Maximum-likelihood localization of a camera network from heterogeneous relative measurements

Joseph Knuth and Prabir Barooah

***Abstract*— This paper proposes an algorithm for estimating the absolute pose (position and orientation) of n cameras using relative measurements between pairs of cameras. Our work is inspired by the recent work [1] where the same problem was considered and a distributed algorithm was proposed. In contrast to [1], which fused relative measurements of orientation and bearing between camera pairs, and produced a least squares estimate, we make two novel contributions. First, our algorithm is capable of fusing any type of relative measurement between cameras: relative orientation, relative position, relative bearing, or relative distance, or any combination thereof. Second, the algorithm determines a maximum likelihood estimate of the camera poses when the measurement noises distributions are Gaussian-like in their corresponding Riemannian manifolds. A gradient descent method on the product manifold $(SO(3) \times R^3)^n$ is used to compute the estimates. Unlike past probabilistic techniques, our assumed distribution for measurement noise on orientation and bearings are defined on the natural manifolds rather then any parameterization. Though the proposed algorithm is centralized in its computation, we discuss how the computations can be distributed among the cameras. Performance of the proposed algorithm is examined through simulations. Comparison with the algorithm in [1] with non-uniform sensor accuracy reveals which algorithm is most appropriate for a given scenario.**

I. INTRODUCTION

A camera network consists of a number of cameras that are geographically distributed such that the images recorded by the cameras are used for a common sensing task. Such networks are useful in a number of applications, such as surveillance of public places for security reasons [2], environmental monitoring of hazardous or remote areas [3], and tracking occupant movement inside smart buildings [4]. The cameras are capable of communicating with one another or a central processor through a communication network.

Images captured by a camera network for the purposes of tracking or localizing objects are only of use if the network is localized. *Localizing* a camera network is to determine all camera poses, i.e., positions and orientations, relative to a common coordinate frame. Localization is also sometimes referred to as calibration or external calibration. With a localized camera network, a common target observed by multiple cameras at distinct time instants can be tracked by fusing measurements provided by the cameras. In fact, localization is required simply to detect when two cameras observe the same target. Manual localization is often inaccurate. Even if positions of cameras can be accurately measured manually, measuring orientations in such a fashion is highly error prone. Automated localization is therefore highly desirable, in which camera network localization is performed by refining initial, perhaps very noisy, estimates using measurements collected by the cameras.

This paper is concerned with automated localization of a static camera networks when the pose of cameras do not change with time. Our work is inspired by the paper [1], in which the authors consider the problem of ensuring globally consistent pose estimates when relative measurements between cameras are available. The papers [5], [6] also consider the same problem in the planar (2-D) case, in which the pose of each camera can be described by three scalars: x, y for the position coordinates and θ for orientation. The question of localizability of planar camera networks with noise free measurements of various types (bearing, distance, etc.) is partially addressed in [5]. The paper [6] proposes two distributed algorithms to compute a least-squares estimate of the camera poses. Compared to the 2-D case examined in these papers, the 3-D case is more practically relevant and considerably more complex. Existing work on 3-D camera calibration has therefore focused on estimation algorithms that lead to some estimates with noisy measurements, even if strong performance guarantees on the estimates obtained cannot be provided.

In [1], a method for 3-D camera network localization is proposed that fuses relative orientation and relative bearing measurements between pairs of cameras in an n camera network. A nonlinear cost function defined on a Riemannian manifold is minimized using gradient descent. The paper [7] proposes a method for distributed 3-D camera network calibration by using belief propagation. The work by [1] deviates from prior algorithmic work on camera network localization in one major aspect. Most of the prior algorithms, such as that in [7], use a minimal parameterization of the orientations (such as Euler angles) which is then used to pose the estimation problem as a vector-space optimization problem. In contrast, the algorithm in [1] does not rely on any parameterization of the orientation. Instead, the cost function whose minimization provides the estimate is defined in the natural space of the problem: in a product Riemannian manifold $(SO(3) \times \mathbb{R}^3)^n$, where $SO(3)$ is the 3-D rotation group.

In this paper we extend the work of [1] on 3-D localization in two ways. First, our proposed algorithm can fuse *heterogeneous* relative measurements between pairs of cameras to improve pose estimates, while the existing work requires that each relative measurement be of the same kind: relative orientation and bearing in [1] and relative pose (orientation and position) in [7]. Obtaining measurements of the relative orientation is extremely difficult in practice, while obtain-

ing other types of relative measurements, such as bearing or distance between two cameras, is usually easier. This makes the proposed method less restrictive than those in [1], [7]. Second, while [1] provides no statistical performance guarantees, we provide the ML estimates for a particular distribution of the measurement noise. The algorithm in [7] in fact computes the MAP (maximum a-posteriori) estimate if measurements, when parameterized to lie in vector spaces, are Normally distributed. In contrast, the distributions we assume are defined on the manifolds that the measurements naturally belong to - the 3D rotation group for orientations and the 2-sphere for bearing - and not on any specific parameterization. Even if the distribution of a measurement that lies in a manifold is Gaussian-like (in the sense that it satisfies the central limit theorem, does not have a heavy tail, etc.), the distributions of the entries of a specific minimal parameterization of the measurement need not be Gaussian. Each parameterization will have a distinct distribution.

Choosing appropriate distributions is critical for maximum likelihood estimation. To this end we propose plausible distributions for each measurement type. Both position and distance measurements are assumed Normally distributed. However, orientation and bearing measurements are not elements of any vector space. Thus no perfect analog to the Gaussian distribution for these cases exist. Therefore we choose Gaussian-like distributions on the respective manifolds. Specifically, bearing measurements are assumed to be Von Mises-Fisher (VMF) distributed. We say the VMF distribution is "Gaussian-like" due the following fact. Just as the Normal distribution is the equilibrium distribution for the Ornstein-Uhlenbeck process in vector spaces, the VMF distribution is the equilibrium distribution of the analogue of the Ornstein-Uhlenbeck process on the sphere [8]. Orientation measurements are chosen to follow the wrapped Gaussian (WG) distribution on $SO(3)$. Though there is still much to learn about this distribution, it is known that the corresponding WG distribution on $SO(2)$ is the solution to the heat equation [9]. The Normal distribution solves the heat equation on a vector space, and so we say the WG distribution is "Gaussian-like."

