2019 18th European Control Conference (ECC)
Napoli, Italy, June 25-28, 2019

Distributed Dual Quaternion Based Localization of Visual Sensor Networks

Luca Varotto, Marco Fabris, Giulia Michieletto, Angelo Cenedese

Abstract— **In this paper we consider the localization problem for a visual sensor network. Inspired by the alternate attitude and position distributed optimization framework discussed in [1], we propose an estimation scheme that exploits the unit dual quaternion algebra to describe the sensors pose. This representation is beneficial in the formulation of the optimization scheme allowing to solve the localization problem without designing two interlaced position and orientation estimators, thus improving the estimation error distribution over the two pose components and the overall localization performance. Furthermore, the numerical experimentation asserts the robustness of the proposed algorithm w.r.t. the initial conditions.**

I. Introduction

A Visual Sensor Network (VSN) is a multi-sensor system composed of a collection of spatially distributed camera-devices capable of communicating over a wireless network [2]. Thanks to the advances in high performance embedded microcontrollers, optimized computer vision techniques and reliable communication protocols, multi-camera networks have increasingly spread out in the last twenty years becoming ubiquitous smart systems in industrial, civil and domestic context. By acquiring information-rich data, these architectures enable vision-based interpretative applications as intelligent IoT surveillance [3], smart living domotics [4], autonomous vehicles assisted driving [5], industrial environments monitoring and control [6], urban perimeter/area patrolling for events detection [7], to cite a few. In all the mentioned scenarios, the efficiency of most of the tasks is strictly conditioned by the knowledge from each device of its pose w.r.t. a global inertial reference frame.

The self-estimation of the position and the attitude of each visual sensor node in a multi-device network corresponds to the *localization* task. This constitutes a classic problem in the literature on autonomous multi-camera systems (see, e.g., [8] and the references therein), mainly motivated by the fact that manual ad-hoc localization is a tedious and error prone task, not suitable to handle wide and/or dynamic networks.

Related works - Various approaches have been developed to solve the distributed VSNs localization problem. Most of these can be categorized according to the dimension of the estimation domain (planar vs. tridimensional), the adopted solution paradigm (centralized vs. distributed), the available measurements (distances, angles of arrival, bearings), some a priori information (e.g., pose of some special nodes, referred as *beacons* or *anchors*) and some assumptions on the environment (e.g., presence of moving objects or markers). From a mathematical perspective, the automated localization task can be recast as an optimization problem over a proper manifold. In this direction, the approach described in [1] consists in the iterative minimization of a suitable cost function through a distributed consensus-based strategy. In [9] the previous method is extended by employing maximum-likelihood estimation techniques. In [10] a generalized Newton scheme is proposed for a non-linear refinement of cameras pose exploiting the intrinsic Riemannian structure of the Essential manifold. Similarly, in [11] a projective Newton optimization is performed on the Special Euclidean manifold $\mathbb{SE}(3)$.

In particular, the solution in [1] (hereafter also referred as *TV algorithm*) represents the starting point of this work. This relies on the assumption that each device in the network retrieves noisy relative pose measurements w.r.t. every other cameras with overlapping field of view (e.g., by employing standard computer vision techniques based on two-view geometry). These available measurements are then exploited to derive the cameras pose through a distributed minimization on $\mathbb{SE}(3)$ resting upon the least-squares (LS) principle. In particular, since $\mathbb{SE}(3) = \mathbb{SO}(3) \times \mathbb{R}^3$, the optimization problem is split into two consecutive interlaced steps: first, a Riemannian gradient descent is performed on the Special Orthogonal manifold $\mathbb{SO}(3)$ to estimate the cameras attitude, then the achieved estimates are used to derive the cameras position through a gradient descent on the Euclidean space $\mathbb{R}^3$. One of the main drawbacks of this approach is that the estimation error is not equally distributed among the two pose components: because of the sequential scheme, the position estimates accumulate much more error than orientation ones. Furthermore, the goodness of the final estimates strictly depends on the initial guess.

Contributions - To overcome the aforementioned issues, we adopt the (unit) dual quaternion formalism to unitarily represent the cameras pose. Through this convention, the LS cost function accounting for the pose estimation error needs to be suitably redefined, thus allowing the design of a single estimator for both position and orientation. Through this approach, our solution guarantees a more balanced estimation error distribution as regards the two pose components and turns out to be more robust w.r.t. the initial conditions.

Paper structure - Sec. II is devoted to the recall of some mathematical preliminaries. In Sec. III, the localization problem is formally stated, providing the distributed dual quaternion based solution whose numerical results are reported in Sec. IV. Finally, the main conclusions are drawn in Sec. V.

All the authors are with the Department of Information Engineering, Università degli Studi di Padova, Italy, Corresponding author: L. Varotto luca.varotto.5@phd.unipd.it.

This work was supported by the University of Padova under grant BIRD168152.

II. Mathematical Preliminaries

This section summarizes some principal foundations about quaternion [12] and dual quaternion [13] algebras to provide a mathematical background for the proposed estimation law.

A. Quaternion Algebra

A quaternion $\mathbf{q}$ is an extension of a complex number in a higher dimensional space. This is defined as the sum of a real and a complex part, i.e.,

$$\mathbf{q} = \underbrace{q_0}_{\text{real part}} + \underbrace{\hat{\imath}q_1 + \hat{\jmath}q_2 + \hat{k}q_3}_{\text{complex part}} \in \mathbb{H}, \tag{1}$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$, $\hat{\imath}, \hat{\jmath}, \hat{k}$ are such that $\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1$, and $\mathbb{H}$ denotes the quaternion space. A quaternion $\mathbf{q} \in \mathbb{H}$ is generally represented as a four-dimensional vector composed by the *scalar part* $q_0 \in \mathbb{R}$ and the *vector part* $\bar{\mathbf{q}} = [q_1 \; q_2 \; q_3]^\top \in \mathbb{R}^3$, i.e., $\mathbf{q} = [q_0 \;\; \bar{\mathbf{q}}^\top]^\top$.

