IEEE Robotics and Automation Letters (RAL) paper presented at the
2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Direct Relative Edge Optimization, a Robust Alternative for Pose Graph Optimization

James Jackson[1], Kevin Brink[2], Brendon Forsgren[1], David Wheeler[3], Timothy McLain[1]

Abstract— **Pose graph optimization is a common problem in robotics and associated fields. Most commonly, pose graph optimization is performed by finding the set of pose estimates which are the most likely for a given set of measurements. In some situations, arbitrarily large errors in pose graph initialization are unavoidable and can cause these pose-based methods to diverge or fail especially in cases where global inputs become available after some time after initialization. This paper details the parameterization of the classic pose graph problem in a relative context, optimizing directly over relative edge constraints between vertices in the pose graph and not on the poses themselves. Unlike previous literature on relative optimization, this paper details relative optimization over an entire pose graph, instead of a subset of edges, resulting in greater robustness to arbitrarily large errors than the classic pose-based or prior relative edge-based methods. Several small-scale simulation comparison studies, along with single and multi-agent hardware experiments, are presented. Results point to relative edge optimization as a strong candidate for solving real-world pose graph optimization problems that contain large heading propagation or initialization errors.**

SLAM, Mapping, Multi-Robot Systems

I. Introduction

A pose graph is a data structure that encodes relative transform constraints between arbitrary poses of one or multiple agents. The nodes of a pose graph represent the pose of each node and the edges represent the relative transformations between nodes, both sequentially and non-sequentially [1]. When using nonlinear optimization techniques to determine the maximum-likelihood configuration of the pose graph, constraints can render the problem as over-constrained.

In the robotics community, pose graphs are commonly used to solve the full simultaneous localization and mapping (SLAM) problem. The construction of pose graphs to solve the SLAM problem is well researched and has been demonstrated extensively. Some notable examples include [1]–[4]. In these applications, pose graph optimization is used to obtain a maximum likelihood estimate for all the poses in the map. Different techniques have been proposed to improve the performance and robustness of pose graph optimization including graph reduction [5], [6], relaxation [7] incremental-pose parameterization [8]–[10] and relative parameterization [11]–[13].

One drawback of global-pose optimization is a strong dependence on the quality of the initialization point for the graph, and there are several instances when large initialization errors may arise. For example, small heading errors in a trajectory compounded over time can cause large errors in the initial pose estimates [9], [14]. This phenomenon is illustrated in Figure 1. Large initialization errors also result when an agent is initialized without global information but later gains global information during operation. In this case all initial pose estimates can be arbitrarily far from their true position. Finally, in multi-agent problems, the initial alignment of each agent's trajectory is unknown and can be arbitrarily far from truth. Kim et al. [15] proposed a method to mitigate this initial error between multiple agents, but requires additional complexity to handle edges between the graphs.

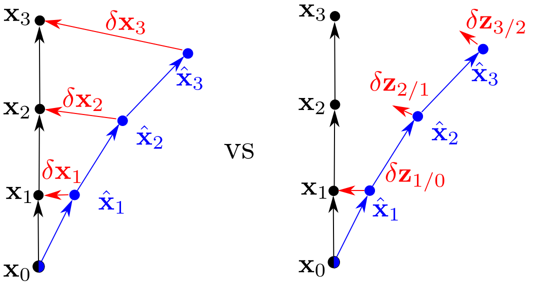

Fig. 1. Comparison of typical errors in global poses (left) vs individual edge constraints (right) given a trajectory with heading error.

Graph initialization is an active area of research and there are several prominent solutions that have been proposed [15]–[21]. Of particular note, [20] and [21] have shown the guaranteed optimization of pose graphs even given large initialization errors. These approaches restructure the cost function of a pose graph optimization problem into a globally convex form with constraints on the involved variables and use semi-definite programming techniques to guarantee the optimal solution.

In this paper, we discuss the benefits of optimizing directly over edge constraints, or *relative edge optimization* (REO), as opposed to the more popular global pose optimization (GPO) and we extend previously described relative approaches [11]–[13] to allow optimization over all edges in pose graph and eliminate the map tears that occur from optimizing over a local subset. We also show that posing the problem in a relative context side-steps the above mentioned initialization problems and keeps the problem well-conditioned even in the worst of these situations due to its more linear representation.

Performing pose graph optimization over all degrees of freedom of the agent and over all measurements can be computationally prohibitive over long distances with high sensor update rates. In cases where an IMU is available, the roll and pitch of an agent become locally observable so to reduce computational requirements sometimes only the locally-unobservable states are estimated using loop closures or global information in an optimization problem over $SE2 \times \mathbb{R}$ [22]–[24]. This optimization also often occurs on a reduced form of marginalized state and measurement information [22]–[24] to further reduce computational requirements. While the work we present here is designed to be applied in such a situation, it could also be applied to other SLAM problems, such as those in $SE3$.

The following sections first derive relative edge optimization and describe how to perform relative batch optimization over all edges in a cyclic graph. Comparisons between REO and GPO are then performed and discussed with the use of a simple simulation study. Finally, results from a hardware experiment with multirotor agents are shown and discussed.

