathcal{V}$ of the graph we associate an angle $\bar{\theta}_v \in [-\pi, \pi)$ and similarly with each edge $e \in \mathcal{E}$ we associate an angle $\eta_e \in [-\pi, \pi)$. We call the angle $\bar{\theta}_v$ the *orientation* of v. Our aim is to find an estimate of the angles $\bar{\theta}_v$, $v \in \mathcal{V}$, from the knowledge of η_e, $e \in \mathcal{E}$, knowing that these represent a measure of the *relative orientation* among $s(e)$ and $t(e)$, in the sense that

$$\eta_e = (\bar{\theta}_{s(e)} - \bar{\theta}_{t(e)} + \varepsilon_e)_{2\pi} \tag{1}$$

where $\varepsilon_e \in [-\pi, \pi)$ are measurement noise terms, which have to be considered small and independent, and where $(x)_{2\pi} := x - 2\pi q_{2\pi}(x)$ with $q_{2\pi}(x) := \lfloor \frac{x+\pi}{2\pi} \rfloor, x \in \mathbb{R}$. Notice that the function $q_{2\pi}(x)$ is a quantizer such that $x \in [-\pi, \pi)$ if and only if $q_{2\pi}(x) = 0$, so that $(x)_{2\pi} = x$. If we stack the angle $\bar{\theta}_v$ and η_e in suitable column vectors we can rewrite (1) in the compact form

$$\boldsymbol{\eta} = (A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon})_{2\pi}, \tag{2}$$

where $(\cdot)_{2\pi}$ is done componentwise.

The estimator is in principle a function $\hat{\boldsymbol{\theta}} : [-\pi, \pi)^{\mathcal{E}} \to [-\pi, \pi)^{\mathcal{V}}$ mapping the available data $\boldsymbol{\eta}$ into an estimate $\hat{\boldsymbol{\theta}}(\boldsymbol{\eta})$ of $\bar{\boldsymbol{\theta}}$. Its performance can be evaluated using the index

$$W(\hat{\boldsymbol{\theta}}) = \frac{1}{N}\|(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})_{2\pi}\|^2 = \frac{1}{N}\sum_{v=1}^{N}(\hat{\boldsymbol{\theta}}_v - \bar{\boldsymbol{\theta}}_v)_{2\pi}^2. \tag{3}$$

Notice that W is in principle a function of $\bar{\boldsymbol{\theta}}, \boldsymbol{\varepsilon}$, namely $W = W(\bar{\boldsymbol{\theta}}, \boldsymbol{\varepsilon})$. If we have a probabilistic model of $\bar{\boldsymbol{\theta}}$ and $\boldsymbol{\varepsilon}$ the considered index is the expected value of W.

Remark 3.1 *The definition of W is reasonable since the values $\bar{\theta}_v$'s and $\hat{\theta}_v$'s are angles, so they are coincident if they differ of integer multiples of 2π, namely if $(\hat{\theta}_v - \bar{\theta}_v)_{2\pi} = 0$.*

The estimator proposed in this paper is based on the minimization of the following least–square cost

$$V(\boldsymbol{\theta}) = \sum_{e\in\mathcal{E}}(\theta_{s(e)} - \theta_{t(e)} - \eta_e)_{2\pi}^2 = \|(A\boldsymbol{\theta} - \boldsymbol{\eta})_{2\pi}\|^2, \tag{4}$$

through which one aims to find a set of estimates $\hat{\theta}_1, \ldots, \hat{\theta}_N$ whose differences along the edges fit the measurements η_e.

Remark 3.2 *Notice that $V(\boldsymbol{\theta} + \alpha\mathbb{1}) = V(\boldsymbol{\theta})$, $\forall \boldsymbol{\theta} \in [-\pi, \pi)^{\mathcal{V}}$ and $\forall \alpha \in \mathbb{R}$, where $\mathbb{1}$ is the vector with all entries equal to 1. This feature naturally arises from the fact that the cost is build on* relative *measurements and is unavoidable unless we impose a further constraint. In fact, we will assume in the sequel that node 1 is the anchor and knows its orientation, which, with no loss of generality, can be assumed to be zero, i.e. $\bar{\theta}_1 = 0$. By adding this constraint, namely imposing that $\hat{\theta}_1 = 0$, we can avoid this source of ambiguity.*

Remark 3.3 *Minimizing the cost $V(\boldsymbol{\theta})$ seems to be reasonable, since, in case of noiseless measurements, we have $V(\boldsymbol{\theta}) = 0$ and $\theta_1 = 0$ if and only if $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$. In other words $V(\boldsymbol{\theta})$ has only one global minimum for $\boldsymbol{\theta} \in (-\pi, \pi]^{\mathcal{V}}$ such that $\theta_1 = 0$, which is $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$. However, $V(\boldsymbol{\theta})$ has, even in the noiseless case, multiple local minima, as it will be shown in the next example. The same problem occurs if we choose a slightly different cost based on the chordal distance, as it has been shown in [5]. In order to avoid such local minima, the classic approach is to initialize the gradient based minimization algorithm in a suitable region in such a way that the convergence to the global minimum is ensured. A deep analysis of this region for a large class of manifolds is done in [13], but it is beyond the scope of this paper. Our approach is different, and it is motivated by the following simple example.*

Example 3.1 *Consider the ring graph with 3 agents in figure 1, and assume for sake of simplicity that $\bar{\theta}_1 = \bar{\theta}_2 = \bar{\theta}_3 = 0$. Consider the ideal noiseless case, so that $\eta_{12} = \eta_{23} = \eta_{31} = 0$. Assuming that we have fixed $\theta_1 = 0$, we have*

to find $\theta_2, \theta_3 \in (-\pi, \pi]$ *such to minimize the cost*

$$V(\theta_1, \theta_2, \theta_3) = \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3)_{2\pi}^2 = \begin{cases} \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3)^2 & \theta_2 - \theta_3 \in (-\pi, \pi], \\ \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3 - 2\pi)^2 & \theta_2 - \theta_3 \geq \pi, \\ \theta_2^2 + \theta_3^2 + (\theta_2 - \theta_3 + 2\pi)^2 & \theta_2 - \theta_3 < -\pi. \end{cases} \tag{5}$$

It can be verified that this cost has three local minima $\theta_2 = \theta_3 = 0$, $\theta_2 = \frac{2}{3}\pi, \theta_3 = -\frac{2}{3}\pi$ *and* $\theta_2 = -\frac{2}{3}\pi, \theta_3 = \frac{2}{3}\pi$ *being* $\theta_2 = \theta_3 = 0$ *the global minimum in which* $V = 0$ *while the other two are only local minima where* $V = \frac{4}{3}\pi^2$.

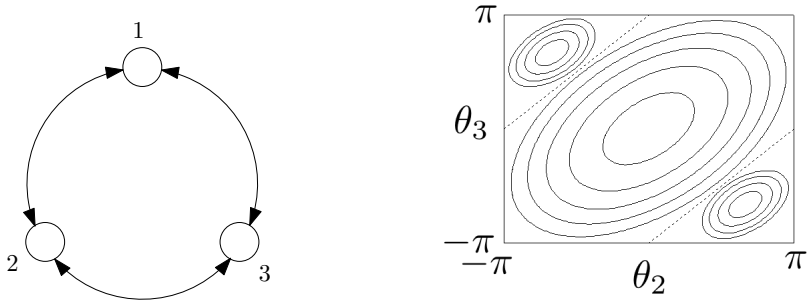

Fig. 1. A simple ring with 3 agents. On the right panel, the three regions in which $[-\pi, \pi)$ is partitioned, with the contour lines of the quadratic functions in (5).

4 Cameras calibration

The proposed problem has a certain interest in applications such as calibration [7] or orientation localization [8] of networks of cameras. In this Section we describe the calibration problem and we relate it to the problem stated above.

Consider a group of cameras perceiving some environment for surveillance purposes. Each camera is modeled as an *ideal pin-hole* device, which consists in a plane, called the image plane, and a point, called the optical center. A point in the environment is sensed by the camera through its radial projection, through the optical center, on the image plane. The sensing region of the camera is the set of points for which this projection exists. To model the perception of a point we endow each camera with a local reference frame, denoted by Σ_v, so that a point in the sensing region of camera v is identified, by v, as a set of coordinates in Σ_v.

Fix now an external reference frame Σ_0. Without loss of generality we can assume that its origin lies on the ground, which is assumed to be a plane, and that it has one axis perpendicular to the ground and pointing "upward". In such a reference frame, the *pose* of the camera v is identified by an element $(g_v, R_v) \in SE(3) = \mathbb{R}^3 \times SO(3)$. The set $SO(3) = \{R \in \mathbb{R}^{3\times 3} : R^T R = I, \det(R) = +1\}$, is the set of rotations in $3D$, while $\mathbb{R}^3$ denotes here the set of translations. The quantity (g_v, R_v) maps the local reference frame Σ_v into the external reference frame Σ_0. More precisely, given a point in the environment whose coordinates in Σ_0 and Σ_v are, respectively, p_0, $p_v \in \mathbb{R}^3$, then

$$p_0 = R_v p_v + g_v. \tag{6}$$

For coordination purposes, it is necessary that the cameras obtain as good as possible estimates of their (g_v, R_v). Assume in fact that an agent is exiting from the sensing region of camera v and is entering in that of camera u, and call p_v its coordinates in Σ_v. Then u can easily find the coordinates of the agent in its reference frame Σ_u using twice (6) to obtain

$$p_u = R_u^T R_v p_v + R_u^T (g_v - g_u). \tag{7}$$

Thus, if each camera has a good estimate of its pose w.r.t. a common external reference frame, they can exchange information on the position of agents in the environment they are monitoring.

