IMPROVING CONSENSUS-BASED DISTRIBUTED CAMERA CALIBRATION VIA EDGE PRUNING AND GRAPH TRAVERSAL INITIALIZATION

G. Michieletto, S. Milani, A. Cenedese, G. Baggio

University of Padova, Dept. of Information Engineering,

e-mail: {giulia.michieletto,simone.milani,angelo.cenedese,giacomo.baggio}@dei.unipd.it

ABSTRACT

Over the past few years, a huge number of distributed camera calibration strategies have been proposed for video surveillance and monitoring systems involving mobile terminals. Many of the proposed solutions rely on consensus-based algorithms, which aim at estimating the configuration of the network via a message passing protocol. In this paper we propose an improved consensus-based distributed camera calibration strategy that exploits a robust initialization, together with a pruning protocol to remove faulty links which could propagate excessively-noisy information through the network reducing the convergence time. The proposed solution seems to improve the state-of-the-art strategies in terms of accuracy, convergence speed, and computational complexity.

***Index Terms*—** camera sensor networks; pose estimation; consensus; camera-in-view, optimization on manifolds

1. INTRODUCTION

The recent employment of patrols of Unmanned Aerial and Surface Vehicles (UAVs and USVs) for several automatic surveillance and delivery tasks has necessitated the design of distributed and fault-tolerant computation and information exchange algorithms [1–3]. As for video surveillance and tracking systems, this need becomes even more significant when referred to the problem of camera calibration. Estimating the orientation and the localization of imaging devices has significant impact on the accuracy of 3D reconstruction strategies [4] and target tracking [5, 6]. Unfortunately, the lack of a centralized entity that facilitates computation, communication, and time-synchronization, together with the time-varying nature of networks and communication links, make traditional multi-camera calibration strategies unsuitable for such systems [7].

Looking for a solution to such inconveniences, scientific research has recently produced a wide variety of distributed calibration protocols that are based on propagating the local information estimated by a single terminal through the network [8]. Many of these rely on the *average consensus algorithm* designed for sensor networks [9, 10]. In its classical formulation, each node measures a scalar quantity, say temperature, and the average temperature over the entire network is obtained by iteratively updating the temperature reading at each node with the average temperature of its neighbours [11]. As for camera calibration, each node/camera estimates its own location/orientation receiving the location/orientation of its neighbours and weighting them using its relative camera pose and rotation. The procedure is iterated multiple times until the estimated parameters does not converge to an optimal configuration.

The work has been supported by the Robotic 3D and by the 3D CloudVision projects, funded by the University of Padova, Italy.

Despite in ideal conditions such approaches work well, in many realistic scenario the accuracy of the final estimate can be significantly impaired. Relative orientations and locations are usually estimated from set of conjugate points acquired by couples of cameras. These estimates can be affected by significant amounts of noise depending on errors in localizing conjugates, on the number of inliers, and on the actual relative position of the cameras. Moreover, since in these iterative approaches the cost function is usually non-convex, the standard descent procedures must be initialized correctly so that the optimization trajectory does not get stuck in some local minima that can be quite far from the real solution.

The proposed approach aims at solving both issues in consensus-based approaches by pruning the communication links in the network which can not be considered reliable and by adopting an appropriate initialization at the beginning of the iterative procedure. Firstly, as far as the former contribution, the quality of relative parameters is tested in order to evaluate whether the communication link is to be kept or not. Secondly, specifically with respect to the initialization issue, various strategies based on different ways to explore the camera network are proposed with the aim of correctly initializing the minimization procedure.

Experimental results show that the proposed solution permits obtaining a more accurate localization of cameras, together with a faster convergence speed.

In the following, Section 2 overviews some of the existing works on consensus-based distributed camera calibration. Section 3 presents the adopted distributed camera calibration strategy, while Section 4 describes the proposed modifications. Section 5 reports some experimental results, and Section 6 draws the final conclusions.

2. RELATED WORKS

Motivated by the fact that manual ad-hoc localization methods are not suitable to handle large number of cameras or dynamic configurations in the network (e.g. a VSN composed of mobile devices), the automation of the pose reconstruction process has become essential to cope with accuracy and real-time requirements.