II. Derivation

Pose graph optimization is often formulated as a least-squares optimization problem and this formulation can be derived using a classical least-squares optimization approach [25] or from a Bayesian perspective in a factor graph [26]. If the noise about graph edges is assumed to be Gaussian, both derivations ultimately lead to the following expression for the global cost function of the optimization given in [1]

$$\begin{aligned} F\left(\hat{\mathbf{z}},\bar{\mathbf{z}}\right) =& \bar{\mathbf{x}}_0^\top \Omega_0^{-1} \bar{\mathbf{x}}_0 + \sum_{i \in \mathcal{N}} \left(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\right)^\top \Omega_i^{-1} \left(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\right) \\ &+ \sum_i \sum_{j \in S_i} \left(\hat{\mathbf{z}}_{j/i} - \bar{\mathbf{z}}_{j/i}\right)^\top \Omega_{j/i}^{-1} \left(\hat{\mathbf{z}}_{j/i} - \bar{\mathbf{z}}_{j/i}\right), \end{aligned} \tag{1}$$

where $\mathbf{z}_{j/i}$ is the edge connecting nodes i and j, calculated as $\mathbf{x}_j \boxminus \mathbf{x}_i$; $\hat{[\cdot]}$, $\bar{[\cdot]}$ are the estimated and measured quantities, respectively; S_i is the set of nodes, x_j, connected to node x_i; $\mathcal{N}$ is the set of all nodes in the graph and Ω is the covariance, or weighting assigned to the respective measurements. Note that if $x_j \in S_i$, then $x_i \notin S_j$, else a $1/2$ weighting on $\mathbf{z}_{j/i}$ would be required to avoid double counting edge constraints.

The problem is then to find the optimal set of poses

$$\hat{\mathbf{x}}^* \quad = \quad \arg\min_{\hat{\mathbf{x}}} F\left(\bar{\mathbf{x}}, \hat{\mathbf{x}}, \bar{\mathbf{z}}, \hat{\mathbf{z}}\right). \tag{2}$$

To solve Eq. 2 in a global context the pose graph is initialized by first defining an initial global pose estimate for each vertex $\hat{\mathbf{x}}_i$ then determining the estimated edges $\hat{\mathbf{z}}_{j/i}$ and then optimizing using an appropriate method. In general, this is done using some variety of Gauss–Newton or Levenberg–Marquardt optimization. Significant work has been done in reducing the complexity of this problem [1], [6], [10] so that it can be performed in real time under realistic computational constraints.

Optimizing over $\hat{\mathbf{x}}$ in Eq. 2 inherently casts the optimization problem into a privileged coordinate frame. While this is often appropriate, when only relative information is available the initial global pose estimates can be arbitrarily far from their true position. If these systems then encounter global inputs, or are operating in a multi-agent environment with unknown intial configuration and encounter constraints between agents, the initial error can cause the Newton method to converge to a local minimum because of linearization error in pose-based Jacobians.

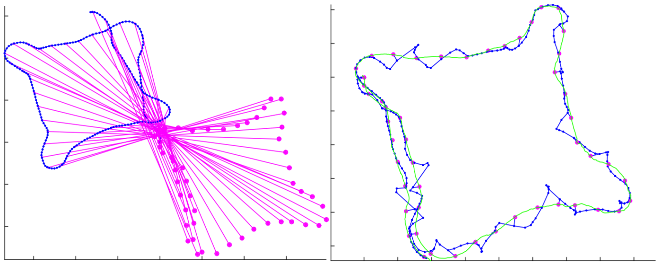

Fig. 2. Divergent behavior observed in global pose optimization of a multirotor flight around a baseball diamond due to poor initialization. Green lines indicate the original trajectory, magenta dots and lines indicate GPS measurements and their associated poses. The blue lines indicate the initial trajectory before optimization on the left and post-optimization on the right.

Figure 2 demonstrates this phenomenon in an exaggerated example. The figure shows results of global-pose optimization on a hardware experiment where a multirotor agent was first given only relative odometry measurements but later receives several delayed GPS inputs. Because no global information was initially available to the system, the agent was initialized with zero heading, although its true heading was approximately 180 degrees. The initial heading error induced large initial errors in global pose and the combination of poor linearization and strong GPS measurement constraints caused the optimization to converge to an incorrect local minima. While this example is exaggerated and there are methods intended to account for this [15]–[19], it illustrates the poor linearization characteristics of global poses, susceptibility to local minima, and a general lack of robustness when operating with sparse globally defined inputs.

Generally speaking, these situations can be difficult to detect and recover from, and are best avoided all together through either better a priori information, or more robust approaches such as the one suggested in this paper.

A. Derivation of Relative Edge Optimization

If one assumes the system only has measurements of transforms between nodes (i.e. $\bar{\mathbf{z}}_{j/i} = \mathbf{x}_j \boxminus \mathbf{x}_i$) with covariance $\Omega_{j/i}$, and that edges are only considered once, then Eq. 1 reduces to the weighted sum of edge constraints and can be rewritten as

$$\begin{aligned} F\left(\hat{\mathbf{x}},\bar{\mathbf{z}}\right) =& \sum_i \sum_{j \in S_i} \left(\left(\hat{\mathbf{x}}_j \boxminus \hat{\mathbf{x}}_i\right) - \bar{\mathbf{z}}_{j/i}\right)^\top \\ & \qquad \Omega_{j/i}^{-1} \left(\left(\hat{\mathbf{x}}_j \boxminus \hat{\mathbf{x}}_i\right) - \bar{\mathbf{z}}_{j/i}\right) \\ =& \sum_i \sum_{j \in S_i} \left\| \left(\left(\hat{\mathbf{x}}_j \boxminus \hat{\mathbf{x}}_i\right) - \bar{\mathbf{z}}_{j/i}\right) \right\|_{\Omega_{j/i}^{\;1}}. \end{aligned} \tag{3}$$

To further generalize the above case, pose-based portions of Eq. 1, $\left(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\right)^\top \Omega_i^{-1} \left(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\right)$, can be rewritten as relative constraints [5], [24], [27]. This makes the simplification in Eq. 3 quite reasonable. Further, if we reparameterize our state in terms of the estimated relative transform between nodes, $\hat{\mathbf{z}}_{j/i} = \hat{\mathbf{x}}_j \boxminus \hat{\mathbf{x}}_i$, as opposed to estimating the nodes themselves, Eq. 3 can be expressed as

$$F\left(\hat{\mathbf{z}}, \bar{\mathbf{z}}\right) = \sum_{i} \sum_{j \in S_i} \left\| \left(\hat{\mathbf{z}}_{j/i} - \bar{\mathbf{z}}_{j/i} \right) \right\|_{\Omega_{j/i}^{1}} . \tag{4}$$

