Collaborative 3D localization of robots from relative pose measurements using gradient descent on manifolds

Joseph Knuth and Prabir Barooah
Technical Report

***Abstract*— We propose a distributed algorithm for estimating the full 3-D pose (position and orientation) of multiple autonomous vehicles with respect to a common reference frame when GPS is not available. This algorithm does not rely on the use of any maps, or the ability to recognize landmarks in the environment. Instead we assume that noisy measurements of the relative pose between pairs of robots are intermittently available. We utilize the additional information about each robot's pose provided by these measurements to improve over self-localization estimates. The proposed method is based on solving an optimization problem in an underlying product manifold $(SO(3)\times\mathbb{R}^3)^{n(k)}$. A provably correct explicit gradient descent law is provided. Unlike many previous approaches, the proposed algorithm is applicable to the 3-D case. The method is also capable of handling a fully dynamic scenario where the neighbor relationships are time-varying. Simulations show that the errors in the localization estimates obtained using this algorithm are significantly lower then what is achieved when robots estimate their pose without cooperation. Results from experiments with a pair of ground robots with vision-based sensors reinforce these findings.**

I. Introduction

In recent years, interest in utilizing teams of autonomous mobile robots has grown rapidly. Multi-robot teams are beneficial in many ways. Utilizing a group of low cost robots may be more economical then risking a single, more costly robot. In search and rescue operations, a group of robots can cover a larger area then a single robot. In hazardous conditions, the innate redundancy of a group of robots may be necessary to prevent catastrophic loss of mission capability. Regardless of the application, localization is a crucial task for any autonomous mobile robot team.

Localization for autonomous robots can be accomplished using a variety of sensors. Some of the more common sensors include Inertial Measurement Units (IMUs), vision based sensors, and Global Positioning System (GPS). Of the three, GPS is the only sensor capable of providing global measurements of a robots position. However in many situations, GPS measurements may not be available, or may only be intermittently available. For example, a group of unmanned aerial vehicles (UAVs) operating in an urban environment may temporarily lose GPS measurements when the signal is blocked by large buildings. In such a situation, the global pose can be found by integrating over the relative pose measurements found using IMUs or vision based sensors. This method of localization through "dead reckoning" can lead to a rapid growth in localization error [1]. When utilizing a team of robots, measurements of the relative pose between pairs of robots may be available. These provide additional information on the robots' pose that can be used to improve localization accuracy.

In this paper we propose a method for collaborative localization after a group of robots loses access to GPS. We assume all robots are equipped with proprioceptive sensors (vision, IMU, etc.) allowing each robot to measure its change in pose between time steps. We refer to these noisy measurements as *inter-time relative pose measurements*. Using these noisy measurements, each robot can perform localization through dead reckoning. In addition, we assume each robot is equipped with exteroceptive sensors, allowing intermittent noisy measurements of the relative pose between pairs of robots. We refer to these measurements as *inter-robot relative pose measurements*. These inter-robot relative pose measurements provide additional information on the absolute pose of each robot. We propose both a centralized and distributed algorithm to perform collaborative localization by fusing the inter-time and inter-robot relative pose measurements to obtain a improved estimate of the global pose of every robot. In the distributed algorithm, communication is only necessary between pairs of robots for which an inter-robot relative pose measurement has been obtained.

Collaborative localization has been considered in the context of simultaneous localization and mapping (SLAM). In one class of approaches, robots exchange local maps which are aligned and merged to improve robots' location estimates as well as to improve the maps; see [2], [3] and references therein. This requires the ability to identify common features in distinct maps generated by the robots. A method based on pose graphs is developed in [4] that does not require exchange of maps. In [5], robots exchange images and an implicit extended Kalman filter is used to update the state of each robot when a common feature is found.

Recognizing common landmarks in distinct maps is often challenging. In addition, exchanging image data or maps between robot requires high bandwidth communication. A second body of work therefore considers the collaborative localization problem as one in which only relative measurements (of pose, position, orientation etc.) between pairs of robots are obtained and used to improve localization accuracy over self-localization. The most common approach to localize a team of robots in 2-D is through the use of the Kalman filter or the Extended Kalman filter; see [6], [7], [8], [9] and references therein. Other approaches include the ML estimator based method of Howard et al. [10], the MAP estimator based approach of Nerurkar et al. [11], and methods based on iterative computation of the best linear unbiased estimator [12]. Leung et al. considers the problem of equivalency between centralized and decentralized collaborative localization algorithms [13]. No specific algorithm is proposed. Instead they allow for an arbitrary algorithm to fit within their framework.

As in the papers cited in the previous paragraph, in this paper we provide a method for collaborative localization of robots that uses noisy relative pose measurements between certain pairs of robots. In contrast to existing work, we make two novel contributions. First, our method is applicable to 3-D pose estimation problem, while to the best of our knowledge all the previous papers on collaborative localization using relative measurements are limited to 2-D pose estimation. Our proposed method relies on solving a non-linear optimization problem on a product manifold $(SO(3) \times \mathbb{R}^3)^{n(k)}$. This is accomplished through a provably convergent gradient descent algorithm on the product manifold. We provide an explicit formula for the gradients as well as the update law. The gradient descent law is provided in a parameterization-independent form, and any parametrization of the rotation operators can be used in its numerical implementation. This problem was previously considered in [14], but the solution provided there used an update law that relied on linear approximation and no proof of correctness could be provided. The second contribution is that the method is applicable to the dynamic scenario, in which the pairs of robots that obtain inter-robot relative pose measurements vary over time. As Leung et al. reports in [13], an important assumption in the papers cited in the previous paragraph (except for [13] itself) is that of a static communication network, or the ability of each robot to send information to all other robots.

We provide two algorithms, a centralized and a distributed one. In the latter, each robot only uses locally available measurements and communicates only with a small number of neighbors. The complexity of the computations performed by a robot is only a function of the number of its neighbors at any given time, not the total number of robots in the group. This makes the distributed algorithm scalable to arbitrarily large groups of robots. In addition, the communication complexity of the algorithm is small. At every update, a pair of neighboring robots needs to exchange only (i) measurement of their relative pose and (ii) their current pose estimates. Since a pose measurement, which is an element of $SE(3)$, can be represented by 6 numbers, the pair of robots have to exchange only 12 numbers.

II. Problem statement

Consider a group of r mobile robots indexed by $i = 1, \dots, r$. Time is measured by a discrete counter $k = 0, 1, 2, \dots$. Measurements of a robot's global pose (position and orientation) from GPS and compass is either not available or only rarely available. Instead, we assume that each robot is equipped with proprioceptive sensors such that, at every time k, the robot is able to measure the Euclidean transformation between its current pose and its pose at the previous time $k-1$. We refer to these measurements as *inter-time relative pose measurements*. Such measurements can be obtained with inertial sensors, vision based sensors, or with a combination thereof. Additionally, they need not be obtained from a sensor alone. Instead, a measurement could also be the estimate of the rotation and translation undergone by the robot in that time interval that is obtained by fusing raw sensor measurements with prediction of the robot's motion from a dynamic/kinematic model.