The proposed algorithm, which is called the ML-CL (maximum likelihood collaborative localization) algorithm, computes the ML estimates by optimization over a Riemannian manifold. Since ML-CL computes the ML estimates, information on varying levels of accuracy of the measurements are taken into account in a principled way. Simulation comparisons with the algorithm in [1] reported here shows how this can be useful. When all the relative measurements have the same noise level, the accuracy of the estimated provided by the ML-CL algorithm are close to that provided by the algorithm in [1]. However, when some of the measurements are known to be quite accurate or inaccurate compared to others, the estimates obtained by the ML-CL algorithm are much more accurate than those by the algorithm in [1].

We only consider the centralized computation case here. Since a static network of cameras is considered where the camera poses do not change with time, pose estimates need to be computed only once. In such a setting, centralized computation is quite feasible. We do discuss how a distributed algorithm can be developed if desired. The scheme is distributed in the sense that each camera can estimate its own pose by iteratively updating its estimate by communicating with a set of "neighboring" cameras.

II. Problem Statement

We wish to determine the pose (position and orientation) of each camera in an n-camera network with respect to a common reference frame, which is called the *absolute reference frame*. We will call such a pose that camera's *absolute pose*. Relative measurements between certain pairs of cameras are available. Such an *inter-camera relative measurement*, say from camera **A** to camera **B**, can of one of the following types:

- **Relative orientation**: The element of the special orthogonal group $SO(3)$ that describes the change in orientation between camera **A** and camera **B**, as viewed from camera **A**'s reference frame. Denoted by the symbol $\mathbf{R}$.
- **Relative position**: The vector in $\mathbb{R}^3$ that describes the change in position between camera **A** and camera **B**, as viewed from camera **A**'s reference frame. Denoted by the symbol $\mathbf{t}$.
- **Relative bearing**: The vector of unit length that points from camera **A**'s position to camera **B**'s position, as viewed from camera **A**'s reference frame. Denoted by the symbol $\boldsymbol{\tau}$.
- **Relative distance**: The distance between camera **A** and camera **B**. Denoted by the symbol δ.

These measurements can be obtained in various ways. If one camera is in the field of view of the other, bearing between them can be easily measured. The distance between them can also be measured if the cameras have targets (tags) of known geometry attached to them. Such methods are used in [10]. In fact, if cameras have targets of known geometry attached to them, the position and/or orientation with respect to the other can also be measured. Distance can also be measured by using radio-frequency (RF) techniques that measure Time of Arrival or Received Signal Strength between the wireless radios attached to the cameras [11]. RF techniques that can measure angle of arrival, such as that in [12] can be used to measure bearing . The more common method of obtaining relative measurements is to detect common feature points in the images collected by a pair of cameras that have overlapping field of view; see [13], [14], [15] for discussion on such methods.

The situation above is best described by a directed fully-labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \ell)$ that shows how the inter-camera relative measurements relate to the absolute pose of each camera. The graph is defined as follows. Each of the n cameras is assigned a unique integer from the set $\mathcal{V} = \{1, \ldots, n\}$. The *node set* for the graph $\mathcal{G}$ is then give by the set $\mathcal{V}_0 = \mathcal{V} \cup \{0\}$ where $0 \in \mathcal{V}_0$ corresponds to the absolute reference frame. The absolute reference frame might

be the reference frame of one of the cameras. The camera associated with node $i \in \mathcal{V}$ will be referred to as *camera* i. Associated with each node $i \in \mathcal{V}$ are the two *node variables* $\mathbf{R}_i \in SO(3)$, the orientation of camera i, and $\mathbf{t}_i \in \mathbb{R}^3$, the position of camera i. Each node variable is given with respect to the absolute reference frame. The problem of localization of the camera network is equivalent to estimating each of the node variables.

The set of directed edges, denoted $\mathcal{E}$, correspond to the inter-camera relative measurements. That is, suppose a measurement from camera i to camera j is available, then there exists an edge $e = (i, j) \in \mathcal{E}$. To delineate the type of measurement, a label from the set $\{$ $\mathbf{R}$ (orientation), $\mathbf{t}$ (position), $\boldsymbol{\tau}$ (bearing), δ (distance) $\}$ is attached to each edge. The map from the set of edges to the set of labels is given by ℓ. Thus if the measurement $e = (i, j)$ between camera i and j is of their relative orientation, then $\ell(e) = \mathbf{R}$. If several different types of relative measurements are available between a camera pair, that is represented by multiple parallel edges, each with one type of measurement.

In certain cases, the absolute camera poses are indeterminate upto a rotation or a translation. For instance, if only relative orientation measurements between nodes in $\mathcal{V}$ are available, absolute orientations can't be determined without ambiguity. We assume that each camera has an initial guess of its absolute pose, even if highly inaccurate. This can be obtained by manual measurements once during deployment. In case the absolute poses are indeterminate, fusing them with the initial guess is likely to lead to a better estimate.

We would like to utilize the information contained in the initial guess of the pose of each camera. To this end, we add two additional edges to our graph for each node, corresponding to each initial estimated node variable. The new edge set is given by $\mathcal{E}_0 = \mathcal{E} \cup \{e_{\mathbf{R}i} = (0, i), e_{\mathbf{t}i} = (0, i) \,|\, i \in \mathcal{V}\}$. Where $\ell(e_{\mathbf{R}i}) = \mathbf{R}$ (orientation) and $\ell(e_{\mathbf{t}i}) = \mathbf{t}$ (position) for all $i \in \mathcal{V}$.

The graph $\mathcal{G} = (\mathcal{V}_0, \mathcal{E}_0, \ell)$ is called the *measurement graph*. Figure 1 shows an example of the graph corresponding to a network of 4 cameras. For each $e = (i, j) \in \mathcal{E}_0$, $\mathbf{M}_e$ is a measurement of the relative orientation, position, bearing, or distance, depending on the value of $\ell(e)$, between nodes i and j. Let $\{\mathbf{M}\}_{\mathcal{E}_0} = \{\mathbf{M}_e \,|\, e \in \mathcal{E}_0\}$ be the set of all noisy inter-camera relative measurements, which from now on includes the initial guess for each camera pose.

