Outlier rejection for pose graph optimization

Joseph Knuth and Prabir Barooah

***Abstract*—We propose an outlier rejection algorithm that functions as a preprocessing step for a pose graph collaborative localization algorithm Outliers in both types of relative pose measurements, from one time to the next for a given robot, and between robots at a given time, are considered. Outliers are identified using only the information contained in the measurements. In particular, no a-priori distribution on the relative measurements is assumed, nor is any information about the absolute pose of the robots utilized. The outlier rejection algorithm exploits properties of pose measurements concatenated over simple cycles in the measurement graph to define an edge consistency metric such that large values are indicative of the presence of an outlier. A hypothesis test approach is then utilized to identify the likely set of outlying measurements. Simulations utilizing the proposed outlier rejection algorithm, along with the collaborative localization algorithm developed in [12] are presented. The outlier rejection algorithm is shown to successfully identify up to 95% of the outliers in the scenario considered, and successfully mitigate the effect of outliers on collaborative localization.**

I. Introduction

In recent years two distinct, though not necessarily mutually exclusive, topics have dominated much of the literature on robot localization, SLAM (simultaneous localization and mapping) and CL (collaborative localization). In each problem, the inclusion of outliers, that is, measurements inconsistent with the set of measurements as a whole, can prove catastrophic to the accuracy of the location estimates.

A subset of the collaborative localization literature specifically deals with the case when noisy relative measurements between a single robot from one time to the next (inter-time relative measurements), and measurements between robots at a given time (inter-robot relative measurements) are utilized to estimate the absolute position and orientation, or pose, of each robot. In such a case, the collaborative localization problem can be stated as an optimization problem over a measurement graph. These algorithms are referred to as pose graph optimization collaborative localization algorithms [12, 10, 1, 7, 18, 5]. Similarly, when the SLAM problem involves inter-robot relative pose measurements, as well as pose measurements with respect to landmarks, the problem is referred to as pose graph SLAM (see [9] and references therein). In either case, the presence of an outlying measurement can lead to grossly inaccurate estimates. This had led to research in both outlier mitigation, and outlier identification and rejection, the latter of which is the topic of this paper.

Many collaborative localization and graph SLAM algorithms attempt to detect data association errors or falsely identified loop closure events. In particular, [17, 16, 13, 2, 3] each attempt to perform simultaneous localization and mapping while remaining robust to outliers. However, since outliers are not explicitly rejected by these algorithms, estimation accuracy is still adversely affected. An outlier identification algorithm is thus useful as a preprocessing step. The set of measurements from which the outliers have been removed can then be utilized by both the robust algorithms listed above, as well as non-robust localization algorithms. In this work we present such a pre-processing algorithm.

The paper [21] is most closely related to our work; they too consider the problem of identifying outliers prior to utilizing them in localizing. The authors of [21] utilize a least squares approach to pose graph optimization, with the inclusion of an additional state for each measurement indicating whether that measurement should be included. The optimal solution to the modified pose graph problem indicates what measurements are likely to be outliers. Though this algorithm is successful in rejecting outliers, it requires an increase in the dimensionality of the optimization problem equal to the number of measurements. When the number of measurements is large, this may be undesirable.

We propose a novel outlier rejection algorithm that functions as a preprocessing step for pose graph collaborative localization. Unlike the method in [21], it entails no increase in the dimension of the pose graph optimization problem. Though the algorithm presented here is equally suitable for both pose graph collaborative localization and graph SLAM, for ease of exposition, we will restrict our focus to only the collaborative localization scenario. We assume all robots are equipped with proprioceptive sensors (vision, IMU, etc.) allowing each robot to measure its change in pose. 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 inter-robot relative pose between pairs of robots. Finally, both inter-time and inter-robots measurements may be corrupted by outliers. For the ease of exposition, we assume that all measurements are of the relative pose. Extending the algorithm to utilize measurements of the relative orientation is straight forward and will be studied in future work.

The proposed algoritm identifies outliers using only the information contained in the relative measurements. In particular, no a priori distribution on the relative measurements is assumed, nor is any information about the absolute pose of the robots utilized. The outlier rejection algorithm exploits properties of pose measurements concatenated over simple cycles in the measurement graph to define an edge consistency metric such that large values are indicative of the presence of an outlier. A hypothesis test approach is then utilized to identify the likely set of outlying measurements. In addition, we indicate how our algorithm can utilize a sliding window approximation. The use of such an approximation constitutes a trade-off between computation time and outlier rejection performance.