Given two generic quaternions $\mathbf{p}, \mathbf{q}$, the conjugation and composition operations in $\mathbb{H}$ are respectively defined as

$$\mathbf{q}^\star = \begin{bmatrix} q_0 & -\bar{\mathbf{q}}^\top \end{bmatrix}^\top, \tag{2}$$

$$\mathbf{p} \circ \mathbf{q} = \begin{bmatrix} p_0 q_0 - \bar{\mathbf{p}}^\top \bar{\mathbf{q}} & (p_0\bar{\mathbf{q}} + q_0\bar{\mathbf{p}} + \bar{\mathbf{p}} \times \bar{\mathbf{q}})^\top \end{bmatrix}^\top. \tag{3}$$

Notably, the composition rule (3) can be expressed as a matrix-vector product. Indeed, it holds that

$$\mathbf{p} \circ \mathbf{q} = \begin{bmatrix} p_0 & -\bar{\mathbf{p}}^\top \\ \bar{\mathbf{p}} & p_0\mathbf{I}_3 + [\bar{\mathbf{p}}]_\times \end{bmatrix} \begin{bmatrix} q_0 \\ \bar{\mathbf{q}} \end{bmatrix} = \mathbf{M}(\mathbf{p})\mathbf{q} \tag{4}$$

$$= \begin{bmatrix} q_0 & -\bar{\mathbf{q}}^\top \\ \bar{\mathbf{q}} & q_0\mathbf{I}_3 - [\bar{\mathbf{q}}]_\times \end{bmatrix} \begin{bmatrix} p_0 \\ \bar{\mathbf{p}} \end{bmatrix} = \mathbf{N}(\mathbf{q})\mathbf{p}, \tag{5}$$

where $\mathbf{M}(\cdot), \mathbf{N}(\cdot) \in \mathbb{R}^{4\times4}$, $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the three dimensional identity matrix and $[\cdot]_\times$ indicates the map from $\mathbb{R}^3$ to the Lie algebra $\mathfrak{so}(3)$ that associates any non-zero vector to its corresponding skew-symmetric matrix. As regards the quaternion composition, we also point out the next results that will be useful in the following:

$$\mathbf{p}^\star \circ \mathbf{q} = \mathbf{M}(\mathbf{p}^\star)\mathbf{q} = \widetilde{\mathbf{N}}(\mathbf{q})\mathbf{p}, \tag{6}$$

where

$$\widetilde{\mathbf{N}}(\mathbf{q}) = \begin{bmatrix} q_0 & \bar{\mathbf{q}}^\top \\ \bar{\mathbf{q}} & -q_0\mathbf{I}_3 + [\bar{\mathbf{q}}]_\times \end{bmatrix} \in \mathbb{R}^{4\times4}. \tag{7}$$

To conclude, a quaternion $\mathbf{q} \in \mathbb{H}$ is called *unit* when $\|\mathbf{q}\|^2 = q_0^2 + \bar{\mathbf{q}}^\top\bar{\mathbf{q}} = 1$. In this case, it belongs to the unit hypersphere $\mathbb{H}_u$ embedded in $\mathbb{R}^4$ and can be used to describe a rotation in 3D space. Formally, given two coordinates systems $\mathscr{F}_i$ and $\mathscr{F}_j$ such that $\mathscr{F}_i$ is rotated by an angle $\theta \in (-\pi, \pi]$ around the unit vector $\mathbf{u} \in \mathbb{R}^3$ w.r.t. $\mathscr{F}_j$, their relative orientation can be represented by the unit quaternion

$$\mathbf{q} = \begin{bmatrix} \cos\frac{\theta}{2} & \mathbf{u}^\top \sin\frac{\theta}{2} \end{bmatrix}^\top \in \mathbb{H}_u. \tag{8}$$

B. Dual Quaternion Algebra

Within the dual number theory context, a dual quaternion $\mathbf{d}$ is generally defined as

$$\mathbf{d} = \mathbf{q}_r + \epsilon\mathbf{q}_d \in \mathbb{DH}, \tag{9}$$

where the (single) quaternions $\mathbf{q}_r, \mathbf{q}_d \in \mathbb{H}$ represent its *real* and *dual parts*, ϵ is the nilpotent dual unit satisfying $\epsilon^2 = 0$ and $\epsilon \neq 0$ and $\mathbb{DH}$ denotes the dual quaternion space.

Considering $\mathbf{d} = \mathbf{q}_r + \epsilon\mathbf{q}_d$ and $\mathbf{d}' = \mathbf{q}'_r + \epsilon\mathbf{q}'_d$ in $\mathbb{DH}$, the composition operation on this manifold is given by

$$\mathbf{d} \odot \mathbf{d}' = \mathbf{q}_r \circ \mathbf{q}'_r + \epsilon(\mathbf{q}_r \circ \mathbf{q}'_d + \mathbf{q}_d \circ \mathbf{q}'_r). \tag{10}$$