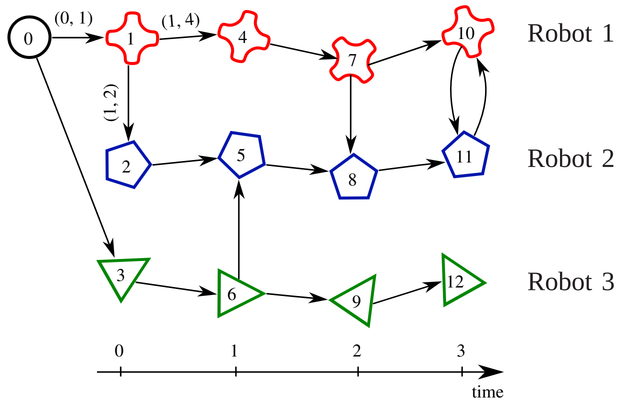


Fig. 1. A time history of three robots up to time $k = 3$ with inter-time and inter-robot relative pose measurements. Each (robot,time) pair is labeled with the corresponding node index from $\mathcal{V}_0(3)$. Arrows indicate edges, i.e., relative measurements, in $\mathcal{E}(3)$. Robots 1 and 3 had GPS measurements at the initial time $k = 0$. No other GPS measurements were available.

In addition to these inter-time measurements, each robot is equipped with exteroceptive sensors so that it can measure the relative pose (Euclidean transformation) of another robot with respect to itself whenever it can "see" the other robot. We call these measurements *inter-robot relative pose measurements*. Various combinations of sensors are able to extract inter-robot relative pose measurements. When information about the robots motion is known a priori, bearing and distance, bearing-only, and distance-only sensors can all be used to find the full pose [15]. If instead each robot is equipped with a camera and a target with known geometry, the full pose can be estimated from visual measurements using the epipolar constraint [16].

The situation above is best described in terms of a time-varying *graph* $\mathcal{G}(k) = (\mathcal{V}_0(k), \mathcal{E}(k))$ that shows how the noisy relative pose measurements relate to the global pose of each robot at every time step. The graph is defined as follows. For each robot $i \in \{1, \dots, r\}$ and each time $t \leq k$, a unique index (call it u) is assigned to the pair (i, t). How this indexing is done is immaterial. The set of these indices $\{1, \dots, rk\}$ define the set $\mathcal{V}(k)$ and the node set of the graph is defined as $\mathcal{V}_0(k) := \mathcal{V}(k) \cup \{0\}$. We then refer to the reference frame attached to robot i at time t as *frame* u. Node u is associated with the pose of robot i at time t relative to some common reference frame, given by the Euclidean transformation between the common reference frame and frame u, expressed in the common frame. We call these poses *node variable* and denote them $\mathbf{T}_u$. The common reference frame with respect to which all node variables are expressed is associated with node 0. If the global pose of at least one robot is known at time 0, perhaps through the use of a GPS and compass, then node 0 can be associated with the global reference frame. When global pose measurements are not available, node 0 could correspond to the initial reference frame of one of the robots. In either case, estimating the node variables is equivalent to determining the robots' poses with respect to frame 0. The node 0 is therefore called the *reference node*.

The set of *directed edges* at time k, denoted $\mathcal{E}(k)$, corresponds to the noisy inter-time and inter-robot measurements collected upto time k. That is, suppose robot i is able to measure robot j's relative pose at time ℓ, and let u, v be the nodes corresponding to robots i, j at time ℓ, respectively. Then $(u, v) \in \mathcal{E}(k)$ for all $k \geq \ell$. This edge is associated with the noisy measurement of the relative pose between robot i and robot j, expressed in robot i's frame at time ℓ. We denote this noisy relative pose measurement by $\hat{\mathbf{T}}_{u\,v}$. Similarly, each inter-time relative pose measurements of a robot also creates an edge in the graph.

The graph $\mathcal{G}(k)$ is called the *measurement graph* at time k. Figure 1 shows an example of the graph corresponding to the measurements collected by 3 robots up to time index 3.

An estimate of robot i's pose at time k can be obtained by chaining together (through the standard pose composition operator) the inter-time relative pose measurements collected from time 0 until time k. This is how a robot localizes itself after losing GPS signal if it is operating alone. In context of the measurement graph, this corresponds to chaining measurements from node 0 to the node u (node u corresponds to the pose of robot i at time k) along the path that only consists of the inter-time measurements of robot i. However, measurement on edges along any undirected path from node 0 to u can lead to such an estimate as well. Because both inter-time and inter-robot relative pose measurements are corrupted by noise, each path from 0 to u will yield a distinct estimate of the node variable associated with u. Fusing each estimate should then yield a more accurate estimate than what is possible by following a single path. When the measurements are linearly related to the node variables, this can be accomplished by using the best linear unbiased estimator, as done in [12]. In our case, the relationship between the measurements and node variables is nonlinear.

In the following sections we propose a method to solve the *collaborative localization problem,* that is, to estimate the pose of each robot at time k with respect to the common reference frame 0 by using all the measurements in the graph $\mathcal{G}(k)$. We first propose a centralized algorithm that takes into account all paths in the graph $\mathcal{G}(k)$ and finds the "best" estimate for each node variable. We then propose a modified algorithm that is fully distributed and only requires communication between robot pairs that obtain inter-robot relative pose measurements.

III. Centralized Algorithm

In this section we present a solution to the collaborative localization problem where all the relative measurements are instantly available to a central processor at each time k. The centralized solution naturally leads to a distributed scheme, which will be described in the next section.

Instead of addressing the problem of estimating the robots' current poses at time k, we examine the more general problem of estimating all the node variables of the measurement graph $\mathcal{G}(k)$ using the robots' past noisy relative pose measurements. We assume the graph $\mathcal{G}(k)$ is *weakly connected* for all time k. That is, for all $k \geq 0$ and all $i, j \in \mathcal{V}(k)$ there exists an undirected path from i to j. An undirected path from a node to another is a path along the edges without respecting the directions of the edges. The problem is posed as an optimization of a cost function over the set of node variables, where the cost function measures how well a given set of global poses explains the noisy measurements collected up to time k. The initial condition for each node variable $i \in \mathcal{V}(k)$ is given by chaining together the noisy relative pose measurements associated with the edges of any undirected path from node 0 to node i. For ease of exposition we assume each robot at time 0 has an initial estimate of its pose at that time with respect to the reference node.