In Section II-A a brief review of the collaborative localization problem is provided, then in Section II-B the outlier identification and rejection problem is explicitly stated. Then is Section III the pose graph outlier rejection algorithm is developed. Finally, in Section IV, simulations utilizing the proposed outlier rejection algorithm in conjunction with the collaborative localization algorithm developed in [12] are presented. The outlier rejection algorithm is shown to successfully identify up to 95% of the outliers in the scenario considered, and successfully mitigate the effect of outliers on the collaborative localization problem. In addition, simulations are presented verifying the simplifying assumptions necessary to make use of hypotheses testing for the method of outlier identification.

II. Problem statement

A. Pose Graph Collaborative Localization

Consider a group of r mobile robots indexed by $i = 1, \ldots, r$. Time is measured by a discrete counter $k = 0, 1, 2, \ldots$. Each robot i is equipped with a local, rigidly attached reference frame, called *frame* i. Localization of robot i consists of estimating the Euclidean transformation $\mathbf{T}_i \in SE(3)$ that relates a robot's local reference frame to an absolute reference frame common to all robots. This transformation is referred to as the robot's *absolute pose*.

Measurements of a robot's absolute pose are either not available or only rarely available. Instead, we assume that each robot is equipped with sensors (such as inertial sensors or vision based sensors) such that, at every time k, a robot is able to obtain a inter-time relative pose measurement: a measurement of the transformation between the previous and the current pose. In addition, each robot is equipped with exteroceptive sensors so that the robot is able to uniquely identify each robot it can "see" (within some sensing radius), and obtain a relative measurement for each such robot with respect to itself. For ease of exposition, we will restrict all inter-robot measurements to be of the relative pose. We call these measurements inter-robot relative measurements.

The *collaborative localization problem* is to estimate the absolute pose of every robot by utilizing both the inter-time and inter-robot measurements. The situation above is best described in terms of a directed, time-varying *graph* $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$ that shows how the noisy relative measurements relate to the absolute pose of each robot at every time step. The graph is defined as follows. For each robot $i \in \{1, \ldots, r\}$ and each time $t \leq k$, a unique index (call it u) is assigned to the pair (i, t). This set of indices define the set $\breve{\mathcal{V}}(k)$ and the node set of the graph is defined as $\mathcal{V}(k) := \breve{\mathcal{V}}(k) \cup \{0\}$. We then refer to the reference frame attached to robot i at time t as *frame* u. Node u is associated with the absolute pose of robot i at time t relative to the common reference frame. We call these poses *node variables* and denote them $\mathbf{T}_u$. We will also refer to the orientation $\mathbf{R}_u$ and position $\mathbf{t}_u$ components of the pose $\mathbf{T}_u$ as node variables. The common reference frame with respect to which all node variables are expressed is associated with node 0. Estimating the node variables is equivalent to determining the robots' poses with respect to frame 0 (the reference frame associated with node 0). Node 0 is therefore called the *reference node*.

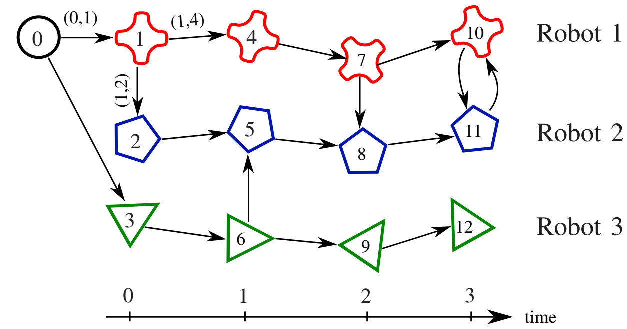

Fig. 1. Snapshot of a measurement graph at time $k = 3$ for a group of three robots when all measurements are of the relative pose. Each (robot,time) pair is labeled with the corresponding node index from $\mathcal{V}(3)$. Arrows indicate edges, i.e., relative measurements, in $\mathcal{E}(3)$. Robots 1 and 3 had GPS measurements at the initial time $k = 0$. Thereafter, no other GPS measurements were available.

The set of *directed edges* at time k, denoted $\mathcal{E}(k)$, corresponds to the noisy inter-time and inter-robot measurements collected up to time k. That is, suppose robot i is able to measure robot j's relative pose at time $\bar{k}$, and let u, v be the nodes corresponding to robots i, j at time $\bar{k}$, respectively. Then the edge $e = (u, v)$ will be in $\mathcal{E}(k)$ for all $k \geq \bar{k}$. Similarly, each inter-time relative pose measurements of a robot also creates an edge in the graph. The noisy relative measurement associated with each edge $e = (u, v) \in \mathcal{E}(k)$ is denoted by $\hat{\mathbf{M}}_e$.

