Autonomous Calibration Algorithms for Networks of Cameras

Domenica Borra, Enrico Lovisari, Ruggero Carli, Fabio Fagnani and Sandro Zampieri

***Abstract*— We deal with the important applicative problem of distributed cameras calibration. We model a network of N cameras as an undirected graph in which communicating cameras can measure their relative orientation in a noisy way. These measures can be used in order to minimize a suitable cost function. The shape of this cost function depends on a vector of integers $\boldsymbol{K}$. We propose two algorithms which in a distributive way estimate such $\boldsymbol{K}$, comparing advantages and disadvantages of both. Simulations are run on a grid network to prove effectiveness of the algorithms.**

I. Introduction

In the last fifteen years many efforts have been spent by the scientific community to distribute tasks over a network of communicating and interacting agents, in order to avoid the major problems of centralized strategies, such as reliability of multi-hops, and reliability of the agents themselves, both physically (they could fail) and under a security point of view (they could be malicious and intentionally damage the network). On the contrary, distributed strategies are based on local exchange of information among hierarchically equal agents. In a network of cameras deployed in a plane one of the most important problems is the calibration. Namely, each camera has to know how it is oriented, at any instant, with respect to a certain common reference frame. The importance of this is clear: assume that an external agent, which has to be tracked, is exiting from the range of the i-th camera and entering in that of the j-th one. In this case camera i has to communicate to camera j to move and follow the agent before camera i looses it. Clearly, both cameras must share the same reference frame. Usually, this is set off-line by a human operator, or by a centralized unit. Instead, the algorithms we propose aim to complete autonomy, and do not require any central control to carry on computations. This allows improved accuracy and possibility of periodical autonomous re–calibration.

The model for the network is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V}$ is the set of cameras, and $\mathcal{E}$ the set of edges, the couples of communicating cameras. The information which is used to calibrate the network, inspired by the works by Barooah and Hespanha [1], [2], [3] on localization, is the relative orientation among cameras, which can be computed if their field of view overlaps. Cameras calibration through relative orientations can be transformed into a consensus like problem over the manifold $\mathcal{S}_1$: this problem has already attracted much attention by research community. In [4], [5] a consensus algorithm on $\mathcal{S}_1$ based on the gradient flow of a potential defined using the chordal distance is proposed. In [6] a similar approach based on the geodesic distance is proposed to study the more general calibration problem on $SE(3)$. The issue with both these approaches is that the defined potentials are characterized by several nontrivial local minima in which, apart from particular initial conditions, it is easy to fall. On the other hand, in [7] the noisy measurements of relative orientations are *a priori* constrained to sum to zero on cycles. Based on this construction, a least-square estimation algorithm, which is proved to be optimal on a ring graph, is presented. We choose to concentrate on the simple case of calibration in $SO(2) \sim \mathcal{S}_1$, and we use the geodesic distance. Our main idea is to break the estimation problem into two parts: first we estimate a sort of combinatorial object which is a vector in $\mathbb{Z}^{\mathcal{E}}$ and takes care of the fact that noise around cycles in general does not sum up to 0. Once this is done we estimate by solving a quadratic optimization problem like in the localization problem. Our method is consistent in the sense that if there is no error, the solution coincides with the true one. We propose two different algorithms: one based on spanning trees, another one based on minimal cycles. This procedure has been used in [8] to improve the estimates for localization in $\mathbb{R}^N$.

The paper is organized as follows: in Section II we present several definitions and algebraic properties of graphs, which will be extensively used throughout the paper. In Section III we formulate the problem we want to deal with. Section IV presents our two algorithms and gives some first properties. In Section V we compare the two algorithms giving closed formulae of the results, and we estimate their performance. Examples and numerical simulations are provided in Section VI and VII.

II. Graph theoretical tools

An undirected graph is a couple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is a subset of unordered pairs of elements of $\mathcal{V}$ called edges. We put $N = |\mathcal{V}|$ and $M = |\mathcal{E}|$. A spanning tree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is simply a tree subgraph of $\mathcal{G}$ which has the same set of nodes, and consequently $|\mathcal{E}_{\mathcal{T}}| = N - 1$.

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 $e \in \mathcal{E}$. In these notations $s(e)$ $(t(e))$ is called the starting (terminating) node of the edge e. Assume from now on that

R. Carli, E. Lovisari and S. Zampieri are with the Department of Information Engineering, University of Padova, via Gradenigo 6/B Padova, Italy, {lovisari, carli, zampi}@dei.unipd.it

D. Borra and F. Fagnani are with Department of Mathematics, Polytechnic of Turin {domenica.borra, fabio.fagnani}@polito.it