To derive a suitable cost function, we break each pose (both noisy relative pose measurements and node variables) into its corresponding rotation $\mathbf{R} \in SO(3)$ [1] and translation $\mathbf{t} \in \mathbb{R}^3$. Note that we consider a rotation $\mathbf{R} \in SO(3)$ to be an abstract operator and not necessarily equated to its matrix representation. When the relative pose measurements are completely error free, $\hat{\mathbf{R}}_{i\,j}$ is the true rotation between frame i and frame j, expressed in frame i. This rotation can also be expressed in terms of the node variables as $\mathbf{R}_i^T \mathbf{R}_j$, where $\mathbf{R}_i^T$ is the adjoint of the operator $\mathbf{R}_i$. Similarly, both $\hat{\mathbf{t}}_{i\,j}$ and $\mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)$ should be equal, which is the translation from frame i to frame j expressed in frame i, if there were no noise in $\hat{\mathbf{t}}_{i\,j}$. When noise is present in the measurements, how much these estimates differ - measured by a suitable distance function - provides a measure of how a given set of node variables explain the noisy measurements. Distance for the translations can be given in terms of the 2-norm of the difference. To measure the distance between $A, B \in SO(3)$, we use a Riemannian distance $d(A, B)$ given by

$$d(A, B) = \sqrt{-\frac{1}{2}\operatorname{Tr}\left(\log^2(A^T B)\right)}. \tag{1}$$

More details on this distance function can be found in the appendix. The cost function is then given by summing over all measurements.

$$f(\{\mathbf{T}\}_{\mathcal{V}(k)}) := \frac{1}{2} \sum_{(i,j)\in\mathcal{E}(k)} \Big(d^2(\hat{\mathbf{R}}_{i\,j}, \mathbf{R}_i^T \mathbf{R}_j) \\ + \|\hat{\mathbf{t}}_{i\,j} - \mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)\|^2 \Big). \tag{2}$$

By minimizing the cost function, we expect to find an improved estimate for the global pose of each robot over what can be found through dead reckoning alone. This cost

[1] $SO(3)$ denotes the set of all bounded linear operators on the Euclidean space $\mathbb{R}^3$ that preserve the length of vectors and orientation of the space. For more information about the group $SO(3)$ and its properties, see [17].

function in (2) is similar to one proposed in [18] for a static camera network; here the cost function changes with time.

Finding the minimum of a function defined over a vector space has been studied extensively. However the function $f(\,\cdot\,)$ is defined on a curved surface, specifically, the product *Riemannian Manifold* $(SO(3)\times\mathbb{R}^3)^{n(k)}$ where $n(k) = |\mathcal{V}(k)|$, the cardinality of the set $\mathcal{V}(k)$. One option for this optimization is to use a parameterization of the rotations using, say, Euler angles 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. However our goal is to find a provably correct algorithm that utilizes the geometry of the space without relying on any particular parameterization. See [19] for a discussion on the merits of such a direct optimization over the alternatives. We accomplish this through use of a gradient descent algorithm on the product manifold.

Given a smooth real valued function f defined on a manifold M, the gradient of f at $p\in M$, denoted $grad\,f(p)$, is a vector in the tangent space of M at p, denoted T_pM. Just as in Euclidean Space, $grad\,f(p)$ points in the direction of greatest rate of increase of f. The following theorem provides the gradient for our cost function. The proof is provided in the appendix.

Theorem 1: The gradient of the cost function shown in (2) at $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_n, \mathbf{t}_n) \in \big(SO(3)\times\mathbb{R}^3\big)^n$ is

$$\begin{aligned} grad\,f(p) = (&grad\,f(\mathbf{R}_1), grad\,f(\mathbf{t}_1), \ldots, \\ &grad\,f(\mathbf{R}_n), grad\,f(\mathbf{t}_n)) \end{aligned} \tag{3}$$

where, for $i = 1, \ldots, n$,

$$\begin{aligned} grad\,f(\mathbf{R}_i) = -\mathbf{R}_i\Bigg(&\sum_{(i,j)\in\mathcal{E}(k)}\Big[\mathbf{R}_i^T(\mathbf{t}_j-\mathbf{t}_i)\hat{\mathbf{t}}_{i\,j}^T \\ &-\hat{\mathbf{t}}_{i\,j}(\mathbf{t}_j-\mathbf{t}_i)^T\mathbf{R}_i + \log(\mathbf{R}_i^T\mathbf{R}_j\hat{\mathbf{R}}_{i\,j}^T)\Big] \\ &+\sum_{(j,i)\in\mathcal{E}(k)}\log(\mathbf{R}_i^T\mathbf{R}_j\hat{\mathbf{R}}_{j\,i})\Bigg) \end{aligned} \tag{4}$$

$$\begin{aligned} grad\,f(\mathbf{t}_i) = &\sum_{(i,j)\in\mathcal{E}(k)}\big(\mathbf{t}_i+\mathbf{R}_i\hat{\mathbf{t}}_{i\,j}-\mathbf{t}_j\big) \\ &+\sum_{(j,i)\in\mathcal{E}(k)}\big(\mathbf{t}_i-\mathbf{R}_j\hat{\mathbf{t}}_{j\,i}-\mathbf{t}_j\big)\,. \end{aligned} \tag{5}$$

Minimizing a function f using gradient descent requires that during each iteration, the current estimate must be updated by moving in the direction of the negative gradient. In a vector space this is accomplished by simply subtracting $\eta\,grad\,f$ from the current estimate for some appropriate scalar η. On a Riemannian manifold, moving in the direction of $-grad\,f$ 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

$$\begin{aligned} \exp_p(\xi) = (&\mathbf{R}_1\exp(\mathbf{R}_1^T\xi_{\mathbf{R}_1}), \mathbf{t}_1+\xi_{\mathbf{t}_1}, \ldots, \\ &\mathbf{R}_n\exp(\mathbf{R}_n^T\xi_{\mathbf{R}_n}), \mathbf{t}_n+\xi_{\mathbf{t}_n}) \end{aligned} \tag{6}$$

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 (6) is the Lie-group exponential map [19]. An exact derivation of this is provided in the appendix. The gradient descent law is

$$p_{t+1} = \exp_{p_t}(-\eta_t grad\,f(p_t)), \quad t = 0, 1, \ldots, \tag{7}$$

where $\eta_t \geq 0$ is chosen to be the *Armijo step size*. More detail on gradient descent algorithms on manifolds, and on the Armijo step size in particular, can be found in [20].

Using the upadate law given in 7, a gradient descent is performed, terminating when the norm of the graident falls below some user specified threshold. Theorem 4.3.1 in [20] guarantees that the estimates found using this algorithm are critical points of the cost function f defined in (2).

It should be noted that while it might be convenient to represent rotations as 3×3 rotation matrices in computation, the algorithm presented above is independent of the parameterization used to represent rotations.

IV. Distributed Algorithm

In this section we propose a modified algorithm capable of running in a fully distributed manner with limited memory, processor power, and communication bandwidth.