The graph $\mathcal{G}(k)$ is called the *measurement graph* at time k; see Figure 1 for an example. We make the following assumption to ensure at least one estimate exists for every robot at each time k:

Assumption 1. *Each robot has access to an estimate of its current absolute pose at time* 0.

Under this assumption, an estimate for the pose of robot i at time time k can be computed by composing the inter-time relative pose measurements obtained by robot i up to time k. This estimate is equivalent to robot i performing dead-reckoning. *The goal of collaborative localization is to fuse information available from all edges in the graph* $\mathcal{G}(k)$ *to obtain estimates of the robots' poses at time* k *that is more accurate than that possible from dead reckoning, which only uses single paths from the reference to the current robot pose.*

B. Outlier Rejection

Consider time varying measurement graph $\mathcal{G}(k) = (\mathcal{V}(k), \mathcal{E}(k))$ as described in section II-A. Let $\left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$

denote the corresponding set of inter-robot and inter-time relative measurements. An *outlier* is a measurement $\hat{\mathbf{M}}_{e'} \in \left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$ that is inconsistent with the remaining measurements. An outlier is most often a grossly inaccurate measurement in a set of measurements that are relatively accurate (for which the noise is small). In such a case, if $\hat{\mathbf{M}}_{e'}$ is an outlier, we expect the estimate of the node variables corresponding to the measurement graph $(\mathcal{V}(k), \mathcal{E}(k)\backslash\{e'\})$ to be more accurate then the estimates corresponding to $\mathcal{G}(k)$, in which the outlier is still included. The *outlier rejection problem* is to identify and remove such outliers using only the information provided by the measurements, and no information about the node variables. In particular, any knowledge of a prior distribution on the value of the node variables is not available.

III. Outlier Rejection Algorithm

A. The First Cut

Consider a set of robots attempting to perform collaborative localization using a pose graph optimization CL algorithm in the presence of outliers when all measurements are of the relative pose. Let $G(k) = (\mathcal{V}(k), \mathcal{E}(k))$ denote the corresponding measurement graph and $\left\{\hat{\mathbf{M}}_e\right\}_{e\in\mathcal{E}(k)}$ the set of all inter-robot and inter-time relative measurements.

A *simple cycle* is an ordered collection of edges $c = (e_0, \cdots, e_\ell)$ such that if, for each i, e_i is an edge from node a_i to node b_i, denoted by $e_i = (a_i, b_i)$, then

- $a_i = b_{i-1},\ \ i = 1, \cdots, \ell,$
- $a_0 = b_k,$
- $a_i \neq a_j,\ \ \forall i \neq j.$

We will refer to the set of all such cycles in the graph $\mathcal{G}(k)$ as $\mathcal{C}(k)$. For a measurement graph in which all measurements are of the relative pose, composing noise-free measurements along any cycle yields the identity. To utilize this fact, for a simple cycle $c = (e_0, \cdots, e_\ell) \in \mathcal{C}(k)$ define the *cycle measurement* $\hat{\mathbf{M}}_c = \hat{\mathbf{M}}_{e_0}\hat{\mathbf{M}}_{e_1}\cdots\hat{\mathbf{M}}_{e_\ell}$. Since $\hat{\mathbf{M}}_c = id \in SE(3)$ whenever the measurements are noise free, a suitable distance metric on the product manifold $(SO(3)\times\mathbb{R}^3)$ provides a measure of the noise encountered in cycle c. Towards this end, we define the *cycle consistency cost* $D_C : \mathcal{C}(k) \to \mathbb{R}^+$ as

$$D_C(c \in \mathcal{C}(k)) = \frac{\sqrt{d^2(id_{SO(3)}, \hat{\mathbf{R}}_c) + \|\hat{\mathbf{t}}_c\|^2}}{|c|} \tag{1}$$

where $\hat{\mathbf{M}}_c = (\hat{\mathbf{R}}_c, \hat{\mathbf{t}}_c)$ and $d(\cdot\,,\cdot)$ is the Riemannian distance [4] given by

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

While $D_C(c)$ provides a measure of the average noise encountered along the cycle $c \in \mathcal{C}(k)$, using $D_C(c)$ alone, little can be said about the accuracy of any particular measurement. Instead, we will consider the cycle consistency costs for all cycles containing a given edge of interest. Let $\mathcal{C}_e(k) \subset \mathcal{C}(k)$ denote the set of all cycles that include edge e. The *edge consistency cost* $D : \mathcal{E}(k) \to \mathbb{R}$ is then defined as

$$D(e \in \mathcal{E}(k)) = \min_{c\in\mathcal{C}_e(k)} \{D_C(c)\}\,. \tag{2}$$

$D(e)$ provides a measure of the noise in measurement $\hat{\mathbf{M}}_e$ based only on other measurements in the graph.