An estimate of the g_v's can be obtained through the consensus-like optimization procedures based on relative measurements as proposed in [3]. In this paper, instead, we concentrate on the estimate of R_v, which we call from now on the *rotational calibration problem.*

In order to estimate the R_v's, the cameras communicate and exchange information with some of the others. The admissible communications are modeled through an undirected communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V} = \{1, \ldots, N\}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. An edge $\{v, u\} \in \mathcal{E}$ exists if and only if the cameras u and v are able to exchange information and, moreover, their sensing regions overlap.

The mutual information used to achieved rotational calibration is the *relative orientation* among pairs of cameras. Given the poses (g_v, R_v) and (g_u, R_u) of cameras v and u, we define $R_{vu} := R_u^T R_v$ the relative orientation of v with respect to u. Notice that R_{vu} appears in (7) when relating the coordinates of a point in Σ_u given those in Σ_v, and that $R_{uv} = R_{vu}^T$.

Assume now $\{u, v\} \in \mathcal{E}$. Then the quantities R_{uv} and R_{vu}, as well as the relative translations $g_u - g_v$ and $g_v - g_u$, can be obtained by the cameras v and u through the so–called eight points algorithm [14]. In general, this relative orientation is corrupted by noise. In our setting, the two cameras can thus only compute the following noisy version of R_{vu}

$$\tilde{R}_{vu} = R_u^T N_{vu} R_v, \tag{8}$$

where $N_{vu} \in SO(3)$ is a rotational error. Since the two cameras compute these quantities on the basis of the same dataset, then we can assume that $\tilde{R}_{uv} = \tilde{R}_{vu}^T$, which implies $N_{uv} = N_{vu}^T$.

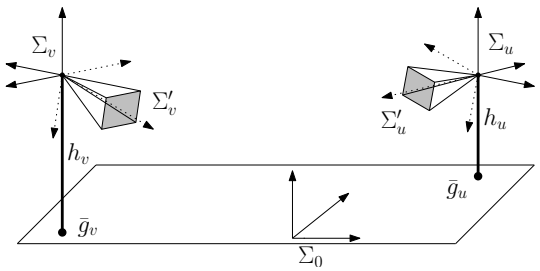

Fig. 2. Description of a network of cameras satisfying the assumptions. It is shown the external reference frame Σ_0 and, for each camera, the "natural" local reference frame Σ'_v (in dotted lines) and the chosen local reference frame Σ_v (in solid lines).

Rotational calibration in $SO(2)$ is related to the previous problem when the following assumptions hold:

- each camera is deployed in the environment at a known height with respect to the ground, which is assumed to be a plane;
- each camera knows the perpendicular to the ground, and sets its reference frame Σ_v to have one specific axis (common to all the cameras) along this direction and pointing "upward".

Remark 4.1 *The natural reference frame Σ'_v of camera v is the optical reference frame, which has two axis lying on the image plane and the z-axis perpendicular to it and crossing the optical center [14]. Implicitly, we assume that the camera v knows its tilt and roll angles, so that it is also able to compute the rotational matrix which transforms Σ'_v in Σ_v.*

In the general case, the element $(g_v, R_v) \in SE(3)$ possesses six degrees of freedom in the sense that g_v and R_v can be parameterized using three independent scalars, and three angles, respectively. The previous assumptions correspond to fixing one of such scalars (the height with respect to the ground) and two of such angles (tilt and roll), in the sense that the pose (g_v, R_v) of the camera v is such that, $\forall v \in \mathcal{V}$,

$$g_v = \begin{bmatrix} \bar{g}_v \\ h_v \end{bmatrix} \qquad R_v = R(\bar{\theta}_v) := \begin{bmatrix} \cos\bar{\theta}_v & -\sin\bar{\theta}_v & 0 \\ \sin\bar{\theta}_v & \cos\bar{\theta}_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Observe that $\bar{g}_v \in \mathbb{R}^2$ is the position, projected on the ground, of camera v, $h_v \in \mathbb{R}$ is the known height of the origin of the reference frame Σ_v, and the angle $\bar{\theta}_v \in [-\pi, \pi)$ represents the rotation about the z axis needed to align Σ_v with Σ_0.

Another consequence of the previous assumptions and the exact knowledge of the tilt and roll angles is that the eight point algorithm naturally provides a matrix $\tilde{R}_{vu}$ having the structure described in (9), namely there exists $\eta_{vu} \in [-\pi, \pi)$ such that $\tilde{R}_{vu} = R(\eta_{vu})$. By exploiting now the relations $R(\theta)R(\varphi) = R(\varphi+\theta)$ and $R(\theta)^T = R(\theta)^{-1} = R(-\theta)$, the model in (8) for measurement of the relative orientation can be rewritten, as in (1), as $\eta_{vu} = (\bar{\theta}_v - \bar{\theta}_u + \varepsilon_{vu})_{2\pi}$, where $\varepsilon_{vu} \in [-\pi, \pi)$ is such that $N_{vu} = R(\varepsilon_{vu})$.

Given two elements $R_1, R_2 \in SO(3)$, their *geodesic* can be be seen as the shortest path joining R_1 and R_2 on the manifold, and the *geodesic distance* $d(R_1, R_2)$ between R_1 and R_2 is simply the length of their geodesic (see [15] for more formal definitions and properties). For elements of $SO(3)$ of the type given in (9) we have

$$d(R(\theta_1), R(\theta_2)) = |(\theta_1 - \theta_2)_{2\pi}|$$

Consequently, one can see that

$$\sum_{\{u,v\}\in\mathcal{E}} d(R(\theta_u)^T R(\theta_v),\ R(\eta_{uv}))^2 = V(\boldsymbol{\theta})$$

namely the cost function proposed in Section 3 corresponds to summing up the squared geodesic distances between $R(\theta_u)^T R(\theta_v)$ and the measurements $R(\eta_{uv})$ along all the edges of the graph. Notice that this cost already appeared in [7].

5 Description of the proposed algorithm

5.1 The regions of convexity of the cost functions

For each $\boldsymbol{K} \in \mathbb{Z}^{\mathcal{E}}$ define the region

$$R_{\boldsymbol{K}}(\boldsymbol{\eta}) := \{\boldsymbol{\theta} \in [-\pi, \pi)^{\mathcal{V}} :\ A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\boldsymbol{K} \in [-\pi, \pi)^{\mathcal{E}}\}. \tag{10}$$

These regions are convex and form a partition of $[-\pi, \pi)^{\mathcal{V}}$. Most of them can actually be empty. It can be seen that, if $\boldsymbol{\theta} \in R_{\boldsymbol{K}}(\boldsymbol{\eta})$, and only for these points, we have

$$V(\boldsymbol{\theta}) = \|A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\boldsymbol{K}\|^2,$$

therefore $V(\boldsymbol{\theta})$ is purely quadratic and convex in $R_{\boldsymbol{K}}(\boldsymbol{\eta})$. For this reason, in each $R_{\boldsymbol{K}}(\boldsymbol{\eta})$ there can be at most one local minimum of $V(\boldsymbol{\theta})$.

Let $\bar{\boldsymbol{K}} \in \mathbb{Z}^{\mathcal{E}}$ be such that $\bar{\boldsymbol{\theta}} \in R_{\bar{\boldsymbol{K}}}(\boldsymbol{\eta})$, namely let $\bar{\boldsymbol{K}}$ identify the region in which the vector of orientations $\bar{\boldsymbol{\theta}}$ lies. Equation (10) yields to

$$q_{2\pi}(A\bar{\boldsymbol{\theta}} - \boldsymbol{\eta}) = \bar{\boldsymbol{K}}.$$

On the other hand, from (2) we have that

$$\boldsymbol{\eta} = A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} - 2\pi q_{2\pi}(A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon}).$$

Since we assumed that $\boldsymbol{\varepsilon} \in [-\pi, \pi)^{\mathcal{E}}$, we can argue that

$$\bar{\boldsymbol{K}} = q_{2\pi}(A\bar{\boldsymbol{\theta}} - \boldsymbol{\eta}) = q_{2\pi}\Big(-\boldsymbol{\varepsilon} + 2\pi q_{2\pi}(A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon})\Big) = q_{2\pi}(A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon})$$

where we used the fact that $q_{2\pi}(x + 2\pi k) = q_{2\pi}(x) + k$, if $k \in \mathbb{Z}$. Consequently, the vector of measurements can be expressed as

$$\boldsymbol{\eta} = A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} - 2\pi\bar{\boldsymbol{K}}. \tag{11}$$

5.2 Estimation of the convexity region

The idea which inspires the algorithm we are going to propose is first to obtain an estimate $\hat{\boldsymbol{K}}$ of $\bar{\boldsymbol{K}}$ and then to obtain $\hat{\boldsymbol{\theta}}$ by minimizing the reshaped cost

$$V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) := \|A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|_2^2 \tag{12}$$

which corresponds to restricting $V(\boldsymbol{\theta})$ to the region $R_{\hat{\boldsymbol{K}}}(\boldsymbol{\eta})$ and then extending the quadratic form to $\mathbb{R}^{\mathcal{V}}$.