The automated calibration task can be casted into an optimization (or a consensus) problem over a Riemannian manifold [12–14]. Conversely to standard calibration techniques, in this case, the optimization is carried out in the natural space of the problem, i.e. the space of rigid-body transformations $SE(3)$ (or the space of rotations $SO(3)$, if only the orientations are considered).

Some interesting results in this direction are presented in the recent literature: in [15, 16], whose approaches consist in iterative procedures based on the minimization of a suitable cost functional through a distributed strategy working in the Riemannian consensus framework; in [17], which extends and refines the previous method by exploiting maximum-likelihood estimation techniques; in [18, 19], which deal with a dynamic scenario consisting of mobile agents; in [20, 21], which narrow the problem to the planar (2D) case.

3. DISTRIBUTED CAMERA CALIBRATION USING CONSENSUS

Given a set of N cameras, the location and the orientation of the i-th camera can be specified by the absolute 3×3 rotation matrix R_i and the 3×1 location vector $\mathbf{t}_i$ with respect to a fixed reference system (*absolute rotation* and *location*). Such parameters can be expressed using their relative counterparts R_{ij} and $\mathbf{t}_{ij}$ (*relative rotation* and *location*), i.e.,

$$R_{ij} = R_i^{-1} R_j = R_i^\top R_j \quad \text{s.t.} \quad R_j = R_i R_{ij} \tag{1}$$

and

$$\mathbf{t}_{ij} = R_i^{-1} \left(\mathbf{t}_j - \mathbf{t}_i \right). \tag{2}$$

It is interesting to note that the relative transformations are invariant to the choice of a global reference frame.

It has already been stated that the solution of the orientation estimation problem for a camera network consists in the absolute rotation reconstruction for each device in the system. Since the absolute rotations are linked to the relative ones through (**??**), the following formal definition of *oriented network* stands:

Definition 1 (Oriented network) *A visual sensors network (VSN) of N agents is said to be oriented if there is a set of relative rotations $\{R_{ij}\}$ between node i and node j such that, when the reference frame of the first node is fixed (e.g. R_1), all the other absolute rotations (R_i, $i = 2 \ldots N$) are uniquely determined.* □

The set of relative rotations $\{R_{ij}\}$ that satisfies the consistency constraints given in Definition 1 can be found following two distinct approaches: a centralized strategy or a distributed one. Clearly, while the former involves the presence of a central unit capable of communicating with all agents that constitute the network, in the second case, the problem is tackled by resorting to local computation (performed by smart cameras) that regards only local interactions between the devices. In actual facts, the approach described in [16] and followed in this work can be classified as partially distributed since the initialization phase is accomplished in a centralized manner.

In detail, the rationale behind such minimization-based solution is a generalization of the classical average consensus procedure that is extended to the case of Riemannian manifold $SO(3)$. The algorithm takes as input the noisy relative measurements $\{\widetilde{R}_{ij}\}$ between each pair of connected cameras in the graph to iteratively compute a set of relative rotations $\{\widehat{R}_{ij}\}$, consistently with the *oriented network* definition. Hence, by applying (1), the absolute rotations $\{\widehat{R}_i\}$ can be estimated for all the network devices, w.r.t. to the same world frame system.

More formally, the suggested criterion is a least squares approach, wherein the cost function φ to minimize rests upon the Riemannian metric, i.e.

$$\varphi = \sum_{i \in \mathcal{V}} \left[\frac{1}{2} \sum_{j:(i,j) \in \mathcal{E}} d^2_{SO(3)}(\widehat{R}_{ij}, \widetilde{R}_{ij}) \right], \tag{3}$$

and to satisfy the consistency constraints of Definition 1 each relative transformation $\widehat{R}_{ij}$ is reparametrized in terms of absolute rotations according to (1):

$$\begin{aligned} \varphi &= \sum_{i \in \mathcal{V}} \left[\frac{1}{2} \sum_{j:(i,j) \in \mathcal{E}} d^2_{SO(3)}(\widehat{R}_i^\top \widehat{R}_j, \widetilde{R}_{ij}) \right] \\ &= \sum_{i \in \mathcal{V}} \varphi_i(\widehat{R}_i). \end{aligned} \tag{4}$$

where $d_{SO(3)}(\cdot, \cdot)$ is the distance metric in SO(3).