As the number of measurements in a graph $\mathcal{G}(k)$ grow, finding all cycles, and subsequently the value of $D(e \in \mathcal{E}(k))$, can become infeasible. Instead, we will will consider some subset found through a depth first search (DFS) on the graph [8]. In particular, let m indicate the number of edges in $\mathcal{E}(k)$ and define a tuning parameter $M > 0$. We then find $m\,M$ random cycles by using the DFS. The set of cycles so found is denoted by $\hat{\mathcal{C}}(k) \subset \mathcal{C}(k)$. We then consider the approximate edge consistency cost $\hat{D}(e \in \mathcal{E}(k))$ given by

$$\hat{D}(e) = \min_{c\in\hat{\mathcal{C}}_e(k)} \{D_C(c)\} \tag{3}$$

where $\hat{\mathcal{C}}_e(k) \subset \hat{\mathcal{C}}(k)$ is the set of all cycles found that contain the edge e.

For the algorithm to perform well, the topology of the graph, the number of outliers, and the value of the tuning parameter M should be such that the following condition is satisfied: If $e \in \mathcal{E}(k)$ is not an outlier, then there exists a cycle $c \in \hat{\mathcal{C}}_e(k)$ such that every edge in $\hat{\mathcal{C}}_e(k)$ is not an outlier. When this condition is satisfied, if a large value of $D_C(c)$ indicates the presence of an outlier in the cycle c, then we expect $\hat{D}(e)$ to be large if and only if e itself is an outlier.

To identify outliers, a hypothesis test of the set $\left\{\hat{D}(e)\,|\,e \in \mathcal{E}(k)\right\}$ will be utilized. Though the values of $\hat{D}(e)$ are not $i.i.d.$ (independent, identically distributed) we will make the simplifying assumption that they in fact are i.i.d. Further, we choose the log-normal distribution to describe the distribution of $\left\{\hat{D}(e)\,|\,e \in \mathcal{E}(k)\right\}$. Evidence supporting this choice of distribution will be presented in Section IV-A. Finally the set $\mathcal{S}(k) = \left\{\log(\hat{D}(e))\,|\,e \in \mathcal{E}(k)\right\}$ is considered. Under the simplifying assumption that the values $\hat{D}(e)$ are distributed $i.i.d.$ log-normal, the set $\mathcal{S}(k)$ will be distributed $i.i.d.$ normal. The one sided version of Grubbs' test for outliers [6] can then be used to identify likely outliers in the measurements as follows. Given the data set $S(k)$, we say a value $s \in S(k)$ is an *outlier (in distribution)* if it is not distributed according to the same $i.i.d.$ normal distribution describing the probability of seeing the other values in $S(k)$. The null hypothesis, H_0, is that there are no large outliers in the set $S(k)$. Here "large" indicates we are only considering outliers to the right of the mean. Outliers that are very negative would not be rejected. The alternative hypothesis is that the largest value in $S(k)$ is an outlier (in distribution). Define the one sided Grubbs' test statistic as

$$G = \frac{s_{\max} - \bar{s}}{\sigma_s}$$

where $s_{\max}$ denotes the maximum value in $S(k)$, and $\bar{s}$, σ_s are the sample mean and sample standard deviation of $S(k)$

respectively The null hypothesis is rejected at a significance level of α if

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/N,N-2}}{N-2+t^2_{\alpha/N,N-2}}}$$

where $t_{\alpha/N,N-2}$ denotes the upper critical value of the t-distribution with $N-2$ degrees of freedom and a significance level of α/N.

If the null hypothesis is rejected, $s_{\max}$ is removed from $S(k)$ and the hypothesis test is repeated until the null hypothesis can not be rejected. Each time an outlier (in distribution) is removed from the set $S(k)$, the edge that generated that particular value is also removed from the graph, and the measurement is discarded as a suspected outlier.

B. Second Cut: Sliding Window Approximation

Straightforward application of the method described in the previous section is only possible up to a certain time, beyond which the size of the graph makes computations infeasible (m becomes too large). Under such a condition, a *sliding window* approximation can be used. Sliding window approximation is commonly used in both pose graph CL and graph SLAM (see [19, 20, 14]). In this section we briefly review the sliding window approximation and provide a modification under which the sliding window approximation can be used in the outlier identification problem.

