Robust incremental SLAM with consistency-checking

Robust Incremental SLAM with Consistency-Checking

Matthew C. Graham[1], Jonathan P. How[2] and Donald E. Gustafson[3]

***Abstract*— Both landmark measurements and loop closures are used to correct for odometry drift in SLAM solutions. However, if any of the measurements are incorrect (e.g. due to perceptual aliasing) standard SLAM algorithms fail catastrophically and can not return an accurate map. A number of algorithms have been proposed that are robust to loop closure errors, but it is shown in this paper that they can not provide robust solutions when landmark measurement errors occur. The root cause of the problem is that most robust SLAM algorithms only focus on creating a locally consistent map (by evaluating whether measurements appear correct individually) rather than a globally consistent map. This paper proposes a new formulation of the robust SLAM problem that explicitly requires finding a globally consistent solution. Motivated by the new cost function, a novel incremental SLAM algorithm is developed that provides accurate solutions for datasets with landmark or loop closure measurement errors. Simulated and experimental results of the new algorithm, called incremental SLAM with consistency-checking, show that the new algorithm provides significantly more accurate results than state-of-the-art robust SLAM methods for datasets with incorrect landmark measurements and can match the performance of current robust SLAM methods for datasets with incorrect loop closures.**

I. INTRODUCTION

The simultaneous localization and mapping (SLAM) problem is central to many robotics applications. The objective of the SLAM problem is to estimate the pose (position and orientation) of a robot as it travels through an unknown environment. Often odometry is used to measure the relative change in pose of the robot over time, but odometry is prone to drift. Two different types of measurements, loop closures and landmark measurements, are used in SLAM systems to correct for the build-up of odometry drift in the solution. Loop closures are generated by the SLAM front-end when the robot has returned to a previously visited location. Loop closures provide a measurement of the relative change in the robot pose between the time of the loop closure detection and the previous time the robot visited that location.[1] In contrast, landmark measurements provide a relative measurement from the robot to a landmark in the environment. In most cases, landmark locations are not known a priori and as a result the SLAM system must also estimate the location of the landmark along with the robot poses.

Most state-of-the-art algorithms pose the SLAM problem as inference on a factor graph where variables nodes correspond to robot poses and landmarks in the environment, and factor nodes represent constraints generated by the measurements on the poses and landmarks [1]–[3]. It is typically assumed that the measurements are noisy but outlier free, in other words the loop closure and landmark measurements are always correct. In practice however, the front-end systems used to identify landmarks and loop closures can generate incorrect measurements caused by self-similar structures in the environment (e.g. visual aliasing [4]). In standard SLAM algorithms, processing even a single incorrect measurement can lead to an inconsistent map or divergence of the algorithm [5,6].

While several robust SLAM algorithms have been proposed in the literature [5]–[11], most have focused on robustness to incorrect loop closures rather than the problem of incorrect landmark measurements. Previously [8], it has been demonstrated that robust SLAM algorithms can fail when applied to problems with incorrect landmarks (see Figure 1). This failure occurs because most current robust algorithms only focus on ensuring local map consistency by evaluating whether each measurement, independent of the others, is incorrect. Ideally, a robust SLAM algorithm should verify that the map is both locally and globally consistent.

The contributions of this paper are (1) a new formulation of the robust SLAM problem, (2) a novel incremental SLAM algorithm called incremental SLAM with consistency-checking (ISCC) that approximately solves the robust SLAM problem and (3) a comprehensive comparison of robust SLAM techniques in the literature. The new robust SLAM formulation focuses on selecting the largest set of measurements that will produce a globally consistent solution. ISCC is designed to ensure that only sets of measurements that are both consistent with each other and the global map are included in the final mapping solution. We demonstrate in the paper that ISCC significantly outperforms existing robust SLAM algorithms when landmark measurement outliers occur and can match the performance of state-of-the-art algorithms when loop closure errors occur.

The outline of the rest of the paper is as follows. Section II provides background on the SLAM problem and a review of existing algorithms for robust SLAM. Section III presents the formulation of the robust SLAM problem. Section IV develops ISCC. Section V presents simulated and experimental results that compares the performance of ISCC to state-of-the-art robust SLAM algorithms. Finally, Section VI provides

[1]Draper Laboratory Fellow, Ph.D. candidate, Dept. of Aeronautics and Astronautics, MIT mcgraham@mit.edu

[2]Richard C. Maclaurin Professor of Aeronautics and Astronautics, MIT jhow@mit.edu

[3]Distinguished Member of the Technical Staff, Draper Laboratory degustafson@draper.com

[1]It should be noted that in landmark-based SLAM, the re-observation of a landmark is also called a loop closure. For clarity in the paper, we will only refer to relative pose measurements between non-adjacent poses as loop closures.