Theorem II.1. *Given a pose graph that can be expressed purely in terms of full relative constraints (not including partial constraints such as range or bearing), the poses reconstructed from the relative optima arrived at by minimizing Eq. 4 starting at the initial node position $\mathbf{x}_0$ are identical to the global optima arrived at by minimizing Eq. 3.*

Proof: Let us assume that our estimated poses are at the optimal value, $\hat{\mathbf{x}} = \hat{\mathbf{x}}^* = \arg\min_{\hat{\mathbf{x}}} F\left(\hat{\mathbf{x}}, \bar{\mathbf{z}}\right)$ and define a compounding function

$$\hat{\mathbf{x}}_i = g_i^k \left(\hat{\mathbf{z}} \in \mathbf{E}_k, \mathbf{x}_0 \right) \tag{5}$$

that compounds some subset of edges $\mathbf{E}_k$ in the appropriate manner from an arbitrary origin (assume $\mathbf{x}_0$ without loss of generality) up to the node $\mathbf{x}_i$. This is illustrated with the green arrow in Figure 3 where the position of $\mathbf{x}_4$ is computed using the function g_4^k with $\mathbf{E}_k = \left(\mathbf{z}_{1/0}, \mathbf{z}_{4/1} \right)$. For non-trivial graphs, there could be multiple selections of $\mathbf{E}_k$ that could construct g_i^k, and these paths can be combined to generate the equation

$$\begin{aligned} \hat{\mathbf{x}}_i &= G_i^M \left(\hat{\mathbf{z}}, \mathbf{x}_0 \right) \\ &= \frac{1}{m} \sum_{\mathbf{E}_k \in M} g_i^k \left(\hat{\mathbf{z}}, \mathbf{x}_0 \right) \end{aligned} \tag{6}$$

where M is a set of possible edges that begin at $\mathbf{x}_0$ and terminate at $\mathbf{x}_i$ with cardinality $m = |M|$. At the optimum we have that $g_i^k \left(\hat{\mathbf{z}}^*, \mathbf{x}_0 \right) = \hat{\mathbf{x}}_i^* \, \forall \, \mathbf{E}_k$, (i.e. every possible edge sequence results in the optimal $\hat{\mathbf{x}}_i^*$ solution when evaluated at $\hat{\mathbf{z}}^*$). Therefore, $G_i^M \left(\hat{\mathbf{z}}^*, \mathbf{x}_0 \right)$ will also have the same optimal solution for $\hat{\mathbf{x}}_i$ regardless of which edge paths are used.

If we re-write Eq. 3 using Eq. 6 we get an expression for our global optimization function in terms of only the relative constraints and the initial pose

$$\begin{aligned} &F\left(\hat{\mathbf{z}}, \mathbf{x}_0, \bar{\mathbf{z}}\right) = \\ &\sum_{i} \sum_{j \in S_i} \left\| \left(\left(G_j^M \left(\hat{\mathbf{z}}, \mathbf{x}_0 \right) \boxminus G_i^M \left(\hat{\mathbf{z}}, \mathbf{x}_0 \right) \right) - \bar{\mathbf{z}}_{j/i} \right) \right\|_{\Omega_{j/i}^{1}} \end{aligned} \tag{7}$$

If we assume that we have optimized Eq. 7 with respect to edges (i.e. $\hat{\mathbf{z}} = \hat{\mathbf{z}}^* = \text{argmin}_{\hat{\mathbf{z}}} \left(F\left(\hat{\mathbf{z}}, \mathbf{x}_0, \bar{\mathbf{z}}\right) \right)$ then

$$\frac{\partial F}{\partial \hat{\mathbf{z}}} = 0,$$

which implies that

$$\frac{\partial F}{\partial \hat{\mathbf{x}}} \frac{\partial \hat{\mathbf{x}}}{\partial \hat{\mathbf{z}}} = 0,$$

and because $\partial \hat{\mathbf{x}} / \partial \hat{\mathbf{z}} \neq 0$, this implies that

$$\frac{\partial F}{\partial \hat{\mathbf{x}}} = 0.$$

The above proof shows that the cost functions Eq. 7 and Eq. 1 share the same minima, regardless of parameterizing with respect to pose or edges. This fact also allows for leeway in the selection of which paths, $\mathbf{E}_k$, are selected. The only requirement is that all edges in the graph are included, or else one cannot fully reconstruct Eq. 7. In practice, the choice of which edges are used could have a significant effect on the resulting Jacobians used in optimization. The choice of edges will therefore affect optimization performance, however the impact of this choice appears to be small compared to the nonlinearities induced by large errors in a pose-based optimization and regardless of loops/paths chosen, the global minima will remain unchanged.

In practice, actually solving Eq. 7 for arbitrary edge paths is not straightforward, especially if loops are present. Other relative approaches to loop closure [11], [12], use a sliding window, or an *active set* of edges that do not traverse the full loop. The advantage to these approaches is that the algorithms run in constant time, even at loop closure (because only a limited set of nodes in the loop are considered). However, these approaches result in map tears at the boundary of the active set of nodes that lead to global map inconsistency. If we instead construct M in such a way that $\mathbf{E}_k$ forms non-trivial loops that includes $\mathbf{z}_{j/i}$ for the evaluation of G_i^M and G_j^M, then we can rewrite Eq. 7 as a sum over loops rather than individual edge constraints and optimize each loop independently. This is possible because any common edges involved in the pose reconstruction of G_i^M and G_j^M are subtracted out in Eq. 7, and only a simple loop remains (see Figure 3). Because some edges may be traversed by more than one loop, edges must be weighted within a loop inversely proportional to the total number of loops in the optimization which traverse that edge. This ultimately leads to the final form of the cost function for relative edge optimization given as