The sliding window measurement graph at time k is given by removing all measurements that occurred before time $k-s$. If left at this point, often the resulting graph will be disconnected, as all edges leading to node 0 may have been removed. To reconnect the graph, the most recent node variable estimates for the nodes introduced at time $k-s$ (the earliest nodes still in the graph) are used as measurements between those nodes and node 0. Specifically, the sliding window measurement graph $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ where

$$\begin{aligned}
\mathcal{V}_s(k) &= \Big(V(k) \setminus \mathcal{V}\big(k-(s+1)\big)\Big) \cup \{0\} \\
\mathcal{E}_s(k) &= \Big(E(k) \setminus \mathcal{E}\big(k-(s+1)\big)\Big) \\
&\quad \cup \{e=(0,j) \,|\, j \in \mathcal{V}(k-s) \setminus \mathcal{V}(k-(s+1))\}.
\end{aligned}$$

and the additional edges $(0,j)$ in $\mathcal{E}_s(k)$ correspond to measurements given by the node variable estimates $(\hat{\mathbf{R}}_j, \hat{\mathbf{t}}_j) \in \Big\{(\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i)\Big\}_{\mathcal{V}(k-1)\setminus\mathcal{V}(k-(s+1))}$ found using a pose graph optimization collaborative localization algorithm.

When utilizing the sliding window approximation to identify outliers, a simple modification is necessary. Let $\mathcal{G}_s(k) = (\mathcal{V}_s(k), \mathcal{E}_s(k))$ denote a sliding window pose graph. Rather then attempting to identify outliers in $\mathcal{G}_s(k)$, a reduced graph $\mathcal{G}_r(k) = (\mathcal{V}_r(k), \mathcal{E}_r(k))$ is used, where $\mathcal{V}_r(k) = \mathcal{V}_s(k) \setminus \{0\}$ and $\mathcal{E}_r(k) = \mathcal{E}_s(k) \setminus \{(0,j) \,|$ the edge $(0,j)$ corresponds to an estimate of the node variable, not a true measurement $\}$. In simple terms, we remove the additional edges connected to 0 added in the construction of the sliding window pose graph. This reduction of the sliding window graph is necessary as, for sufficiently large k, the measurements (really node variable estimates) corresponding to these edges are expected to be more noisy then the inter-robot and inter-time measurements. In fact, in many cases the uncertainty in these node variable estimates will grow without bound while the uncertainty in inter-robot and inter-time measurements remains constant. Finally, the outlier rejection algorithm described in Section III is now applied to the reduced sliding window graph so constructed at every time index k. The original (non-reduced) sliding window graph, minus the identified outliers, can then be passed to any appropriate pose graph CL algorithm.

Remark 1. *Though only the centralized solution is discussed in this paper, the outlier rejection algorithm utilizing the sliding window approximation can be distributed in such a way as to have each robot perform its own outlier rejection using only measurements for neighboring robots. In particular, each robot will maintain contact with any robots it has measured (or that has measured it) since time $k-s$ and construct a local sliding window measurement graph consisting of inter-robot measurements between its self and its neighbors, as well as inter-time measurements for itself and for each of its neighbors. Outlier rejection can then be performed on the local sliding window measurement graph.*

Remark 2. *Although the outlier rejection algorithm is developed for the case when all measurement are of the relative pose, extending the algorithm to the case when measurements are of the relative orientation instead is straightforward. Utilizing the fact that noise free orientation measurements composed over cycles also yield the identity, the algorithm as presented need only be modified by redefining the cycle consistency cost function D_c as*

$$D_c(c \in \mathcal{C}(k)) = \frac{d(id_{SO(3)}, \hat{\mathbf{R}}_c)}{|c|}.$$

IV. Simulation

In this section we first present results justifying the use of the log-normal distribution to describe the edge consistency cost distribution, which is necessary in the application of Grubbs' test for outliers. Then in section IV-B we present simulations in which outliers are present during collaborative localization and the results of applying the outlier rejection algorithm.