For each robot i, let $N_i^{(+)}(k)$ denote the set of all robots $j\in\{1,\ldots,r\}$ such that, at time k, robot i can measure its relative pose with respect to j. Let $N_i^{(-)}(k)$ denote the set of all robots $j\in\{1,\ldots,r\}$ such that, at time k, robot j can measure its relative pose with respect to robot i. The *neighbors* of robot i at time k are then given by the set $N_i(k) = N_i^{(+)}(k)\cup N_i^{(-)}(k)$. *We assume that each robot can communicate with its neighbors during each time step.*

To facilitate the description of the distributed algorithm, consider the time varying *local* measurement graph $\mathcal{G}_i(k) = (\mathcal{V}_i(k), \mathcal{E}_i(k))$ of robot i, whose node set is simply the neighbors of i at time k along with the reference node 0 and i itself: $\mathcal{V}_i(k) = N_i(k)\cup\{0,i\}$. The edges of $\mathcal{G}_i(k)$ correspond to the inter-robot measurements at time k between i and its neighbors, along with an edge $(0,j)$ for each $j\in\mathcal{V}_i(k)$. Thus if robot i can "see" robot j at time k, then $(i,j)\in\mathcal{E}_i(k)$. Similarly, if j can see i, $(j,i)\in E_i(k)$. Each node in the local measurement graph $\mathcal{G}_i(k)$ is associated with a global pose for a robot at time k. Thus an edge $(p,q)\in\mathcal{G}_i(k)$ (where $i=p$ or q) is associated with the noisy inter-robot relative pose measurement between robots p and q at time k. The additional edges $(0,j)$, $j\in\mathcal{V}_i(k)$ are associated with the initial estimate for each robots global pose, denoted $\hat{\mathbf{T}}_{0\,j}$. Each robot i obtains $\hat{\mathbf{T}}_{0\,i}$ at time k by concatenating the robot's pose estimate obtained at time $k-1$ with the noisy inter-time relative pose measurement describing the robots motion from $k-1$ to k. We consider this estimate as a measurement on the edge $(0, i)$. The graph $\mathcal{G}_i(k)$ is a now a valid measurement graph since each edge has an associated noisy relative measurement. The edge $(0, i)$ ensures that $\mathcal{G}_i(k)$ is weakly connected.

The decentralized algorithm works as follows. At each time k, every robot $i \in \{1, \ldots, r\}$ forms an initial estimate of its global pose $\hat{\mathbf{T}}_{0\,i}(k)$ as described above and measures the inter-robot relative pose $\hat{\mathbf{T}}_{i\,j}$ for each robot $j \in N_i^{(+)}(k)$. It then transmits the pair $(\hat{\mathbf{T}}_{i\,j}, \hat{\mathbf{T}}_{0\,i})$ to each robot $j \in N_i^{(+)}(k)$ and receives in turn robot j's estimate of its current global pose $\hat{\mathbf{T}}_{0\,j}(k)$. Similarly, for each robot $j \in N_i^{(-)}(k)$ robot i will receive the pair $(\hat{\mathbf{T}}_{j\,i}, \hat{\mathbf{T}}_{0\,j})$ and transmit to robot j the estimate $\hat{\mathbf{T}}_{0\,i}$. Robot i then executes the algorithm present in section III on the local measurement graph $\mathcal{G}_i(k)$. After the computation, only the estimated value for the node variable $\mathbf{T}_i$ is retained. No other value need be stored in robot i's local memory. Note that if robot i has no neighbors at time k, the distributed collaborative localization algorithm is equivalent to performing self-localization from inter-time relative measurements. Since the distributed algorithm is simply the centralized algorithm applied to a local measurement graph, it inherits the correctness property from the algorithm presented in section III.

V. Simulation Results

In this section we present simulations studying the improvement in localization accuracy with collaborative localization. The distributed, rather then the centralized, algorithm was chosen because it is more practical to implement on large teams of robots. We also explore the difference between the centralized and distributed algorithms' performance through simulations.

A. *Centralized vs. Distributed*

To compare the centralized and distributed algorithms, localization of a group of 5 robots capable of measuring inter-robot pose and exchanging information was simulated. Each of the 5 robots travel along a distinct zig-zag path, shown in figure 2. Two robots can obtain relative pose measurements at time k if the Euclidean distance between them at that time is less than $7\ m$. Furthermore, 25% of these potential measurements were dropped to simulate random failure. A plot of the number of neighbors for robot 1 over time is shown in Figure 3. The rotation measurements for each relative pose (both inter-robot and inter-time) were corrupted by independent identically distributed (i.i.d.) unit quaternions drawn from a Von Mises-Fisher distribution [21] centered around the zero-rotation quaternion and with a concentration parameter of $10,000$. Noise in the relative translation measurements was simulated by adding i.i.d zero-mean normal random variables with covariance matrix $I_{3\times 3} \times 10^{-6}\ m^2$.

Figure 4 shows the orientation and position error of robot 1 in the 5-robot team, when its position is estimated with (i) self-localization, (ii) centralized collaborative localization, and (iii) distributed collaborative localization, algorithms. The plots indicate that, first, a significant improvement in

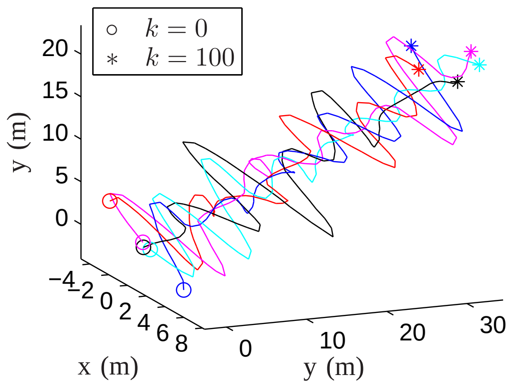


Fig. 2. The 3D trajectories for robots 1 through 5, used in all simulations.

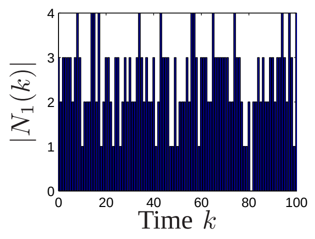


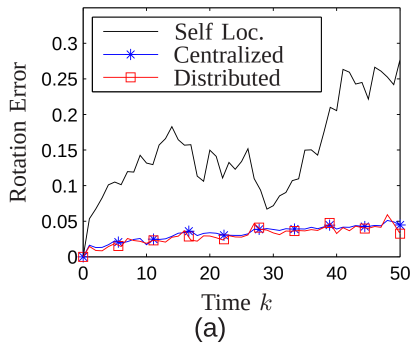
Fig. 3. The number of neighbors of robot 1 as a function of time, in a 5-robot team.

localization accuracy is achieved with collaboration. Second, the improvement in localization accuracy with the distributed algorithm is quite close to that using the centralized algorithm. This is promising since the distributed algorithm is applicable to large teams of robots in highly dynamic scenarios that can lead to arbitrary time variation in neighbor relationships. The centralized algorithm is not applicable in a realistic setting, but it provides a measure of the best possible performance. Due to these results, we only study the distributed algorithm from now on.

B. *Localization accuracy vs. Number of robots*

In this section we estimate the bias and variance for a group of robots using a Monte Carlo simulation with $1,000$ samples. The distributed, rather then the centralized, algorithm was chosen because it is more practical to implement on larger reams of robots. The robots were simulated traveling along distinct zig-zag paths in 3-D space, so that all three translational and rotational coordinates varied along time for each robot. Measurement noise was introduced in the same manner as in the previous simulations, and varied randomly from run to run. Neighbor relations were again determined as described earlier, and was kept the same from run to run to preclude this from being an additional source of randomness.