Let $\{(\mathbf{R}, \mathbf{t})\}_{\mathcal{V}} = \{(\mathbf{R}_i, \mathbf{t}_i) \,|\, i \in \mathcal{V}\}$ denote the set of all node variables. *Our goal is to determine the most likely value of* $\{(\mathbf{R}, \mathbf{t})\}_{\mathcal{V}}$ *given the noisy measurements* $\{\mathbf{M}\}_{\mathcal{E}_0}$ *for appropriate models of the measurement noise in terms of their probability density functions (pdfs).*

In this paper $\mathbf{R} \in SO(3)$ is to be understood as a linear operator from the Euclidean space $\mathbb{R}^3$ (with the 2-norm) to itself that preserves the length of vectors and the orientation of the space. It is not to be understood as 3×3 rotation matrices or any other representation of 3D rotations. Numerical implementation of the algorithm we propose in the next section can be performed with any representation of $SO(3)$, such as rotation matrices or unit quaternions.

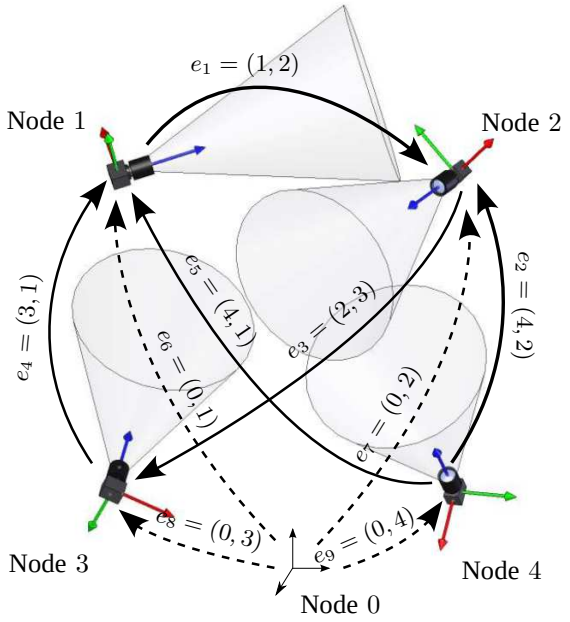


Fig. 1. A camera network consisting of 4 cameras shown as the corresponding graph. The sold lines indicate edges from $\mathcal{E}$ (inter-camera measurements) while the dashed lines indicate edges from $\mathcal{E}_0 \setminus \mathcal{E}$ (initial pose guesses). Node 0 represents the absolute reference frame in which the pose of each camera is to be estimated. Usually node 0 is simply the frame of one of the cameras.

III. The ML estimates

We assume that measurement on distinct edges are statistically independent, so that the joint pdf of all the measurements satisfies:

$$p(\{\mathbf{M}\}_{\mathcal{E}_0}) = \prod_{e \in \mathcal{E}_0} p_e(\mathbf{M}_e), \tag{1}$$

where $p_e(\,\cdot\,)$ is the pdf of the measurement on edge e. Since there are four types of measurements on the edges, we need four classes of pdfs, which are denoted by $p_{\mathbf{R}}$, $p_{\mathbf{t}}$, $d_{\boldsymbol{\tau}}$ and p_δ, for measurement of orientation, position, bearing and distance, respectively. Choosing appropriate pdfs for orientations and bearing measurements is challenging since these densities are not defined over vector spaces but over curved surfaces. In particular, the density $p_{\mathbf{R}}$ is defined over $SO(3)$ and the function $p_{\boldsymbol{\tau}}$ is defined over $\mathbb{S}^2$, the 2-sphere (unit sphere in $\mathbb{R}^3$ with Euclidean norm). We assume that each relative orientation measurement $\hat{\mathbf{R}}_{ij}$ comes from a *wrapped Gaussian distribution* on $SO(3)$ with mean $\mathbf{R}_i^T \mathbf{R}_j$ and covariance matrix $\sigma^2 I$. The density function $p_{\mathbf{R}} : SO(3) \to \mathbb{R}^+$ is given by

$$p_{\mathbf{R}}(\hat{\mathbf{R}}_{ij}) = K_R \sum_{k=-\infty}^{\infty} \exp\Big(-\frac{1}{2\sigma^2} \big(d(\hat{\mathbf{R}}_{ij}, \mathbf{R}_i^T \mathbf{R}_j) - 2\pi k\big)^2 \Big) \tag{2}$$

for appropriate normalizing constant $K_R(\sigma)$ [9]. Here the distance function $d(\,\cdot\,,\,\cdot\,)$ in $SO(3)$ is given by the Rieman-

nian distance

$$d(A,B) = \sqrt{-\frac{1}{2}\operatorname{Tr}\left(\log^2(A^TB)\right)}, \quad A,B \in SO(3). \quad (3)$$

This density has been proposed in [9] as an extension of the Normal distribution in Euclidean spaces to $SO(3)$.

For bearing measurements, we choose the Von Mises-Fisher distribution [16]. This is a well-known density in the literature on directional statistics and considered a close analog of the Gaussian density in the d-Sphere. Specifically, each relative bearing measurement $\hat{\boldsymbol{\tau}}_{ij}$ is assumed to be distributed according to the Von Mises-Fisher distribution with mean direction $\mu_{\boldsymbol{\tau}} := \frac{\mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)}{\|\mathbf{t}_j - \mathbf{t}_i\|}$ and concentration parameter k_e. The density function $p_{\boldsymbol{\tau}} : S^2 \to \mathbb{R}^+$ is given by

$$p_{\boldsymbol{\tau}}(\hat{\boldsymbol{\tau}}_{ij}) = K_{\boldsymbol{\tau}} \exp\Big(\frac{k_e}{\|\mathbf{t}_j - \mathbf{t}_i\|}(\mathbf{t}_j - \mathbf{t}_i)^T\mathbf{R}_j\hat{\boldsymbol{\tau}}_{ij}\Big) \quad (4)$$

for appropriate normalization constant $K_{\boldsymbol{\tau}}(k_e)$.

Each relative position measurement $\hat{\mathbf{t}}_{ij}$ is assumed to be multivariate normal with mean $\mu_{\mathbf{t}} := \mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)$ and covariance matrix Σ_e. The density function $p_{\mathbf{t}} : \mathbb{R}^3 \to \mathbb{R}^+$ is given by

$$p_{\mathbf{t}}(\hat{\mathbf{t}}_{ij}) = K_{\mathbf{t}} \exp\Big(-\frac{1}{2}(\hat{\mathbf{t}}_{ij} - \mu_{\mathbf{t}})^T\Sigma_e^{-1}(\hat{\mathbf{t}}_{ij} - \mu_{\mathbf{t}})\Big) \quad (5)$$

for appropriate normalization constant $K_{\mathbf{t}}(\Sigma_e)$.