$$F\left(\hat{\mathbf{z}}, \mathbf{x}_0, \bar{\mathbf{z}}\right) = \sum_{\mathcal{L} \in M} \sum_{j,i \in \mathcal{L}} \frac{1}{n_{j/i}} \left\| \hat{\mathbf{z}}_{j/i} - \bar{\mathbf{z}}_{j/i} \right\|_{\Omega_{j/i}^{1}} , \tag{8}$$

where M is the set of loops that must fully span the set of edges, $\mathcal{L}$ is a single loop and $n_{j/i}$ is the number of times the edge $z_{j/i}$ appears in M. The process of segmenting the graph into simple loops is illustrated in Figure 4. The graph is split into two simply-connected loops that share some common edges. The sectioning of the graph is arbitrary, and two potential options are shown in the figure.

Starting at Eq. 8, we can re-construct Eq. 7 by selecting any subset of loops that contain every $\mathbf{x}_i$ and rewriting edge estimates in terms of the compounding functions G_i^M (i.e. $\hat{\mathbf{z}}_{j/i} = G_j^M \boxminus G_i^M$). If we expand this and regroup on nodes $\mathbf{x}_i$, we arrive back to Eq. 7, which we know is equivalent to Eq. 3.

The cost function in Eq. 8 provides the user with significant flexibility and can make the optimization tractable. If all loops are considered, the optimization can become computationally expensive. However, users can select a subset of loops for edge-based optimization that maintains the global minima, and is also likely to provide good structure, accurate Jacobians, and tractable computational loads to the optimization algorithm. While this this paper does not

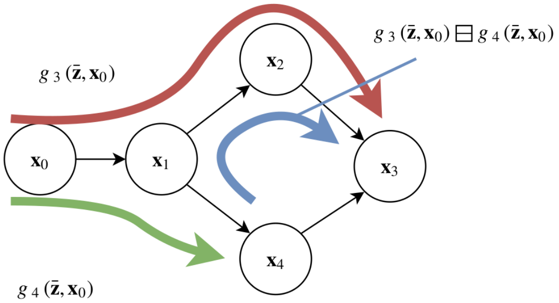
$g_3(\bar{\mathbf{z}}, \mathbf{x}_0)$
$g_3(\bar{\mathbf{z}}, \mathbf{x}_0) \boxminus g_4(\bar{\mathbf{z}}, \mathbf{x}_0)$
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$
$g_4(\bar{\mathbf{z}}, \mathbf{x}_0)$

Fig. 3. Subtracting two concatenation functions removes any common edges between the two paths, resulting in a simple loop. The blue path is the result of subtracting the green path from the red path and does not contain the common edge $\mathbf{z}_{1/0}$.

address selection of desired loops specifically, it will be demonstrated in Sec. III that even for an arbitrary selection of loops the edge-based optimization applied to Eq. 8 shows a dramatic increase in robustness when compared to pose-based optimization of Eq. 7. We now discuss the optimization of Eq. 8.

B. Solving REO

We wish to find the optimal update $\Delta\mathbf{z}^*$ to our initial edge estimate $\hat{\mathbf{z}}$. Letting $\hat{\mathbf{z}}^+_{j/i} = h\left(\hat{\mathbf{z}}_{j/i}, \Delta\mathbf{z}\right)$ where h properly considers the way that $\Delta\mathbf{z}$ maps into the edge update, we see

$$\begin{aligned} F\left(\hat{\mathbf{z}}^+, \bar{\mathbf{z}}\right) &= \sum_{\mathcal{L}\in M} \sum_{i,j\in\mathcal{L}} \frac{1}{n_{j/i}} \left\|\hat{\mathbf{z}}^+_{j/i} - \bar{\mathbf{z}}_{j/i}\right\|_{\Omega_{j/i}} \\ &= \sum_{\mathcal{L}\in M} \sum_{i,j\in\mathcal{L}} \frac{1}{n_{j/i}} \left\|h\left(\hat{\mathbf{z}}_{j/i}, \Delta\mathbf{z}\right) - \bar{\mathbf{z}}_{j/i}\right\|_{\Omega^{1}_{j/i}} . \end{aligned} \tag{9}$$

If we take the derivative of Eq. 9 with respect to $\Delta\mathbf{z}$ and set it to zero, we can solve for the optimal edge update $\Delta\mathbf{z}^*$.

Defining and evaluating $H = {}^{\partial h}/_{\partial\hat{\mathbf{z}}}$. is non-trivial because there is no straightforward way to find the interactions between edges $h\left(\hat{\mathbf{z}}_{j/i}, \Delta\mathbf{z}\right)$ in the loop. Previously published approaches to REO [11]–[13] side-step this issue by optimizing a local subset of edges during a loop closure. However, as noted in [11]–[13], this often results in a globally inconsistent map. In contrast, we address these interdependent constraints by first reparameterizing our cost function into the sum of loops. We approximate $h\left(\hat{\mathbf{z}}_{j/i}, \Delta\mathbf{z}\right)$ by modeling each loop as an independent loop closure with a series of *odometry* edges and a single *loop closure* edge. The odometry edges and their associated perturbations are concatenated normally and all error is lumped into the loop closure edge resulting in the following segmented definition for h:

$$\begin{aligned} h^{\text{odom}}\left(\mathbf{z}_{j/i}, \Delta\mathbf{z}\right) &= \mathbf{z}_{j/i} + \Delta\mathbf{z}_{j/i} \\ h^{\text{LC}}\left(\mathbf{z}_{j/i}, \Delta\mathbf{z}\right) &= \left(\mathbf{z}_{b/i} + \Delta\mathbf{z}_{b/i}\right) \boxplus \left(\mathbf{z}_{c/b} + \Delta\mathbf{z}_{c/b}\right) \\ &\quad \cdots \boxplus \left(\mathbf{z}_{j/z} + \Delta\mathbf{z}_{j/z}\right). \end{aligned}$$