Simulations for robot teams of size $1, 2, 3, 4$ and 5 were carried out. When only one robot is present in the team,



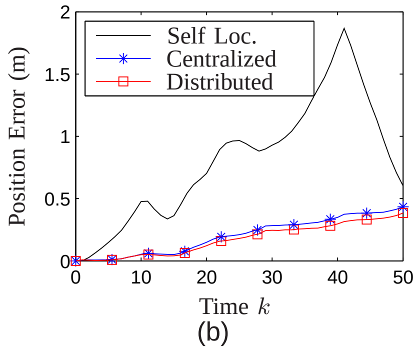


Fig. 4. A plot of (a) the Rotation error given by $d(\hat{\mathbf{R}}_1, \mathbf{R}_1)$ and (b) the position error given by $\|\hat{\mathbf{t}}_1 - \mathbf{t}_1\|$ for robot 1 over 50 time steps using 3 separate localization techniques. Self Loc. indicates the robot localized without the use of any inter-robot relative pose measurements. Centralized and Distributed indicate the robot utilized the inter-robot relative pose measurement from upto 4 other robots using either the centralized or distributed algorithm.

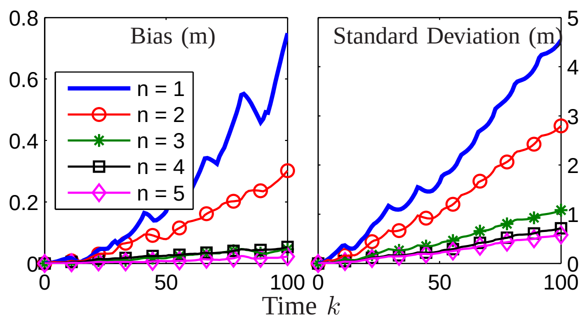


Fig. 5. The bias and standard deviation in position error for robot 1 when the distributed algorithm is applied to a group of $1, 2, 3, 4$ or 5 robots. Note, what we refer to as standard deviation is given by $\sqrt{\text{Tr}\,(Cov(\mathbf{e}(k), \mathbf{e}(k)))}$

collaborative localization is equivalent to self-localization without the aid of any inter-robot relative pose measurement.

The position estimation error of robot i is $\mathbf{e}_i(k) := \hat{\mathbf{t}}_i(k) - \mathbf{t}_i(k)$, where $\mathbf{t}_i(k)$ is its global position at time k and $\hat{\mathbf{t}}_i(k)$ is the estimate of this position. The bias and standard deviation in the position estimation error $\mathbf{e}_i(k)$ for robot 1 ($i = 1$) are shown in Figure 5. Both bias and standard deviation show significant improvement with distributed collaborative localization over self-localization. This is evident even for a team of only two robots. As the number of robots in the team increases, the localization error of robot 1 decreases. The improvement in accuracy however, shows a diminishing return with increasing team size.

VI. Experimental results

In this section we present results for two experiments conducted using the two Pioneer P3-DX robots shown in Figure 7. Each robot was equipped with a calibrated monocular Prosillica EC 1020 camera and wheel odometers. Measurements from these sensors were fused to find the noisy inter-time relative pose measurements. Each robot is additionally equipped with a target allowing the on-board cameras to measure the inter-robot relative pose by exploiting the known geometry of each target. The true pose of each robot was determined using an overhead camera capable of tracking each robot's target. The sensor were polled every 0.2 seconds with the noisy inter-robot relative pose measurements available at most, but not all times.

In the first experiment, each robot moved in a straight line with their paths approximately parallel. Two different pose estimates of the robots were obtained. One with self-localization (with the inter-time relative pose measurements alone) and the other with collaborative localization with the distributed algorithm, which utilized the inter-robot relative pose measurements. The resulting global position estimates, along with the true positions, for robot 1 are reported in Figure 8. The errors in the rotation estimates for robot 1 using each localization technique are shown in Figure 9. A distinct improvement in localization accuracy is seen when collaborative localization is performed. Simulations presented in Section V indicate that we should see a significant improvement in localization accuracy even in this small team, and the experimental results are consistent with that conclusion.

In the second experiment, the robots each moved along a circular path. The noisy relative pose measurements were used to perform both self localization and collaborative localization using the distributed algorithm. The resulting global position estimates, along with the true position for robot 2 are reported in Figure 6. A distinct improvement in localization accuracy is seen when collaborative localization is performed.

VII. Conclusion and Future Work

We introduced a novel distributed algorithm for estimating the full 3-D pose of multiple robots when noisy measurements of the relative pose between pairs of robots are intermittently available. The proposed algorithm does not rely on any particular parameterization of the underlying manifold. The algorithm is provably correct: the solution converges to a minimum of the cost function that measures how well the estimates explain the noisy relative measurements. The distributed algorithm requires communication only between neighbors. Both memory capacity and processing power requirements are small, and only depends on the number of neighbors, not on the total number of robots. The algorithm is applicable to a dynamic scenario in which the neighbors

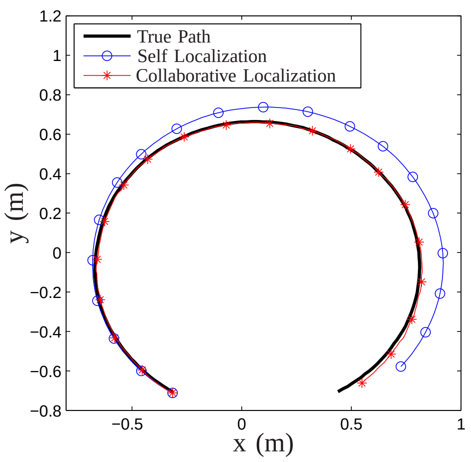


Fig. 6. A plot of the location of robot 2 in the overhead camera reference frame when both robots move in circles. The true path (found using the overhead camera), estimated path using self localization, and estimated path using the distributed collaborative localization algorithm are all reported.

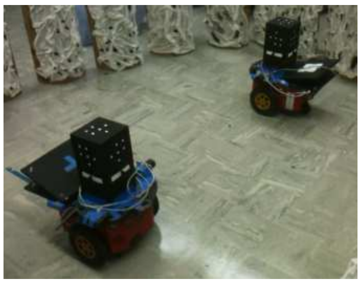


Fig. 7. Two Pioneer P3-DX robots equipped cameras and targets. Robot 1 is shown on the left, while robot 2 is on the right.

of a robot can vary arbitrarily with time. The novel contributions of this work compared to much of earlier work on collaborative localization are (i) ability to perform 3-D localization and (ii) ability to handle a time-varying network of robots.

Simulations show a significant increase in localization accuracy with the distributed collaborative localization algorithm over self-localization. The improvement is significant even for a small team of robots (2 or 3), with diminishing returns with increasing number of robots. Experimental results verify that indeed significant accuracy improvement can be achieved even with two robots.