With these positions, each camera has to solve the non-linear minimization of $\varphi_i(\widehat{R}_i)$ over $\widehat{R}_i \in SO(3)$, which is achieved through an iterative two steps procedure. At each iteration, every node i of the network computes the Riemannian gradient of $\varphi_i(\widehat{R}_i)$ w.r.t. $\widehat{R}_i$, $\mathrm{grad}_{\widehat{R}_i} \varphi_i(\widehat{R}_i)$ [16]. Then, if $\widehat{R}_i(t)$ denotes the estimate of R_i at the t-th iteration, $\widehat{R}_i(t+1)$ is determined by performing a gradient descent step, moving along the geodesic in the direction $-\mathrm{grad}_{\widehat{R}_i(t)} \varphi_i(\widehat{R}_i(t))$ with a properly chosen step-size ε, i.e.

$$\widehat{R}_i(t+1) = \exp_{\widehat{R}_i(t)}(-\varepsilon \, \mathrm{grad}_{\widehat{R}_i(t)} \varphi_i(\widehat{R}_i(t))). \tag{5}$$

The rotation estimate is finally communicated to the neighboring nodes. The iterative procedure stops after a pre-defined number T of iterations.

4. THE PROPOSED MODIFICATIONS

As shown in the previous section, consensus-based protocols rely on an pervasive propagation of the local information, which is assumed to lead the manifold to an optimal state. Unfortunately, this process can be significantly impaired by the presence of errors affecting the estimated relative poses [23]. The robustness of the estimation algorithm can be improved by increasing the redundancy of the information exchanged in the network, i.e., enabling multiple routes that spread the data across the different nodes. A less computationally-demanding solution implies preventing unreliable links from spreading polluted data. Departing from previous solutions, which entail a randomized message propagation to bypass unreliable connections, the current approach adopts a new quality metric to test the reliability of links, i.e., the accuracy of the relative poses between couples of cameras acquiring a common set of 3D points.

4.1. Camera-In-View

Let us assume that the relation between the poses of cameras i and j is defined by the relative orientation matrix $R_{i,j}$ and the relative translation $\mathbf{t}_{i,j}$. The proposed metric relies on the so-called "Camera-In-View" (CIV) condition: whenever the j-th camera falls within the field-of-view of the i-th camera, it is possible to relate the camera visual center $\boldsymbol{\tau}_j$ to its projection $\boldsymbol{\mu}_{i,j}$ via the following equation.

$$\boldsymbol{\mu}_{i,j} \simeq K_i \left[I | \mathbf{0} \right] \left[R_{i,j} | \mathbf{t}_{i,j} \right] \boldsymbol{\tau}_j = K_i \left[R_{i,j} | \mathbf{t}_{i,j} \right] \boldsymbol{\tau}_j \tag{6}$$

where $\boldsymbol{\tau}_j$ is defined with respect to the coordinate system of camera j. Note that we assume that $\boldsymbol{\tau}_j$ and K_i are known since they are intrinsic characteristics of the j-th device.

CIV conditions can be verified in different ways: in [24], a visual signalling protocol permits identifying the other cameras in the network, while in [25] the authors assume that a description of the mobile device carrying the camera is broadcasted at the beginning of the acquiring session. Both strategies leads to the identification of the point $\hat{\boldsymbol{\mu}}_{i,j}$ in the image acquired by camera i.

From these premises, it is possible to evaluate the correctness of the estimated $R_{i,j}$ and $\mathbf{t}_{i,j}$ by computing the distance

$$d_{ij} = \|\hat{\boldsymbol{\mu}}_{i,j} - \boldsymbol{\mu}_{i,j}\|_2 \tag{7}$$

where $\|\cdot\|^2$ denotes the Euclidean norm.

Whenever the j-th camera is in the field-of-view of the i-th camera (CIV condition) and $d_{i,j} > \epsilon_{Th}$, the link $i - j$ is considered unreliable and removed from the set of edges in the connectivity graph.