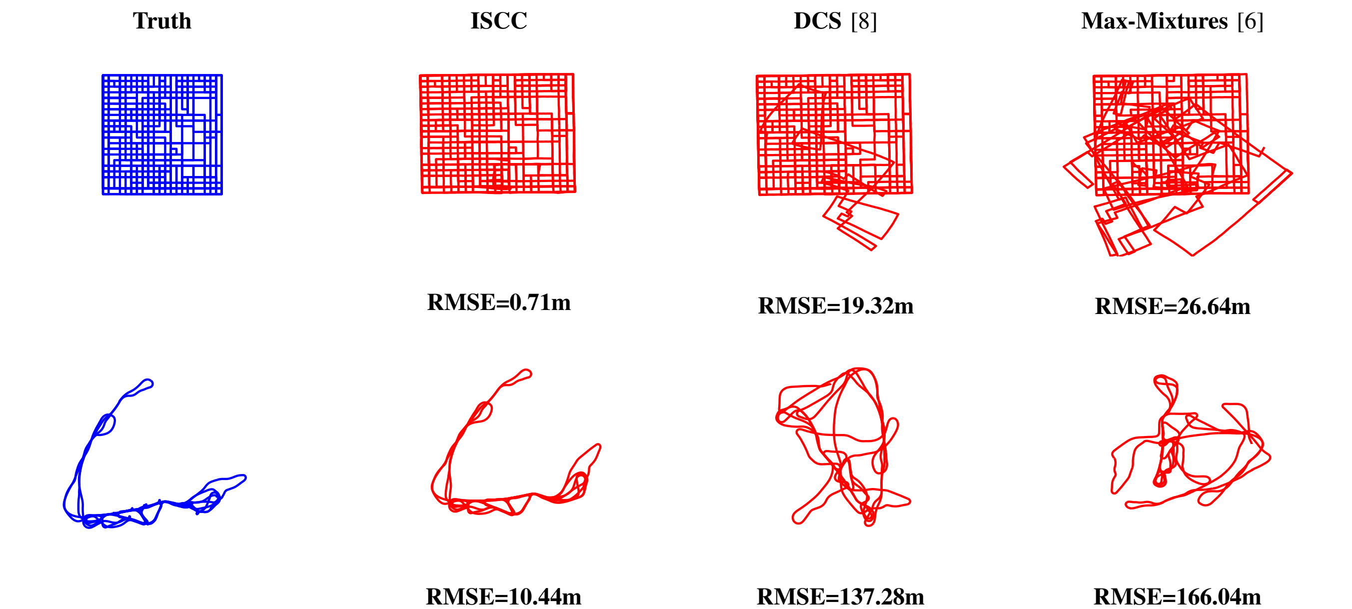


Fig. 1: Typical optimization results for cityTrees10000 (top row) and Victoria Park (bottom row) with 1000 incorrect landmark measurements. Root mean-squared error values for each solution are shown below each figure. Both max-mixtures and dynamic covariance scaling fail to correctly identify the incorrect measurements and the SLAM solution suffers as a result. New ISCC method can correctly detect the incorrect measurements and provides substantially better results.

a brief summary as well as areas of future work.

II. Background and Related Work

This section provides a brief review of SLAM factor graphs and an overview of the robust SLAM literature.

A. *Graph-based SLAM*

The following notation will be used throughout the paper. $\mathbf{x}_i$ and $\mathbf{l}_j$ represent the ith robot pose and jth landmark location respectively. $\mathbf{x}$ and $\mathbf{l}$ without subscripts are the full sets of poses and landmark locations respectively. An odometry measurement from $\mathbf{x}_i$ to $\mathbf{x}_{i+1}$ will be expressed as $\mathbf{y}_i^o$. A Landmark measurement between $\mathbf{x}_i$ and $\mathbf{l}_j$ will be written as $\mathbf{y}_{i,j}^L$. A loop closure measurement between poses $\mathbf{x}_i$ and $\mathbf{x}_k$ will be written as $\mathbf{y}_{i,k}^{LC}$ Finally, the set of landmark and loop closure measurements will be denoted L and LC respectively.