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

we have fixed an orientation (s,t) on $\mathcal{G}$. The *incidence matrix* $B \in \{\pm 1,\, 0\}^{\mathcal{E}\times\mathcal{V}}$ of $\mathcal{G}$ is defined by putting $B_{e\, s(e)} = 1$, $B_{e\, t(e)} = -1$, and $B_{ev} = 0$ if $v \neq s(e), t(e)$. An *oriented cycle* h of length n (with $n > 2$) is an ordered sequence of nodes $h = (v_1 v_2 \cdots v_n)$ (up to cyclic permutations) such that $\{v_i, v_{i+1}\} \in \mathcal{E}$ for all $i = 1, \ldots, n$ (interpreting $n+1 = 1$). The set of cycles is denoted by $\mathcal{H}$. Given a cycle h, we denote by $-h$ the same cycle with reversed orientation. Given $h = (v_1\, v_2 \cdots v_n) \in \mathcal{H}$, we associate a row vector $\boldsymbol{r}_h \in \{\pm 1,\, 0\}^{\mathcal{E}}$ as follows. Put $e_i = \{v_i, v_{i+1}\}$ and define $r_h(e_i) := B_{e_i\, v_i}$ for every $i = 1, \ldots, n$. While, $r_h(e) = 0$ for any other edge not in the cycle. It is immediate to see that $\boldsymbol{r}_{-h} = -\boldsymbol{r}_h$. Given two oriented cycles h, h' such that $r_h(e) r_{h'}(e) = 0$ for all edges except one e^* for which $r_h(e^*) r_{h'}(e^*) = -1$, we can consider the *sum cycle* denoted by $h + h'$ determined by setting $\boldsymbol{r}_{h+h'} := \boldsymbol{r}_h + \boldsymbol{r}_{h'}$. The sum cycle $h + h'$ is simply obtained by joining the edges of the two cycles and dropping e^*. An oriented cycle is said to be *minimal* if it can not be written as the sum of two other cycles. Clearly, if h is minimal, also $-h$ is minimal. We denote by $\mathcal{H}_0$ any subset of minimal cycles with the property that $h \in \mathcal{H}_0$ iff $-h \notin \mathcal{H}_0$. A standard result says that $|\mathcal{H}_0| = M - N + 1$.

Given a spanning tree $\mathcal{T} = (\mathcal{V},\, \mathcal{E}_\mathcal{T})$ of the graph $\mathcal{G}$, a cycle h is a *($\mathcal{T}$-) fundamental cycle* if $h = (v_1 v_2 \cdots v_n)$ where $\{v_i, v_{i+1}\} \in \mathcal{E}_\mathcal{T}$ for every $i = 1, \ldots, n-1$. Notice that $\{v_1, v_n\}$ is the only edge in the cycle which is not in $\mathcal{E}_\mathcal{T}$. Clearly, for each edge $e \in \mathcal{E} \setminus \mathcal{E}_\mathcal{T}$, there are two fundamental cycles sharing the edge e, h_e and $-h_e$, and we assume that h_e has been chosen so that $r_{h_e}(e) = 1$. $\mathcal{H}_\mathcal{T}$ will denote the set of h_e's, and evidently also $|\mathcal{H}_\mathcal{T}| = M - N + 1$. This is not a coincidence as the following result will show.

Let $R \in \mathbb{Z}^{\mathcal{H}\times\mathcal{E}}$ be an integer matrix whose rows are all the vectors $\boldsymbol{r}_h$ as h varies among all possible cycles, and let R_0 and $R_\mathcal{T}$ be the sub-matrices of R consisting of those rows in $\mathcal{H}_0$ and in $\mathcal{H}_\mathcal{T}$, respectively. Interpreting R, R_0, and $R_\mathcal{T}$ as group homomorphisms on $\mathbb{Z}^{\mathcal{E}}$ and the matrix B as a group homomorphism $\mathbb{Z}^{\mathcal{V}} \to \mathbb{Z}^{\mathcal{E}}$, we have the following.

Lemma 2.1: It holds $\ker R_0 = \ker R_\mathcal{T} = \ker R = \operatorname{Im} B$.

While there is an obvious bijection between $\mathcal{H}_\mathcal{T}$ and $\mathcal{E} \setminus \mathcal{E}_T$, the construction of a bijection between $\mathcal{H}_0$ and $\mathcal{E} \setminus \mathcal{E}_T$ needs a bit of extra work, as stated in the next proposition.

Proposition 2.1: Given the graph $\mathcal{G}$ with an orientation and a spanning tree $\mathcal{T}$, there is an ordering $e_1, \ldots, e_{M-N+1}$ of the edges in $\mathcal{E} \setminus \mathcal{E}_\mathcal{T}$ such that, for every i, e_i forms a minimal cycle in $\mathcal{G}$ with all remaining edges lying in $\mathcal{E}_\mathcal{T} \cup \{e_1, \ldots e_{i-1}\}$.

III. Problem formulation

We model the network of cameras as a connected undirected graph $\mathcal{G} = (\mathcal{V},\, \mathcal{E})$ equipped with an orientation (s,t). Each node is equipped with a camera. Fix an external reference frame and let $\bar{\theta}_v \in \mathbb{R}$ to be the orientation of the camera of agent v w.r.t. such reference frame.

We fix a reference node v^*, called *anchor node*, that exactly knows its orientation, but the chosen external reference frame does not depend on the anchor node. In the following the orientation of the anchor node w.r.t. the external reference frame will be set to 0 without losing generality since the actual orientations have to be computed up to global translations.