From a mathematical perspective, a dual quaternion can be suitably represented as an eight-dimensional vector, e.g., $\mathbf{d} = [q_{r,0} \; \bar{\mathbf{q}}_r^\top \; q_{d,0} \; \bar{\mathbf{q}}_d^\top]^\top$, where $q_{r,0}, q_{d,0} \in \mathbb{R}$ and $\bar{\mathbf{q}}_r, \bar{\mathbf{q}}_d \in \mathbb{R}^3$ according to the single quaternion notation. Hence, similarly to the quaternion case, the composition (10) can be expressed as matrix-vector multiplication, namely

$$\begin{aligned} \mathbf{d} \odot \mathbf{d}' &= \begin{bmatrix} \mathbf{M}(\mathbf{q}_r) & \mathbf{0}_4 \\ \mathbf{M}(\mathbf{q}_d) & \mathbf{M}(\mathbf{q}_r) \end{bmatrix} \begin{bmatrix} \mathbf{q}'_r \\ \mathbf{q}'_d \end{bmatrix} = \mathbf{U}(\mathbf{d})\mathbf{d}' \\ &= \begin{bmatrix} \mathbf{N}(\mathbf{q}'_r) & \mathbf{0}_4 \\ \mathbf{N}(\mathbf{q}'_d) & \mathbf{N}(\mathbf{q}'_r) \end{bmatrix} \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_d \end{bmatrix} = \mathbf{V}(\mathbf{d}')\mathbf{d}. \end{aligned} \tag{11}$$

where $\mathbf{U}(\cdot), \mathbf{V}(\cdot) \in \mathbb{R}^{8\times8}$ and the notation $\mathbf{0}_4$ is used to indicate a (4×4) matrix whose entries are all zeros. Note that, analogously to (6), it also holds that

$$\mathbf{d}^\star \odot \mathbf{d}' = \mathbf{U}(\mathbf{d}^\star)\mathbf{d}' = \widetilde{\mathbf{V}}(\mathbf{d}')\mathbf{d}, \tag{12}$$

where $\mathbf{d}^\star = \mathbf{q}_r^\star + \epsilon\mathbf{q}_d^\star \in \mathbb{DH}$ is the conjugate of the dual quaternion $\mathbf{d} \in \mathbb{DH}$ and $\widetilde{\mathbf{V}}(\mathbf{d}) \in \mathbb{R}^{8\times8}$ is defined as

$$\widetilde{\mathbf{V}}(\mathbf{d}) = \begin{bmatrix} \widetilde{\mathbf{N}}(\mathbf{q}_r) & \mathbf{0}_4 \\ \widetilde{\mathbf{N}}(\mathbf{q}_d) & \widetilde{\mathbf{N}}(\mathbf{q}_r) \end{bmatrix}. \tag{13}$$

Finally, a dual quaternion $\mathbf{d} \in \mathbb{DH}$ is called *unit* when $\|\mathbf{q}_r\|^2 = 1$ (its real part is a unit quaternion) and $\mathbf{q}_r^\top\mathbf{q}_d = 0$ (its real and dual part are orthogonal). In this case, the dual quaternion belongs to the set $\mathbb{DH}_u$ and can be used to describe a pose transformation in 3D space. Formally, given two coordinates systems $\mathscr{F}_i$ and $\mathscr{F}_j$, their relative orientation $\mathbf{q}_r \in \mathbb{H}_u$ and the position $\mathbf{p} \in \mathbb{R}^3$ of the origin of $\mathscr{F}_i$ in $\mathscr{F}_j$ (given in $\mathscr{F}_j$) can be represented by the dual quaternion

$$\mathbf{d} = \mathbf{q}_r + \frac{\epsilon}{2}\mathbf{q}_t \circ \mathbf{q}_r \in \mathbb{DH}_u, \tag{14}$$

where $\mathbf{q}_t = \begin{bmatrix} 0 & \mathbf{p}^\top \end{bmatrix}^\top \in \mathbb{H}$ is the (pure) quaternion embedding the vector $\mathbf{p} \in \mathbb{R}^3$.

III. Distributed Localization of a VSN

In this section the localization problem for a VSN is formally stated and, inspired by [1], a suitable distributed solution is provided resting upon a gradient descent minimization. The original and innovative aspect relies on the cameras pose representation via the dual quaternion formalism and its exploitation in the optimization framework for the design of the localization algorithm.

A. Problem Statement

Consider a VSN made up of n cameras. According to the most agreed pinhole model [10], we assume that each device $i \in \{1 \dots n\}$ is associated to a local frame $\mathscr{F}_i = \{O_i, (X_i, Y_i, Z_i)\}$ so that the origin O_i coincides with the center of projection of the camera, while the Z_i-axis (named *optical axis*) is pointed in the viewing direction, the Y_i-axis is up faced and the X_i-axis is oriented according to the left-handed coordinates system. The pose $g_i \in \mathbb{SE}(3)$ of the camera is thus identified by the position and the orientation of $\mathscr{F}_i$ w.r.t. the global inertial frame.

We then model the network as a connected undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ so that the i-th device is identified by i-th node in the vertex set $\mathcal{V}$ and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is determined by accounting for the network visibility constraints, meaning that $(i, j) \in \mathcal{E}$ if the i-th and j-th cameras have (partially) overlapping field of view. For the sake of simplicity, we also assume that the neighboring cameras (corresponding to adjacent nodes in the graph) are able to communicate according to a preset protocol. This means that $\mathcal{E}$ describes both the cameras communication and sensing interactions.

The distributed localization problem consists in the determination of the pose (w.r.t. the inertial frame) of each camera composing the given VSN, by exploiting the available measurements, in a distributed manner. To this end, it is useful to observe that for each pair of cameras i and j with overlapping fields of view (i.e., such that $(i, j) \in \mathcal{E}$), the pose g_i of the i-th camera is linked to the pose g_j of the neighboring j-th camera. In fact, the relative change of coordinates $g_{ij} = g_i^{-1} \circ g_j$ holds and, consistently, the pose of camera j w.r.t. that of i results as $g_j = g_i \circ g_{ij}$. In the following, g_{ij} is referred as the *relative* pose between the i-th and the j-th neighboring cameras, whereas the pose g_i of camera i is said *absolute*, and represents its state. Moreover, each camera in the network is assumed to be able to employ standard computer vision algorithms (see, e.g., [14], [15], [16]) to recover a noisy measurement $\widetilde{g}_{ij} \in \mathbb{SE}(3)$ of the relative pose w.r.t. to any device in its neighborhood $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. Note that in general, it occurs that $\widetilde{g}_{ij} \neq (\widetilde{g}_{ji})^{-1}$, due to noise effects.