Factor graphs are a compact way of representing the posterior pdf of a robot's pose and landmark locations given odometry, loop closure and landmark measurements. Each variable node in the graph corresponds to a robot pose or landmark location. The factor nodes in the graph represent relative constraints on the poses and landmarks derived from the measurements. A standard assumption in the SLAM literature is that the measurement noise is independent and Gaussian [14].

Given the Gaussian and independence assumptions, the maximum likelihood estimates of the poses and landmark locations can be expressed as

$$(\hat{\mathbf{x}}, \hat{\mathbf{l}}) = \underset{\mathbf{x},\mathbf{l}}{\operatorname{argmin}} \sum_i \chi^2_{o,i} + \sum_{\mathbf{y}^L_{i,j} \in L} \chi^2_{L,ij} + \sum_{\mathbf{y}^{LC}_{i,k} \in LC} \chi^2_{LC,ik} \quad (1)$$

where

$$\chi^2_{o,i} = ((\mathbf{x}_i \oplus \mathbf{y^o_i}) \ominus \mathbf{x}_{i+1})^T \Lambda_o ((\mathbf{x}_i \oplus \mathbf{y^o_i}) \ominus \mathbf{x}_{i+1}) \quad (2)$$

$$\chi^2_{L,ij} = ((\mathbf{x}_i \oplus \mathbf{y^L_{i,j}}) \ominus \mathbf{l}_j)^T \Lambda_L ((\mathbf{x}_i \oplus \mathbf{y^L_{i,j}}) \ominus \mathbf{l}_j) \quad (3)$$

$$\chi^2_{LC,ik} = ((\mathbf{x}_i \oplus \mathbf{y^{LC}_{i,k}}) \ominus \mathbf{x}_k)^T \Lambda_{LC} ((\mathbf{x}_i \oplus \mathbf{y^{LC}_{i,k}}) \ominus \mathbf{x}_k) \quad (4)$$

Λ_o, Λ_L and Λ_{LC} are the information matrices of the odometry, landmark and loop closure measurements respectively, and $\oplus$ and $\ominus$ are standard pose composition operators [15].[2]

The cost function in Eq. 1 is a nonlinear least-squares (NLS) problem and can be efficiently solved by specialized software packages such as g2o [3] and iSAM [1]. However, if incorrect landmark or loop closure measurements occur in the data, standard NLS solvers will fail to converge to the correct solution [5,6]. In those cases a robust SLAM algorithm must be used instead.

B. *Robust SLAM Algorithms*

Robust SLAM algorithms can be divided into two main categories. The first category approach the problem by adding additional latent variables to the factor graph to account for outliers in the measurements (augmented model approaches). The second set focuses on choosing sets of measurements that lead to a consistent SLAM solution.

1) Augmented Model Approaches: Sünderhauf et al. [5, 18] introduced additional variables into the SLAM problem formulation, called *switch variables*, that can take any value in the interval $[0\ 1]$ and are used as weights for each of the loop closure measurements. The switch variables provide robustness by determining whether to accept or reject a

[2] In general, Eqs. 2 and 4 should also include a log-map that maps the residuals in $SE2$ or $SE3$ to their respective tangent spaces. For more details see [16,17].

potential loop closure measurement. When a switch variable is equal to zero its corresponding loop closure measurement will not have any impact on the SLAM solution. Agarwal et al. presented a generalization of switch variables called dynamic covariance scaling (DCS) that provides a closed form update for the switch variables and results in faster convergence [8].

Another approach developed by Olson and Agarwal [6], *max-mixtures*, modifies the conditional probability distribution of the measurements so that the noise is represented by a Gaussian mixture instead of a single Gaussian. The algorithm then selects the most likely mixture component before each update. Finally, several robust algorithms have been proposed that add additional variables to the robust SLAM problem and solve the augmented problem by applying the Expectation-Maximization algorithm [9,10].

Augmented model approaches can be applied to datasets with both landmark and loop closures. However, previous results [8] (as well as results in this paper) show that they are only robust to a small number of incorrect landmark measurements.

One problem with augmented model approaches is that they only focus on local consistency in the graph (by determining whether each measurement is correct independently of the others) rather than global consistency. This strategy works well for loop closures because typically only one loop closure measurement exists between any two robot poses. Thus, if the measurement is not locally consistent with its associated pose estimates, the loop closure is likely incorrect and can be ignored.