Example 5.1 *Let's consider again the Example 3.1. Assume we are still in the ideal noiseless case, so that* $\eta_{12} = \eta_{23} = \eta_{31} = 0$. *Since we fixed* $\theta_1 = 0$, *then* V *is function only of* θ_2, θ_3. *As shown above* $V(0, \theta_2, \theta_3)$ *is quadratic in the three regions corresponding to the case* $\theta_2 - \theta_3 \in (-\pi, \pi]$, *to the case* $\theta_2 - \theta_3 \geq \pi$ *and to the case* $\theta_2 - \theta_3 < -\pi$. *These three regions correspond to* $R_{\boldsymbol{K}}(0)$ *in the three cases* $\boldsymbol{K} = (K_{12}, K_{23}, K_{31}) = (0, 0, 0)$, $\boldsymbol{K} = (0, -1, 0)$, *and* $\boldsymbol{K} = (0, 1, 0)$. *The other regions are empty. This is depicted in right panel on figure 1.*

The idea of how to estimate $\bar{\boldsymbol{K}}$ comes from the observation that the relative differences of the actual orientations $\bar{\theta}_v$ along a cycle must necessarily sum up to a multiple of 2π. More precisely, let h be an oriented cycle and let $\boldsymbol{r}_h$ be its representative vector as explained in Section 2. Then, from (11), using the fact that $\boldsymbol{r}_h A = 0$, we obtain that

$$\boldsymbol{r}_h\boldsymbol{\eta} + 2\pi\boldsymbol{r}_h\bar{\boldsymbol{K}} = \boldsymbol{r}_h\boldsymbol{\varepsilon}$$

and so we can argue that

$$q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta}) + \boldsymbol{r}_h\bar{\boldsymbol{K}} = q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta} + 2\pi\boldsymbol{r}_h\bar{\boldsymbol{K}}) = q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\varepsilon}).$$

On the other hand, if it happens that the algebraic sum of the noise along the cycle h is below π in modulus, namely if $|\boldsymbol{r}_h\boldsymbol{\varepsilon}| < \pi$, then we obtain that $q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\varepsilon}) = 0$. This yields

$$\boldsymbol{r}_h\bar{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta}). \tag{13}$$

Observe that in this case $\boldsymbol{r}_h\bar{\boldsymbol{K}}$ could be exactly computed on the basis of the measurements $\boldsymbol{\eta}$ along the cycle h. This would suggest to define implicitly the estimation $\hat{\boldsymbol{K}}$ by imposing $\boldsymbol{r}_h\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta})$ for any cycle h. However, the resulting system of equations would not be solvable in general due to the non–coherence of the noises along cycles. What can be done is to choose a basis for Γ, and impose the constraint on the corresponding cycles.

Inspired by the previous observations, assume that a generic basis of Γ is given, and let $R \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times\mathcal{E}}$ the matrix whose rows are the elements of the basis. Denote by $\hat{\boldsymbol{K}}_R$ the estimate of $\bar{\boldsymbol{K}}$ we want to obtain using R. More precisely, we want to find $\hat{\boldsymbol{K}}_R$ satisfying the following system of equations on $\mathbb{Z}^{\mathcal{E}}$ in matrix form

$$R\hat{\boldsymbol{K}}_R = -q_{2\pi}(R\boldsymbol{\eta}). \tag{14}$$

By Propositions 2.1 and 2.2 we obtain that the general solution of this equation is

$$\hat{\boldsymbol{K}}_R = -X_R q_{2\pi}(R\boldsymbol{\eta}) + A\boldsymbol{h} \tag{15}$$

where X_R is the matrix, introduced in Proposition 2.2 such that $RX_R = I$ and $\boldsymbol{h}$ is any column vector in $\mathbb{Z}^{\mathcal{V}}$.

Once $\hat{\boldsymbol{K}}_R$ has been selected, the final solution $\hat{\boldsymbol{\theta}}_R$ (which depends on the choice of R) is determined by minimizing the cost

$$V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) = \|A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}_R\|_2^2.$$

The solution of the previous minimization is the solution of the following linear equation

$$A^T A\hat{\boldsymbol{\theta}}_R = A^T\boldsymbol{\eta} + 2\pi A^T\hat{\boldsymbol{K}}_R. \tag{16}$$

As done above, in order to avoid the non uniqueness due to the kernel of A, we fix $\theta_1 = 0$. Then consider the vector $\boldsymbol{\xi} \in \mathbb{R}^{\mathcal{V}}$ defined by $\xi_1 = 1$ and $\xi_v = 0$ for any $v \neq 1$. It is well-known [16] that there exists a symmetric matrix $G \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ such that

$$\begin{cases} GA^T A = I - \mathbb{1}\boldsymbol{\xi}^T, \\ G\boldsymbol{\xi} = 0. \end{cases} \tag{17}$$

This matrix is called the Green matrix associated to the graph $\mathcal{G}$. Using the properties of G and the expression of $\hat{\boldsymbol{K}}_R$ we obtain

$$\hat{\boldsymbol{\theta}}_R = GA^T\boldsymbol{\eta} + 2\pi GA^T\hat{\boldsymbol{K}}_R = GA^T\boldsymbol{\eta} - 2\pi GA^T X_R q_{2\pi}(R\boldsymbol{\eta}) + 2\pi GA^T A\boldsymbol{h}.$$

Recall that we look for an estimate in $[-\pi,\pi)^{\mathcal{V}}$, thus the nodes need to project onto this set the computed solution. This can be done entry–wise, and corresponds to the choice

$$\boldsymbol{h} := -q_{2\pi}\left(GA^T\boldsymbol{\eta} - 2\pi GA^T X_R q_{2\pi}(R\boldsymbol{\eta})\right).$$

Using the definition of Green matrix, one can show that $h_1 = 0$, thus obtaining

$$\hat{\boldsymbol{\theta}}_R = GA^T\boldsymbol{\eta} - 2\pi GA^T X_R q_{2\pi}(R\boldsymbol{\eta}) + 2\pi\boldsymbol{h} = \left(GA^T\boldsymbol{\eta} - 2\pi GA^T X_R q_{2\pi}(R\boldsymbol{\eta})\right)_{2\pi} \in [-\pi,\pi)^{\mathcal{V}}. \tag{18}$$

A simple continuity argument then shows that, when the threshold $\bar{\varepsilon}$ tends to 0, the estimate $\hat{\boldsymbol{\theta}}_R$ converges to $\bar{\boldsymbol{\theta}}$. In other terms, if the noise is small, we have a guarantee that our solution is close to the true $\bar{\boldsymbol{\theta}}$. In the following Sections we formally prove this statement.

5.3 Performance analysis of the proposed algorithm

As mentioned in Section 3, in order to evaluate the performance of the proposed algorithm, we need to estimate the index $W = \frac{1}{N}\|(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})_{2\pi}\|^2$ defined in (3). Observe that, since $\boldsymbol{\eta} = A\bar{\boldsymbol{\theta}} + \boldsymbol{\varepsilon} - 2\pi\bar{\boldsymbol{K}}$ and $X_R q_{2\pi}(R\boldsymbol{\eta}) = -\hat{\boldsymbol{K}}_R + A\boldsymbol{h}$ then (18) can be rewritten

$$\begin{aligned}\hat{\boldsymbol{\theta}}_R &= GA^T\boldsymbol{\eta} - 2\pi GA^T X_R q_{2\pi}(R\boldsymbol{\eta}) + 2\pi\boldsymbol{h}\\ &= GA^TA\bar{\boldsymbol{\theta}} + GA^T\boldsymbol{\varepsilon} - 2\pi GA^T\bar{\boldsymbol{K}} + 2\pi GA^T\hat{\boldsymbol{K}}_R - 2\pi GA^TA\boldsymbol{h} + 2\pi\boldsymbol{h}\end{aligned}$$

Since $\bar{\theta}_1 = 0$ and $h_1 = 0$ we can argue that

$$\hat{\boldsymbol{\theta}}_R = \bar{\boldsymbol{\theta}} + GA^T\boldsymbol{\varepsilon} + 2\pi GA^T(\hat{\boldsymbol{K}}_R - \bar{\boldsymbol{K}}), \tag{19}$$

which shows that the computed estimate differs from the vector of actual orientations by a term depending only on the measurement noise. The second term is instead related to the ability to correctly identify the region in which $\bar{\boldsymbol{\theta}}$ lies.