In the light of these observations, the main objective of the localization task for a given VSN represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the computation of a set of the relative transformations $\{\hat{g}_{ij}, (i, j) \in \mathcal{E}\}$ that satisfy the consistency constraint along network cycles[1] and, simultaneously, are as close as possible to the available measurements $\{\widetilde{g}_{ij}, (i, j) \in \mathcal{E}\}$. To this aim, employing a LS approach, it is convenient to minimize the following cost function

$$\rho(\{\hat{g}_{ij}\}) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \tfrac{1}{2} d_g^2(\hat{g}_{ij}, \widetilde{g}_{ij}), \tag{15}$$

where $d_g(\cdot, \cdot)$ defines the distance on the $\mathbb{SE}(3)$ manifold. Note that the available measurements may not be consistent. To enforce this constraint, the relative transformations can be reparametrized using the absolute ones, i.e.,

$$\rho(\{\hat{g}_i\}) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \tfrac{1}{2} d_g^2(\hat{g}_i^{-1} \circ \hat{g}_j, \widetilde{g}_{ij}). \tag{16}$$

In particular, we observe that the cost function (16) can be rewritten in a form more convenient for the application of a distributed paradigm, namely

$$\begin{aligned} \rho(\{\hat{g}_i\}) &= \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{N}_i} \tfrac{1}{4} \left(d_g^2(\hat{g}_i^{-1} \circ \hat{g}_j, \widetilde{g}_{ij}) + d_g^2(\hat{g}_j^{-1} \circ \hat{g}_i, \widetilde{g}_{ji}) \right) \\ &= \sum_{i \in \mathcal{V}} \rho_i(\hat{g}_i). \end{aligned} \tag{17}$$

[1]The consistency constraint requires the transformation along any cycle in the network results to be identical.

It is straightforward that the self-localization problem can be solved in a distributed way: each camera is required to solve the (non-linear) minimization of the local cost function $\rho_i(\hat{g}_i)$. This depends only on information available locally or through 1-hop communication: the actual pose estimates $\hat{g}_i$ and $\{\hat{g}_j,\ j \in \mathcal{N}_i\}$, and the measurements sets $\{\tilde{g}_{ij}, j \in \mathcal{N}_i\}$ and $\{\tilde{g}_{ji}, j \in \mathcal{N}_i\}$. In particular, this optimization can be achieved using an iterative consensus framework, as in [1].

B. Dual Quaternion Based Solution

In [1], the minimization of the cost function (17) is addressed by exploiting the fact that $\mathbb{SE}(3) = \mathbb{SO}(3) \times \mathbb{R}^3$. Each pose $g_i \in \mathbb{SE}(3)$ corresponds to the pair $(\mathbf{R}_i, \mathbf{p}_i)$ where $\mathbf{R}_i \in \mathbb{SO}(3)$ is a rotation matrix and $\mathbf{p}_i \in \mathbb{R}^3$ is a position vector. The distance function $d_g(\cdot, \cdot)$ can then be redefined by considering the metrics on $\mathbb{SO}(3)$ and $\mathbb{R}^3$ as

$$\begin{aligned} d_g^2(\hat{g}_{ij}, \widetilde{g}_{ij}) &= d_{\mathbb{SO}(3)}^2(\hat{\mathbf{R}}_{ij}, \widetilde{\mathbf{R}}_{ij}) + \|\hat{\mathbf{R}}_i^\top \hat{\mathbf{p}}_{ij} - \widetilde{\mathbf{p}}_{ij}\|^2 \qquad (18) \\ &= d_{\mathbb{SO}(3)}^2(\hat{\mathbf{R}}_i^\top \hat{\mathbf{R}}_j, \widetilde{\mathbf{R}}_{ij}) + \|\hat{\mathbf{R}}_i^\top (\hat{\mathbf{p}}_j - \hat{\mathbf{p}}_i) - \widetilde{\mathbf{p}}_{ij}\|^2 \end{aligned}$$

and consequently, substituting (18) in (17), it follows that

$$\rho(\{\hat{g}_i\}) = \rho_R(\{\hat{\mathbf{R}}_i\}) + \rho_T(\{\hat{\mathbf{R}}_i, \hat{\mathbf{p}}_i\}). \tag{19}$$

The TV algorithm described in [1] solves the distributed localization task by first minimizing $\rho_R(\cdot)$ through a Riemannian gradient descent and then $\rho_T(\cdot)$ applying a Euclidean gradient descent, thus splitting the optimization problem in two consequent steps. Following this scheme, it is perceivable that the estimation error accumulated during the first step (minimization of $\rho_R(\cdot)$) is inherited by the second step (minimization of $\rho_T(\cdot)$), hence, the estimation inaccuracy is not equally distributed over the two pose components. Moreover, it is verified that the performance of the optimization procedure strongly depends on the estimates initialization.

To overcome all the aforementioned issues, we intend to reformulate the cost function (17) using the algebra of unit dual quaternions, although this implies to consider $\mathbb{DH}_u$ instead of $\mathbb{SE}(3)$ as poses domain. To this end, we consider the following distance function

$$\begin{aligned} d_{\mathbb{DH}_u}(\hat{\mathbf{d}}_{ij}, \widetilde{\mathbf{d}}_{ij}) &= \|\widetilde{\mathbf{d}}_{ij}^\star \odot \hat{\mathbf{d}}_{ij}\| = \|\widetilde{\mathbf{d}}_{ij}^\star \odot (\hat{\mathbf{d}}_i^\star \odot \hat{\mathbf{d}}_j)\| \qquad (20) \\ &= \|\mathbf{U}(\widetilde{\mathbf{d}}_{ij}^\star)\widetilde{\mathbf{V}}(\hat{\mathbf{d}}_j)\hat{\mathbf{d}}_i\|, \qquad (21) \end{aligned}$$

where $\hat{\mathbf{d}}_{ij}, \widetilde{\mathbf{d}}_{ij} \in \mathbb{DH}_u$ denote the estimate and the measurement of the relative pose between the i-th and the j-th neighboring cameras, respectively, while $\hat{\mathbf{d}}_i, \hat{\mathbf{d}}_j \in \mathbb{DH}_u$ indicate the estimate of their absolute poses.