In contrast, landmark nodes are densely connected in SLAM factor graphs. By myopically evaluating each landmark measurement, augmented model algorithms miss the opportunity to exploit the additional information available from the other measurements. For instance, an incorrect landmark measurement could appear consistent given only the landmark and pose estimates associated with it, but not agree with the other landmark measurements.

Based on these observations, ISCC focuses on determining which measurements are correct by searching for a consistent subset of measurements. Additionally ISCC emphasizes both local consistency of the measurements as well as global consistency of the graph.

2) Consistency Based Approaches: An alternative set of robust SLAM algorithms attempt to remove incorrect measurements by performing consistency checks on subsets of the measurements. The RRR algorithm clusters loop closure measurements spatially and then uses a series of χ^2 tests to determine whether the clusters are consistent with each other and with the overall solution [7,13]. Any measurements that are inconsistent are ignored when calculating the final SLAM solution.

ISCC is similar to RRR (i.e. both use χ^2 tests as a means of evaluating consistency) but there are several key differences. One of the biggest differences between RRR and ISCC is the cost function that each algorithm uses. The only goal of the RRR algorithm is to produce a solution that is consistent. There is no explicit requirement or goal to maximize the number of measurements included in the final solution. As a result, RRR often ignores a large number of correct loop closures as the number of outliers increases [19], which in turn leads to less accurate solutions. In contrast, ISCC attempts to find the largest set of measurements that leads to a consistent solution. By optimizing the number of measurements as well as requiring consistency, ISCC tends to retain more correct measurements and produces more accurate solutions than RRR as the number of outliers increases.

Recently, Carlone et al. [11] posed the robust SLAM problem as an optimization with the goal of selecting the largest subset of measurements in the graph that admit a solution that is consistent. The authors of that paper focus on 2D SLAM scenarios where an approximate solution to the optimization can be calculated using linear programming. While their algorithm is fast and effective, it relies heavily on the assumption that all the measurements are linear and as a result there is no clear way to extend the algorithm to 3D SLAM datasets.

While the approach of Carlone et al. is the most similar in spirit to our method, there are several key differences. First, motivated by the standard Gaussian noise assumption for SLAM, we apply a different consistency test (weighted l_2 norm vs. l_1 norm) in our robust SLAM problem formulation. The l_1-norm constraint in Carlone et al.'s formulation was chosen so that the robust SLAM problem could be solved using linear programming, but it also restricts the scope of their algorithm to 2D datasets. Using a weighted l_2-norm in the problem formulation allowed us to develop a robust SLAM algorithm, ISCC, that can be applied to both 2D and 3D SLAM problems.

III. Robust SLAM Problem Formulation

There are two primary criteria for robust SLAM:

1) Generate a solution that is *consistent* both locally and globally,
2) Generate a solution that is as accurate as possible.

In this context, consistent means that the solution generated by the SLAM algorithm agrees with the measurements used in the solution. Consistency is defined more formally in the next section.

The approach taken in this paper is to decide which measurements should be included in the factor graph and then solve the SLAM problem on that graph. Clearly, the solution will be consistent and accurate if none of the incorrect measurements are included.

However, it should be noted that the two goals do not necessarily align. For instance, a consistent solution could be generated by only processing the odometry measurements, but it would not be metrically accurate due to the accumulation of odometry drift. Therefore consistency is not sufficient to guarantee an accurate solution. To generate the most accurate solution, ideally the SLAM solution would include as many of the measurements as possible so long as the solution remained consistent.

The rest of this section defines a set of consistency tests and then poses the robust SLAM problem based on the criteria described above.

A. Consistency Tests

If the factor graph were composed entirely of odometry measurements, a consistent solution could be calculated by setting the pose estimates such that they satisfy the odometry exactly. However, when landmark and loop closure measurements are added to the graph, the SLAM solution will not be able to satisfy all of the measurements exactly because of measurement noise. If all of the measurements are correct, the resulting measurement residuals should still be small. But, if any of the measurements are incorrect, we would expect the measurement residuals to be larger because some pose estimates will be set to values that are inconsistent with their respective constraints. Therefore, the sum of measurement residuals provides a tool for determining whether the graph is consistent.