Nodes can obtain relative noisy measurements along the available edges in the way we are going to describe. First, given any real number x, we set

$$(x)_{2\pi} = x - 2\pi q_{2\pi}(x)$$

where $q_{2\pi}(x) = \lfloor \frac{x+\pi}{2\pi} \rfloor \in \mathbb{Z}$ is such that $(x)_{2\pi} \in [-\pi,\, \pi)$. The floor function is chosen in order to guarantee that $q_{2\pi}(x) = x$, if $x \in [-\pi, \pi)$. It would be equivalent to project $\mathbb{R}$ onto $(-\pi, \pi]$ just imposing $q_{2\pi}(x) = \lceil \frac{x-\pi}{2\pi} \rceil$.
If $e = \{v,\, w\} \in \mathcal{E}$, we assume that the fields of view of v and w overlap, so that by means of known algorithms, the following noisy measurement can be computed

$$\eta_e = (\bar{\theta}_{s(e)} - \bar{\theta}_{t(e)} - \varepsilon_e)_{2\pi} = \bar{\theta}_{s(e)} - \bar{\theta}_{t(e)} - \varepsilon_e - 2\pi \bar{K}_e \quad (1)$$

where $\bar{K}_e \in \mathbb{Z}$. The assumption that the measurement is an angle $\eta_e \in [-\pi,\, \pi)$ does not entail any loss of generality.

The incidence matrix B of the graph $\mathcal{G}$ allows to rewrite this relation in vector form as

$$\boldsymbol{\eta} = B\bar{\boldsymbol{\theta}} - \boldsymbol{\varepsilon} - 2\pi \bar{\boldsymbol{K}}, \quad (2)$$

where $\bar{\boldsymbol{K}} \in \mathbb{Z}^{\mathcal{E}}$.

We assume noises along different edges to be independent and to be equally distributed as $\varepsilon_e \sim \mathcal{U}[-\bar{\varepsilon},\, \bar{\varepsilon}]$. This last choice is made for the sake of simplicity. Notice that it is a realistic hypothesis, since each camera has a well defined resolution, and by construction it has a failure tolerance thus it cannot mislead more than a certain number of pixels.

The *calibration* problem consists in giving an estimate $\hat{\theta}_v \in \mathbb{R}$ of the correspondent $\bar{\theta}_v$ for each node $v \in \mathcal{V}$. These estimates will be constructed using the available relative measurements and exchanging information along the graph. Notice that $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\theta}} + 2\pi \boldsymbol{l}$ for some $\boldsymbol{l} \in \mathbb{Z}^{\mathcal{V}}$, will give rise to the same measurements $\boldsymbol{\eta}$, so even in the case when no noise is present, $\bar{\boldsymbol{\theta}}$ can only be determined up to these 2π integer translations. The vectors of the form $\bar{\boldsymbol{\theta}} + 2\pi \boldsymbol{l}$ will be called *representatives* of $\bar{\boldsymbol{\theta}}$.

A. Cost function

To address the calibration problem we consider the cost

$$V(\boldsymbol{\theta}) = \sum_{e\in\mathcal{E}} (\theta_{s(e)} - \theta_{t(e)} - \eta_e)^2_{2\pi} = \|(B\boldsymbol{\theta} - \boldsymbol{\eta})_{2\pi}\|_2^2. \quad (3)$$

The cost $V(\boldsymbol{\theta})$ attains the value zero for any representative of $\bar{\boldsymbol{\theta}}$ in case of noiseless measurements. However, even in this ideal case, it has multiple local minima. In order to let the algorithm avoid these points, we define the following regions

$$R_{\boldsymbol{K}}(\boldsymbol{\eta}) := \{\boldsymbol{\theta} \in \mathbb{R}^{\mathcal{V}} :\, |B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi \boldsymbol{K}| \le \pi \mathbb{1}\}^1, \quad (4)$$

where $\boldsymbol{K} \in \mathbb{Z}^{\mathcal{E}}$. These regions are convex and form a partition of $\mathbb{R}^{\mathcal{V}}$. However, some of them can be empty, since they are defined by M constraints on N variables and in

[1] $|\boldsymbol{v}| \le \boldsymbol{p}$, where both $\boldsymbol{v}, \boldsymbol{p} \in \mathbb{R}^n$, means $-p_i \le v_i < p_i$ for all $i = 1, \ldots, n$.

general $M > N$. It is trivial to see that if $\boldsymbol{\theta} \in R_{\boldsymbol{K}}(\boldsymbol{\eta})$, and only for these points, then $V(\boldsymbol{\theta}) = \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\boldsymbol{K}\|_2^2$, which is purely quadratic and convex in $R_{\boldsymbol{K}}(\boldsymbol{\eta})$, where thus there can be at most one local minimum of $V(\boldsymbol{\theta})$. The main idea is the following: first, obtain a reasonable estimate $\hat{\boldsymbol{K}}$ of $\bar{\boldsymbol{K}}$. Then minimize the reshaped cost

$$V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) := \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|_2^2 \qquad (5)$$

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

Notice that by the way $\boldsymbol{\eta}$ has been defined,

$$|B\bar{\boldsymbol{\theta}} - \boldsymbol{\eta} - 2\pi\bar{\boldsymbol{K}}| = |\boldsymbol{\varepsilon}| \leq \pi\mathbb{1} \qquad (6)$$

so that $\bar{\boldsymbol{\theta}} \in R_{\bar{\boldsymbol{K}}}(\boldsymbol{\eta})$. If we have obtained an estimation $\hat{\boldsymbol{K}}$ such that $\hat{\boldsymbol{K}} = \bar{\boldsymbol{K}} + B\boldsymbol{l}$ for some $\boldsymbol{l} \in \mathbb{Z}^{\mathcal{V}}$, then, clearly, $\bar{\boldsymbol{\theta}} + 2\pi\boldsymbol{l} \in R_{\hat{\boldsymbol{K}}}(\boldsymbol{\eta})$. A simple continuity argument then shows that, when the threshold $\bar{\varepsilon}$ tends to 0, the estimation $\hat{\boldsymbol{\theta}}$ converges to $\bar{\boldsymbol{\theta}} + 2\pi\boldsymbol{l}$. In other terms, with such a $\hat{\boldsymbol{K}}$, we have a guarantee that our solution is close to a feasible one, namely to a representative of the true $\bar{\boldsymbol{\theta}}$. In Section V, we analyze in more detail the performance of the algorithms.