Employing (21), the cost function (17) turns out to be

$$\rho(\{\hat{\mathbf{d}}_i\}) = \sum_{i \in \mathcal{V}} \rho_i(\hat{\mathbf{d}}_i), \qquad \text{where} \tag{22}$$

$$\rho_i(\hat{\mathbf{d}}_i) = \sum_{j \in \mathcal{N}_i} \tfrac{1}{4} \Big(\|\mathbf{U}(\widetilde{\mathbf{d}}_{ij}^\star)\widetilde{\mathbf{V}}(\hat{\mathbf{d}}_j)\hat{\mathbf{d}}_i\|^2 + \|\mathbf{U}(\widetilde{\mathbf{d}}_{ji}^\star)\widetilde{\mathbf{V}}(\hat{\mathbf{d}}_i)\hat{\mathbf{d}}_j\|^2 \Big).$$

Adopting the dual quaternion formalism, the minimization of (22) can still be performed following the distributed paradigm; in particular, each device in the network can locally exploit an iterative optimization strategy to minimize $\rho_i(\hat{\mathbf{d}}_i)$. Hence, we suppose that, after an initialization step

Algorithm 1: DDQL

// t_{max} total number of iterations
// n number of cameras in the network
Input: relative noisy measurements $\{\widetilde{\mathbf{d}}_{ij}, (i,j) \in \mathcal{E}\}$
Initialization $(t = 0)$:
initial pose estimates $\{\hat{\mathbf{d}}_i(0), i \in \mathcal{V}\}$, $\hat{\mathbf{d}}_1(0) = \mathbf{d}_I$
initial cost function value $\rho(0) = \rho(\{\hat{\mathbf{d}}_i(0)\}, \{\widetilde{\mathbf{d}}_{ij}\})$
for $t = 1 : t_{max}$ **do**
  $\hat{\mathbf{d}}_1(t) = \mathbf{d}_I$
  for $i = 2 : n$ **do**
    Update: $\hat{\mathbf{d}}_i(t) = \hat{\mathbf{d}}_i(t-1) - \delta \frac{\partial \rho_i(t-1)}{\partial \hat{\mathbf{d}}_i(t-1)}, \quad \delta > 0$
    Normalization: $\hat{\mathbf{d}}_i(t) = \hat{\mathbf{q}}_{r,i}(t) + \epsilon \hat{\mathbf{q}}_{d,i}(t)$
    $\rightarrow \hat{\mathbf{d}}'_i(t) = \hat{\mathbf{q}}'_{r,i}(t) + \epsilon \hat{\mathbf{q}}'_{d,i}(t)$ with
      $\hat{\mathbf{q}}'_{r,i}(t) = \frac{\hat{\mathbf{q}}_{r,i}(t)}{\|\hat{\mathbf{q}}_{r,i}(t)\|}$
      $\hat{\mathbf{q}}'_{d,i}(t) = \hat{\mathbf{q}}_{d,i}(t) - \hat{\mathbf{q}}_{r,i}(t)(\hat{\mathbf{q}}'_{d,i}(t)^\top \hat{\mathbf{q}}_{r,i}(t))$
  end
  $\rho(t) = \rho(\{\hat{\mathbf{d}}'_i(t)\}, \{\widetilde{\mathbf{d}}_{ij}\})$
end

required to set the initial pose estimates, at every iteration t each node i in the network performs the following procedure.

1) Denoting the current estimate of the absolute pose $\mathbf{d}_i$ by $\hat{\mathbf{d}}_i(t)$, it first determines the derivative of $\rho_i(t) = \rho_i(\hat{\mathbf{d}}_i(t))$ w.r.t. $\hat{\mathbf{d}}_i(t)$, whose expression is given by

$$\frac{\partial \rho_i(t)}{\partial \hat{\mathbf{d}}_i(t)} = \sum_{j \in \mathcal{N}_i} \Big(\widetilde{\mathbf{V}}(\hat{\mathbf{d}}_j(t))^\top \mathbf{U}(\widetilde{\mathbf{d}}^\star_{ij})^\top \mathbf{U}(\widetilde{\mathbf{d}}^\star_{ij}) \widetilde{\mathbf{V}}(\hat{\mathbf{d}}_j(t)) + \\ + \mathbf{U}(\hat{\mathbf{d}}^\star_j(t))^\top \mathbf{U}(\widetilde{\mathbf{d}}^\star_{ji})^\top \mathbf{U}(\widetilde{\mathbf{d}}^\star_{ji}) \mathbf{U}(\hat{\mathbf{d}}^\star_j(t)) \Big) \hat{\mathbf{d}}_i(t).$$

2) Then, it determines the estimation $\hat{\mathbf{d}}_i(t+1)$ related to the subsequent iteration by performing a steepest gradient descent step, so that

$$\hat{\mathbf{d}}_i(t+1) = \hat{\mathbf{d}}_i(t) - \delta \frac{\partial \rho_i(t)}{\partial \hat{\mathbf{d}}_i(t)}, \tag{23}$$

where $\delta > 0$ constitutes a properly chosen step-size.