A straight-forward means of testing whether the graph is globally consistent is to apply a χ^2 test to the weighted sum of measurement residuals. Given pose and landmark estimates, odometry, landmark and loop closure measurements, the weighted sum of measurement residuals of the graph can be expressed as

$$\chi^2_G = \sum_i \chi^2_{o,i} + \sum_{\mathbf{y}^L_{i,j} \in L} \chi^2_{L,ij} + \sum_{\mathbf{y}^{LC}_{i,k} \in LC} \chi^2_{LC,ik} \quad (5)$$

where $\chi^2_{o,i}$, $\chi^2_{L,ij}$ and $\chi^2_{LC,ik}$ are given by Eqs. 2- 4. If the solution is consistent we would expect χ^2_G to satisfy the following inequality with probability p:

$$\chi^2_G \leq \chi^2(p, n_{dof}) \quad (6)$$

where $\chi^2(p, n_{dof})$ is the inverse χ-squared cdf with n_{dof} degrees of freedom evaluated at p. By setting p to a value close to 1 (i.e. 0.95) we can verify that the graph is consistent with high probability as long as it satisfies Eq. 6.

In addition to being globally consistent, ideally each measurement included in the factor graph should be locally consistent with its associated pose and landmark estimates. Thus, we also define local consistency tests for each landmark and loop closure measurement

$$\chi^2_{L,ij} \leq \chi^2(p, n_{dof,L}) \quad \forall j \quad (7)$$

$$\chi^2_{LC,ik} \leq \chi^2(p, n_{dof,LC}) \quad \forall k \quad (8)$$

where $n_{dof,L}$ and $n_{dof,LC}$ are the number degrees of freedom of the landmark and loop closure measurements. If a graph satisfies Eqs. 6–8 then we declare the graph to be consistent.

B. Robust SLAM Cost Function

With the consistency tests defined, the final step is to formulate a cost function for robust SLAM. We make the standard assumptions that the measurement noise is Gaussian and that the odometry measurements are not corrupted with outliers. We also define a set of binary variables s^L_j and s^{LC}_k that indicate whether the jth landmark measurement and kth loop closure measurement are included ($s^L_j = 1$,$s^{LC}_k = 1$) in the graph. Given the intuition that a robust SLAM solution should include as many measurements as possible while remaining consistent, we define the robust SLAM cost function as:

$$\max_{\mathbf{x},\mathbf{l},\mathbf{s}} \quad \sum_{j=1}^{n_L} s_j \quad (9)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{i=1}^{n_o} \chi^2_{o,i} + \sum_{j=1}^{n_L} s^L_j \chi^2_{l,j} + \sum_{k=1}^{n_{LC}} s^{LC}_k \chi^2_{LC,ik} \leq \chi^2(p, n_{dof}) \\ & s^L_j \chi^2_{L,ij} \leq \chi^2(p, n_{dof,L}) \quad \forall j \\ & s^{LC}_k \chi^2_{LC,ik} \leq \chi^2(p, n_{dof,LC}) \quad \forall k \\ & s^L_j \in \{0,1\} \quad \forall j, s^{LC}_k \in \{0,1\} \quad \forall k \end{aligned}$$

where n_{dof} is the number of degrees of freedom in the factor graph. This cost function ensures that any solution will meet the criteria for a robust SLAM solution. Maximizing the sum of the indicator variables encourages a solution that includes as many measurements as possible, which will lead to an accurate solution, and the χ^2 constraints ensure that the solution will be consistent.

While the problem formulation in Eq. 9 meets the criteria for a robust SLAM cost function, it is not practical to optimize it directly. In particular, Eq. 9 is a mixed integer nonlinear program and is NP-hard to solve [20]. The remainder of this paper proposes and evaluates an algorithm that approximately solves Eq. 9.

IV. Robust Incremental SLAM with Consistency Checking

This section proposes an incremental greedy solution to the robust SLAM problem in 9 called ISCC. By processing the data incrementally and verifying that the current graph satisfies Eqs. 6–8 we ensure that a consistent graph is maintained throughout the solution process. Additionally, as new measurements are added to the graph they can be identified as outliers by evaluating whether the updated graph is inconsistent or not. If the graph is inconsistent after adding a new measurement, a greedy search is performed to find the minimum number of measurements that can be removed from the graph in order to make the graph consistent again.