A. Justification of the log-normal distribution

A pose measurement graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with 50 nodes and 202 edges was used in two simulations in which all edges correspond to relative pose measurements. Outliers were generated by corrupting 3% of the relative pose measurements by 10m in position and 90° in orientation (about a random axis). The robots moved at an average speed of $.1\text{m}$ per time step, so a 10m error is enough to make a measurement an outlier. The 90° error in the orientation measurement was chosen because it is commonly seen when utilizing vision-based sensors (see [15]). The average angular speed of the robots was 8° per time step, and so the error in orientation is

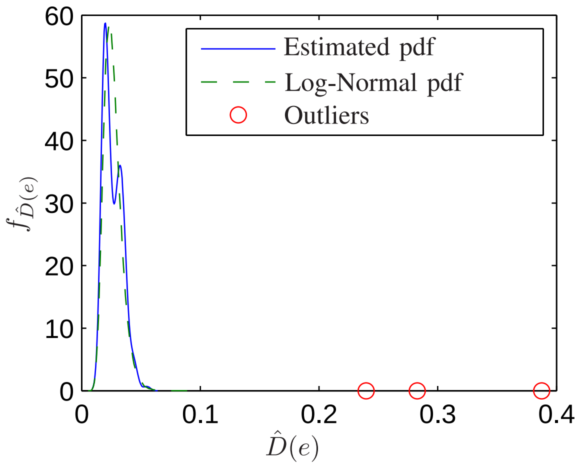

Fig. 2. A comparison between the estimated pdf for the values of $\hat{D}(e)$ as compared to the corresponding log-likelihood distribution pdf. The location of the outliers are also indicated.

also sufficiently large to be classified as an outlier. The tunning parameters M and α were set at 10 and 0.025 respectively.

The values of $\left\{\hat{D}(e)\,|\,e \in \mathcal{E}\right\}$ were computed for the graph $\mathcal{G}$. Gaussian kernel density approximation was then used to compute an estimate of the pdf describing the values found. In addition, the sample mean and variance were computed, and the log-normal density with the equivalent mean and variance was identified.

In Figure 2 both the estimated pdf, along with the corresponding log-normal density function can be seen. A comparison of the estimated and log-normal pdf shows that, while not an exact match, the approximate shape is adequately captured. Most importantly to the success of the hypothesis test based algorithm presented in Section III, the value of $\hat{D}(e)$ when e is out outlier is also an outlier for the identified log-normal distribution.

B. Outlier Rejection

We now present a set of simulations that provide some insight into the effectiveness of the proposed algorithm. First we define a set of convenient performance metrics. The position estimation error of robot i is defined as $\mathbf{e}_i(k) := \hat{\mathbf{t}}_i(k) - \mathbf{t}_i(k)$, where $\mathbf{t}_i(k)$ is its absolute position at k and $\hat{\mathbf{t}}_i(k)$ is the estimate. The bias in the position estimation error of robot i is defined as $\|\,\mathrm{E}[\mathbf{e}_i(k)]\|$, where $\|\cdot\|$ is the 2-norm and E denotes expectation. The standard deviation is defined as $\sqrt{\mathrm{Tr}\left(Cov(\mathbf{e}_i(k), \mathbf{e}_i(k))\right)}$, where $Cov(\cdot)$ stands for covariance. In each scenario described below, the bias and standard deviation in position estimation error is estimated through the use of a Monte Carlo simulation with 300 sample runs.

A group of 4 robots were simulated to move along distinct 3-D paths. Two robots were able to obtain noisy relative pose measurements at time k if the Euclidean distance between them at that time was less then $7\mathrm{m}$. The Riemannian pose graph optimization algorithm presented in [12] was used to perform the collaborative localization in three case. In the first simulation, all measurements were equally accurate, that is, no outliers were present. In the second simulation, outliers were generated by corrupting approximately 4% of the measurements by $90°$ in orientation (about a random axis) and $10\mathrm{m}$ in position. Finally, the same set of outlier induced measurements was considered, but the outlier rejection algorithm presented in Section III was utilized to identify and reject likely outliers. For this simulation the sliding window approximation described in Section III-B was utilized with a window length (s) of 3. The tunning parameters M and α were set at 10 and 0.025 respectively. The bias and standard deviation of the position estimation error, defined above, estimated using a 300-iteration Monte-Carlo simulation are presented in Figure 3. A summary of the average number of measurements, outliers, and false negatives/positives is reported in Table I.