IV. Description of the algorithms

Both algorithms we now describe are based on the idea of breaking the estimation problem into two steps: first give an estimate $\hat{\boldsymbol{K}}$ of $\bar{\boldsymbol{K}}$ and then minimize the quadratic function defined in Eq. 5. The two algorithms only differ in the first step, as the first uses the fundamental cycles of a given spanning tree, the second instead the minimal cycles.

Regarding the first step, the main idea underlying both algorithms exploits the fact 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. Then, from Eq. 2, using the fact that $\boldsymbol{r}_h B = 0$, we obtain that $\boldsymbol{r}_h\bar{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta}) - q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\varepsilon})$. Therefore, if it happens that the algebraic sum of the noise along the cycle h is below π in modulus, i.e. $|\boldsymbol{r}_h\boldsymbol{\varepsilon}| < \pi$, we obtain that

$$\boldsymbol{r}_h\bar{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta}). \qquad (7)$$

In other terms, in this case $\boldsymbol{r}_h\bar{\boldsymbol{K}}$ can be exactly computed on the basis of the measurements $\boldsymbol{\eta}$ along the cycle h. This would suggest to define $\hat{\boldsymbol{K}}$ in such a way that $\boldsymbol{r}_h\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta})$ for any cycle h, but this in general will not be possible since the various $\boldsymbol{r}_h$'s are linearly dependent. What must be done is to restrict the cycles to a subset for which the corresponding $\boldsymbol{r}_h$'s form a $\mathbb{Z}$-basis for the $\mathbb{Z}$-module generated by the rows of the matrix R. The choice of this basis is the essential difference among the two algorithms.

A. The Tree-algorithm

Fix a spanning graph $\mathcal{T}$ and consider the corresponding fundamental cycles. Let us impose that $\boldsymbol{r}_h\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta})$ for any fundamental cycle h. From Lemma 2.1 we know that this determines $\hat{\boldsymbol{K}}$ up to elements in the image of B as required. A concrete solution can be easily found by imposing $\hat{\boldsymbol{K}}_e = 0$ for every $e \in \mathcal{E}_{\mathcal{T}}$. Then, we easily obtain that, for any $e \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$,

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

where, we recall, h_e is the fundamental cycle associated with e such that $r_{h_e}(e) = 1$. This algorithm is very simple and easily implementable. As we will point out, however, its performances for large graphs are rather poor.

A distributed way to compute $\hat{\boldsymbol{K}}$ is proposed below. Fix an anchor node, denoted by v^*, which will serve as a root in the tree $\mathcal{T}$. First of all, we propagate the measurements along the tree starting from the root, namely, given a node v and called $f(v)$ its father, we set

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

As a side effect, we also obtain a first estimate $\hat{\boldsymbol{\theta}}_{FE}$ of $\bar{\boldsymbol{\theta}}$.

Now we construct $\hat{\boldsymbol{K}}$. For each edge $e = \{v, w\} \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$, the nodes v and w exchange their first estimates $\hat{\theta}_{FE,\,v}$, $\hat{\theta}_{FE,\,w}$ and compute $\hat{K}_e$ as

$$\hat{K}_e = \left\lfloor \frac{\hat{\theta}_{FE,s(e)} - \hat{\theta}_{FE,t(e)} - \eta_e}{2\pi} \right\rceil.$$

This is the only choice of $\hat{\boldsymbol{K}}$ for which $\hat{\boldsymbol{\theta}}_{FE} \in R_{\hat{\boldsymbol{K}}}(\boldsymbol{\eta})$.

Finally we obtain a final estimate of $\bar{\boldsymbol{\theta}}$, call it $\hat{\boldsymbol{\theta}}$, by minimizing the quadratic cost function in Eq. 5. This problem can be for example easily solved using a distributed Jacobi algorithm as shown in [2], [3].

Algorithm 1 Tree-Algorithm

```
(Input variables)
$\bar{\theta}_{v^*}$, value of the anchor
$\eta_e, e = 1, \dots, M$
$\mathcal{T}$ spanning tree

(Step A: first estimate $\hat{\boldsymbol{\theta}}_{FE}$)
$\hat{\theta}_{FE,v^*} = \bar{\theta}_{v^*}$;
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
    $\hat{K}_e = \left\lfloor \frac{\hat{\theta}_{FE,s(e)} - \hat{\theta}_{FE,t(e)} - \eta_e}{2\pi} \right\rceil$

(Step C: second estimate $\hat{\boldsymbol{\theta}}$)
Initial condition: $\hat{\boldsymbol{\theta}}(0) = \hat{\boldsymbol{\theta}}_{FE}$
compute $\hat{\boldsymbol{\theta}} = \text{argmin}\left\|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\right\|_2^2$
```