3) Since the additive update rule (23) may violate the constraints of $\mathbb{DH}_u$ space, a normalization step is performed on the computed $\hat{\mathbf{d}}_i(t+1) = \hat{\mathbf{q}}_{r,i}(t+1) + \epsilon \hat{\mathbf{q}}_{d,i}(t+1)$. The new estimates thus results $\hat{\mathbf{d}}'_i(t+1) = \hat{\mathbf{q}}'_{r,i}(t+1) + \epsilon \hat{\mathbf{q}}'_{d,i}(t+1)$ with

$$\hat{\mathbf{q}}'_{r,i} = \frac{\hat{\mathbf{q}}_{r,i}}{\|\hat{\mathbf{q}}_{r,i}\|}, \quad \hat{\mathbf{q}}'_{d,i} = \hat{\mathbf{q}}_{d,i} - \hat{\mathbf{q}}_{r,i}(\hat{\mathbf{q}}'^\top_{d,i} \hat{\mathbf{q}}_{r,i}), \tag{24}$$

dropping the time dependency for the sake of compactness. The pose estimate $\hat{\mathbf{d}}'_i(t+1)$ is finally communicated to all the neighboring nodes $j \in \mathcal{N}_i$.

The iterative algorithm stops after a preset number $t_{max} \in \mathbb{N}$ of iterations, which has to be large enough to guarantee the achievement of a minimum for the cost function.

Alg. 1 summarizes the described procedure (referred as *Distributed Dual Quaternion based Localization*, DDQL) that allows to estimate the absolute pose of each camera in a VSN starting from the noisy and inconsistent pose measurements. Note that, w.l.o.g., we assume that these are expressed according to the dual quaternion formalism.

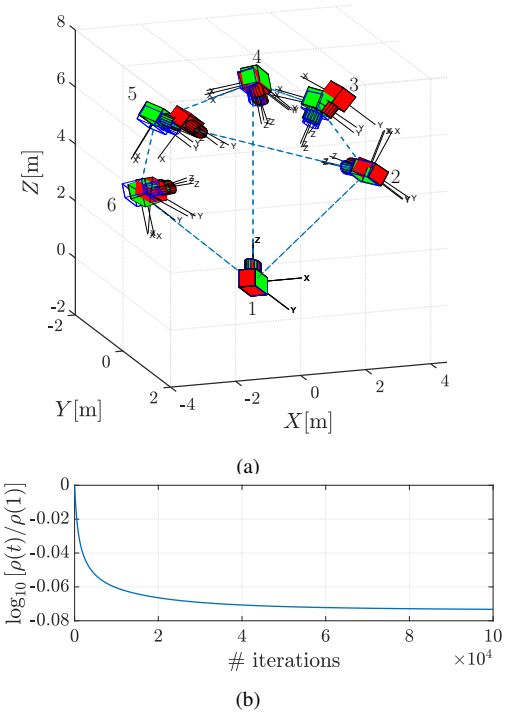

Fig. 1: Performance of DDQL algorithm: (a) real cameras poses (blue), initial estimates (red) and final estimates (green) of the Alg. 1; (b) cost function trend in logarithmic scale.

Moreover, we suppose that the body frame of camera 1 coincides with the global reference frame, namely its pose is always encoded by $\mathbf{d}_I \in \mathbb{DH}_u$, i.e., the dual quaternion corresponding to identical rotation and the zero position. This assumption is necessary to solve the localization problem when no a priori knowledge on the pose of a camera is available and it implies that all the final estimates are biased by the same quantity w.r.t. to the inertial frame.

IV. Numerical Results

In this section some simulative results are provided to show the performance of the DDQL algorithm. Its effectiveness is confirmed by the comparison with the TV algorithm described in [1], focusing on the advantages carried out by the cost function reformulation through the dual quaternion formalism and the optimal solution that follows.

A. Performance of DDQL Algorithm

To illustrate the effectiveness of Alg. 1, we consider the camera network in Fig. 1(a) composed of $n = 6$ devices located at the same height from the ground, i.e., a planar VSN that can be used, for instance, for monitoring a certain area. The simulation campaign is carried out accounting for the cameras interactions defined by the graph depicted in light blue (dashed lines) in the aforementioned figure, where we also report the real absolute cameras poses, namely the ground truth of the test. We generate the noisy relative measurements by composing the real absolute poses corrupted by noise. In detail, the (absolute) orientations are altered by additional white Gaussian noise with standard deviation

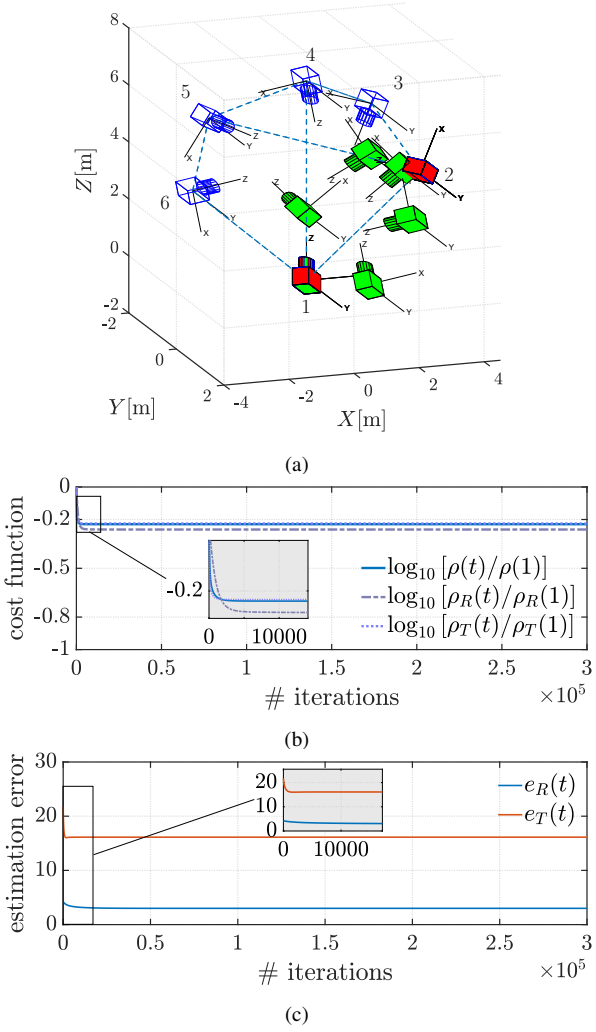

Fig. 2: Performance of TV algorithm with bad initial conditions: (a) real cameras poses (blue), initial estimates (red) and final estimates (green); (b) trend of the cost function and its components in logarithmic scale; (c) trend of the average estimation error on orientations (blue line) and positions (red line).