Finally, each relative distance measurement $\hat{\delta}_{ij}$ is assumed Normally distributed with mean $\|\mathbf{t}_j - \mathbf{t}_i\|$ and variance σ_e^2. The density function $p_\delta : \mathbb{R} \to \mathbb{R}^+$ is given by

$$p_\delta(\hat{\delta}_{ij}) = K_\delta \exp\Big(\frac{-(\hat{\delta}_{ij} - \|\mathbf{t}_j - \mathbf{t}_i\|)^2}{2\sigma_e^2}\Big) \quad (6)$$

for appropriate normalizing constant $K_\delta(\sigma_e)$.

Among these distributions, the wrapped Gaussian is the most cumbersome due to the infinite series in its definition. We therefore approximate $p_{\mathbf{R}}$ by the function

$$\bar{p}_{\mathbf{R}}(\hat{\mathbf{R}}_{ij}) = K_R \exp\big(-\frac{1}{2\sigma^2}d^2(\hat{\mathbf{R}}_{ij}, \mathbf{R}_i^T\mathbf{R}_j)^2\big). \quad (7)$$

Note that $\bar{p}_{\mathbf{R}}$ is not a probability density function. In the context of maximum likelihood estimation, however, there is no need for $\bar{p}_{\mathbf{R}}$ to be a pdf as it is not meant to describe a distribution, only closely approximate the function $f_{\mathbf{R}}$ that does.

To justify the approximation of $p_{\mathbf{R}}$ by $\bar{p}_{\mathbf{R}}$, the 1-norm of $p_{\mathbf{R}} - \bar{p}_{\mathbf{R}}$ was computed using Monte-Carlo integration with $100,000$ samples. The value of σ_e was in the range $[0^+, 2]$. The results are reported in figure 2. For $\sigma < 0.7$ radians, the norm of the difference is near enough to zero to be indistinguishable.

We therefore conclude that the approximation $\bar{p}_{\mathbf{R}}$ of the wrapped Gaussian distribution $p_{\mathbf{R}}$ is quite accurate for values of $\sigma < 0.7$.

We are now ready to characterize the ML estimate of the camera poses, which is given in the next proposition.

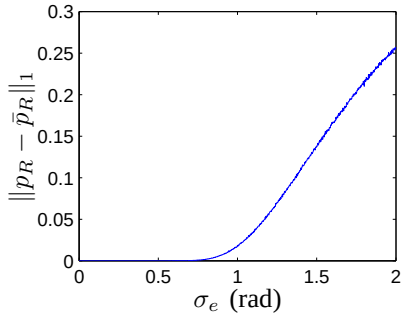


Fig. 2. The magnitude of the difference between the pdf for orientation measurements, $p_{\mathbf{R}}$, and its approximation $\bar{p}_{\mathbf{R}}$.

Proposition 1: An approximation of the maximum likelihood ML estimate $\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}$ of the node variables $\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}$ based on the measurements $\{\mathbf{M}\}_{\mathcal{E}_0}$ is given by

$$\{(\hat{\mathbf{R}},\hat{\mathbf{t}})\}_{\mathcal{V}} = \underset{\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}} \in (SO(3)\times\mathbb{R})^{|\mathcal{V}|}}{\arg\min} f(\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}) \quad (8)$$

where $f : (SO(3)\times\mathbb{R})^{|\mathcal{V}|} \to \mathbb{R}$ is a cost function given by

$$f(\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}) := \sum_{(i,j)=e\in\mathcal{E}_0} g_e(\mathbf{R}_i,\mathbf{t}_i,\mathbf{R}_j,\mathbf{t}_j) \quad (9)$$

in which $g_e(\mathbf{R}_i,\mathbf{t}_i,\mathbf{R}_j,\mathbf{t}_j)$ is the cost for edge e defined as

$$g_e(\mathbf{R}_i,\mathbf{t}_i,\mathbf{R}_j,\mathbf{t}_j) = \begin{cases} \frac{1}{2\sigma_e^2}d^2(\hat{\mathbf{R}}_{ij},\mathbf{R}_i^T\mathbf{R}_j) & \text{if } \ell(e)=\mathbf{R} \\ \frac{1}{2}\Big((\hat{\mathbf{t}}_{ij} - \mathbf{R}_i^T(\mathbf{t}_j-\mathbf{t}_i)) \\ \quad\times\Sigma_e^{-1}(\hat{\mathbf{t}}_{ij} - \mathbf{R}_i^T(\mathbf{t}_j-\mathbf{t}_i))\Big) & \text{if } \ell(e)=\mathbf{t} \\ \frac{-k_e}{\|\mathbf{t}_j-\mathbf{t}_i\|}(\mathbf{t}_j-\mathbf{t}_i)^T\mathbf{R}_i\hat{\boldsymbol{\tau}}_{ij} & \text{if } \ell(e)=\boldsymbol{\tau} \\ \frac{1}{2\sigma_e^2}\big(\hat{\delta}_{ij} - \|\mathbf{t}_j-\mathbf{t}_i\|\big)^2 & \text{if } \ell(e)=\delta \end{cases} \quad (10)$$

Proof: We rewrite (1) as

$$p(\{\mathbf{M}\}_{\mathcal{E}_0}|\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}) = \prod_{e\in\mathcal{E}_0} p_e(\mathbf{M}_e \,|\, \{(\mathbf{R},\mathbf{t})\}_{\mathcal{E}_0})$$

where the dependency on the unknown parameters $\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}$ is shown clearly. The likelihood function $L(\,\cdot\,)$ is the density viewed as a function of the unknown parameters. The log-likelihood function $\log L$ satisfies

$$\log L(\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}} \,|\, \{\mathbf{M}\}_{\mathcal{E}_0}) = \log\Big(p(\{\mathbf{M}\}_{\mathcal{E}_0}|\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}})\Big)$$
$$\propto \log(\prod_{e\in\mathcal{E}_0} \operatorname{Ker}\big(p_e(\mathbf{M}_e \,|\, \{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}})\big)$$

where $\operatorname{Ker} p_{\mathbf{M}}$ is the kernel of the corresponding pdf. The max-likelihood estimate is obtained by maximizing the right hand side of the relation above. When we use the approximation $\bar{p}_{\mathbf{R}}$ instead of $p_{\mathbf{R}}$, it turns out that that the right hand side is equal to $-f(\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}})$, were f is as defined in (9). The corresponding (approximate) maximum likelihood estimate for $\{(\mathbf{R},\mathbf{t})\}_{\mathcal{V}}$ given $\{\mathbf{M}\}_{\mathcal{E}_0}$ is computed by minimizing f. This estimate is not strictly equal to the maximum likelihood

estimate because of the approximation of $f_{\mathbf{R}}$ by $\bar{f}_{\mathbf{R}}$. Since the approximation is quite accurate for $\sigma < 0.7$, we expect the estimate obtained to be a close approximation of the ML estimate for $\sigma < 0.7$. ■