B. Minimal cycles-algorithm

The second algorithm is based on the minimal cycles of the graph $\mathcal{G}$. Let us impose that $\boldsymbol{r}_h\hat{\boldsymbol{K}} = -q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta})$ for any minimal cycle h. From Lemma 2.1 we know that this determines $\hat{\boldsymbol{K}}$ up to elements in the image of B as required. A concrete method for constructing such $\hat{\boldsymbol{K}}$ once the values $-q_{2\pi}(\boldsymbol{r}_h\boldsymbol{\eta})$ have been computed for all minimal cycles, can be easily based on Proposition 2.1. We start, as in the previous algorithm, from a spanning subtree $\mathcal{T} = (\mathcal{V}, \mathcal{E}_\mathcal{T})$ of $\mathcal{G}$ and we assign $\hat{K}_e = 0$ on the edges in $\mathcal{E}_\mathcal{T}$. We then consider the remaining edges $e_1, e_2, \dots, e_{M-N+1}$ ordered as in Proposition 2.1 and we define $\hat{\boldsymbol{K}}$ iteratively as

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

where, h_i is the minimal cycle that e_i forms with edges in $\mathcal{E}_\mathcal{T} \cup \{e_1, \dots e_{i-1}\}$ oriented so that $r_{h_i}(e_i) = 1$.

The last step of the *Minimal cycle-algorithm* is the same as that of the *Tree-algorithm*. Since we have no particular initial condition, it can be set to $(0, \dots, 0)$ for simplicity.

Algorithm 2 Minimal cycles-algorithm

```
(Input variables)
$\eta_e, e = 1, \dots, M$
$\mathcal{T}$ spanning tree
$\mathcal{H}_0 = \{\boldsymbol{r}_{h_1}, \dots, \boldsymbol{r}_{h_{M-N+1}}\}$ minimal cycles set

(Step A: computation of $\boldsymbol{b} = -q_{2\pi}(R_0\boldsymbol{\eta})$)
for $h \in \mathcal{H}_0$ do $b_h = -\left\lfloor \frac{\sum_{e\in h}\eta_e + \pi}{2\pi} \right\rfloor$

(Step B: estimate $\hat{\boldsymbol{K}}$)
for $e \in \mathcal{E}_\mathcal{T}$ do $\hat{K}_e = 0$
for $h \in \mathcal{H}_0$ s.t. $\hat{K}_e$ is known for all $e \in h$ except for
    one $\bar{e}$ do $\hat{K}_{\bar{e}} = b_h - \sum_{e\in h, e\neq\bar{e}} \boldsymbol{r}_{h_e}(e)\hat{K}_e$

(Step C: second estimate $\hat{\boldsymbol{\theta}}$)
Initial condition: $\hat{\boldsymbol{\theta}}(0) = (0, \dots, 0)$
compute $\hat{\boldsymbol{\theta}} = \mathrm{argmin}\left\|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\right\|_2^2$
```

This second algorithm allows much better performances than the *Tree-algorithm*, but it requires a greater order of collaboration among nodes. In fact, we assume that, through some local collaboration among nodes, each minimal cycle corresponds to a “superagent”, able to sense all the measurements along the edges of its cycle. Clearly, this is far more than just locally exchanging information.

V. Analysis and comparison of the algorithms

Recall that the final estimate $\hat{\boldsymbol{\theta}}$ must minimize the cost $V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) = \|B\boldsymbol{\theta} - \boldsymbol{\eta} - 2\pi\hat{\boldsymbol{K}}\|_2^2$. Since B has a kernel (which is spanned by $\mathbb{1}$) of course $\hat{\boldsymbol{\theta}}$ is only determined up to multiples of $\mathbb{1}$[2]. This non uniqueness can be avoided as follows. As already done in Section IV, fix an anchor node $v^* \in \mathcal{V}$ that knows the true value of its orientation, and assume that it never changes its estimate. Then consider the vector $\boldsymbol{\xi} \in \mathbb{R}^\mathcal{V}$ defined by $\xi(v^*) = 1$ and $\xi(v) = 0$ for any $v \neq v^*$. Now define the Green matrix associated to $\mathcal{G}$ and v^* as the solution of the following equations

$$\begin{cases} GB^TB = I - \mathbb{1}\boldsymbol{\xi}^T \\ G\boldsymbol{\xi} = 0. \end{cases} \tag{8}$$

Writing down the stationary point equation for our quadratic problem and using the Green matrix, it is straightforward to show that the following result holds true.

Proposition 5.1: If $\hat{\boldsymbol{K}}$ is the estimate of $\bar{\boldsymbol{K}}$, the minimum of $V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta})$ is attained at

$$\hat{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}} + GB^T\boldsymbol{\varepsilon} - 2\pi GB^T(\bar{\boldsymbol{K}} - \hat{\boldsymbol{K}}). \tag{9}$$