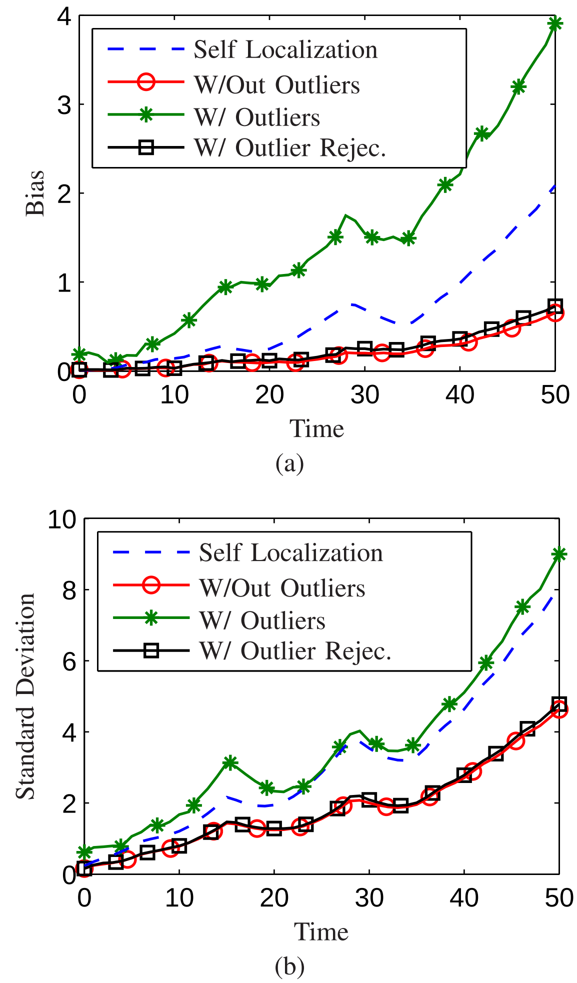

Fig. 3. A comparison of localization accuracy without outliers, with outliers and no rejection, and with outliers when the rejection algorithm is utilized. The (a) Bias and (b) Standard Deviation in position estimation error are estimated using a 300-iteration Monte-Carlo Simulation.

A number of observations can be made from the plots in

	Average	Percentage
Measurements	48.1	-
Outliers	1.85	3.7%
False Rejects	0.22	0.5%
False Accepts	0.087	5%

TABLE I
THE AVERAGE NUMBER OF MEASUREMENTS, OUTLIERS, MEASUREMENTS FALSELY REJECTED, AND OUTLIER MEASUREMENTS FALSELY ACCEPTED WHEN THE OUTLIER REJECTION ALGORITHM IS SIMULATED AS A PREPROCESSING STEP OF COLLABORATIVE LOCALIZATION. THE AVERAGES ARE WITH RESPECT TO ALL 300 MONTE-CARLO SIMULATIONS OVER ALL 50 TIME STEPS. THE PERCENTAGES ARE WITH RESPECT TO TOTAL MEASUREMENTS FOR OUTLIERS, NON-OUTLIER MEASUREMENTS FOR FALSELY REJECTED MEASUREMENTS, AND OUTLIER MEASUREMENTS FOR FALSELY ACCEPTED MEASUREMENTS.

Figure 3. The first observation is that the presence of outliers can cause the estimates found using collaborative localization to become *even less accurate* than those found using dead reckoning alone. To see why this is so, it is important to note that the collaborative localization algorithm used in this simulation is essentially a least squares optimization problem. As such, the algorithm is sensitive to outliers, to the extent that the estimated node variables may better reflect the information contained in the outliers, rather then the remaining, more accurate measurements.

The second observation to be made from Figure 3 is the considerable improvement to localization accuracy when the proposed outlier rejection algorithm is utilized before performing collaborative localization. In fact, accuracy is almost as good as that if no outliers were present. This is unsurprising as, from Table I it is evident that the majority of outliers were correctly identified, and very few additional measurements were falsely rejected. A sufficient number of relative measurements, approximately 96%, remained after the outlier rejection preprocessing; enough to allow the collaborative location algorithm to perform well.

V. SUMMARY AND FUTURE WORK

In this paper we presented an algorithm to identify and rejection outliers as a preprocessing step to pose graph optimization based collaborative localization. The algorithm utilized properties of relative pose measurements composed over cycles to develop a metric on the set of edges indicative of the presence of an outlier. A hypothesis test was then utilized to identify the likely set of outliers.

Simulations were presented that studied the effectiveness of the outlier rejection algorithm when utilized before performing collaborative location on a set of robots. It was shown that, while the presence of outliers can cause collaborative localization to perform even worse then dead reckoning, the outlier rejection algorithm succeeds in the task of removing outliers to such a extent that performance is increased nearly to that of the no-outlier case.

In addition, simulations were presented that explored the validity of the assumption that edge consistency costs were i.i.d. log-normal. The critical conclusion of this simulation was that, if an edge e' is an outlier, the corresponding approximate edge consistency cost $\hat{D}(e')$ is an outlier of the log-normal distribution with mean and variance given by the sample mean and variance of the set $\left\{\hat{D}(e)\right\}$. Because of this, we expect to correctly identify outlying measurements using Grubbs' test for outliers.