A. Computing the Estimate

To compute the ML estimate, we have to solve the optimization problem (8). Finding the minimum of a function defined over a vector space has been studied extensively. However the function $f(\cdot)$ in (9) is defined on a curved surface, specifically, the product *Riemannian Manifold* $(SO(3) \times \mathbb{R}^3)^n$. One option for this optimization is to use a parameterization of the rotations using, say, 3×3 rotation matrices or unit quaternions, and then embedding the manifold in an vector space of higher dimension. Optimization techniques applicable to vector spaces can then be used, with the constraints on the parameterization of rotations appearing as Lagrange multipliers. This however, leads to an increase in the dimensionality of the problem. In addition, since the constraints are non-linear equality constraints, the problem becomes non-convex. Any local convexity the original problem might possess is lost in this transformation. We therefore search for the minima directly on the product manifold by employing manifold optimization techniques from [17], in particular, a gradient descent method.

Given a smooth real valued function f defined on a manifold M, the gradient of f at $q \in M$, denoted $grad\, f(q)$, is a vector in the tangent space of M at q, denoted T_qM. Just as in Euclidean Space, $grad\, f(q)$ points in the direction of greatest rate of increase of f. Using linearity of the gradient operator, finding the gradient of f defined in (9) reduces to finding the gradient of the edge cost g_e defined in (10) for each of the 4 measurement types. The gradient at a point $q = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n) \in \big(SO(3) \times \mathbb{R}^3\big)^n$ is

$$\begin{aligned} grad\ g_e(q) = \Big(&grad\ g_e(\mathbf{R}_1), grad\ g_e(\mathbf{t}_1), \ldots, \\ &grad\ g_e(\mathbf{R}_n), grad\ g_e(\mathbf{t}_n)\Big) \end{aligned} \tag{11}$$

As an illustrative example, the gradients $grad g_e(\mathbf{R}_k), grad g_e(\mathbf{t}_k)$ when $e = (i,j)$ corresponds to an orientation measurement $\hat{\mathbf{R}}_{ij}$ are given by

$$grad\, g_e(\mathbf{R}_k) = \begin{cases} -\frac{1}{\sigma_e^2}\mathbf{R}_k \log(\mathbf{R}_k^T \mathbf{R}_j \hat{\mathbf{R}}_{ij}^T) & \text{if } k = i \\ -\frac{1}{\sigma_e^2}\mathbf{R}_k \log(\mathbf{R}_k^T \mathbf{R}_i \hat{\mathbf{R}}_{ij}) & \text{if } k = j \\ 0 & \text{o.w.} \end{cases}$$

$$grad\ g_e(\mathbf{t}_k) = 0.$$

We omit the expression for the all the gradients due to lack of space; but they can be computed in a similar manner.

Minimizing a function f using gradient descent requires that the current estimate be updated during each iteration by moving in the direction of $-\eta\, grad\, f$ for some appropriate scalar (*step size*) η. On a Riemannian manifold, this requires the notion of *parallel transport*. The parallel transport map at a point $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n) \in (SO(3) \times \mathbb{R}^3)^n$, denoted by $\exp_p$, is given by $\exp_p(\xi) = (\mathbf{R}_1 \exp(\mathbf{R}_1^T \xi_{\mathbf{R}_1}), \mathbf{t}_1 + \xi_{\mathbf{t}_1}, \ldots, \mathbf{R}_n \exp(\mathbf{R}_n^T \xi_{\mathbf{R}_n}), \mathbf{t}_n + \xi_{\mathbf{t}_n})$ where $\xi = (\xi_{\mathbf{R}_1}, \xi_{\mathbf{t}_1}, \ldots, \xi_{\mathbf{R}_n}, \xi_{\mathbf{t}_n})$ is an element of the *tangent space* $T_p\big[(SO(3) \times \mathbb{R}^3)^n\big] = T_{\mathbf{R}_1} SO(3) \times \cdots \times T_{\mathbf{t}_n} \mathbb{R}^3$, and the $\exp(\cdot)$ function appearing in the right hand side of the above equation is the Lie-group exponential map [18]. The gradient descent update law is then given by

$$q_{t+1} = \exp_{q_t}(-\eta_t grad\, f(q_t)), \quad t = 0, 1, \ldots. \tag{12}$$

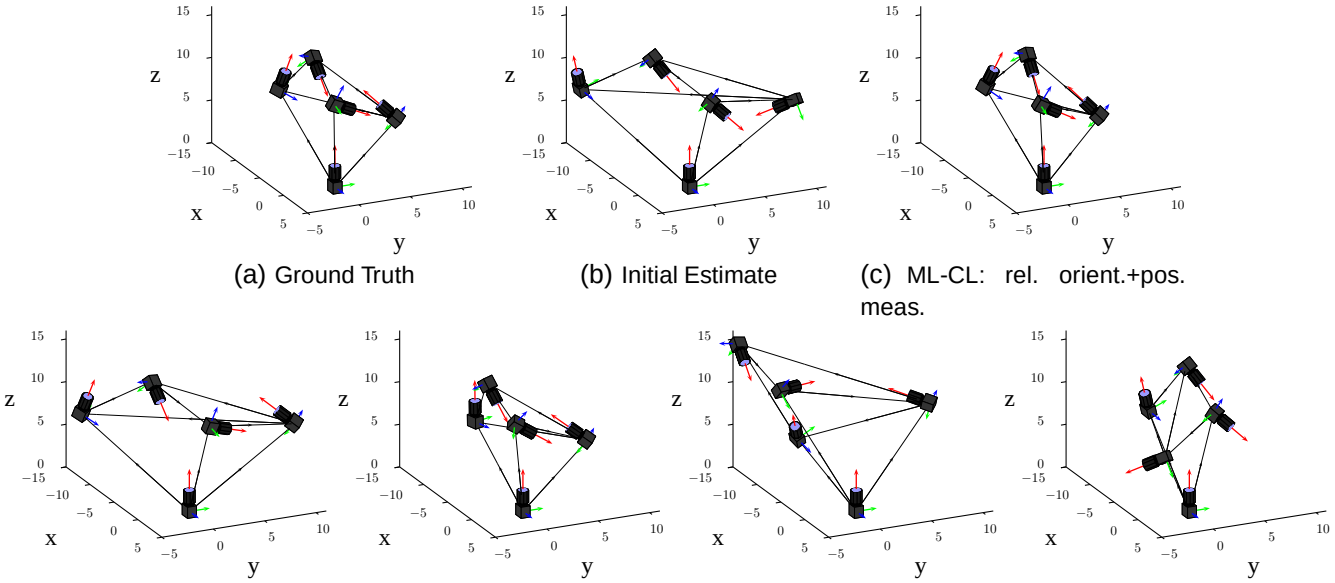
The step-size η_t is chosen by a backtracking line search (the Armijo step). Using the update law given in (12), a gradient descent is performed, terminating when the norm of the gradient falls below some user specified threshold. Theorem 4.3.1 in [17] guarantee that this algorithm converges to a critical point of the cost function f defined in (9).