The previous proposition basically says that the difference between the final estimate $\hat{\boldsymbol{\theta}}$ and the actual orientations $\bar{\boldsymbol{\theta}}$ is made of two terms. The first one, $GB^T\boldsymbol{\varepsilon}$, is unavoidable and only depends on the fact that the measurements are noisy. This term is the localization error in the works by Barooah and Hespanha on $\mathbb{R}^N$. The second term, $-2\pi GB^T(\bar{\boldsymbol{K}} - \hat{\boldsymbol{K}})$ depends on the estimation $\hat{\boldsymbol{K}}$, and it is due to the geometry of $\mathcal{S}_1$. If $\hat{\boldsymbol{K}} = \bar{\boldsymbol{K}} + B\boldsymbol{l}$, with $\boldsymbol{l} \in \mathbb{Z}^\mathcal{V}$, it is easy to see that

$$\hat{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}} + 2\pi(I - \mathbb{1}\boldsymbol{\xi}^T)\boldsymbol{l} + GB^T\boldsymbol{\varepsilon} = \tilde{\boldsymbol{\theta}} + GB^T\boldsymbol{\varepsilon}$$

where $\tilde{\boldsymbol{\theta}} = \bar{\boldsymbol{\theta}} + 2\pi(I - \mathbb{1}\boldsymbol{\xi}^T)\boldsymbol{l}$ is a representative of $\bar{\boldsymbol{\theta}}$.

Let L_0 be the maximum length of a minimal cycle and $L_\mathcal{T}$ the maximum length of a $\mathcal{T}$-fundamental cycle. It is clear that $L_0 \leq L_\mathcal{T}$, and in general $L_\mathcal{T}$ depends on the number of nodes of the graph, as the examples in Section VI will point out. We present now the main result of this section, which basically gives a noise threshold for both algorithms, under which it is guaranteed to have $\hat{\boldsymbol{K}} - \bar{\boldsymbol{K}} \in \mathrm{Im}_\mathbb{Z} B$.

Proposition 5.2: If

$$\text{Tree algorithm}: \ \bar{\varepsilon} < \frac{\pi}{L_\mathcal{T}}$$

$$\text{Cycle algorithm}: \ \bar{\varepsilon} < \frac{\pi}{L_0}$$

we have

$$\hat{\boldsymbol{K}} = \bar{\boldsymbol{K}} + B\boldsymbol{l}, \boldsymbol{l} \in \mathbb{Z}^\mathcal{V}.$$

Remark 5.1: A closed formula for $\hat{\boldsymbol{K}}$ can actually be derived for both algorithms, and it will be objective of our future research to deepen the performance analysis, centered on the error in the estimation of $\bar{\boldsymbol{K}}$.

If the assumptions of Proposition 5.2 hold, we have as a straightforward consequence $\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}} + GB^T\boldsymbol{\varepsilon}$, where $\tilde{\boldsymbol{\theta}}$ is a representative of $\bar{\boldsymbol{\theta}}$. In this case the error term is exactly the same appearing in the vector space case and can be analyzed exploiting the probabilistic assumptions made on the noise in Section III, namely that $\varepsilon_e \sim \mathcal{U}[-\bar{\varepsilon}, \bar{\varepsilon}]$, $\forall\, e \in \mathcal{E}$. In case this holds, from paper [1] we can obtain an estimate of the variance of the final estimate error in terms of the effective resistance of a suitable electrical network. Namely, consider an electrical network whose nodes are those of the graph $\mathcal{G}$ and for each edge of $\mathcal{G}$ we have a resistance of 1 Ohm. Denote by $\mathcal{R}_{uv}$ the effective resistance among any pair of nodes $u, v \in \mathcal{V} \times \mathcal{V}$. Then we have the following result

Proposition 5.3 ([1]): The estimate $\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}} + GB^T\varepsilon$ is unbiased, namely $\mathbb{E}\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}}$, and its v-th component has

[2] $\mathbb{1}$ represents a vector of suitable dimension whose entries are all equal to 1.

variance $\mathbb{E}[(\hat{\theta}_v - \tilde{\theta}_v)^2] = \mathcal{R}_{vv^*}$ where v^* is the anchor. As a consequence, the normalized scalar estimation variance is

$$\frac{1}{N}\mathrm{var}(\hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}) = \frac{1}{N}\sum_{v\in\mathcal{V}} \mathcal{R}_{vv^*} \tag{10}$$

which is the average effective resistance among the anchor and the other nodes of the network.

Remark 5.2: The previous Proposition gives mean and variance of the estimation error $\boldsymbol{\delta} = \hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}}$. However, this is not entirely correct, since what we are really interested in is $\boldsymbol{\delta}_{2\pi} = (\hat{\boldsymbol{\theta}} - \tilde{\boldsymbol{\theta}})_{2\pi}$, which has still zero mean, and variance less then that in Eq. (10). Nonetheless, if the noise is big and $\hat{\boldsymbol{\theta}}$ is not near $\tilde{\boldsymbol{\theta}}$, the probability to end up in a point near another representative of $\bar{\boldsymbol{\theta}}$ is intuitively very small, so we preferred to give only the results in Proposition 5.3.

VI. Examples

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.

First of all, we show that our algorithms can avoid the local minima in the original cost. Consider the simple ring graph with 3 agents in Fig. 1, and assume $\bar{\theta}_1 = \bar{\theta}_2 = \bar{\theta}_3 = 0$ for sake of simplicity (the problem becomes thus consensus on $\mathcal{S}_1$). Consider the ideal noiseless case, so that $\eta_{12} = \eta_{23} = \eta_{31} = 0$ and $\bar{K}_{12} = \bar{K}_{23} = \bar{K}_{31} = 0$, and assume $\hat{\boldsymbol{K}} = \bar{\boldsymbol{K}}$.