The test is run on all the edges of the graph before starting the consensus strategy. In case connectivity is lost at the end of the procedure, the erased links with the lowest $d_{i,j}$ are re-included until connectivity is restored. Then, standard consensus strategy is run on the resulting graph.

The overall performance of the proposed scheme is compared with the standard consensus strategy in the following section.

4.2. Averaging in $SO(3)$

There are plenty of ways to define the (weighted) mean of a set of rotations (see [26] for an extensive discussion on the subject). In this work, following [27], given a set of N elements in $SO(3)$, $\{R_1, \dots, R_N\}$, and a set of weights $\mathbf{w} = \{w_1, \dots, w_N\}$ s.t. $\sum_{i=1}^N w_i = 1$, we characterize the *weighted mean* of $\{R_1, \dots, R_N\}$ as

$$\bar{R}_{\mathbf{w}} = \arg \min_{R \in SO(3)} \sum_{i=1}^{N} w_i d^2_{SO(3)}(R, R_i). \tag{8}$$

It is worth to observe that, in the case $w_i = 1/N$ for all i, the previous quantity reduces to the *simple mean* of $\{R_1, \dots, R_N\}$.

A globally convergent algorithm to compute the mean of a set of rotations is presented in [28]. This method consists of two main steps:

1. the computation of the mean in the tangent space,
2. its projection back onto $SO(3)$ via exponential map.

When dealing with weighted mean, this procedure can be easily generalized, as reported in Algorithm 1.

Algorithm 1 Weighted rotation mean of $\{R_1, \dots, R_N\}$

```
1: Set R := R_1 and choose a tolerance ε > 0.
2: loop
3:     Compute r = Σ_{i=1}^N w_i log(R^T R_i)
4:     if ||r|| < ε then
5:         return R
6:     end if
7:     Update R = exp(r)
8: end loop
```

4.3. Initialization

The minimization-based algorithm leverages on the noisy relative rotations among cameras and envisages the iterative communication among nodes of the estimated absolute rotations. Given the non-convexity of the involved cost functions, in order to allow for the convergence towards a correct estimate, it is therefore necessary to initialize the matrices $\{\widehat{R}_i(0)\}$ appropriately.

This section is entirely devoted to the presentation of some initialization methods that differ for the a priori information requirement, the computational load and the robustness to measurement noise. The underlying idea shared by all these methods is that it is necessary to extract a subgraph from the VSN graph in order to assign initial values consistently with the requirement stated by Definition 1.

4.3.1. Single Spanning Tree Method

The easiest way to design an initialization strategy is the *single spanning tree method* (SST). As illustrated in Algorithm 2, it consists of three steps:

1. choose any node as a reference/root (e.g. node 1) and to impose $\widehat{R}_1(0) = I_3$;
2. find a spanning tree $\mathcal{ST}_{\mathcal{G},1}$ that provides the simple paths ℓ_{1i} from the root node to any other node i in the network;
3. set $\widehat{R}_i(0) = \widehat{R}_1(0)\widetilde{R}^\top_{\ell_{1i}}$ for all $i \in \mathcal{V}$, where $\widetilde{R}_{\ell_{1i}}$ is the relative rotations composition along the path ℓ_{1i} in the designed rooted spanning tree, (**??**).

Algorithm 2 SST

```
1: Set root = 1 and R̂_1 = I
2: Define a spanning tree ST_{G,1}
3: for i ← 2 to N do
4:     Compute R̂_i = R̂_1 R̃^T_{ℓ_1i}
5: end for
```

There is a level of arbitrariness in the choice of the reference node, which eventually may affect the reconstruction accuracy. In fact, it is important to observe that the estimated absolute rotation (and thus the error w.r.t. the true value) is obtained through a composition law similar to (1).The lack of accuracy increases with the distance of the i-th node from the reference node, and thus, it is advisable to select a spanning tree as balanced as possible, wherein the differences of the paths lengths are as small as possible.

In summary, the initialization method based on SST has the advantage of being fast, nevertheless the robustness of the whole algorithm crucially depends on the root node that must be manually selected in a centralized fashion or through a *leader election* procedure [29].