The algorithm presented above is independent of the parameterization used to represent rotations. One could use unit quaternions, 3×3 rotation matrices, etc.

Remark 1 (Distributed Implementation): The computations involved in the Riemannian gradient descent described above can be distributed among the cameras. For a camera $i \in \mathcal{V}$, define the set of *neighbors* of i, denoted $N_i \subset \mathcal{V}$, as the set of all $j \in \mathcal{V}$ such that an inter-camera measurements exists between i and j, or $e = (i,j) \in \mathcal{E}$. We assume that each camera is equipped with radios and local processors, and that the edges in the measurement graph $\mathcal{G}$ also define communication links among cameras. From the gradient formula (11) one observes that the portion of the gradient corresponding to a camera $i \in \mathcal{V}$ at iteration counter t only depends on the current pose estimates of neighboring cameras and relative measurements between i and its neighbors. The only hurdle in distributing the gradient descent algorithm is the step-size calculation since that may require the current estimate of all the poses and the full gradient. This hurdle can be overcome by using a subgradient method, in which a predetermined sequence of step sizes $\{\eta_t\}_{t \in \mathbb{N}}$ are used by all the cameras that are equipped with synchronized clocks. In particular, any sequence of positive scalars that satisfies $\lim_{t\to\infty} \eta_t = 0$, $\sum_{t=1}^{\infty} \eta_t = \infty$, $\sum_{t=1}^{\infty} \eta_t^2 < +\infty$ guarantees that the sequence given in (12) will converge to a critical point of the cost function (9)[19]. Using the subgradient allows performing gradient descent to minimize (9) in a distributed manner since at the update stage between iterations t and $t+1$, a camera only needs to communicates with its neighboring cameras $j \in N_i$ to obtain the current estimates of the neighbors' node variables. This is enough to compute its own portion of the gradient. □

IV. Simulation studies

In this section we present simulations studying the performance of the ML-CL algorithm in terms of localization accuracy.



(a) Ground Truth (b) Initial Estimate (c) ML-CL: rel. orient.+pos. meas.

(d) ML-CL: rel. orient. meas. (e) ML-CL: rel. pos. meas. (f) ML-CL: rel. bearing meas. (g) ML-CL: rel. dist. meas.

Fig. 3. (a) An image of a camera network along with (b) the initial guess for the poses of the cameras, and (c-g) the estimated poses after the ML-CL algorithm has been used with inter-camera relative measurements. Each of the plots in (c)-(g) correspond to a distinct type of relative measurement.

	Orient.(σ_e)	Pos.(Σ_e)	Bearing (k_e)	Dist. (σ_e)
Initial Est.	0.26 rad	$4I$ m	–	–
Inter-Cam.	0.087 rad	$0.25\,I$ m	20	0.5m

TABLE I

THE PARAMETERS OF THE MEASUREMENT PDFS (SEE (2)- (6)) USED IN ALL SIMULATIONS.

A. Performance in a single experiment

We consider a network of 5 cameras. One possible graph on such a network is presented in figure 3(a). Both the initial guess for the pose of each camera, and the noisy inter-camera measurements are drawn from the distributions described in Section III. The parameters for each distribution are reported in Table I. The variances of the initial guess of poses are chosen to be much higher then those for the inter-camera measurements to simulate a realistic situation when the initial guess is poor.

The ML-CL algorithm is applied to one realization of the noisy measurements for each of the measurement types, as well as for the case when both relative orientation and position measurements are available. The resulting estimated positions and orientations are shown in Figures 3(c-g). We see that relative orientation measurements allow for an accurate estimation of the orientation of each camera without having any effect on the estimated position. In contrast, relative position measurements improve both the orientation and position estimates for each camera. This is expected since absolute orientations of the cameras affect the relative positions while absolute positions do not affect relative orientations. When noisy measurements of both orientation and position are available, the pose of every camera can be estimated with high accuracy. In contrast to the position and orientation measurements, both the bearing and distance measurements lead to poor estimates, at least for this set of measurements.

B. Performance evaluation through MC simulations

We next examine the following questions. One, how does estimation accuracy of the ML-CL algorithm change as the connectivity of the measurement graph increases due to the increase in the number of relative measurements for the same number of cameras, and how does accuracy depend on the type of those measurements? Two, how does ML-CL perform compared to the alternative method proposed in [1]? Some results relevant to the first question have been already presented in the previous section. In this section we examine the question through Monte-Carlo simulations to verify the trends already observed are not random occurrences.