In this paper we assume availability of relative pose measurements between robots. The "measurement graph"-based framework we use is also applicable when relative measurements of distance, bearing, or orientation between robots are available. In these cases, the cost function has to be changed suitably so that it is a function only of the available measurements on the edges of the graph. Preliminary work along these lines shows promising results; details will be presented elsewhere.

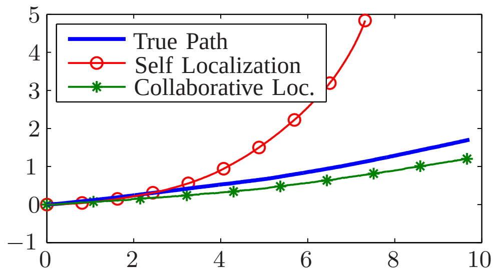


Fig. 8. A plot of the location of robot 1 when both robots move in a straight line. The true path (found using the overhead camera), estimated path using self localization, and estimated path using the distributed collaborative localization algorithm are all reported.

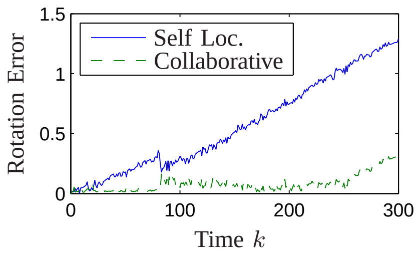


Fig. 9. The error in rotational estimates for robot 1 in a group of two robots while moving in a straight line. The rotation error between true rotation $R \in SO(3)$ and its estimate $\hat{R}$ is measured by the Riemannian distance $d(R, \hat{R})$ defined in (10). The legend "Self Loc." refers to robot 1 performing self localization while the legend "Collaborative" refers to the scenario when the distributed collaborative algorithm is used.

APPENDIX

A. *Riemannian Manifolds*

In this section we provide greater detail on the building blocks of our algorithm including Riemannian manifolds. A full introduction to the study of Riemannian geometry is outside the scope of this paper, and thus a basic knowledge of Riemannian geometry is assumed. For those unfamiliar with this material, the following may prove useful [22].

Let $\mathbb{L}(\mathbb{R}^3)$ denote the set of all linear operators on the vector space $\mathbb{R}^3$. We can then describe the manifold of 3-D rotations, denoted $SO(3)$, by the set $\{R \in \mathbb{L}(\mathbb{R}^3) : R^T R, det(R) = 1\}$. Given a point $p \in SO(3)$, the tangent space of $SO(3)$ at p, denoted $T_pSO(3)$ is given by

$$T_pSO(3) = \{p\hat{v} : \hat{v} \in \mathbb{L}(\mathbb{R}^3), \hat{v}^T = -\hat{v}\}.$$

A *Riemannian metric* on a Manifold M is given by defining $\forall p \in M$ $g_p : T_pM \times T_pM$ such that

- $\forall p \in M$, g_p is an inner product
- given two vector fields X and Y on M, the map $g : M \to \mathbb{R}$, $p \mapsto g_p(X_p, Y_p)$ is smooth.

This function g gives rise to an inner-product norm on the tangent space T_pM for each $p \in M$. That is, for $\xi \in T_pM$, $\|\xi\|^2 = g_p(\xi, \xi)$.

A *Riemannian manifold* is given by a Manifold equipped with a Riemannian metric. The first Riemannian manifold we consider is $(SO(3), g)$ were g is the Riemannian metric given by

$$g_p(A,B) = \frac{1}{2}\operatorname{Tr}\left(A^T B\right) \tag{8}$$

for all $p \in SO(3)$, $A, B \in T_pSO(3)$. From this point on, when we refer to $SO(3)$ we mean this Riemannian manifold. To further simplify the notation, when the argument $p \in SO(3)$ is clear, we will denote $g_p(X_p, Y_p)$ by $g(X_p, Y_p)$.

For a Riemannian manifold, the analog to a straight line in Euclidean space is given by a *geodesic*. For two points p and q on a manifold M, the geodesic from p to q, denoted γ_{pq}, is the "shortest" path from p to q. More precisely, if we consider the set of all parameterized paths from p to q, given by $\Gamma = \{\gamma : [0,1] \to M : \gamma(0) = p, \gamma(0) = q\}$ then a parameterized path for γ_{pq} is one that minimizes

$$\int_0^1 \|\gamma'(t)\| dt$$

over all paths in Γ.

For the manifold $SO(3)$, the geodesic from p to q ($\in SO(3)$) is given by

$$\gamma_{pg}(t) = p\,\exp(tv),\ t \in [0,1]$$

for

$$v = \log(p^{-1}q)$$

where exp is the Lie algebra to Lie group map given by the standard matrix exponential and log is the inverse map of exp. [2]

We next introduce the concept of distance on the manifold $SO(3)$. For an arbitrary Riemannian manifold, the distance between two points $p, q \in M$ is given by

$$d_M(p,q) = \int_0^1 \|\gamma_{pq}(t)\| dt \tag{9}$$

For $SO(3)$ this is given by

$$\begin{aligned} d_{SO(3)}(p,q) &= \int_0^1 \|p\,\exp(tv)v\| dt \\ &= \sqrt{-\frac{1}{2}\operatorname{Tr}\left(\log^2(p^T q)\right)}. \end{aligned} \tag{10}$$

The subscript $SO(3)$ on the distance function will be omitted when the arguments make it clear what manifold the distance refers to.

We next consider the notion of parallel transport. For $p \in M$ and $\xi \in T_pM$, the *parallel transport* function, denoted $\exp_p$, finds a new point $q \in M$ given by moving along a geodesic beginning at p and with instal velocity ξ. More precisely,

$$q = \exp_p(\xi) = \gamma(1) \tag{11}$$

where γ is a parameterized geodesic with $\gamma(0) = p$ and $\gamma'(0) = \xi$. In the case of $SO(3)$, $\exp_p(\xi) = p\exp(p^T\xi)$.

[2]This use of $\log(p)$ for $p \in SO(3)$ is only valid on the region where exp is a diffeomorphism. Manthon shows that exp is a diffeomrphism at least on the open ball about $I \in SO(3)$ of radius π [23].

The second manifold we consider is $(\mathbb{R}^3, <\cdot,\cdot>)$, standard Euclidean 3-space with the Riemannian metric given by the inner product on $\mathbb{R}^3$. Given points $p, q \in \mathbb{R}^3$ the (unique) geodesic connecting p and q is the straight line from p to q and $d(p,q) = \|p-q\| := <p-q, p-q>^{1/2}$.

Finally, we consider the Riemannian manifold $SE(3)$. Unfortunately, no bi-invariant Riemannian metric exists for $SE(3)$. Instead, we will use the equivalent manifold

$$M := SO(3) \times \mathbb{R}^3 \tag{12}$$

There exists a natural way to define the Riemannian metric on M so that we can deal with geodesics and gradients on M. More details on this will be given in Appendix C.