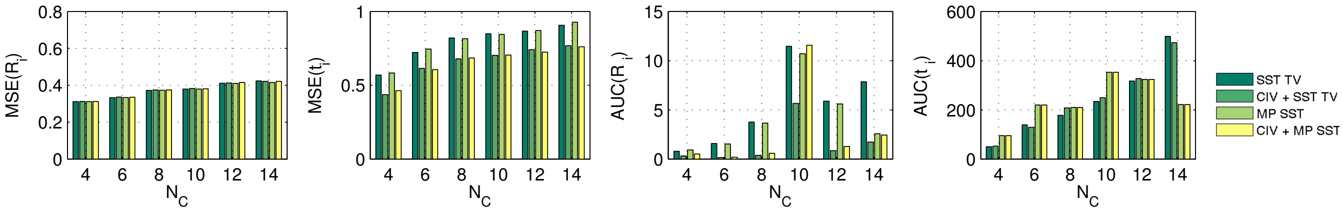
4.3.2. Multipath Method

A more complex initialization strategy that aims at reducing the arbitrariness in the choice of the root node is the *multipath method* (MP). This consists in the definition of several paths in the network and the averaging of multiple absolute rotation estimates for each camera.

With reference to Algorithm 3, the procedure starts similarly to SST approach as the root node (e.g. node 1) is fixed setting $\widehat{R}_1(0) = I$. Then, for each other node i, four steps are performed. In detail,

1. all possible m_i paths ℓ^k_{1i} ($k = 1, \dots, m_i$) from the reference node to node i are determined;
2. rotation $\widehat{R}^k_i(0) = \widehat{R}_1(0)\widetilde{R}^\top_{\ell^k_{1i}}$ is computed using the relative rotation composition rule (**??**) along the k-th path in order to obtain m_i different estimates;
3. each k-th estimate is associated to a weight w^k_i equal to the reciprocal of the k-th path length, up to a normalization factor, i.e.

$$w^k_i = \frac{1}{|\ell^k_{1i}|} \frac{1}{\sum_{k=1}^{m_i} \frac{1}{|\ell^k_{1i}|}}, \tag{9}$$



(a) (b) (c) (d)

Fig. 1. Comparison of different initialization algorithms in different camera settings. Evaluation considers the MSE on the reconstructed rotation matrices (a), location vector (b), together wit the area of the curve cost-vs-iteration for the estimation of rotation (c) and translation (d).

4. the final rotation estimate $\widehat{R}_i(0)$ is thus derived as the mean of $\{\widehat{R}_i^1(0), \dots \widehat{R}_i^{m_i}(0)\}$ weighted by $\{w_i^1, \dots, w_i^{m_i}\}$ accordingly to Algorithm 1.

Algorithm 3 MP

Set root $= 1$ and $\widehat{R}_1 = I$
for $i \leftarrow 2$ **to** N **do**
  Compute all possible ℓ_{1i}^k, $k = 1, \dots, m_i$
  for $k \leftarrow 1$ **to** m_i **do**
    Compute $\widehat{R}_i^k = \widehat{R}_1 \widetilde{R}_{\ell_{1i}^k}^\top$
    Compute $w_i^k = \frac{1}{|\ell_{1i}^k|} \frac{1}{\sum_{k=1}^{m_i} \frac{1}{|\ell_{1i}^k|}}$
  end for
  Compute the weighted mean of $\{\widehat{R}_i^1, \dots \widehat{R}_i^{m_i}\}$
end for

The main advantage of MP is that the uncertainty on the initial absolute rotation is generally reduced for two reasons. Firstly, by averaging on different rotations, a priori information about the relative transformations is better exploited. Secondly, the adoption of the weighting factors (9) allows to mainly consider the estimates computed using the shortest paths, which provide the less noisy estimates, for each device. On the other hand, the computational burden can in principle become prohibitive because of the calculation of the rotations mean and the identification of all the paths connecting two nodes (which is known to be a NP-hard problem). To avoid this latter issue, only a subset of all the possible paths is evaluated, for instance the subset of paths having fixed length $\lambda << N$. By suitably selecting the parameter λ according to the network topology, the benefits of MP are preserved, while the growth of overall computational complexity is controlled.