The following definitions are required. The error in an estimate $\hat{\mathbf{R}}_i$ of the orientation for a camera i is $e_R(i) := d(\mathbf{R}_i, \hat{\mathbf{R}}_i)$, where $d(\cdot\,,\cdot)$ is defined in (3). The error in an estimate $\hat{\mathbf{t}}_i$ of the position of camera i is $e_p(i) := \|\mathbf{t}_i - \hat{\mathbf{t}}_i\|_2$. The total r.m.s. error in the orientation and position estimate is defined as $\sqrt{\mathrm{E}[\sum_{i=1}^{n} e_R^2(i)]}$ and $\sqrt{\mathrm{E}[\sum_{i=1}^{n} e_p^2(i)]}$, respectively, where $\mathrm{E}[\,\cdot\,]$ denotes expectation. The expected value $\mathrm{E}[e_{[\,\cdot\,]}(i)]$ is also referred to as the bias in that error. All expectations are computed from appropriate averaging from random samples obtained through simulations.

a) Effect of graph connectivity and measurement type: We again consider a network of 5 cameras. Starting with a graph containing no relative measurements (i.e., edges), where the initial guess is the best we can do, measurements are randomly added until the graph is fully connected. For each graph, r.m.s. errors are estimated using a Monte-Carlo simulation with 2000 samples. In each sample, the initial guess and noisy inter-camera relative measurements are again drawn from the distributions described in Section III with the parameters shown in Table I. The experiment is repeated for

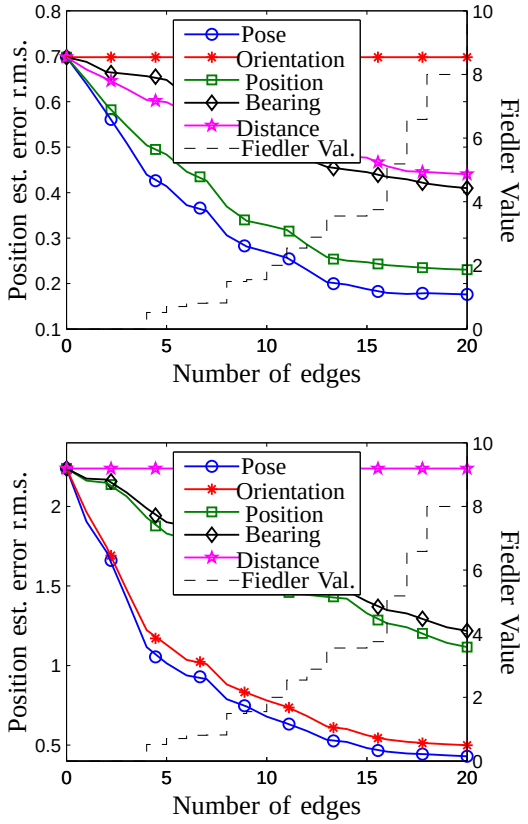


Fig. 4. Total r.m.s. error of the (a)position estimates and (b) orientation estimates with Ml-CL in a 5 camera network, as a function of number of measurements. Each curve in the plot corresponds to the type of relative measurement available that is specified by the legend.

each type of relative measurement, keeping the initial guess the same. The total r.m.s. error in the position and orientation estimates are reported in Figure 4. We also show the Fiedler value of the network in the figures. The Fiedler value is the second smallest eigenvalue of the graph Laplacian matrix, and is a scalar measure of the connectivity [20].

We see from the figures that as the number of measurements increase, the estimation error decreases, as expected. The ML-CL algorithm is seen to improve absolute orientation and position estimate over the initial guess for almost every type of relative measurement. The few exceptions are as follows. Measurements of the relative distance has no effect on the estimated orientation. The cause of this is immediately obvious from an examination of the cost function 9 as the edge cost corresponding to distance measurements does not contain an orientation node variable. Stated another way, same inter-camera distances can be maintained with arbitrary absolute camera orientations. This supports what was seen in figure 3 for a single realization of the noise. Similarly, relative orientation measurements do not improve absolute position estimates, which can be explained in a similar manner.

C. Effect of measurement noise, and comparison with [1]

We now examine the effect on localization accuracy when some of the measurements are noisier than others. We also compare ML-CL's performance with the algorithm in [1] for varying measurement noise levels.

A network of 3 cameras is considered, in which every camera has a measurement for every other camera in the network. That is, the corresponding graph is fully connected. Each inter-camera measurement is of the relative orientation and bearing. These measurement types are chosen to enable comparison with the algorithm in [1], since the same measurement types are considered in [1]. The parameters σ_e and k_e for distributions of relative orientation and bearing measurements of camera 2 and 3 are given in Table I. However, the noise parameters for measurements obtained by camera 1 are allowed to vary as follows: for orientation measurements, $\sigma_e = 0.087 \times K$, and for bearing measurements, $k_e = 20 \times K$ where $K \in [2^{-8}, 2^8]$. The wrapped Gaussian distribution (for relative orientation measurements) and Von Mises - Fisher distributions (for relative bearing measurements) produce more or less noisy measurements depending on the value of K. For larger values of K, orientation measurements become more noisy, while bearing measurements become less noisy. For each value of K considered, noisy measurements are generated from the corresponding distributions. These measurements are then used by the ML-CL algorithm and the algorithm in [1] to estimate the pose of each camera. To coincide with the assumptions made in [1], the initial pose estimates are not used as measurements in the ML-CL algorithm. This will be reflected in our choice of distance metric. For each method of estimation and each value of K, the bias and variance of the distance between pose estimates is computed from a Monte-Carlo simulation with 200 samples. The distance between a pose $T = \{(\mathbf{R}_i, \mathbf{t}_i)\}_{i=1}^n$ of a network of n cameras and its estimate $\hat{T}$ is defined as

$$d(T, \hat{T}) := \left(\sum_{i=1}^{n} (d^2(\mathbf{R}_i, \hat{R}_i) + \left\| \frac{\mathbf{t}_i}{\|\mathbf{t}_i\|} - \frac{\hat{t}_i}{\|\hat{t}_i\|} \right\|^2) \right)^{1/2} \tag{13}$$

Using the orientation and bearing measurements alone, the pose of all cameras can only be estimated up to a scale ambiguity. For that reason we have normalized the position estimates to adjust for the ambiguity in scale.