Consider the case in which we have as initial conditions $\hat{\theta}_1(0) = 0$, $\hat{\theta}_2(0) = \frac{2}{3}\pi$ and $\hat{\theta}_3(0) = \frac{4}{3}\pi$. If we use directly the original cost in Eq. (3), it is easy to see that the initial condition is a local maximum (in case of 5 or more agents the analogous configuration is a local minimum), so any gradient-descent like algorithm gets stuck. However, if we reshape the cost using our guess $\hat{\boldsymbol{K}}$, we have to minimize $V_{\hat{\boldsymbol{K}}}(\boldsymbol{\theta}) = \|B\boldsymbol{\theta} - 2\pi\hat{\boldsymbol{K}}\|_2^2 = \|B\boldsymbol{\theta}\|_2^2$, and we converge to the actual orientations fixing the anchor at $\hat{\theta}_1 = \bar{\theta}_1 = 0$.

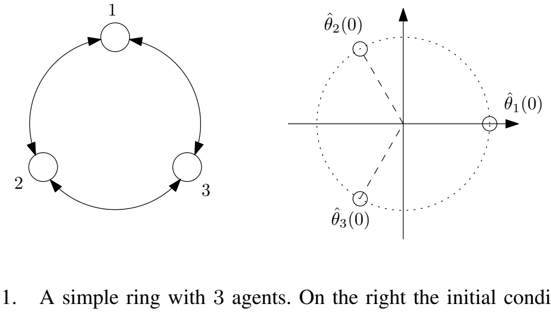

Fig. 1. A simple ring with 3 agents. On the right the initial conditions.

In order to draw now a comparison among the two algorithms, consider the graphs shown in Fig.2. 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 line. They are rooted on the anchor on the most left-top node. The set of minimal cycles $\mathcal{H}_0$ is simply the set of squares which form the graph.

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 $L_\mathcal{T} = N$. On the contrary, the minimal cycles are of length $L_0 = 4$. As an immediate consequence, the *Minimal cycles-algorithm* has much better performances since the upper bound $\bar{\varepsilon} < \frac{\pi}{4}$ is independent on the number of nodes. On the contrary, 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.

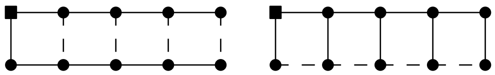

Fig. 2. Two examples of spanning trees for a line–like graph. The proposed algorithms work in a similar manner for the one on the right, while the *Minimal cycles-algorithm* is far more effective for the one on the left. The square node is the anchor.

If we consider instead the spanning tree on the right, we can see that $L_\mathcal{T} = 4$ as well, since the spanning tree is chosen in a much better way. In this case, the two algorithms have comparable and good performance. It is not always true, however, that the *Minimal cycles-algorithm* has good performances. For example, if we consider the ring graph in Fig. 3 we can easily see that there is only one minimal cycle. Here the two proposed algorithms basically coincide, comparing performances. In such a case, the *Tree-algorithm* is better, since it is easier to implement and completely distributed, and it requires less information on the topology of the network, as well as less communications.

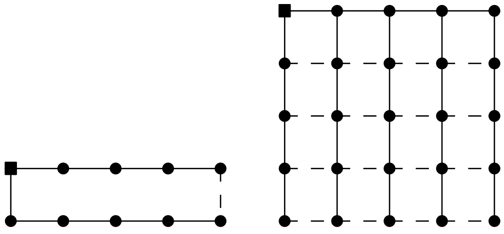

Fig. 3. On the left a ring graph, for which the two algorithms have the same performance. On the right, a grid graph. The square node is the anchor.

As a last example, consider the $2D$ grid on the right in Fig. 3. The comb-shaped spanning tree is the one in thick line. As before, here $L_\mathcal{T} \sim \sqrt{N}$ adding one of the edges on the bottom, while $L_0 = 4$. However, it can be shown that for the grid $L_\mathcal{T} \sim \sqrt{N}$ is actually the best one can do. So in this case the *Minimal cycles-algorithm* has always better performances than the *Tree-algorithm*. Notice that the choice of the spanning tree is fundamental to draw a comparison between the algorithms. Even if the tree is such that $L_\mathcal{T}$ is minimum, the choice of the better algorithm depends on the topology of the graph, since it could hold $L_\mathcal{T} > L_0$, as highlighted in the previous example.

VII. Numerical results

In this Section we provide a numerical comparison between the two approaches we propose in this paper. Specifically, in the experiments we simulate the *Tree-algorithm* and the *Minimal cycles-algorithm* on square grid graphs of size

$N = n^2$ for n ranging from 3 up to 19. An example of square grid graph is depicted in Fig. 4 (left panel), where $n = 5$ and, in turn, $N = 25$.

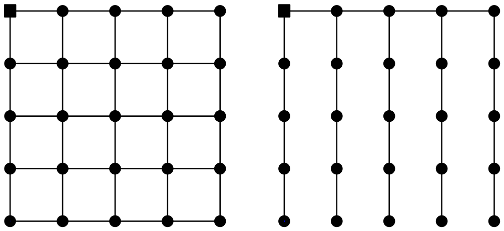

Fig. 4. On the left a square grid graph. On the right the correspondent spanning tree used in simulations. The square node is the anchor.