The previous equation yields

$$W = \frac{1}{N}\|(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}})_{2\pi}\|^2 = \frac{1}{N}\|(GA^T\boldsymbol{\varepsilon} + 2\pi GA^T(\hat{\boldsymbol{K}}_R - \bar{\boldsymbol{K}}))_{2\pi}\|^2.$$

This equality implies that, in case $R\hat{\boldsymbol{K}}_R = R\bar{\boldsymbol{K}}$, namely if we have that $\hat{\boldsymbol{K}}_R = \bar{\boldsymbol{K}} + A\boldsymbol{h}$ for some $\boldsymbol{h}$, then

$$\begin{aligned}W &= \frac{1}{N}\|(GA^T\boldsymbol{\varepsilon} + 2\pi GA^TA\boldsymbol{h})_{2\pi}\|^2 = \frac{1}{N}\|(GA^T\boldsymbol{\varepsilon} + 2\pi\boldsymbol{h})_{2\pi}\|^2\\ &= \frac{1}{N}\|(GA^T\boldsymbol{\varepsilon})_{2\pi}\|^2 \leq \frac{1}{N}\|GA^T\boldsymbol{\varepsilon}\|^2.\end{aligned}$$

Notice that the term $\frac{1}{N}\|GA^T\boldsymbol{\varepsilon}\|^2$ has been already studied in the literature (see [2,4,17]), since it characterizes the performance of the calibration algorithm over vector spaces. In these papers the electrical analogy is used. The graph

$\mathcal{G}$ is considered as an electrical network where there is a resistance of 1 Ohm along all the edges in $\mathcal{G}$. If we denote by $\mathcal{R}_{v1}$ the effective resistance among the node v and the anchor node 1, then it can be shown that

$$\frac{1}{N}\mathbb{E}\|GA^T\boldsymbol{\varepsilon}\|^2 = \frac{\sigma^2}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1},$$

where we are assuming that the components of the noise vector $\boldsymbol{\varepsilon}$ have zero mean and variance σ^2. This result allows to obtain the estimate of $\frac{1}{N}\mathbb{E}\|GA^T\boldsymbol{\varepsilon}\|^2$ as a function of the number of nodes for some families of graphs. For example, if the graph is a line, then

$$\frac{1}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1} \approx N.$$

If the graph is a 2D grid, then

$$\frac{1}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1} \approx \log N,$$

while if the graph is a grid in dimension greater or equal to 3, then $\frac{1}{N}\sum_{v\in\mathcal{V}}\mathcal{R}_{v1}$ is bounded from above by a constant independent of N. This qualitative behavior can be of interest for the designer. We will show this feature in the examples proposed in Section 8.

Remark 5.1 *Observe that, since*

$$R\hat{\boldsymbol{K}}_R = -q_{2\pi}(R\boldsymbol{\eta}) = -q_{2\pi}(R\boldsymbol{\varepsilon} - 2\pi R\bar{\boldsymbol{K}}) = R\bar{\boldsymbol{K}} - q_{2\pi}(R\boldsymbol{\varepsilon}),$$

then $R\hat{\boldsymbol{K}}_R = R\bar{\boldsymbol{K}}$ occurs if the components of the vector $R\boldsymbol{\varepsilon}$ belong to $[-\pi,\pi)$, and so if $||R\boldsymbol{\varepsilon}||_\infty < \pi$ [2]. If the components of the noise vector $\boldsymbol{\varepsilon}$ are supported in $[-\bar{\varepsilon},\bar{\varepsilon}]$, since $||R\boldsymbol{\varepsilon}||_\infty \leq \bar{\varepsilon}||R||_\infty$, then

$$||R||_\infty < \frac{\pi}{\bar{\varepsilon}}$$

implies that $R\hat{\boldsymbol{K}}_R = R\bar{\boldsymbol{K}}$. Notice finally that, if the rows of R are obtained from a set of closed paths, then $||R||_\infty$ is simply the maximum length of the paths in this set. For this reason it is convenient to choose R whose rows are obtained from paths of minimum length. This fact inspires the algorithms proposed in the next Section.

6 Distributed algorithms for rotational calibration

In this Section we will propose two possible distributed implementations of the rotational calibration algorithm proposed in the previous Section. Observe that the proposed algorithm is based on two steps. The first consists in the estimation of $\bar{\boldsymbol{K}}$ based on formulas (14) and (15). The second step consists in the estimation of $\bar{\boldsymbol{\theta}}$ based on formula (16). Distributed implementations of the second step have already been proposed in the literature (see [4,18]). We propose here two distributed implementations of the first step, which are based on two ways to select the set of cycles as a basis of Γ.

6.0.0.1 Fundamental cycles Fix a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_\mathcal{T})$ of $\mathcal{G}$. Order arbitrarily the edges in $\mathcal{E}\setminus\mathcal{E}_\mathcal{T}$ as $e_1,\ldots,e_{M-N+1}$ and consider cycles $h_1,\ldots,h_{M-N+1}$ in $\mathcal{G}$ such that for each i we have that $\mathrm{supp}(h_i) \subseteq \mathcal{E}_\mathcal{T}\cup\{e_i\}$ and $\boldsymbol{r}_{h_i}(e_i) = 1$. In words, cycle h_i is constructed with the edges in $\mathcal{E}_\mathcal{T}\cup\{e_i\}$ and oriented in such a way that $\boldsymbol{r}_{h_i}(e_i) = 1$. Such cycles are called *($\mathcal{T}$-)fundamental cycles*.

[2] We recall that, if $x\in\mathbb{R}^n$, then $||x||_\infty = \max_i\{|x_i|\}$ and if $M\in\mathbb{R}^{n\times m}$, then $||M||_\infty = \max_i\{\sum_j |M_{ij}|\}$. According to these definitions we have that $||Mx||_\infty \leq ||M||_\infty||x||_\infty$.

6.0.0.2 Minimal cycles Fix again a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G}$. Another possible construction is the following iterative one:

- Among all cycles whose edges are all in $\mathcal{E}_{\mathcal{T}}$ except one, choose one of minimal length. Call it h_1 and call e_1 the only edge in h_1 which is not in $\mathcal{E}_{\mathcal{T}}$.
- Suppose edges $e_1, \dots, e_i$ and cycles $h_1, \dots, h_i$ have been selected. Among all cycles whose edges are all in $\mathcal{E}_{\mathcal{T}} \cup \{e_1, \dots e_i\}$ except one, choose one of minimal length. Call it h_{i+1}, and call e_{i+1} the only edge in h_{i+1} which is not in $\mathcal{E}_{\mathcal{T}} \cup \{e_1, \dots e_i\}$. Such cycles are called *($\mathcal{T}$-)minimal cycles*.

Both the fundamental and the minimal cycles provide a basis of Γ as shown in the next Lemma whose proof is given in the Appendix.

Lemma 6.1 *Fix a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G}$. The subsets of Γ:*

$$\{r_h \,|\, h \ \mathcal{T}\text{-fundamental cycle}\}\,,\ \{r_h \,|\, h \ \mathcal{T}\text{-minimal cycle}\}$$

are both $\mathbb{Z}$-basis of Γ.

6.1 The Tree-algorithm

In this distributed implementation the agents are supposed to be the nodes and the edges composing the graph $\mathcal{G}$. Fix a spanning graph $\mathcal{T}$ rooted at the anchor node 1. In order to obtain $\hat{\boldsymbol{K}}$, we need first to obtain the value $\boldsymbol{r}_{h_i}\boldsymbol{\eta}$, for each cycle h belonging to the set of the fundamental cycles. This value will be stored by the agent associated to the edge e_i, which is the edge in h_i not belonging to $\mathcal{T}$. For the computation of $\boldsymbol{r}_{h_i}\boldsymbol{\eta}$, first the measurements are propagated along the tree starting from the root. In other words, given a node v and called $f(v)$ its father, we set $\hat{\theta}_{FE,\,1} = 0$ and

$$\hat{\theta}_{FE,\,v} = \hat{\theta}_{FE,\,f(v)} + \eta_{v,f(v)}. \tag{20}$$

As a side effect, we also obtain a first estimate $\hat{\boldsymbol{\theta}}_{FE}$ of $\bar{\boldsymbol{\theta}}$. Then each edge e_i not belonging to $\mathcal{T}$ can compute $\boldsymbol{r}_{h_i}\boldsymbol{\eta}$ as

$$\boldsymbol{r}_{h_i}\boldsymbol{\eta} = \hat{\theta}_{FE,\,s(e_i)} - \hat{\theta}_{FE,\,t(e_i)} + \eta_{e_i}.$$

Now we build $\hat{\boldsymbol{K}}$ as follows. For each edge e in $\mathcal{T}$, we assign $\hat{\boldsymbol{K}}_e = 0$. Instead, for each edge e_i not belonging $\mathcal{T}$, we assign

$$\hat{\boldsymbol{K}}_{e_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta}).$$

Observe that this assignment of $\hat{\boldsymbol{K}}$ ensures that, for all the fundamental cycles h_i, it holds

$$\boldsymbol{r}_{h_i}\hat{\boldsymbol{K}} = \hat{\boldsymbol{K}}_{e_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta})$$

that is needed to correctly estimate $\bar{\boldsymbol{K}}$. Once $\hat{\boldsymbol{K}}$ is obtained, the nodes can compute in a distributed way the components $\left(\text{argmin}_{\boldsymbol{\theta}}\|A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|\right)_{2\pi}$.