Finding an approximation of $H_{j/i}$ is now possible, even in non-trivial spaces for all edges in the graph, enabling a globally consistent solution. For example, in $SE(2)$, $H^{\text{odom}}_{j/i} = I$, and an example algorithm for calculating $H^{\text{LC}}_{j/i}$ is given in Algorithm 1. This algorithm could be modified to accommodate other systems such as $SE3$

The distinction between loop closure and odometry edges in a simple loop is somewhat arbitrary. In our implementation we have chosen the *loop closure* edge as the most recently acquired measurement in the loop. This is an area of future study because, as mentioned before, the choice of this edge likely influences optimization performance.

Algorithm 1. Calculation of $H^{\ell}_{a/z}$ for a pose graph cycle with reversed edges in $SE(2)$

```
for $\mathbf{z}_{j/i} \in \ell_{\text{odom}}$ do
    if $\text{dir}\left(\mathbf{z}_{j/i}\right) > 0$ then
        $\frac{\partial\Delta t_{a/z}}{\partial\Delta t_{j/i}} = \prod_{m,n<i,j} R^n_m$
    else
        $\frac{\partial\Delta t_{a/z}}{\partial\Delta t_{j/i}} = -\prod_{m,n\leq i,j} R^n_m$
    end if
end for
for $\mathbf{z}_{j/i} \in \ell_{\text{odom}}$ do
    if $\text{dir}\left(\mathbf{z}_{j/i}\right) > 0$ then
        $\frac{\partial\Delta t_{a/z}}{\partial\theta_{j/i}} = \sum_{m,n>i,j} \frac{\partial\Delta t_{a/z}}{\partial\Delta t_{n/m}} \Delta t_{n/m}$
    else
        $\frac{\partial\Delta t_{a/z}}{\partial\theta_{j/i}} = \sum_{m,n\geq i,j} -\frac{\partial\Delta t_{a/z}}{\partial\Delta t_{n/m}} \Delta t_{n/m}$
    end if
    $\frac{\partial\theta_{a/z}}{\partial\theta_{j/i}} = \text{dir}\left(\mathbf{z}_{j/i}\right)$
    $\frac{\partial\Delta t_{a/z}}{\partial\theta_{j/i}} = \mathbf{0}$
end for
```

III. Results Comparing Pose and Edge-based Optimization

Several simulation studies and a hardware experiment were performed to illustrate some of the similarities and differences of global pose and edge-based optimization. The first study illustrates the equivalence between REO and GPO in well-conditioned problems and the second study demonstrates the improved robustness of REO compared to GPO in the presence of noisy edge measurements. The third study illustrates the robustness of REO to gross initial heading error, where GPO often fails completely. Finally, the hardware experiment demonstrates that the algorithm works on real-world data collected by an autonomous system and is well-suited for multi-agent scenarios.

Each simulation study focused on the optimization of the pose graph of a house-shaped trajectory in $SE(2)$ shown in Figure 5 that contains nine nodes and five loop closures. This trajectory was chosen because it is simple enough that properly converged graphs are easy to identify, but also complex enough to illustrate non-trivial items of discussion. The Levenberg-Marquardt optimizer implemented in GTSAM [28] was used as the GPO algorithm, while a

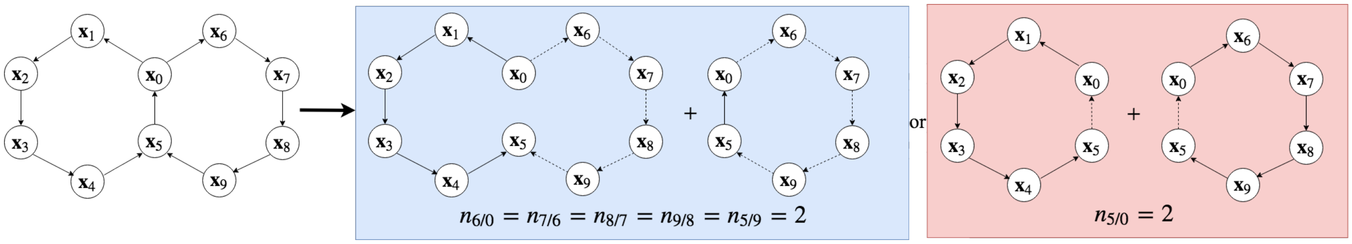

Fig. 4. A pose graph, composed of two simply-connected loops that share a single edge. We model the graph as the sum of a subset of simply-connected loops. Two ways to segment this graph are shown here for reference, but there is a third perfectly valid configuration which is not shown.

prototype implementation of Algorithm 1 was used for REO. To evaluate performance of the optimization routines, we considered the RMS error of the final optimized position of each node in the graph

$$J = \frac{1}{N} \sum_{i=0}^{N} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|, \tag{10}$$

where the heading state of each node was not considered in the RMS error metric to avoid scaling ambiguity between position and heading error. The pose of the solution optimized by REO was calculated by starting at the same origin pose estimated by GPO and compounding the optimized edge estimates to put both solutions into the same frame of reference.

A. Simulation Example 1: Well-Conditioned

The first study illustrates that for a typical, well-conditioned trajectory, the optimizations perform equivalently. The house trajectory was corrupted with small amounts of random Gaussian noise on both translation and heading (see Table I) and a loop closure was placed between nodes at each corner of the square in the house. One thousand of these trajectories were solved by both optimization routines and an example is shown in Figure 5. Every one of these optimizations produced virtually identical results between global pose and edge-based optimization. A histogram of RMS error of the solution found by both optimization algorithms is also shown in Figure 5. Because of the noisy inputs, neither approach perfectly resolves the initial trajectory.