5. EXPERIMENTAL RESULTS

The performance of the proposed solution was evaluated in terms of estimation accuracy (both for camera poses and for the reconstructed 3D point clouds that represent the acquired scene) and computational effort. More precisely, a virtual 3D model of the scene was acquired and a set of N_C cameras has been randomly placed around it. For every set of N_C cameras, we considered 10 different random settings in order to obtain an averaged performance. Each camera in the network is connected to the two closest cameras, together with all the cameras that fall within its field-of-view: for every couple of connected cameras, conjugate points are identified and used to estimate the relative rotation $R_{i,j}$ and translation $\mathbf{t}_{i,j}$. At the beginning of the message propagation algorithm, a spanning-tree based propagation of the extrinsic parameter values propagate the initial absolute pose and orientation through the camera network (all the absolute rotations and poses are initialized with the identity matrix I and the array $\mathbf{0}$). Then, the consensus-based strategy described in [15] is run on the network.

Performances were evaluated measuring the average Mean Square Error (MSE) between the estimated orientations $\hat{R}_i$ and the real ones R_j, together with the MSE between estimated locations $\hat{\mathbf{t}}_i$ and the real ones $\mathbf{t}_i$. Accuracy of the reconstruction is parameterized by the MSE between the reconstructed 3D point cloud $\hat{\mathbf{P}}_k$ and the ground truth $\mathbf{P}_k$. Moreover, algorithm speed has been parameterized by the Area Under the Curve (AUC) of the cost-vs-iteration plot. This parameter was evaluated for the estimation cost plots of both $\hat{R}_i$ and $\hat{\mathbf{t}}_i$. A smaller area value denotes a faster convergence speed.

Figure 1 reports the average MSE(R), MSE($\mathbf{t}$), MSE($\mathbf{P}$), AUC(R), and AUC($\mathbf{t}$) for different camera networks with increasing N_C. We considered two different initializations (SST and MP) to the Tron-Vidal (TV) algorithm, eventually combined with Camera-In-View (CIV) link test. As a result, we had 4 different combinations which are erported in the legend of Fig. 1.

It is possible to notice that the both the MP initialization and the CIV test permit improving the accuracy for the standard algorithm (which corresponds to the label SST + TV) [15]. No significant differences can be appreciated for camera orientations (see Fig. 1 (a)). Moreover, the computational speed is significantly improved since the are of the cost curves is significantly lower than that of Tron-Vidal solution (see Fig. 1 (c) and (d)). It is also possible to infer that MP strategies permits reducing the convergence speed without affecting the final accuracy significantly. Combined with CIV pruning, both advantages can be combined reducing the estimation time and refining the parameter values.

6. CONCLUSIONS

This work concerns the orientation estimation problem for a N-camera network, which consists in the reconstruction of the absolute rotation and location of each device in the system, given the availability of noisy relative measurements. The proposed solutions adopts a graph traversal initialization strategy, together with a link pruning algorithm that aims at removing the noisiest relative measurements. Experimental results show that the proposed approach improves both in terms of final accuracy and convergence speed allowing a more accurate calibration of cameras, together with a reduced computational complexity. Future work will be devoted to testing the tracking ability of such system and test it in a multicamera set-up defined by a set of drones.

7. REFERENCES

[1] T. Nageli, C. Conte, A. Domahidi, M. Morari, and O. Hilliges, “Environment-independent formation flight for micro aerial vehicles,” in *Proc. of IEEE IROS 2014*, Sept 2014, pp. 1141–1146.

[2] F. Poiesi and A. Cavallaro, “Self-positioning of a team of flying smart cameras,” in *Proc. of IEEE ISSNIP 2015*, April 2015, pp. 1–6.

[3] A. Kushleyev, D. Mellinger, and C. Powers, “Towards a swarm of agile micro quadrotors,” *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, Nov. 2013.

[4] S. N. Sinha and M. Pollefeys, “Visual-hull reconstruction from uncalibrated and unsynchronized video streams,” in *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, Sept 2004, pp. 349–356.