6.2 Minimal cycles-algorithm

The second algorithm exploits the construction of a set of minimal cycles for the graph. The procedure in this case is not fully distributed, in the sense that we must assume that each *minimal cycle* is associated with an agent which carries on the computations associated with this cycle.

Fix a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G}$ and let $h_1, \dots, h_{M-N+1}$ be the set of minimal cycles. Benote by $e_1, \dots, e_{M-N+1}$ the associated edges forming the set $\mathcal{E} \setminus \mathcal{T}$ ordered in the way described above. As in the previous case, assign $\hat{\boldsymbol{K}}_e = 0$ for any $e \in \mathcal{T}$. Consider now the remaining edges $e_1, e_2, \dots, e_{M-N+1}$. We know that cycle h_1 has edges in $\mathcal{T} \cup \{e_1\}$. Since $\hat{\boldsymbol{K}}_e = 0, \forall e \neq e_1$ for any e in the support of h_1, we can assign

$$\hat{\boldsymbol{K}}_{e_1} = -q(\boldsymbol{r}_{h_1}\boldsymbol{\eta}).$$

Algorithm 1 Tree-Algorithm

(Input variables)
$\bar{\theta}_1$, value of the anchor;
$\eta_e, e = 1, \dots, M$;
$\mathcal{T}$ spanning tree;

(Step A: first estimate $\hat{\boldsymbol{\theta}}_{FE}$)
$\hat{\theta}_{FE,1} = 0$;
for $i = 1, \dots, N$ **do**
  for $j = 2, \dots, N$ **do**
    if j is a son of i in $\mathcal{T}$ **then** $\hat{\theta}_{FE,j} = \hat{\theta}_{FE,i} + \eta_{j,i}$;

(Step B: estimate $\hat{\boldsymbol{K}}$)
for $e \in \mathcal{E}$ **do**
  $\hat{\boldsymbol{K}}_e = q_{2\pi}(\hat{\theta}_{FE,s(e)} - \hat{\theta}_{FE,t(e)} - \eta_e)$;

(Step C: second estimate $\hat{\boldsymbol{\theta}}$)
compute $\left(\text{argmin}_{\boldsymbol{\theta}} \|A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|\right)_{2\pi}$ s.t. $\theta_1 = 0$.

Then, we proceed iteratively, and at the i-th iteration we already computed $\hat{\boldsymbol{K}}_{e_1}, \dots, \hat{\boldsymbol{K}}_{e_{i-1}}$. In order to assign $\hat{\boldsymbol{K}}_{e_i}$, observe that the following relation must be satisfied

$$\boldsymbol{r}_{h_i}\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta}).$$

Therefore, we derive

$$\hat{\boldsymbol{K}}_{e_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta}) - \sum_{j \neq i} \boldsymbol{r}_{h_i}(e_j)\hat{\boldsymbol{K}}_{e_j}.$$

In order to achieve this computation, the agent associated with the cycle h_i has to know η_{e_j} and $\hat{\boldsymbol{K}}_{e_j}$ for each edge e_j belonging to h_i. As a consequence, this agent needs to receive the values of $\hat{\boldsymbol{K}}_{e_j}$ from the agents associated to adjacent cycles, before being able to compute $\hat{\boldsymbol{K}}_{e_i}$. After the estimation of $\hat{\boldsymbol{K}}$, the remaining steps are the same as those of the previous algorithm.

Algorithm 2 Minimal cycles-algorithm

(Input variables)
$\eta_e, e = 1, \dots, M$;
$\mathcal{T}$ spanning tree;
$\boldsymbol{r}_1, \dots, \boldsymbol{r}_{M-N+1}$ minimal cycles set;

(Step A: computation of $\boldsymbol{b} = -q_{2\pi}(R\boldsymbol{\eta})$)
for $i = 1, \dots, M - N + 1$ **do** $b_{h_i} = -q_{2\pi}(\boldsymbol{r}_{h_i}\boldsymbol{\eta})$;

(Step B: estimate $\hat{\boldsymbol{K}}$)
for $e \in \mathcal{E}_{\mathcal{T}}$ **do** $\hat{\boldsymbol{K}}_e = 0$;
for $i = 1, \dots, M - N + 1$ **do** $\hat{\boldsymbol{K}}_{\bar{e}} = b_{h_i} - \sum_{j=1}^{i-1} \boldsymbol{r}_{h_i}(e_j)\hat{\boldsymbol{K}}_{e_j}$;

(Step C: second estimate $\hat{\boldsymbol{\theta}}$)
compute $\left(\text{argmin}_{\boldsymbol{\theta}} \|A\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|\right)_{2\pi}$ s.t. $\theta_1 = 0$.

This second algorithm allows much better performance than the first one, but it requires a greater amount of communication and collaboration among nodes. In fact, as already mentioned, it is required that an agent is associated with each cycle, and it must know all the measurements along the edges of its cycle.

Let us illustrate how the *Minimal cycles-algorithm* works with the following simple example.

Example 6.1 *Consider the simple graph in Figure 3. For such a graph, the minimal cycles are* $h_1, \dots, h_5$*, and the edges are* $1, \dots, 13$*. Assume that* $b_1 = 1$*,* $b_2 = 2$*,* $b_3 = b_4 = b_5 = 0$*, where* $\boldsymbol{b} = -q_{2\pi}(R\boldsymbol{\eta})$*. Edges* $1, \dots, 8$ *form a spanning tree* $\mathcal{T}$ *of the graph. First of all, set* $\hat{\boldsymbol{K}}_1 = \dots \hat{\boldsymbol{K}}_8 = 0$*. Now, cycles* h_1 *and* h_2 *are made of edges of the tree apart from the edges* 9 *and* 10 *respectively. Thus* h_1 *sets* $\hat{\boldsymbol{K}}_9 = 1$*, while* h_2 *sets* $\hat{\boldsymbol{K}}_{10} = -2$*, since the direction of* 10 *is incoherent with the orientation of* h_2*. Once this is done, we know the value of* $\hat{\boldsymbol{K}}$ *along all the edges of* h_3 *and* h_4*, apart from* 12 *and* 11 *respectively. In order to guarantee that the sum over the cycles* h_3 *and* h_4 *is equal to* b_3 *and* b_4 *respectively, we can derive* $\hat{\boldsymbol{K}}_{12} = 2$ *and* $\hat{\boldsymbol{K}}_{11} = 1$*. Finally, all the edges* $e \neq 13$ *of* h_5 *have have their corresponding* $\hat{K}_e$ *already assigned, so it suffices to set* $\hat{\boldsymbol{K}}_{13} = -3$*. Now the sum of* $\hat{\boldsymbol{K}}$ *over the five minimal cycles corresponds to* $\boldsymbol{b}$ *entry–wise.*

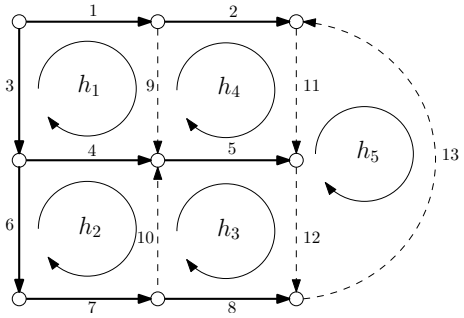

Fig. 3. A simple graph to show how the second algorithm works.

7 Resilience against measurement noise for different graph topologies

In this Section we compare the two algorithms we have proposed for several different graph topologies. We concentrate on grid-like topologies, since they can be used to model real networks of cameras. In order to draw the comparison, consider the graphs shown in figure 4. In both cases we have a line–like graph with many nodes deployed along one dimension, and the chosen spanning trees are shown in thick lines. They are rooted on the anchor on the most left-top node. The set of the minimal cycles basis is simply the set of squares which form the graph.

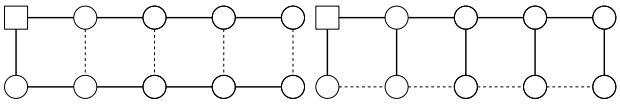

Fig. 4. Two examples of spanning trees for a line–like graph. The proposed algorithms work in a similar manner for the one on the right, while the *Minimal cycles-algorithm* is far more effective for the one on the left.