Sim #	σ_x^2 (m^2)	σ_y^2 (m^2)	σ_ψ^2 (rad^2)
1	1×10^{-5}	1×10^{-5}	1×10^{-5}
2	1×10^{-3}	1×10^{-3}	1×10^{-1}
3	1×10^{-5}	1×10^{-5}	1×10^{-2}

TABLE I

TABLE OF NOISE PARAMETERS USED IN THE SIMULATION EXAMPLES

B. Simulation Example 2: Large Noise

As noted earlier, a problematic situation for global pose optimization occurs when significant heading error compounds over several edges to produce poor initial pose estimates. To model this situation, the second simulation study was to optimize random house trajectories that were generated with significant noise on translation and heading (see Table I). Figure 6 shows an example of one of these trajectories, and a histogram of the RMS error of the two approaches.

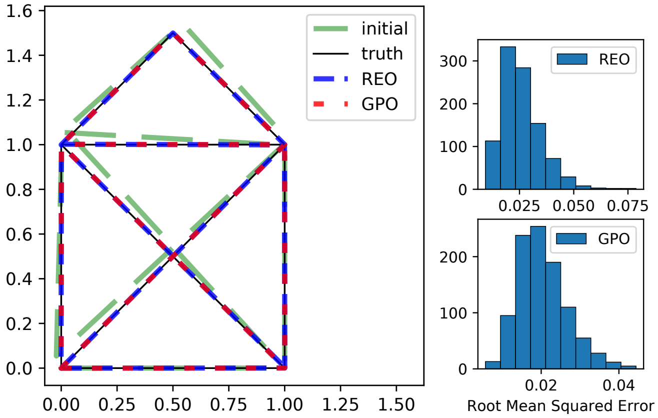

Fig. 5. One sample and summary results of simulation example 1 where trajectories were corrupted with small errors in both translation and heading. All trials had nearly identical optimization results between relative-edge and global-pose optimization. The histogram shows the spread of the RMS error for both REO and GPO over 1000 trials.

In this situation, while GPO was able to find the correct solution much of the time (as shown by the large bin close to zero in the histogram in Figure 6), it sometimes failed to find the solution at all and resulted in divergent behavior. REO, however, found the appropriate minima every time. Although the fit quality is reduced compared to the low noise case, as expected, REO displays significantly more robustness than GPO in this case.

C. Simulation Example 3: Global Heading Misalignment

Many global pose optimization methods struggle when the initial heading is inaccurate. This sort of situation often occurs in GPS-denied navigation or multi-agent problems when, for lack of better information, an agent is initialized in a nominal direction that is out of alignment with its true global heading. To illustrate this, a simulation study was performed where the house trajectory was initialized with 90 deg of error in global orientation. Small amounts of translation noise and heading noise were applied to each edge (see Table I) and loop closures between the corners of the house were replaced with loop closures to a *virtual node* co-located at $\mathbf{x}_0$ and connected to $\mathbf{x}_0$ with a zero-information

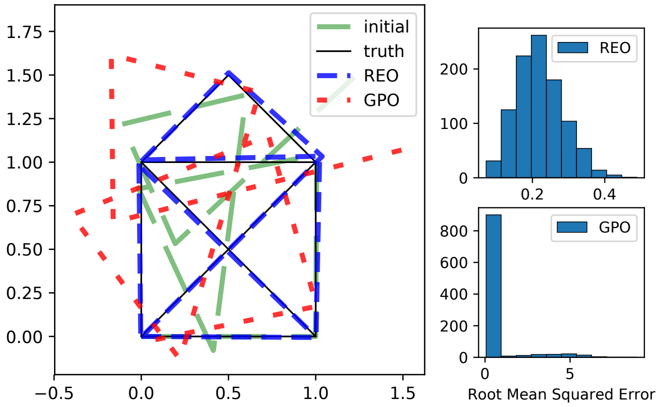

Fig. 6. One sample and summary results of simulation example 2 where 1000 trajectories were corrupted with large errors in both translation and heading. GPO was often able to find the solution, however, it diverged in approximately 10% of cases (as shown in the RMSE histograms). REO found the optimal solution each time.

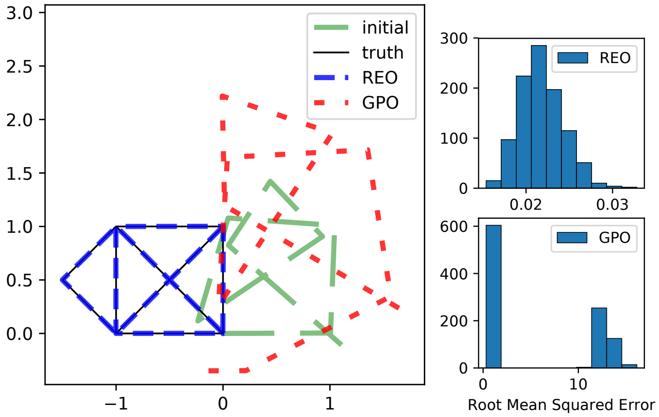

Fig. 7. One sample and summary results of simulation example 3: the house trajectory simulation study with 90° initial heading error. GPO was often able to find the solution, however in approximately 40% of cases, it diverged (as shown in the RMSE histogram). REO found the optimal solution for each of the 1000 trajectories.

edge as described in [5], [24], [27]. Figure 7 shows a sample trajectory and summary results from this study. The results of this study illustrate how GPO struggles in this situation and may end up stalling in a local, and incorrect, minima.

The fundamental problem illustrated in Sec. III-B and Sec. III-C is the compounding of errors and uncertainty that occurs in the pose parameterization. As a result of these large errors and uncertainty, pose-based Jacobians must be evaluated far from the solution and incur significant linearization error (See Figure 1). In contrast, the error and uncertainty about any relative constraint is independent of any other edges. This means that a relative parameterization is much more linear in nature, and easily deals with problems such as arbitrarily large heading misalignment.