Pseudocode for ISCC is shown in Algorithm 1. Measurements are processed as they are generated by the SLAM front-end. Odometry measurements are assumed to be outlier free and are automatically added to the factor graph. Landmark and loop closure measurement are added to graph and then the graph is checked for consistency using Eqs. 6–8. If the graph is found to be inconsistent, ISCC performs a greedy search (lines 8–9 of Algorithm 1) that seeks to remove the fewest number of measurements from the graph such that it becomes consistent. This process removes outliers from the graph and ensures that the graph is always consistent after each measurement is processed. The rest of this section describes in detail the greedy search procedure for removing outliers from the graph.

Algorithm 1 ISCC

Require: Measurement queue $\mathbf{Y}$

1: Initial graph $G = \emptyset$
2: **for** (each $\mathbf{y}_i \in \mathbf{Y}$) **do**
3: Add $\mathbf{y}_i$ to G and update
4: **if** ($\mathbf{y}_i$ is not odometry) **then**
5: Check if G is consistent using Eqs. 6–8
6: **if** (G is not consistent **then**
7: $\mathbf{Y}_{test} \leftarrow$ findCandidateOutliers($\mathbf{y}_i$,G)
8: $G \leftarrow$ findConsistentMeasurementSet($\mathbf{y}_i$,$\mathbf{Y}_{test}$,G)
9: **end if**
10: **end if**
11: **end for**
12: **return** Optimized graph G

Algorithm 2 findCandidateOutliers

Require: current measurement $\mathbf{y}$, factor graph G

1: **if** ($\mathbf{y} \in L$) **then**
2: $j^* \leftarrow$ Index of landmark associated with $\mathbf{y}$
3: $Y_{test} \leftarrow \left\{\mathbf{y}_{i,j}^{L} \in L \mid j = j^*\right\}$
4: **else**
5: $Y_{test} \leftarrow \left\{\mathbf{y}_{i,k}^{LC} \in LC \mid \chi^2_{LC,ik} > \chi^2(p, n_{dof,LC})\right\}$
6: **end if**
7: **return** Set of potential outliers Y_{test}

Algorithm 3 findConsistentMeasurementSet

Require: Current Measurement $\mathbf{y}$, test set Y_{test}, factor graph G

1: $outliers \leftarrow \mathbf{y}$
2: **for** (each $\mathbf{y}_i \in Y_{test}$) **do**
3: Remove $\mathbf{y}_i$ from G, Update G
4: **if** (G is consistent) **then**
5: $outliers \leftarrow outliers \cup \mathbf{y}_i$
6: **end if**
7: Add $\mathbf{y}_i$ into G
8: **end for**
9: Remove $outliers$ from G, Update G
10: **return** Updated factor graph G

A. *Outlier Removal*

If the graph becomes inconsistent after adding a new measurement, $\mathbf{y}_i$, there are two possible explanations: 1) $\mathbf{y}_i$ is an outlier or 2) a previously processed measurement is an outlier and was erroneously accepted. To make the graph consistent, ISCC first determines which previously accepted measurements are most likely to be outliers (using Algorithm 2) and then removes as few measurements as possible from the graph while ensuring that the resulting graph is consistent (using Algorithm 3).

ISCC applies two different strategies for determining which measurements are potential outliers depending on the measurement type. If $\mathbf{y}_i$ is a landmark measurement, then logically the other measurements of that landmark are the most likely candidates to be outliers. Therefore, if $\mathbf{y}_i$ is a landmark measurement Algorithm 2 simply returns the set of measurements of that landmark (Step 3).

In the case of loop closures, note that if the graph has become inconsistent after adding $\mathbf{y}_i$, any loop closure measurements that are locally inconsistent (i.e. they fail the test in Eq. 8) must contain information which is not consistent with $\mathbf{y}_i$. This means that if $\mathbf{y}_i$ is not an outlier, then at least one of the measurements that failed the local consistency tests is an outlier. Using this information, if $\mathbf{y}_i$ is a loop closure measurement, Algorithm 2 returns the set of loop closures in the graph that are locally inconsistent (Step 5).

Given a set of candidate outliers Y_{test}, the next step (Algorithm 3) is to determine which measurements should be removed from the graph. Since ISCC is optimizing Eq. 9, it should remove as few measurements as possible to make the graph consistent. Note that since the graph was consistent before adding $\mathbf{y}_i$, an admissible solution to the outlier removal problem is to remove $\mathbf{y}_i$. However that solution may not be unique because $\mathbf{y}_i$ could in fact be correct in which case there should be at least one other measurement that could be removed from the graph to generate a consistent solution. Therefore, in order to remove any potential outliers from the graph, ISCC tests each measurement in Y_{test} to determine if any of them can be removed from the graph to make it consistent (Steps 2–8 of Algorithm 3). If there are any measurements in Y_{test} that can be removed from the graph to make it consistent, ISCC removes them from the graph along with $\mathbf{y}_i$ since there is no clear way to decide which measurements are outliers (Step 9 of Algorithm 3). If no other measurements can be removed, $\mathbf{y}_i$ is removed from the graph.