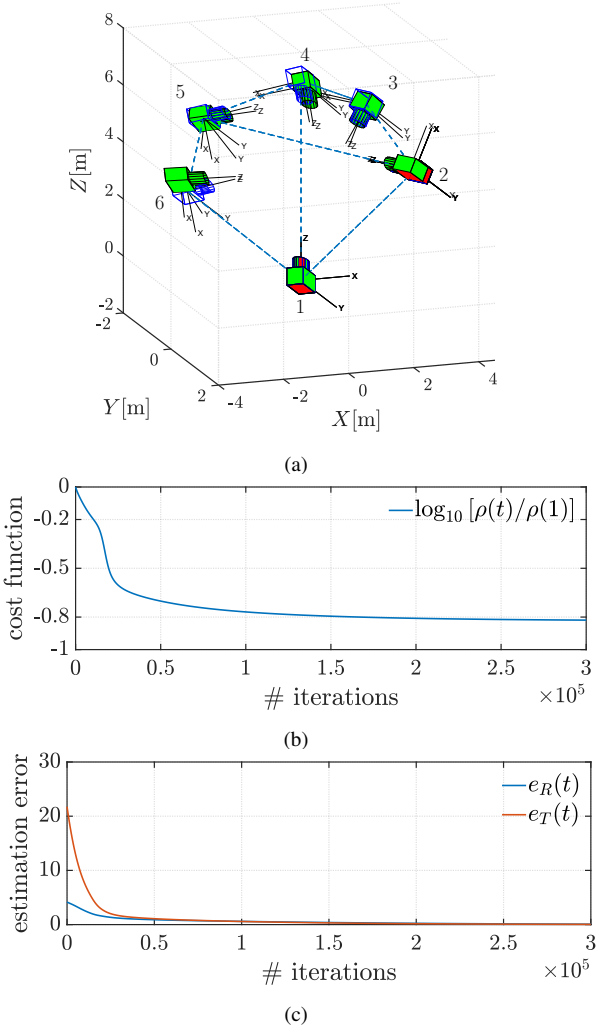

Fig. 3: Performance of DDQL algorithm with bad initial conditions: (a) real cameras poses (blue), initial estimates (red) and final estimates (green), (b) cost function trend in logarithmic scale; (c) trend of the average estimation error on orientations (blue line) and positions (red line).

$\sqrt{5}$deg on the X and Z-axis (tilt and roll angles) and 5deg on the Y-axis (pan angle), whereas a white Gaussian noise with standard deviation $\sqrt{0.005}$m along all the axes is added to the (absolute) positions. The DDQL algorithm is run by setting $\delta = 10^{-4}$ and $t_{max} = 10^5$ iterations, and assuming to start by the initial pose estimates marked in red in Fig. 1(a).

The output of the proposed method, i.e., the final pose estimates reported in green in Fig. 1(a), approaches the ground truth, although a small error affects the cameras orientation estimate along the Y-axis. This observation can be justified by the fact that the noise on the relative rotations is assumed to have a larger variance along this axis, nonetheless the estimation error results to be well distributed between the orientations and positions estimates. It is also worth to notice that the cost function (17), whose trend is depicted in Fig. 1(b), decreases by almost 16% from its initial value: this represents a sensible minimization.

B. Comparison between TV and DDQL algorithm

To highlight the advantages of adopting $\mathbb{DH}_u$ instead of $\mathbb{SE}(3)$ as pose manifold, we compare the performance of the DDQL algorithm w.r.t. the TV one. In particular, we consider the previously described network but we account for a noise-free setup wherein the relative measurements correspond to the real relative poses. Moreover, to stress the robustness of Alg. 1 w.r.t. the initial conditions when compared to the approach described in [1], we suppose to initialize all the poses estimates so that they correspond to the true pose of one of the devices (except for the first camera that is initialized to its own true pose).

Fig. 2 shows the performance of the TV-algorithm. We observe that the final estimates (green cameras in Fig 2(a)) are very far from the real poses (blue profiled cameras in the same figure). This behaviour is consistent with the fact that the cost function (17) and its components ($\rho_R(\cdot)$ and