This idea is illustrated by a loop containing four edges with heading error in Figure 8. Note the size of $\Delta \mathbf{x}_1$ and associated uncertainty in the global update (upper right) compared the incremental edge updates $\Delta \mathbf{z}_{j/i}$ (bottom). Because the uncertainty estimates for edges do not compound

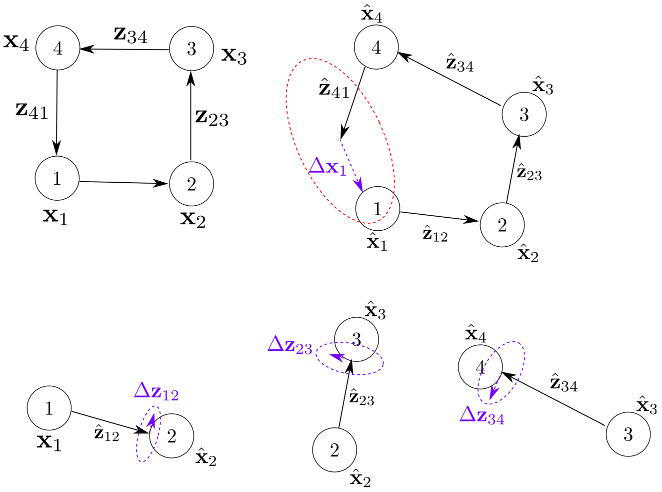

Fig. 8. A simple loop of four nodes and the associated updates to pose (upper right) and edges (bottom).

like they do for poses, both the error and uncertainty for a relative edge constraint are typically lower than its connected poses, and therefore, linearization errors are smaller. For this reason we believe that the recent globally guaranteed methods in global pose optimization [20], [21] could be applied to our relative parameterization of the cost function. Such an approach would combine the benefits of guaranteed global convergence with the smaller initial error in the relative parameterization and potentially improve the speed of convergence.

D. Hardware Experiment

To demonstrate REO on a non-trivial data set, the algorithm was used to optimize data collected by an experimental autonomous navigation system flying two loops through an indoor/outdoor environment with an RGBD camera using a multirotor aircraft, as described in [23] and shown in Figure 9. The data includes 891 nodes and 30 loop closures calculated using the methods described in [23] and is shown in Figure 10. Each loop is approximately 100 meters around and the noise parameters for the experiment are given in Table II.

Fig. 9. The hexacopter used to collect the data in the hardware experiment.

Both algorithms produced similar results to the ones found in [23]. Because the results were not identical they were both evaluated using the original cost function without global

Edge Type	$\sigma_x^2(\mathrm{m}^2)$	$\sigma_y^2(\mathrm{m}^2)$	$\sigma_\psi^2(\mathrm{rad}^2)$
Odometry	8.2×10^{-4}	8.2×10^{-4}	2.2×10^{-7}
Loop Closure	1×10^{-5}	1×10^{-5}	1×10^{-3}

TABLE II
TABLE OF NOISE PARAMETERS IN THE HARDWARE EXPERIMENT

information, given in Eq. 4. The the value of the cost function for REO was 0.00303 and for GPO was 0.115. The specific reason for the difference between the result given by two algorithms is unclear but may be due to GPO reaching a local minima close to the global solution.

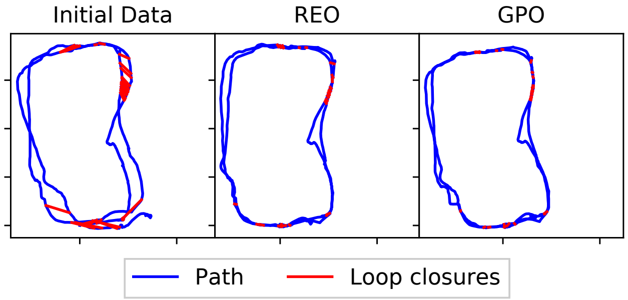

Fig. 10. The results of the optimization when one agent is collecting data.

In a second experiment, to illustrate a collaborative mapping problem, the same data set was used but the data was split in two as if it had been collected by independent agents. The initial position of each agent was assumed to be unknown and therefore one agent was initialized with a heading error of approximately 180 deg. The associated paths and loop closures for each agent can be seen in Figure 11.

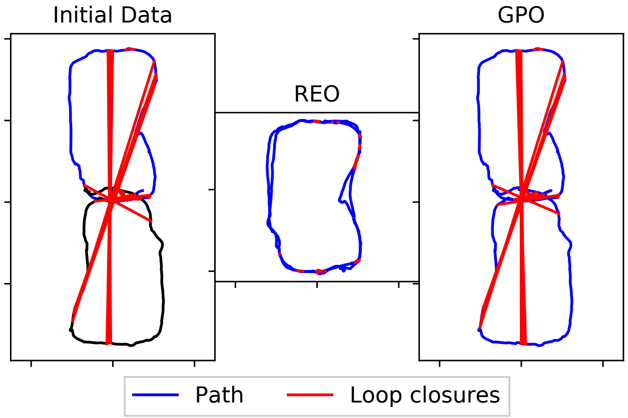

Fig. 11. The results of the optimization when multiple agents are collecting data. In the plot of the initial data the blue and black lines are the data collected by the first and second agent respectively.

In the multi-agent case, GPO failed to converge to any solution other than the initial guess. Additional testing revealed that GPO only converged if the initial heading error was less than 100 deg. This indicates that the jacobians of the cost function were so ill-conditioned that the cost function appeared locally flat and the stopping condition was satisfied in GPO after the first iteration. REO, on the other hand, converged for the given initial heading error between agents and closely matches the results in Figure 10 and [23].