The results are reported in Figure 5. Though both algorithms provide accurate estimates, we see that when all measurements have equal amount of noise, the algorithm in [1] proves to be more accurate. This is likely due to an additional optimization step found in [1] in which the initial estimates are improved by minimizing an additional cost function. The ML-CL algorithm does not perform this additional step, though it could be implemented if desirable. However the ML-CL algorithm provides more accurate estimates when the difference between the noise levels in the various measurements is large. This occurs since the ML-CL algorithm takes into account the noise in each measurement in a principled way to compute the most likely estimates given this information. This reduces the effect of measurements that are highly noisy, while heightening the effects of

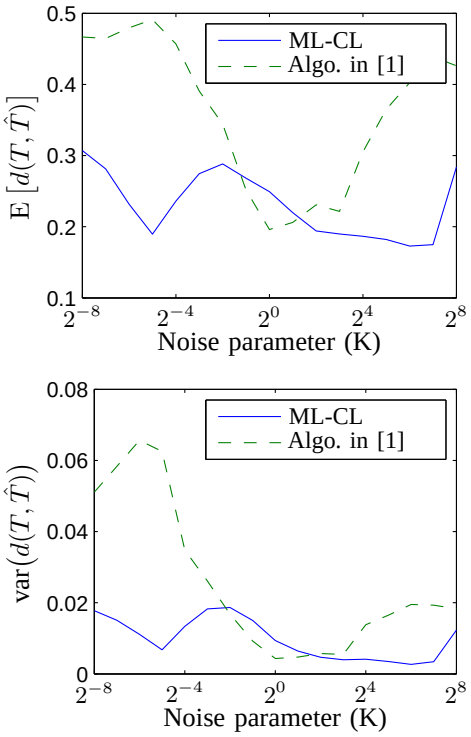


Fig. 5. Comparison of ML-CL and the algorithm in [1] for varying levels of noise in some of the relative measurements in a 3- camera network. Larger K corresponds to noisier relative orientation measurements and less noisy bearing measurements.

measurements with lower noise. No such weights are present in the algorithm in [1]. While it is possible to modify the cost function in [1] to include weights, due to the non-Euclidean nature of the relative orientation measurements, it is not clear how one would determine these weights.

V. Conclusion

Relative measurements between pairs of cameras can be fused with absolute measurements to improve pose estimates of all the cameras. We introduced an algorithm for doing so that computes the (approximate) maximum likelihood estimate of the absolute poses of the cameras for certain measurement noise distributions. Earlier approaches for 3-D camera network localization required each measurement to be of a specific type, and only considered "least-squares" type estimates that did not provide statistical guarantees on the estimates. The novel contributions of this work are (i) the ability to fuse various types of inter-camera relative measurements (orientation, bearing, position, distance, and any combination thereof), and (ii) a maximum likelihood (ML) approach that considers a distribution on the group $SO(3)$ rather then on one of its parameterizations. The ML formulation has the advantage over least-squares type approaches that it allows the algorithm to emphasize the low noise measurements over those with high noise in a principled way.

In this paper we only considered a centralized algorithm since that is adequate for a static network of cameras. The algorithm can be distributed in such a way that communication is only necessary between neighboring cameras. Future research will compare performance of the distributed algorithm with the centralized one. We also intend to study the performance of the proposed algorithm on a camera network experimentally in the future. Some of the noise distributions assumed in deriving the ML estimates, particularly those defined over $SO(3)$ and $\mathbb{S}^2$, needs further study to test how well they model noise in sensor measurements. We intend to do so through the use of hypothesis testing on experimental data.

References

[1] R. Tron and R. Vidal, "Distributed Image-Based 3-D Localization of Camera Sensor Networks," *Proceedings of the 48th IEEE Conference on Decision and Control*, pp. 1–11, Sept. 2009.

[2] U. Ramachandran, K. Hong, L. Iftode, R. Jain, R. Kumar, K. Rothermel, J. Shin, and R. Sivakumar, "Large-Scale Situation Awareness With Camera Networks and Multimodal Sensing," in *Proceedings of the IEEE*, 2012, pp. 1–15.

[3] S. Soro and W. Heinzelman, "A Survey of Visual Sensor Networks," *Advances in Multimedia*, vol. 2009, pp. 1–21, 2009.

[4] V. Erickson, M. A. Carreira-Perpinan, and A. Cerpa, "OBSERVE: Occupancy-based system for efficient reduction of HVAC energy," in *10th International Conference on Information Processing in Sensor Networks (IPSN 2010)*, April 2011, pp. 258–269.

[5] G. Piova, I. Shames, B. Fidan, F. Bullo, and B. Anderson, "On frame and orientation localization for relative sensing networks," in *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 2008, pp. 2326–2331.

[6] D. Borra, E. Lovisari, R. Carli, F. Fagnani, and S. Zampieri, "Autonomous calibration algorithms for networks of cameras," in *American Control Conference*, 2012.

[7] D. Devarajan and R. J. Radke, "Calibrating Distributed Camera Networks Using Belief Propagation," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 1, pp. 060 696–221, 2007.

[8] J. Kent, "Some probabilistic properties of bessel functions," *The Annals of Probability*, vol. 6, no. 5, pp. pp. 760–770, 1978.

[9] P. T. Fletcher, S. Joshi, C. Lu, and S. Pizer, "Gaussian Distributions on Lie Groups and Their Application to Statistical Shape Analysis," in *Information Processing in Medical Imaging*, May 2003, pp. 450–462.

[10] D. Lymberopoulos, A. Barton-Sweeny, and A. Savvides, "Sensor localization and camera calibration using low power cameras, ENALAB technical report," ENALAB (Yale University), Tech. Rep., September 2005.

[11] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529 – 2553, 2007.

[12] B. N. Hood and P. Barooah, "Estimating doa from radio-frequency RSSI measurements using an actuated reflector," *IEEE Sensors*, vol. 11, no. 2, pp. 413–417, February 2011.

[13] J. Kassebaum, N. Bulusu, and W.-C. Feng, "3-D Target-Based Distributed Smart Camera Network Localization," *Image Processing, IEEE Transactions on*, vol. 19, no. 10, pp. 2530–2539, 2010.

[14] G. Kurillo, Z. Li, and R. Bajcsy, "Wide-area external multi-camera calibration using vision graphs and virtual calibration object," in *Distributed Smart Cameras, 2008. ICDSC 2008. Second ACM/IEEE International Conference on*, 2008, pp. 1–9.

[15] H. Medeiros, H. Iwaki, and J. Park, "Online distributed calibration of a large network of wireless cameras using dynamic clustering," in *2nd ACM/IEEE Int. Conf. on Distributed Smart Cameras*, September 2008, p. 110.

[16] N. I. Fisher, T. Lewis, and B. J. J. Embleton, *Statistical Analysis of Spherical Data*. Cambridge University Press, 1997.

[17] P. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.

[18] Y. Ma, J. Košecká, and S. Sastry, "Optimization criteria and geometric algorithms for motion and structure estimation," *International Journal of Computer Vision*, vol. 44, no. 3, pp. 219–249, 2001.

[19] L. Yang, "Riemannian median and its estimation," *LMS Journal of Computation and Mathematics*, vol. 13, pp. 461–479, Jan. 2011.

[20] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, pp. 298–305, 1973.