The algorithm was developed for the case when all measurements are of the relieve pose. However, it is important to note that the extension to rejection of orientation measurement outliers is trivial whenever all inter-robot measurements *include* an orientation measurement; see Remark 2. In fact, other measurement types are permissible and can be included in the collaborative location computations (as done in [11]), but will be neglected by the outlier rejection algorithm.

While it is clear that the number of edges affect the robustness of the outlier rejection algorithm, an in-depth study of the connection between the maximum number of outliers for which the rejection algorithm succeeds versus the topology of the measurement graph remains an active area for future work. Future studies will also attempt to extend the proposed outlier rejection scheme to other measurement types: position, bearing, and distance.

Finally, in addition to the centralized algorithm presented in Section III, a method to distributed the outlier rejection algorithm was also briefly outlined in Remark 1. In such a distributed scheme, each robot poses a different local measurement graph that is only a small part of the total graph. How the outlier rejection performs on these local graphs as compared to the full centralized solution presented here requires further study.

REFERENCES

[1] 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*, Shanghai, China, Mar 2011.

[2] Cesar Dario Cadena Lerma, John McDonald, John J Leonard, and José Neira. Place recognition using near and far visual information. In *World Congress*, volume 18, pages 6822–6828, 2011.

[3] Mark Cummins and Paul Newman. Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[4] Sylvestre Gallot, Dominique Hulin, and Jacques LaFontaine. *Riemannian Geometry*. Springer, 3rd edition, 2004. ISBN 3540204938.

[5] Giorgio Grisetti, Cyrill Stachniss, Slawomir Grzonka, and Wolfram Burgard. A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent. *Proc. of robotics science and systems (RSS)*, pages 1–8, July 2007.

[6] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[7] A. Howard, M.J. Mataric, and G.S. Sukhatme. Putting the 'i' in 'team': an ego-centric approach to cooperative localization. In *IEEE International Conference on Robotics and Automation*, 2003.

[8] Donald B Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4(1):77–84, 1975.

[9] Ayoung Kim and Ryan Eustice. Pose-graph visual slam with geometric model selection for autonomous underwater ship hull inspection. In *Intelligent Robots and Systems, 2009. IROS*

2009. IEEE/RSJ International Conference on*, pages 1559–1565. IEEE, 2009.

[10] J. Knuth and P. Barooah. Distributed collaborative localization of multiple vehicles from relative pose measurements. In *47th Annual Allerton Conference on Communication, Control, and Computing*, pages 314 –321, Allerton, IL, October 2009. doi: 10.1109/ALLERTON.2009.5394785.

[11] Joseph Knuth and Prabir Barooah. Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization. In *IEEE Conf. on Rob. and Auto. (ICRA)*, 2013.

[12] Joseph Knuth and Prabir Barooah. Collaborative localization with heterogeneous inter-robot measurements by Riemannian optimization. In *IEEE International Conference on Robotics and Automation*, May 2013.

[13] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[14] Seungwook Lim, Tae kyeong Lee, Seongsoo Lee, Shounan An, and Se young Oh. Adaptive sliding window for hierarchical pose-graph-based slam. In *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*, pages 2153–2158, Oct. 2012.

[15] Yi Ma, Jana Košecká, and Shankar Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.

[16] J.B. McDonald, M. Kaess, C. Cadena, J. Neira, and J.J. Leonard. 6-DOF multi-session visual SLAM using anchor nodes. In *European Conference on Mobile Robots, ECMR*, pages 69–76, Orebro, Sweden, Sep 2011.

[17] Edwin Olson, Matthew Walter, John Leonard, and Seth Teller. Single cluster graph partitioning for robotics applications. In *Proceedings of Robotics Science and Systems*, pages 265–272, 2005.

[18] S.I. Roumeliotis and G.A. Bekey. Distributed multirobot localization. *IEEE Transactions on Robotics and Automation*, 18(5): 781 – 795, oct 2002.

[19] Gabe Sibley, Gaurav S Sukhatme, and Larry Matthies. Constant time sliding window filter slam as a basis for metric visual perception. In *Proc. IEEE International Conference on Robotics and Automation Workshop*, pages 10–14, 2007.

[20] Gabe Sibley, Larry Matthies, and Gaurav Sukhatme. A sliding window filter for incremental slam. *Unifying perspectives in computational and robot vision*, pages 103–112, 2008.

[21] N. Sunderhauf and P. Protzel. Towards a robust back-end for pose graph slam. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1254–1261, May 2012.