V. Experimental Evaluation

Simulated and real-world datasets were used to evaluate ISCC and compare it to state-of-the-art robust SLAM algorithms. The evaluations focused on the accuracy of the solutions as well as how accurately the robust SLAM methods identified outliers in the datasets.

A. *Datasets and Evaluation Set-Up*

Several standard benchmark SLAM datasets (Manhattan3500, Intel, City10000, Sphere2500, CityTrees10000, Victoria Park, and Torus2000Points) were used for the algorithm evaluations. Manhattan3500, Intel, City10000, and Sphere2500 contain loop closure measurements while CityTrees10000, Victoria Park and Torus2000Points contain landmark measurements. Sphere2500, City10000, CityTrees10000, Victoria Park and Torus2000Points are available as part of the iSAM package [2]. The Manhattan3500 and Intel datasets are available as part of the g2o distribution [21]. We also compared the algorithms on the real-world Bicocca dataset using the processed data files released with the RRR package [22].

Since the simulated datasets do not contain incorrect measurements, additional outliers were added to the datasets artificially. For the datasets containing loop closures, additional outliers were generated using the random, local, grouped and local grouped approaches proposed in [19]. The random outlier generation strategy chooses nodes uniformly at random and generates a false loop closure measurement between the poses. Local outliers produce loop closure measurements between nodes that are in close proximity to each other in the graph. The grouped outlier strategy creates clusters of mutually consistent false loop closures. The local grouped strategy combines the local and grouped outlier generation approaches. Landmark measurement outliers were generated by choosing landmarks and poses at random and generating measurements between them as in the comparisons from [8]. The number of outliers added to each dataset was varied between 200 and 1000 in 200 outlier increments. 30 Monte Carlo trials were performed for each dataset, number of outliers and outlier selection strategy for a total of 600 trials.

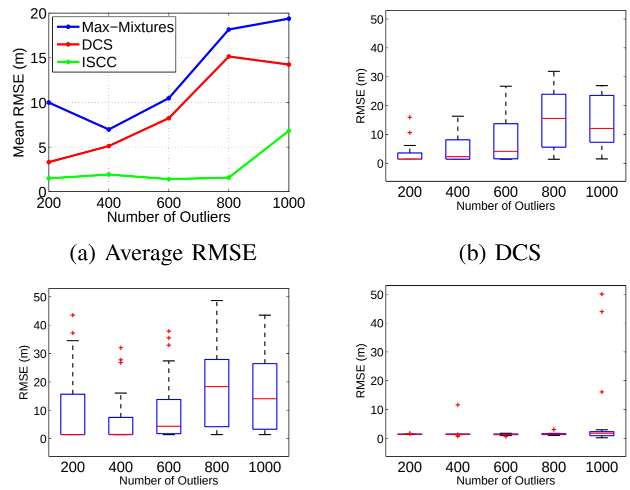
The Bicocca dataset was collected experimentally and contains loop closures that were generated by a visual bag-of-words based place recognition system. The false loop closures in the dataset were caused by visual aliasing.

Five different robust SLAM algorithms were compared: DCS [8], max-mixtures [6], RRR [7,13], Carlone et al.'s linear programming approach (l_1-SLAM) [11], and ISCC. All of the algorithms (with the exception of l_1-SLAM) were implemented using the g2o package [3]. The robust kernel implementation of DCS that is included with g2o was used for all evaluations. The nominal value of $\Phi = 1$ was used for all of the DCS evaluations except for the Bicocca dataset where $\Phi = 5$ as specified by the authors in their original evaluation [8]. An open source version of max-mixtures was used for the experiments [23]. Two mixture components were used for max-mixtures. The first mixture corresponds to a nominal measurement and had a weight equal to 1. The second mixture corresponds to an outlier and had a weight equal to 0.01 with an information matrix equal to the nominal information matrix scaled by 10^{-6}. Finally, the χ^2 probability threshold for ISCC was $p = 0.95$.