For the graph on the left, if we take the tree and we add the last edge on the right we obtain a cycle with maximum length N. On the contrary, the minimal cycles are of length $L_0 = 4$. As an immediate consequence, the *Minimal cycles-algorithm* has much better performance since the upper bound $\bar{\varepsilon} < \frac{\pi}{4}$ is independent on the number of nodes. On the other hand, in order the *Tree-algorithm*, to produce a good estimate $\hat{\boldsymbol{K}}$, the magnitude of the noise should decrease with the dimension of the graph.

If we consider instead the spanning tree on the right, we can see that the fundamental cycles have length 4 as well, since the spanning tree is chosen in a much better way. In this case, the two algorithms have the same performance.

If we consider the ring graph in Figure 5, we can easily see that there is only one minimal cycle. Here the two proposed algorithms coincide. In such a case, the *Tree-algorithm* is better, since it is easier to implement and completely distributed, it requires less information on the topology of the network, as well as less communications.

Indeed, consider the $2D$ grid on the right in Figure 5. The comb-shaped spanning tree is the one in thick line. Here we see that the maximum length of the fundamental cycles grows as $\sqrt{N}$. Since it can be proved that the maximum length of fundamental cycles in any spanning tree on a $2D$ grid grows *at least* as $\sqrt{N}$, the chosen spanning tree already achieves the maximum resilience to measurement noise. On the contrary, the minimum cycles always have length equal to 4. Thus, in this case the *Minimal cycles-algorithm* has always better performance than the *Tree-algorithm*. Notice that the choice of the spanning tree is fundamental to draw a comparison between the algorithms.

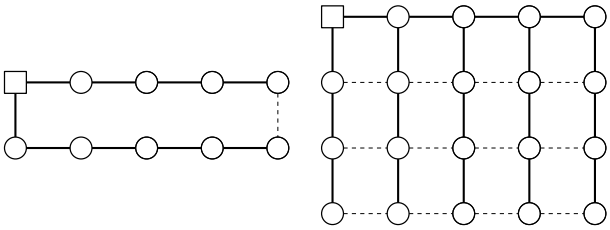

Fig. 5. On the left a ring graph, for which the two algorithms have the same performance. On the right, a grid graph.

8 Numerical results

In this Section we provide numerical comparison between the two approaches we propose in this paper, and the frame localization algorithm proposed in [8].

In the first experiment we simulate the *Tree-algorithm* and the *Minimal cycles-algorithm* on square grids of size $N = n^2$, for n ranging from 3 up to 20. An example of square grid is depicted in Figure 6 (left panel), where $n = 4$. In the same Figure, we also depict the type of spanning tree we use to build the $\mathcal{T}$-fundamental cycles. The construction is analogous for different values of n.

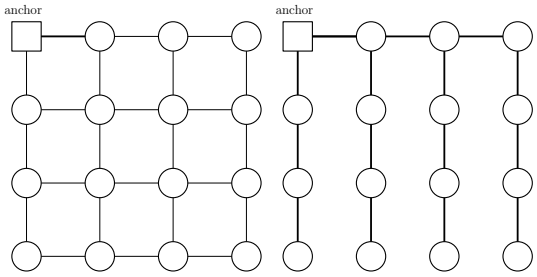

Fig. 6. On the left a square grid graph for $n = 4$. On the right the correspondent spanning tree used in simulations.

In all simulations we set $\bar{\theta}_1 = 0$, while, for $v \in \{2, \dots, N\}$, $\bar{\theta}_v$ is randomly sampled from a uniform distribution on $[-\pi, \pi)$. The values of the noises ε_e, $e \in \mathcal{E}$, are randomly sampled from a uniform distribution on $[-\bar{\varepsilon}, \bar{\varepsilon}]$, where $\bar{\varepsilon} = \frac{\pi}{8}$. For each n, the values we plot are averaged over 200 trials. Different $\bar{\boldsymbol{\theta}}$ and a different set of noises are generated for each trial.

On the left panel of Figure 7, we plot the estimate error

$$W(\hat{\theta}) = \frac{1}{N} \|(\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}})_{2\pi}\|^2$$

for both the *Tree-algorithm* and the *Minimal cycles-algorithm*. On the right panel, we plot the error on $\bar{\boldsymbol{K}}$ defined as

$$e_K = \frac{1}{M} \|\bar{\boldsymbol{K}} - \hat{\boldsymbol{K}}\|^2.$$

The choice for $\bar{\varepsilon}$ was dictated by the inequality

$$\frac{\pi}{L_0} = \frac{\pi}{4} > \frac{\pi}{8} = \bar{\varepsilon},$$

which implies that the *Minimal cycles-algorithm* always correctly estimates $\bar{\boldsymbol{K}}$. On the contrary, observe that, in the case of the *Tree-algorithm*, e_K is zero only for low values of n. To conclude, notice that the behavior of $W(\hat{\theta})$ in the case of minimal cycles is approximately logarithmic in the number of nodes N, as predicted in Section 5.3. As expected, the *Minimal cycles-algorithm* outperforms the *Tree-algorithm*.

The second experiment concerns the comparison with the procedure proposed in [8], which we call FL for Frame Localization. The goal of this algorithm, as ours, is to obtain an estimate of $\bar{\boldsymbol{\theta}}$ from the measurements $\boldsymbol{\eta}$. The basic

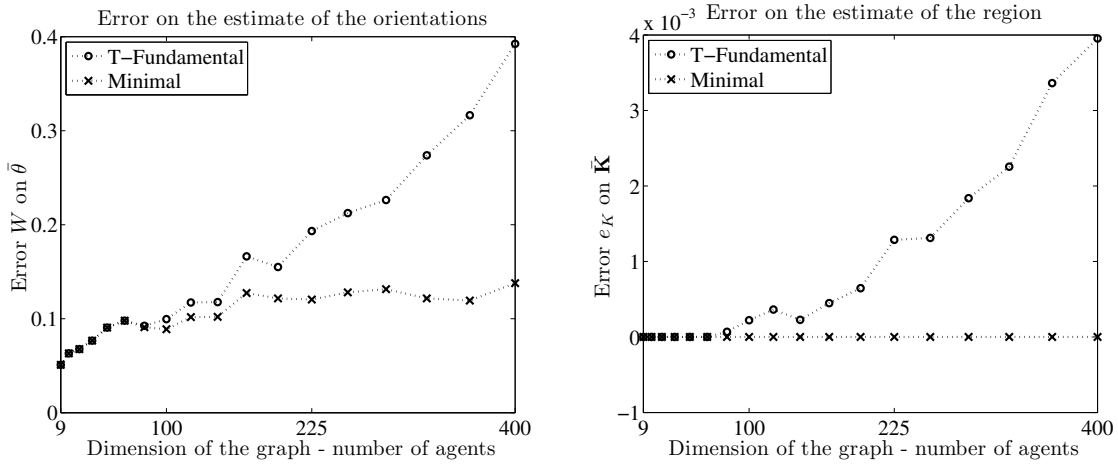

Fig. 7. Average error on the orientations (modulo 2π) and error on $\bar{\boldsymbol{K}}$ in case of $2-D$ grid with $N = 9, \ldots, 400$. The circle–marked plot corresponds to the *Minimal cycles-algorithm*, the crossed-marked one to the *Tree-algorithm*.

idea of the FL algorithm is that measurements along cycles of the graph should sum up to multiples of 2π, as it happens in a noise–free scenario. Thus, in the FL algorithm first of all a $\mathbb{Z}$–basis of Γ is selected. Then, it aims to solve the following minimization problem

$$\hat{\boldsymbol{\psi}} = \underset{\boldsymbol{\psi}\in\mathbb{R}^{\mathcal{E}} \text{ s.t. } (R\boldsymbol{\psi})_{2\pi}=0}{\arg\min} ||(\boldsymbol{\psi}-\boldsymbol{\eta})_{2\pi}||_2, \tag{21}$$

where the rows of $R \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times\mathcal{E}}$ are the $\mathbb{Z}$–basis of Γ. In order to minimize the expression in (21), the authors propose the following iterative procedure

$$\begin{cases} \boldsymbol{\psi}(0) = \boldsymbol{\eta} \\ \boldsymbol{\psi}(t+1) = \boldsymbol{\psi}(t) - \kappa R^T (R\boldsymbol{\psi}(t))_{2\pi} \end{cases}$$

where κ is a strictly positive real number. If κ is small enough, then the algorithm converges to a vector $\boldsymbol{\psi}_\infty$ such that $(R\boldsymbol{\psi}_\infty)_{2\pi} = 0$ (Theorem 12, [8]). Once $\boldsymbol{\psi}_\infty$ is determined, the estimate $\hat{\boldsymbol{\theta}}$ can be found spreading the information from the anchor, similarly to what is done in (20).