In REO the majority of error in the graph at initialization exists only in the loop closure edges and there is little incentive for the optimization to modify odometry edges until the maps are first aligned. GPO does not have this property, so previous approaches deliberately inform the optimization of the alignment error through the addition of extra anchor nodes and modification of the loop closure edges [15]. We require no such modification, and all constraints can be considered homogeneously just by nature of the parameterization.

In a real-time application, it is likely that an optimized implementation of REO would perform slower than the GPO implementation in GTSAM because of the dense Jacobian REO produces. However, given the demonstrated robustness of REO over traditional methods, optimization of the REO algorithm is of interest and we believe that REO can be refined to run in real time on problems of practical importance.

IV. CONCLUSION

In summary, we have shown that optimizing with respect to relative-edge constraints is robust to large initial and propagated heading errors. We have extended the relative optimization technique presented in [11] to avoid map tears that occur when optimizing only a subset of edges. With these improvements, we conclude that REO should be considered for use in place of, or in tandem with GPO, for solving graph optimization problems that exhibit large initial and propagated heading errors that have proven problematic to global approaches.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2010.

[2] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, “An Atlas framework for scalable mapping,” *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, pp. 1899–1906, 2003.

[3] M. Bosse, P. Newman, J. Leonard, and S. Teller, “Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework,” *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, 2004.

[4] F. Lu and E. Milios, “Globally consistent scan matching for environment mapping,” *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[5] K. Konolige, “Large-scale map making,” in *National Conference on Artificial Intelligence*, pp. 457–463, AAAI Press, 2004.

[6] N. Carlevaris-Bianco, M. Kaess, and R. M. Eustice, “Generic node removal for factor-graph SLAM,” *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1371–1385, 2014.

[7] U. Frese, P. Larsson, and T. Duckett, “A multilevel relaxation algorithm for simultaneous localization and mapping,” *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 196–207, 2005.

[8] G. Grisetti, “Nonlinear constraint network optimization for efficient map learning,” *Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428–439, 2009.

[9] E. Olson, J. Leonard, and S. Teller, “Fast iterative optimization of pose graphs with poor initial estimates,” *IEEE International Conference on Robotics and Automation*, no. May, pp. 2262–2269, 2006.

[10] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “iSAM2: incremental smoothing and mapping using the Bayes tree,” *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[11] D. Sibley and C. Mei, “Adaptive relative bundle adjustment,” *Robotics: Science and Systems*, vol. 220, no. 3, pp. 1–8, 2009.

[12] S. Anderson, K. MacTavish, and T. D. Barfoot, “Relative continuous-time SLAM,” *International Journal of Robotics Research*, vol. 34, no. 12, pp. 1453–1479, 2015.

[13] S. Anderson and T. D. Barfoot, “Towards Relative Continuous-Time SLAM,” *IEEE International Conference on Robotics and Automation*, vol. 34, no. 12, pp. 1033–1040, 2013.

[14] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, “A fast and accurate approximation for planar pose graph optimization,” *International Journal of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014.

[15] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, “Multiple relative pose graphs for robust cooperative mapping,” *IEEE International Conference on Robotics and Automation*, pp. 3185–3192, 2010.

[16] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, “Pairwise-consistent measurement set maximization for robust multi-robot map merging,” *IEEE International Conference on Robotics and Automation*, 2018.

[17] L. Carlone, R. Tron, K. Daniilidis, and F. Dellaert, “Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization,” *IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 4597–4604, 2015.

[18] P. Agarwal, G. Grisetti, G. Diego Tipaldi, L. Spinello, W. Burgard, and C. Stachniss, “Experimental analysis of dynamic covariance scaling for robust map optimization under bad initial estimates,” *IEEE International Conference on Robotics and Automation*, pp. 3626–3631, 2014.

[19] J. Wang and E. Olson, “Robust pose graph optimization using stochastic gradient descent,” *IEEE International Conference on Robotics and Automation*, pp. 4284–4289, 2014.

[20] J. G. Mangelson, J. Liu, R. M. Eustice, and R. Vasudevan, “Guaranteed Globally Optimal Planar Pose Graph and Landmark SLAM via Sparse-Bounded Sums-of-Squares Programming,” *arXiv e-prints*, p. arXiv:1809.07744, Sep 2018.

[21] D. Rosen, L. Carlone, A. Bandeira, and J. Leonard, “SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group,” Tech. Rep. MIT-CSAIL-TR-2017-002, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, Feb. 2017.

[22] T. Qin, P. Li, and S. Shen, “VINS-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[23] D. O. Wheeler, D. P. Koch, J. S. Jackson, G. J. Ellingson, P. W. Nyholm, T. W. McLain, and R. W. Beard, “Relative navigation of autonomous GPS-degraded micro air vehicles,” *All Faculty Publications*, 2017. `http://scholarsarchive.byu.edu/facpub/1962/`.

[24] D. O. Wheeler, P. W. Nyholm, D. P. Koch, G. J. Ellingson, T. W. McLain, and R. W. Beard, “Relative navigation in GPS-degraded environments,” in *Encyclopedia of Aerospace Engineering*, pp. 1–10, John Wiley & Sons, Ltd, May 2016.

[25] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “g2o: A general framework for graph optimization,” in *IEEE International Conference on Robotics and Automation*, pp. 3607–3613, May 2011.

[26] M. Kaess, A. Ranganathan, and F. Dellaert, “iSAM: Incremental smoothing and mapping,” *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[27] J. Rehder, K. Gupta, S. Nuske, and S. Singh, “Global pose estimation with limited GPS and long range visual odometry,” *IEEE International Conference on Robotics and Automation*, pp. 627–633, 2012.

[28] F. Dellaert, “Factor graphs and GTSAM : A hands-on introduction,” tech. rep., Georgia Institute of Technology, 2012.