$\rho_T(\cdot)$ introduced in (19)) attain a local minimum, as depicted in Fig 2(b). Fig. 2(c) describes the behaviour of $e_R(\cdot)$ and $e_T(\cdot)$, namely the mean estimation errors on orientations and positions, respectively. These quantities are computed according to the following expressions

$$e_R(t) = \frac{1}{n}\sum_{i\in\mathcal{V}} \|\mathbf{R}_i - \hat{\mathbf{R}}_i(t)\|_F^2, \tag{25}$$

$$e_T(t) = \frac{1}{n}\sum_{i\in\mathcal{V}} \|\mathbf{p}_i - \hat{\mathbf{p}}_i(t)\|^2, \tag{26}$$

where $\hat{\mathbf{R}}_i(t) \in \mathbb{SO}(3)$ and $\hat{\mathbf{p}}_i(t) \in \mathbb{R}^3$ denote the i-th camera estimation of its real rotation matrix $\mathbf{R}_i \in \mathbb{SO}(3)$ and position vector $\mathbf{p}_i \in \mathbb{R}^3$ at iteration t, respectively, and $\|\cdot\|_F^2$ indicates the Frobenius norm. As expected, both $e_R(\cdot)$ and $e_T(\cdot)$ decrease only during the first iterations, namely before the local minimum is reached, moreover, in both cases the reduction w.r.t. initial value is moderate. Fig. 2(b) and Fig. 2(c) highlight one of the main drawbacks of TV-algorithm: position estimates are influenced by rotation estimates, hence, $\rho_T(\cdot)$ and $e_T(\cdot)$ experience a smaller decrease compared to $\rho_R(\cdot)$ and $e_R(\cdot)$. Remarkably, this result is independent on the specific initialization, being related to the adopted sequential optimization strategy.

Fig. 3 reports the results of the DDQL algorithm. It clearly appears that this approach is more robust w.r.t. the initial conditions as compared to the TV algorithm: the final poses estimates are very close to the real ones (Fig. 3(a)) and the cost function significantly decreases (Fig. 3(b)). In addition, Fig. 3(c) shows that both orientation and position errors[2] significantly decrease w.r.t. their initial values. In particular, it can be observed that both $e_R(\cdot)$ and $e_T(\cdot)$ tend to zero. As a consequence, at the steady-state we have that $e_R(\cdot) \approx e_T(\cdot)$: this is due to the balanced estimation error distribution, i.e., there is not a larger error accumulation in position estimates than in orientation ones, contrarily to the TV case.

V. Conclusions

In this work, we propose a distributed algorithm solving the self-localization problem for a VSN. The original and innovative aspect of the provided method, termed DDQL algorithm, relies on the exploitation of the (unit) dual quaternion algebra to represent the pose of the devices in the network, which yields a modification to the optimization framework w.r.t the current literature. In particular, as compared to the approach described in [1], that constitutes the starting point for the designed solution, the DDQL method does not require to split the pose estimation problem into two (rotation and position) contributes entailing a better distribution of the estimation error over the two pose components. Numerical simulations support this observation and, in addition, prove that DDQL inherently exhibits an increased robustness w.r.t. the estimation initial conditions.

[2]In this case $\hat{\mathbf{R}}_i(t)$ and $\hat{\mathbf{p}}_i(t)$ are extracted from $\hat{\mathbf{d}}_i(t)$. Indeed, given $\mathbf{d} = \mathbf{q}_r + \epsilon\mathbf{q}_d$, the real quaternion is associated to the rotation matrix $\mathbf{R}(\mathbf{q}_r) = \mathbf{I}_3 + 2q_{r,0}[\bar{\mathbf{q}}_r]_\times + 2[\bar{\mathbf{q}}_r]_\times^2$, according to the Rodrigues formula. Furthermore, from the quaternion composition $2\mathbf{q}_d \circ \mathbf{q}_r$, the position vector can be extract employing (14).

References

[1] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec 2009, pp. 901–908.

[2] C.-M. Kyung *et al.*, *Theory and applications of smart cameras*. Springer, 2016.

[3] M. Pavithra and R. Kathirvel, "Smart camera surveillance system monitoring based internet of things," *Imperial Journal of Interdisciplinary Research*, vol. 3, no. 6, 2017.

[4] S. Gruenwedel, V. Jelaca, J. O. Niño-Castañeda, P. Van Hese, D. Van Cauwelaert, P. Veelaert, and W. Philips, "Decentralized tracking of humans using a camera network," in *Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, vol. 8301. International Society for Optics and Photonics, 2012, p. 83010D.

[5] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *arXiv preprint arXiv:1704.05519*, 2017.

[6] G. Di Leo, C. Liguori, A. Pietrosanto, and P. Sommella, "A vision system for the online quality monitoring of industrial manufacturing," *Optics and Lasers in Engineering*, vol. 89, pp. 162–168, 2017.

[7] N. Bof, R. Carli, A. Cenedese, and L. Schenato, "Asynchronous distributed camera network patrolling under unreliable communication," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 5982–5989, 2017.

[8] F. Bajramovic, *Self-Calibration of Multi-Camera Systems*. Logos Verlag Berlin GmbH, 2010.

[9] J. Knuth and P. Barooah, "Maximum-likelihood localization of a camera network from heterogeneous relative measurements." in *ACC*, 2013, pp. 2374–2379.

[10] Y. Ma, J. Koseck, and S. Sastry, "Optimization criteria and geometric algorithms for motion and structure estimation," *International Journal of Computer Vision*, vol. 44, pp. 219–249, 1999.

[11] M. Sarkis and K. Diepold, "Camera-pose estimation via projective newton optimization on the manifold," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1729–1741, April 2012.

[12] Y.-B. Jia, "Quaternions and rotations," *Com S*, vol. 477, no. 577, p. 15, 2008.

[13] ——, "Dual quaternions," 2013.

[14] H. C. Longuet-Higgins, "Readings in computer vision: Issues, problems, principles, and paradigms," M. A. Fischler and O. Firschein, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987, ch. A Computer Algorithm for Reconstructing a Scene from Two Projections, pp. 61–62. [Online]. Available: http://dl.acm.org/citation.cfm?id=33517.33523

[15] A. Chaudhury, A. Gupta, S. Manna, S. Mukherjee, and A. Chakrabarti, "Multiple view reconstruction of calibrated images using singular value decomposition," *CoRR*, vol. abs/1011.0596, 2010. [Online]. Available: http://arxiv.org/abs/1011.0596

[16] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, p. 13301334, December 2000. [Online]. Available: https://www.microsoft.com/en-us/research/publication/a-flexible-new-technique-for-camera-calibration/