B. Gradients

Given a Riemannian manifold M and function $f : M \to \mathbb{R}$, the gradient of f, denoted $grad\, f$, is defined by the relation

$$df(\xi) = g(grad\, f, \xi) \tag{13}$$

for all vector fields ξ on M. When evaluated at a point $p \in M$, we have $df(\xi)(p) = (f \circ \gamma)'(0)$ for any curve $\gamma : [0,1] \to M$ such that $\gamma(0) = p$ and $\gamma'(0) = \xi_p$. We then have the following point wise definition for the gradient of f.

$$g_p(grad\, f(p), \xi_p) = (f \circ \gamma)'(0) \quad \forall p \in M. \tag{14}$$

Consider the function $f : SO(3) \to \mathbb{R}$, $p \mapsto d^2(p,q)$ for some fixed $q \in SO(3)$. For an arbitrary $p \in SO(3)$, we consider the geodesic $\gamma_{pq}(t) = p\exp(tv)$ where $v = log(p^{-1}q)$. Then using (14)

$$(f \circ \gamma)(t) = \big(d(p,q) - t\,d(p,q)\big)^2$$

and

$$(f \circ \gamma)'(0) = -d^2(p,q) = \operatorname{Tr}\left(v^2\right). \tag{15}$$

The corresponding tangent vector to the curve at p is given by

$$\xi_p := \gamma'(0) = pv.$$

Applying the Riemannian metric at p gives us

$$g_p(grad\, f(p), \xi_p) = \frac{1}{2}\operatorname{Tr}\left((grad\, f(p))^T pv\right). \tag{16}$$

Applying (14) we have

$$grad\, f(p) = -2p\log(p^{-1}q) \quad \forall p \in SO(3). \tag{17}$$

Using equation 17 and the bi-invariance of the distance function, we can find the following gradient that will be necessary in the sequel. For $(i,j) \in \mathcal{E}$, $k \in \mathcal{V}$ let $f = \frac{1}{2}d^2(\mathbf{R}_i^T\mathbf{R}_j, \hat{\mathbf{R}}_{ij})$, then

$$grad\, f(\mathbf{R}_k) = \begin{cases} -2\mathbf{R}_k \log\left(\mathbf{R}_k^T\,\mathbf{R}_j\,\hat{\mathbf{R}}_{ij}^T\right) & \text{if } k = i \\ -2\mathbf{R}_k \log\left(\mathbf{R}_k^T\,\mathbf{R}_i\,\hat{\mathbf{R}}_{ij}\right) & \text{if } k = j \\ 0 & \text{o.w.} \end{cases} \tag{18}$$

Next consider for $(i,j) \in \mathcal{E}$, $k \in \mathcal{V}$ let $f = \frac{1}{2}\|\hat{\mathbf{t}}_{ij} - \mathbf{R}_i^T(\mathbf{t}_j - \mathbf{t}_i)\|^2$. Yi Ma shows in [19] that for a function $f : SO(3) \to \mathbb{R}$ the gradient at $\mathbf{R} \in SO(3)$ is given by

$$grad\, f(\mathbf{R}) = f_{\mathbf{R}} - \mathbf{R} f_{\mathbf{R}}^T \mathbf{R} \tag{19}$$

where $(f_{\mathbf{R}})_{ij} = \frac{\partial f}{\partial \mathbf{R}_{ij}}$. Using this equation, we find

$$grad\, f(\mathbf{R}_k) = \begin{cases} -\mathbf{R}_k \Big[\mathbf{R}_k^T (\mathbf{t}_j - \mathbf{t}_k) \hat{\mathbf{t}}_{i\,j}^T & \text{if } k = i \\ \quad -\hat{\mathbf{t}}_{i\,j} (\mathbf{t}_j - \mathbf{t}_k)^T \mathbf{R}_k \Big] & \\ 0 & \text{o.w.} \end{cases} \tag{20}$$

Finally, the gradient in $\mathbb{R}^3$ is given by

$$grad\, f(\mathbf{t}_k) = \begin{cases} \left(\mathbf{t}_k + \mathbf{R}_k \hat{\mathbf{t}}_{i\,j} - \mathbf{t}_j \right) & \text{if } k = i \\ \left(\mathbf{t}_k - \mathbf{R}_i \hat{\mathbf{t}}_{i\,j} - \mathbf{t}_i \right) & \text{if } k = j \\ 0 & \text{o.w.} \end{cases}$$

C. Product Manifold

In this section we consider manifolds that can be described as products of two or more simple manifolds.

Theorem 2: Let $(M_1, g^{(1)})$ and $(M_2, g^{(2)})$ be two Riemannian Manifolds and define the product manifold (M, g) as

$$M = M_1 \times M_2 \tag{21}$$

$$g_p(\xi, \zeta) = g_{p_1}^{(1)}(\xi_1, \zeta_1) + g_{p_2}^{(2)}(\xi_2, \zeta_2) \tag{22}$$

for all $p = (p_1, p_2) \in M$ and all $\xi = (\xi_1, \xi_2),\ \zeta = (\zeta_1, \zeta_2) \in T_pM$. Consider a function $f \in C^\infty(M)$ and for all $(p, q) \in M$ define

$$f_1^q : M_1 \to \mathbb{R}, \quad a \mapsto f(a, q)$$
$$f_2^p : M_2 \to \mathbb{R}, \quad b \mapsto f(p, b).$$

Then for all $(p, q) \in M$

$$grad\, f(p, q) = \left(grad\, f_1^q(p), grad\, f_2^p(q) \right).$$

Proof: Fix $(p, q) \in M$ and consider an arbitrary tangent vector $\xi = (\xi_1, \xi_2) \in T_{(p,q)}M$. Choose a parameterized path $(\gamma_1, \gamma_2) = \gamma : [0, 1] \to M$ such that ξ_1 is tangent to $\gamma_1(t)$ at $p = \gamma_1(t_0)$ and ξ_2 is tangent to $\gamma_2(t)$ at $q = \gamma_2(t_0)$ (and thus ξ is tangent to $\gamma(t)$ at $(p, q) = \gamma(t_0)$).

The following lemma is necessary for the proof. The proof of this lemma can be found in [22].

Lemma 1: $\frac{d}{dt} f(\gamma(t))|_{t_0} = \frac{d}{dt} f(\gamma_1(t), \gamma_2(t_0))\big|_{t_0} + \frac{d}{dt} f(\gamma_1(0), \gamma_2(t))\big|_{t_0}$.