Remark 8.1 *It is worth remarking that, for both the* Tree-algorithm *and the* Minimal cycles-algorithm*, it is possible to provide a closed form expression for the estimate of $\bar{\boldsymbol{\theta}}$, see (19). Further research might lead to a better understanding of the influence of the measurement noise on the algorithm performance. Instead, as far as the FL algorithm is concerned, no closed formula is known for $\boldsymbol{\psi}_\infty$, which seems to depend on the choice of the parameter κ. Moreover, there is no guarantee that $\boldsymbol{\psi}_\infty$ is the solution to the minimization problem in (21).*

In the comparisons, we use the same set of measurements for all the estimation algorithms, and the FL algorithm is run taking the families of $\mathcal{T}$-fundamental cycles and the minimal cycles. The results of the first simulation are shown in Figure 8, left panel. In this case, the threshold for the error is again set at $\bar{\varepsilon} = \frac{\pi}{8}$. The plots are the averaged value of $W(\hat{\boldsymbol{\theta}})$ over 200 simulations. The Figure depicts the plots for the *Tree-algorithm*, the *Minimal cycles-algorithm*, and the FL algorithm run using the $\mathcal{T}$-fundamental cycles. The plot for FL algorithm run using the minimal cycles is not shown, since it overlaps the plot of the *Minimal cycles-algorithm*.

It can be seen from the Figure that, for low values of n, $\bar{\boldsymbol{K}}$ is correctly estimated by the *Tree-algorithm* and all the algorithms seem to provide similar performance. For higher values of n, the FL algorithm seems to perform better on the $\mathcal{T}$-fundamental cycles. This suggests that also for the FL algorithm the length of the cycles influences the performance.

On right panel in Figure 8, we propose a second comparison in case the measurements are corrupted by a large noise, setting $\bar{\varepsilon} = \frac{\pi}{3}$. Notice thus that, in this case, there is no guarantee that $\bar{\boldsymbol{K}}$ is correctly estimated, even if we

exploit the minimal cycles. The plotted values are obtained averaging over 100 simulations on grids of dimension $n = 3, 4, \ldots, 10$. In this case, we show the plots of the results using the FL algorithm on both $\mathcal{T}$-fundamental cycles and minimal cycles. As in the previous simulations, the *Tree-algorithm* is always outperformed by the FL algorithm, but it is also true that the *Minimal cycles-algorithm* gives the best performance. From our numerical experiments, the FL algorithm has somehow an intermediate performance, and they show anyway that the shorter the cycles are, the better the estimates are.

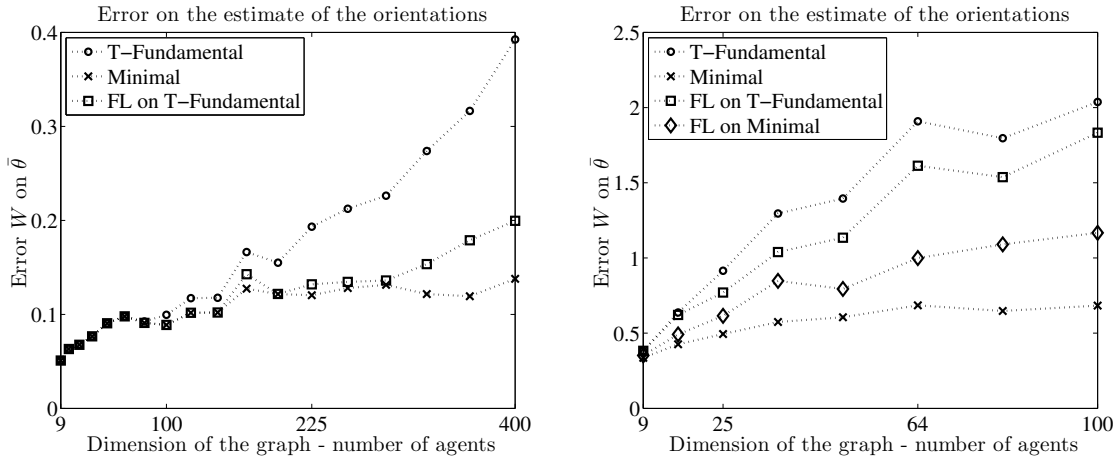

Fig. 8. Comparison with the Frame Localization algorithm. On left panel, average error on the orientations (modulo 2π) in case of $2-D$ grid with $N = 9, \ldots, 400$ and small measurement noise. On the right panel, in the case of $N = 9, \ldots, 100$ with large measurement noise. The plot compares the *Minimal cycles-algorithm* (cross–marks), the *Tree-algorithm* (circle-marks), the FL algorithm using $\mathcal{T}$-fundamental cycles (square-marks), and the FL algorithm using minimal cycles (diamond-marks). On the left panel, the results of FL algorithm on minimal cycles are not shown since they overlap those of *Minimal cycles-algorithm*.

9 Conclusions

This paper proposes two versions of an algorithm which allows a network of cameras to autonomously calibrate, more precisely to determine an estimate of their orientations w.r.t. a common reference frame. This algorithm is suitable to be distributed rather easily over the cameras which can act as agents in a multi-agent system. The proposed algorithm is based on a two steps procedure. In the first step, the algorithm estimates a vector of integers $\bar{\boldsymbol{K}}$ coding the region of convexity which the global minimum is believed to belong to. In the second step, the algorithm performs the computation of the minimizer belonging to the selected region of convexity. Compared with other existing algorithms, this procedure permits to understand the properties of the proposed solution and to understand when this solution is correct.

More investigations are needed for obtaining in this field of research, in the following directions.

- We believe that the proposed algorithm could be improved by obtaining a better estimate of the vector of integers $\bar{\boldsymbol{K}}$ based on the maximum a posteriori MAP estimation or on the maximum likelihood ML estimation.
- A more refined performance analysis has to be done, in terms of the error to gain the estimate $\bar{\boldsymbol{K}}$, and in terms of the index $W(\theta)$ proposed in (3).
- Finally, the more general case of cameras deployed in $\mathbb{R}^3$, and thus rotational calibration in $SO(3)$, needs to be addressed.

References

[1] H. Aghajan and A. Cavallaro. *Multi-Camera Networks: Principles and Applications*. Academic Press., 2009.

[2] P. Barooah and J. P. Hespanha. Distributed estimation from relative measurements in sensor networks. In *Proceedings of the 2nd International Conference on Intelligent Sensing and Information Processing*, Dec. 2005.

[3] P. Barooah, N. M. da Silva, and J. P. Hespanha. Distributed optimal estimation from relative measurements for localization and time synchronization. In *Distributed Computing in Sensor Systems*, volume 4026 of *Lect. Notes in Comput. Science*, pages 266–281. Springer, Berlin, June 2006.

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

[5] A. Sarlette. *Geometry and Symmetries in Coordination Control.* PhD thesis, University of Liège, 2009.

[6] A. Sarlette and R. Sepulchre. Consensus optimization on manifolds. *SIAM Journal on Control and Optimization*, 58:56–76, 2009.

[7] Roberto Tron and René Vidal. Distributed image-based 3-d localization of camera sensor networks. In *Proceedings of the 49th IEEE Conference on Decision and Control CDC'09*, pages 901–908, 2009.

[8] G. Piovan, I. Shames, B. Fidan, F. Bullo, and B. D. O. Anderson. On frame and orientation localization for relative sensing networks. *Automatica*, 2011.

[9] A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, 30(1):20–36, 2011.

[10] W.J. Russell, D.J. Klein, and J.P. Hespanha. Optimal estimation on the graph cycle space. In *American Control Conference. ACC'10*, pages 1918 –1924, 2010.

[11] D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri. Autonomous calibration algorithms for networks of cameras. In *Proceedings of the Americal Control Conference, ACC'12.*, July 2012.

[12] R. Diestel. *Graph Theory.* Springer. Graduate Texts in Mathematics., August 2005.

[13] R. Tron, B. Afsari, and R. Vidal. Average consensus on riemannian manifolds with bounded curvature. In *Proceedings of the 51th IEEE Conference on Decision and Control CDC'11*, 2011.

[14] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry. *An Invitation to 3D Vision: From Images to Geometric Models.* Springer Verlag, 2003.

[15] M.P. do Carmo. *Riemannian Geometry.* Birkhäuser. Mathematics: Theory & Applications, 1992.

[16] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs.* , .

[17] E. Lovisari, F. Garin, and S. Zampieri. Resistance-based performance analysis of the consensus algorithm over geometric graphs. *SIAM Journal on Control and Optimization*, 2012.

[18] S. Bolognani, S. Del Favero, L. Schenato, and D. Varagnolo. Consensus-based distributed sensor calibration and least-square parameter identification in wsns. *International Journal of Robust and Nonlinear Control*, 20(2), January 2010.

[19] M. Newman. *Integral Matrices.* Academic Press. Pure and Applied Mathematics. A Series of Mongraphs and Textbooks., 1972.

[20] T.W. Hungeford. *Algebra.* Springer. Graduate Texts in Mathematics., December 1980.