In all our simulations we set $\bar{\theta}_1 = 0$, while, for $i \in \{2, \dots, N\}$, $\bar{\theta}_i$ is randomly sampled from a uniform distribution on $[-\pi, \pi]$. The values of the noises ε_e, $e \in \mathcal{E}$, are also randomly sampled, in this case from a uniform distribution on $[-\bar{\varepsilon}, \bar{\varepsilon}]$ where $\bar{\varepsilon} = \frac{\pi}{8}$.

The simulation results obtained are reported in Fig. 5 and in Fig. 6. For each n the values we plot are averaged over 200 trials (a different $\bar{\boldsymbol{\theta}}$ and a different set of noises are generated for each trial). The kind of spanning tree we use to run our algorithms is illustrated in Figure 6. Here we have $n = 5$, but for different values of n the spanning tree used is similarly built. Observe that, for the square grid graphs and the corresponding spanning tree we consider, we have $L_0 = 4$ independently from n, and $L_{\mathcal{T}} = 2n + 2$.

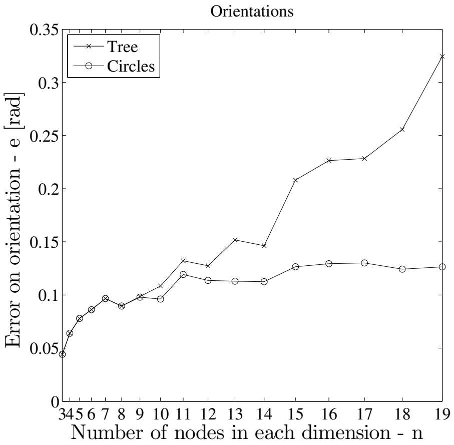

Fig. 5. Average error on the orientations (modulo 2π).

In Fig. 5 we show the value of the estimate error $e = \frac{1}{N}\|(\bar{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}})_{2\pi}\|^2$ for both the *Tree-algorithm* and the *Minimal cycles-algorithm*; in Fig. 6 we plot the value $e_K = \frac{1}{M}\|(\bar{\boldsymbol{K}} - \hat{\boldsymbol{K}})_{\mathrm{Im}_{\mathbb{Z}}B}\|^2$, where if $\boldsymbol{X} \in \mathbb{Z}^M$, $(\boldsymbol{X})_{\mathrm{Im}_{\mathbb{Z}}B}$ represents the projection out of the $\mathbb{Z}$-submodule spanned by the columns of B. This quantity is taken as a measure of the distance between the actual value $\bar{\boldsymbol{K}}$ and the estimates obtained through the algorithms. Notice that, since

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

it follows from Proposition 5.2 that the *Minimal cycles-algorithm* always correctly estimates $\bar{\boldsymbol{K}}$, thus $e_K = 0$. On the contrary, in the case of the *Tree-algorithm* e_K is increasing with the dimension of the graph. This is not surprising since $L_{\mathcal{T}}$ grows linearly with $\sqrt{N}$, and so intuitively the probability of the estimate to be bad becomes larger and larger. As expected, one can check from Fig. 5 that the *Minimal cycles-algorithm* outperforms the *Tree-algorithm*.

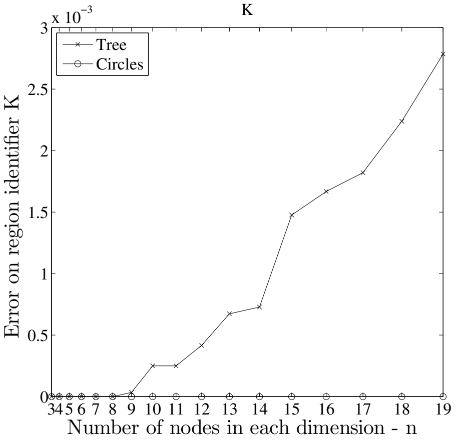

Fig. 6. Average error on $\hat{\boldsymbol{K}}$.

VIII. Conclusions

This paper deals with the problem of distributively calibrate a network of cameras deployed in a plane. Two algorithms are proposed to reshape a suitable cost function which is used by each camera to obtain an estimate of its actual orientation w.r.t. an external reference frame. Future research will focus on deeper characterization of such estimates and on the more general case of calibration on $SO(3)$.

References

[1] P. Barooah and J. P. Hespanha, "Distributed estimation from relative measurements in sensor networks," in *Proc. of the 2nd Int. Conf. on Intelligent Sensing and Information Processing*, Dec. 2005.

[2] 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*, ser. Lect. Notes in Comput. Science. Berlin: Springer, June 2006, vol. 4026, pp. 266–281.

[3] P. Barooah and J. P. Hespanha, "Estimation on graphs from relative measurements: Distributed algorithms and fundamental limits," *IEEE Contr. Syst. Mag.*, vol. 27, no. 4, pp. 57–74, Aug. 2007.

[4] A. Sarlette, "Geometry and symmetries in coordination control," Ph.D. dissertation, University of Liège, 2009. [Online]. Available: http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2009/Sar09

[5] A. Sarlette and R. Sepulchre, "Consensus optimization on manifolds," *SIAM J. Control and Optimization*, vol. 58, no. 1, pp. 56–76, 2009. [Online]. Available: http://www.montefiore.ulg.ac.be/services/stochastic/pubs/2009/SS09a

[6] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *CDC*, 2009, pp. 901–908.

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

[8] W. Russell, D. Klein, and J. Hespanha, "Optimal estimation on the graph cycle space," in *American Control Conference (ACC), 2010*, 30 2010-july 2 2010, pp. 1918 –1924.