Recall the following relation,

$$g(grad\, f(p, q), \xi) = \frac{d}{dt} (f \circ \gamma)\big|_{t_0}. \tag{23}$$

Using lemma 1 we can rewrite the right hand side of equation 23 as

$$\frac{d}{dt} (f_1^q \circ \gamma_1)|_{t_0} + \frac{d}{dt} (f_2^p \circ \gamma_2)|_{t_0}.$$

Using the relation in equation 23 again, this time for the individual Riemannian Manifolds, we find

$$g(grad\, f(p, q), \xi) = g^{(1)}(grad\, f_1^q(p), \xi_1) + g^{(2)}(grad\, f_2^p(q), \xi_2). \tag{24}$$

Now consider $grad\, f(p, q) = (A, B) \in T_{(p,q)M}$ for some unknown $A \in T_pM,\ B \in T_qM$. Using the definition of g in equation 22 equation 24 can be rewritten

$$g^{(1)}(A, \xi_1) + g^{(2)}(B, \xi_2) = g^{(1)}(grad\, f_1^q(p), \xi_1) + g^{(2)}(grad\, f_2^p(q), \xi \tag{25}$$

Equation 25 is true for all $\xi_1 \in T_pM$ and all $\xi_2 \in T_qM$ and so

$$A = grad\, f_1^q(p)$$
$$B = grad\, f_2^p(q).$$

■

As the number of Manifolds increases, the notation $grad\, f_1^q(p)$ becomes more cumbersome. We will therefore simplify the notation by writing $grad\, f(p)$ when we mean $grad\, f_1^q(p)$ any time the manifold the gradient found with respect to is made obvious by the argument p and q is clear from context.

The following corollary is a direct consequence of Theorem 2.

Corollary 1: Given a set of n Riemannian Manifolds $\{(M_i, g_i)\}_{i=1}^n$. If the product Riemannian metric on the product manifold (M, g)

$$M = M_1 \times \cdots \times M_n \tag{26}$$

is defined as

$$g_p(X, Y) = g_1(X_1, Y_1) + \cdots + g_n(X_n, Y_n) \tag{27}$$

for $p = (p_1, \ldots, p_n) \in M$ and $X = (X_1, \ldots, X_n), Y \in T_pM = T_{p_1}M_1 \times \ldots T_{p_n}M_n$, then

$$grad\, f(p) = \left(grad\, f(p_1), \ldots, grad\, f(p_n) \right). \tag{28}$$

It is also immediately obvious from the definition, that if the product Riemannian metric is defined as in (28) then geodesics on M will be the product of geodesics on the M_i's. Parallel transport on M is then given by parallel transport on each individual M_i.

The function we want to minimize, shown in (2) is a function from M to $\mathbb{R}$ where M is the product manifold given by

$$M := \left(SO(3) \times \mathbb{R}^3 \right)^{|\nu(k)|} \tag{29}$$

It can be shown that for $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_k, \mathbf{t}_k) \in M$, the tangent space of the product manifold at p is given by

$$T_pM = (T_{\mathbf{R}_1} SO(3) \times T_{\mathbf{t}_1} \mathbb{R}) \times \cdots \times (T_{\mathbf{R}_k} SO(3) \times T_{\mathbf{t}_k} \mathbb{R}) \tag{30}$$

where $k = |\nu|$.

We define the Riemannian metric on the product manifold as follows. For $p = (\mathbf{R}_1, \mathbf{t}_1, \ldots, \mathbf{R}_k, \mathbf{t}_k) \in M$ where $k = \#(\nu)$, and $X = (X_{\mathbf{R}_1}, X_{\mathbf{t}_1}, \ldots, X_{\mathbf{R}_k}, X_{\mathbf{t}_k}), Y \in T_pM$

$$g_p(X, Y) := \sum_{i \in \nu} \left(g_{\mathbf{R}_i}(X_{\mathbf{R}_i}, Y_{\mathbf{R}_i}) + < X_{\mathbf{t}_i}, Y_{\mathbf{t}_i} > \right) \tag{31}$$

This meats the conditions of Corollary 1 and so for $p = (\mathbf{R}_0^0 1, \mathbf{t}_{0,1}^0, \ldots, \mathbf{R}_(^0 k), \mathbf{t}_{0,k}^0) \in M$

$$grad\, f(p) = \left(grad\, f(\mathbf{R}_1^0, \mathbf{t}_{0,1}^0, \ldots, \mathbf{R}_k^0, \mathbf{t}_{0,k}^0) \right) \qquad (32)$$

where $grad\, f(\mathbf{R}_i^0)$ and $grad\, f(\mathbf{t}_{0,i}^0)$ are given by equations 18 and 20 respectively.

References

[1] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, June 2003.

[2] K. Ni, D. Steedly, and F. Dellaert, "Tectonic SAM: Exact, out-of-core, submap-based SLAM," in *IEEE International Conference on Robotics and Automation*, 2007, p. 16781685.

[3] L. A. A. Andersson and J. Nygards, "C-SAM : Multi-robot SLAM using square root information smoothing," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 2798–2805.

[4] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *IEEE International Conference on Robotics and Automation*, 2010.

[5] V. Indelman, P. Gurfil, E. Rivlin, and H. Rotstein, "Distributed vision-aided cooperative localization and navigation based on three-view geometry," in *IEEE Aerospace Conference*, 2011.

[6] R. Sharma and C. Taylor, "Cooperative navigation of mavs in gps denied areas," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008.

[7] V. Caglioti, A. Citterio, and A. Fossati, "Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach," in *IEEE Workshop on Distributed Intelligent Systems*, 2006.

[8] A. Martinelli, F. Pont, and R. Siegwart, "Multi-robot localization using relative observations," in *IEEE International Conference on Robotics and Automation*, 2005.

[9] S. Roumeliotis and G. Bekey, "Distributed multirobot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781 – 795, oct 2002.

[10] A. Howard, M. Matark, and G. Sukhatme, "Localization for mobile robot teams using maximum likelihood estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, 2002.

[11] E. Nerurkar, S. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 1402–1409.

[12] R. Aragues, L. Carlone, G. Calafiore, and C. Sagues, "Multi-agent localization from noisy relative pose measurements," in *IEEE International Conference on Robotics and Automation*, 2011.

[13] K. Y. Leung, T. Barfoot, and H. Liu, "Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 62 –77, 2010.

[14] J. Knuth and P. Barooah, "Distributed cooperative vision-based localization of mobile robot teams," in *47th Annual Allerton Conference on Communication, Control and Computing*, September 2009.

[15] N. Trawny, X. Zhou, K. Zhou, and S. Roumeliotis, "Interrobot transformations in 3-d," *IEEE Transactions on Robotics*, vol. 26, no. 2, April 2010.

[16] M. Monda and C. Woolsey, "Pose estimation from visual measurements using the epipolar constraint," in *IEEE Conference on Decision and Control*, 2010.

[17] A. W. Joshi, *Elements of Group Theory for Physicists*, 3rd ed. John Wiley & Sons Inc., 1982.

[18] R. Tron and R. Vidal, "Distributed image-based 3-d localization of camera sensor networks," in *IEEE Conference on Decision and Control*, 2009.

[19] Y. Ma, J. Košecká, and S. Sastry, "Optimization criteria and geometric algorithms for motion and structure estimation," *International Journal of Computer Vision*, vol. 44, pp. 219–249, 2001.

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

[21] K. V. Mardia and P. E. Jupp, *Directional Statistics*, ser. Wiley series in probability and statistics. Wiley, 2000.

[22] None, "A temporary bib as a placeholder," in *TEMP*, 0 0000.

[23] J. Manton, "A globally convergent numerical algorithm for computing the centre of mass on compact lie groups," in *IEEE International conference on control, automation, robotics and vision*, 2004.