A Appendix: Proofs

In this Appendix, we give some algebraic properties of the row vectors $\boldsymbol{r}_h$ associated with the paths h. We start by observing that,

$$(\boldsymbol{r}_{(v_1,v_2)}A)_v = \sum_{e\in\mathcal{E}} \boldsymbol{r}_{(v_1,v_2)}(e)A_{ev} = A_{\{v_1,v_2\},v}A_{\{v_1,v_2\},v} = \begin{cases} 1, & \text{if } v = v_1, \\ -1, & \text{if } v = v_2, \\ 0, & \text{if } v \neq v_1, v_2. \end{cases}$$

In other words, $\boldsymbol{r}_{(v_1,v_2)}A = \mathbb{1}_{v_1} - \mathbb{1}_{v_2}$, where the symbol $\mathbb{1}_v$ means the column vector in $\mathbb{R}^{\mathcal{V}}$ with the entry of position v equal to 1 and all the other entries equal to 0. Observe moreover that, if $h = (v_1, v_2, \cdots v_n, v_{n+1})$, then

$$\boldsymbol{r}_h A = \sum_{i=1}^{n} \boldsymbol{r}_{(v_i,v_{i+1})}A = \sum_{i=1}^{n} \mathbb{1}_{v_i} - \mathbb{1}_{v_{i+1}} = \mathbb{1}_{v_1} - \mathbb{1}_{v_{n+1}}.$$

The latter equality proves that $\boldsymbol{r}_h A = 0$, if h is a closed path.

Proof [Proof of Proposition 2.1]

(a) The fact that $\Gamma \subseteq \{\boldsymbol{r} \in \mathbb{Z}^{\mathcal{E}} \mid \boldsymbol{r}A = 0\}$ follows from the previous arguments.

(b) Fix a spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G}$ and let $e_1, \ldots, e_{M-N+1}$ be the edges in $\mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$. Select the family cycles $h_1, \ldots, h_{M-N+1}$ by taking h_i to be the only cycle in $\mathcal{G}$ with edges in $\mathcal{E}_{\mathcal{T}} \cup \{e_i\}$ and such that $\boldsymbol{r}_{h_i}(e_i) = 1$. We now prove that the $\boldsymbol{r}_{h_i}$'s generate $\{\boldsymbol{r} \in \mathbb{Z}^{\mathcal{E}} \mid \boldsymbol{r}A = 0\}$. Let $\boldsymbol{r} \in \mathbb{Z}^{\mathcal{E}}$ such that $\boldsymbol{r}A = 0$ and let $\tilde{\boldsymbol{r}} := \boldsymbol{r} - \sum_{i=1}^{M-N+1} \boldsymbol{r}(e_i)\boldsymbol{r}_{h_i}$. Notice that $\tilde{\boldsymbol{r}}A = 0$ and that $\mathrm{supp}(\tilde{\boldsymbol{r}}) \subseteq \mathcal{E}_{\mathcal{T}}$. We show now that these facts imply $\tilde{\boldsymbol{r}} = 0$. Indeed, if $\mathrm{supp}(\tilde{\boldsymbol{r}}) \subseteq \mathcal{E}_{\mathcal{T}}$

and $\mathrm{supp}(\tilde{\boldsymbol{r}}) \neq \emptyset$, then $\mathrm{supp}(\tilde{\boldsymbol{r}})$ would include at least one leaf, namely a node v^* such that there exists only one edge $e^* \in \mathrm{supp}(\tilde{\boldsymbol{r}})$ containing v^*. In this case

$$0 = (\tilde{\boldsymbol{r}}A)_{v^*} = \sum_e \tilde{\boldsymbol{r}}(e)A_{ev^*} = \boldsymbol{r}(e^*)A_{e^*v^*},$$

which yields $\tilde{\boldsymbol{r}}(e^*) = 0$, a contradiction.

(c) It remains to prove the $\mathbb{Z}$-independence of the row vectors $\boldsymbol{r}_{h_1}, \dots \boldsymbol{r}_{h_{M_N+1}}$. Assume that $\alpha_i \in \mathbb{Z}$ are such that $\sum_i \alpha_i \boldsymbol{r}_{h_i} = 0$. Then, for any $\ell = 1, \dots, M - N + 1$, we have that

$$0 = \sum_i \alpha_i \boldsymbol{r}_{h_i}(e_\ell) = \alpha_\ell \boldsymbol{r}_{h_\ell}(e_\ell) = \alpha_\ell,$$

which shows that $\alpha_i = 0$.

Proof [Proof of Proposition 2.2] 1) First observe that, if we have two matrices $R, R' \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times\mathcal{E}}$ formed by two different $\mathbb{Z}$-bases of Γ, then $\ker R = \ker R'$ easily follows from the algebra of matrices over $\mathbb{Z}$ (see [19]). It remains to prove the assertion for a particular choice of the $\mathbb{Z}$-basis of Γ. Consider the $\mathbb{Z}$-basis $\boldsymbol{r}_{h_1}, \dots \boldsymbol{r}_{h_{M-N+1}}$ of Γ, built from a spanning tree $\mathcal{T}$ of $\mathcal{G}$ as described in point b) of the proof of Proposition 2.1. The fact that $\mathrm{Im}A \subseteq \ker R$ follows from Proposition 2.1. At this point, the only thing to be shown is that $\ker R \subseteq \mathrm{Im}A$. Let $\boldsymbol{K} \in \ker R$. Since rows in R form a basis of Γ, we have that $\boldsymbol{r}_h \boldsymbol{K} = 0$ for every closed path h. Let us fix a node $v_0 \in \mathcal{V}$. Then for each node $v \in \mathcal{V}$, let γ_v be the path in the tree $\mathcal{T}$ connecting v_0 to v. Define now the column vector $\boldsymbol{h} \in \mathbb{Z}^\mathcal{V}$ with components

$$h_v := \boldsymbol{r}_{\gamma_v}\boldsymbol{K} = \sum_{e\in\mathcal{E}} \boldsymbol{r}_{\gamma_v}(e)\boldsymbol{K}_e \ ,$$

where $\boldsymbol{r}_{\gamma_v}$ is the row vector in $\mathbb{Z}^\mathcal{E}$ associated with the path γ_v. We now show that $\boldsymbol{K}_e = h_{t(e)} - h_{s(e)}$, for each $e \in \mathcal{E}$. A straightforward consequence is that $\boldsymbol{K} = A\boldsymbol{h}$, i.e. $\boldsymbol{K} \in \mathrm{Im}A$. Consider the closed path h defined merging the paths $\gamma_{s(e)}$, $(s(e), t(e))$ and $-\gamma_{t(e)}$, where we recall that $-\gamma_{t(e)}$ denotes the path obtained by reversing $\gamma_{t(e)}$. Observe that $\boldsymbol{r}_h = \boldsymbol{r}_{\gamma_{s(e)}} + \boldsymbol{r}_{(s(e),t(e))} - \boldsymbol{r}_{\gamma_{t(e)}}$. It follows that

$$\begin{aligned} 0 = \boldsymbol{r}_h\boldsymbol{K} &= \boldsymbol{r}_{\gamma_{s(e)}}\boldsymbol{K} + \boldsymbol{r}_{(s(e),t(e))}\boldsymbol{K} - \boldsymbol{r}_{\gamma_{t(e)}}\boldsymbol{K} \\ &= h_{s(e)} + K_e - h_{t(e)} \end{aligned}$$

whence the thesis holds.

b) This follows from the fact that $\mathbb{Z}^\mathcal{E}/\Gamma$ is a torsion free module, and consequently it is free, being $\mathbb{Z}$ a principal ideal domain (see [20]).

Proof [Proof of Lemma 6.1] The fact that the rows in $\mathbb{Z}^\mathcal{E}$ obtained from a set of fundamental cycles form a basis of Γ follows from the the part (b) of the proof of Proposition 2.1. The fact the rows in $\mathbb{Z}^\mathcal{E}$ obtained from the minimal cycles form a basis of Γ is consequence of the following arguments. We order the edges putting first the edges in $\mathcal{T}$ and the edges $e_1, \dots, e_{M-N+1}$. Let R_F be the matrix having rows $\boldsymbol{r}_{h_1^F}, \dots, \boldsymbol{r}_{h_{M-N+1}^F}$, and R_M be the matrix having rows $\boldsymbol{r}_{h_1^M}, \dots, \boldsymbol{r}_{h_{M-N+1}^M}$, where h_i^F and h_i^M are the fundamental and the minimal cycles, respectively. Then, from the proposed construction, R_F and R_M can be partitioned as

$$R_F = [R'_F \ \ I], \qquad R_M = [R'_M \ \ T],$$

where I is the identity matrix of dimension $M - N + 1$, $T \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times\mathcal{E}}$ is a square lower triangular matrix with 1 on the diagonal and $R'_F, R'_M \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times\mathcal{E}_\mathcal{T}}$. Observe that $T^{-1} \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})}$ and, since the rows of R_F generate Γ while the rows of R_F belong to Γ, then there exists a matrix $Z \in \mathbb{Z}^{(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})\times(\mathcal{E}\setminus\mathcal{E}_\mathcal{T})}$ such that $R_M = ZR_F$. As a consequence, $Z = T$ and $R'_M = ZR'_F = TR'_F$. Finally, we can argue that $R_F = T^{-1}R_M$, which shows that the rows of R_M generate the same submodule generated by the rows of R_F which is Γ.