[5] S. Katragadda and A. Cavallaro, “Neighbour consensus for distributed visual tracking,” in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, April 2015, pp. 1–6.

[6] Y. Wang, S. Velipasalar, and M. Casares, “Cooperative object tracking and composite event detection with wireless embedded smart cameras,” *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2614–2633, Oct 2010.

[7] Xue Wang, Sheng Wang, Dao-Wei Bi, and Jun-Jie Ma, “Distributed peer-to-peer target tracking in wireless sensor networks,” *Sensors*, vol. 7, no. 6, pp. 1001–1027, 2007.

[8] M. Taj and A. Cavallaro, “Distributed and decentralized multi-camera tracking,” *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 46–58, May 2011.

[9] F. Garin and L. Schenato, “A survey on distributed estimation and control applications using linear consensus algorithms,” *Networked Control Systems, Springer Lecture Notes in Control and Information Sciences*, vol. 406, pp. 75–107, 2011.

[10] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, Jan 2007.

[11] N. Sadeghzadeh. N and A. Afshar, “Consensus-based distributed particle filters in sensor networks,” in *2009 Chinese Control and Decision Conference*, June 2009, pp. 4333–4338.

[12] Alain Sarlette and Rodolphe Sepulchre, “Consensus optimization on manifolds,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 56–76, 2009.

[13] P-A Absil, Robert Mahony, and Rodolphe Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.

[14] R. Tron, B. Afsari, and R.Vidal, “Riemannian Consensus for Manifolds With Bounded Curvature,” *Automatic Control, IEEE Transactions on*, vol. 58, no. 4, pp. 921–934, Apr. 2013.

[15] R. Tron and R. Vidal, “Distributed Image-Based 3D Localization of Camera Sensor Networks,” in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, Dec. 2009, pp. 901–908.

[16] Roberto Tron and Rene Vidal, “Distributed 3-D localization of camera sensor networks from 2-D image measurements,” *Automatic Control, IEEE Transactions on*, vol. 59, no. 12, pp. 3325–3340, 2014.

[17] Joseph Knuth and Prabir Barooah, “Maximum-likelihood localization of a camera network from heterogeneous relative measurements,” in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 2374–2379.

[18] Joseph Knuth and Prabir Barooah, “Collaborative 3D localization of robots from relative pose measurements using gradient descent on manifolds,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1101–1106.

[19] Joseph Knuth and Prabir Barooah, “Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1534–1539.

[20] Domenica Borra, Enrico Lovisari, Raffaele Carli, Fabio Fagnani, and Sandro Zampieri, “Autonomous calibration algorithms for networks of cameras,” in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 5126–5131.

[21] Giulia Piovan, Iman Shames, Barış Fidan, Francesco Bullo, and Brian DO Anderson, “On frame and orientation localization for relative sensing networks,” *Automatica*, vol. 49, no. 1, pp. 206–213, 2013.

[22] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry, *An invitation to 3-d vision: from images to geometric models*, vol. 26, Springer Science & Business Media, 2012.

[23] Lorenzo Alvisi, Jeroen Doumen, Rachid Guerraoui, Boris Koldehofe, Harry Li, Robbert van Renesse, and Gilles Tredan, “How robust are gossip-based communication protocols?,” *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 5, pp. 14–18, Oct. 2007.

[24] S. Milani, “Three-dimensional reconstruction from heterogeneous video devices with camera-in-view information,” in *Proc. of ICIP 2015*, Sept. 2015, pp. 2050–2054.

[25] S. Milani and A. Memo, “Impact of drone swarm formations in 3d scene reconstruction,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 2598–2602.

[26] R. Hartley, J. Trumpf, Y. Dai, and H. Li, “Rotation Averaging,” *International Journal of Computer Vision*, vol. 103, no. 3, pp. 267–305, July 2013.

[27] Maher Moakher, “Means and averaging in the group of rotations,” *SIAM journal on matrix analysis and applications*, vol. 24, no. 1, pp. 1–16, 2002.

[28] J. H. Manton, “A globally convergent numerical algorithm for computing the centre of mass on compact Lie groups,” in *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, Dec. 2004, vol. 3, pp. 2211 – 2216.

[29] Nancy A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.