The metrics of performance used for the evaluations were root mean-squared position error (RMSE) of the poses and precision and recall of the measurements. RMSE was calculated by aligning the SLAM solution with truth and calculating the position error for each node in the graph. Precision in this context measures the fraction of measurements included in the final graph that were correct, while recall measures the fraction of the total correct measurements that were included in the graph. Note that an ideal robust SLAM algorithm would achieve precision and recall values of 1. Precision and recall could not be used to evaluate DCS because it does not make binary decisions about whether each measurement in the graph is an outlier.

B. Landmark Dataset Results

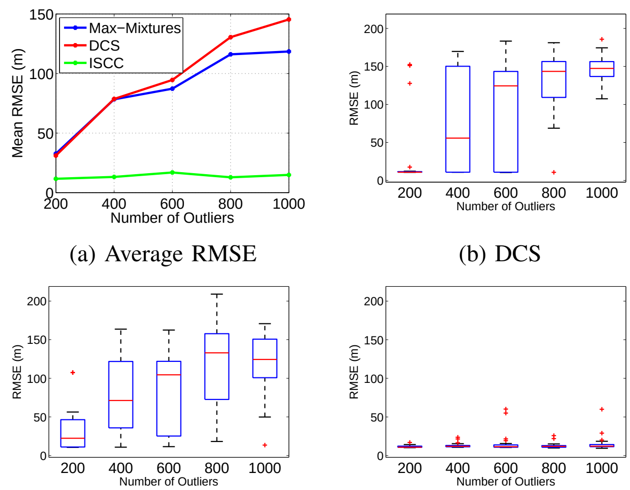
Results for the landmark datasets are shown in Figures 2–4. Results for l_1-SLAM and RRR are not shown because they are not designed for landmark-based SLAM datasets and could not be applied. Overall, ISCC significantly outperforms the other robust SLAM algorithms. ISCC achieved the



(a) Average RMSE (b) DCS

(c) Max-Mixtures (d) ISCC

Fig. 2: Comparison of Monte Carlo Results for the CityTrees10000 dataset.



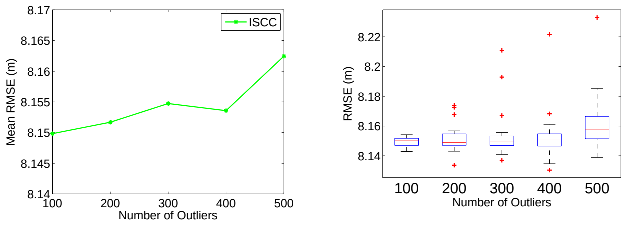
(a) Average RMSE (b) DCS

(c) Max-Mixtures (d) ISCC

Fig. 3: Comparison of Monte Carlo Results for the Victoria Park dataset.

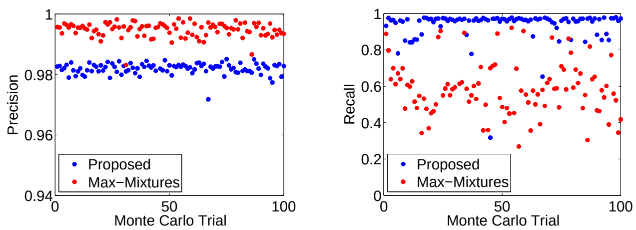
lowest average RMSE performance across every dataset by a significant margin. Moreover, in the case of the Torus2000 dataset, DCS and max-mixtures diverged for nearly every Monte Carlo trial while ISCC consistently converged to a good solution. These results indicate that searching for solutions that are both globally and locally consistent leads to significantly more robust solutions when landmark errors occur.

Figure 5 shows the precision and recall values for max-mixtures and ISCC applied to the Victoria Park dataset with 1000 incorrect landmark measurements. The precision values indicate that most of the measurements included in the graph by both algorithms are correct, but the recall values indicate that max-mixtures ignores a significant fraction of the correct landmark measurements. Ignoring so many correct measurements can cause the SLAM solution to be more heavily impacted by incorrect measurements and allows for the local build-up of substantial odometry drift errors. Also, note that the optimal solution to Eq. 9 should have



(a) Average RMSE (b) ISCC

Fig. 4: RMSE Results for the Torus2000Points Dataset. Max-mixtures and DCS could not converge to a valid solution on this dataset.



(a) Precision (b) Recall

Fig. 5: Precision and recall results for the Victoria Park dataset. Max-mixtures shows a significant drop in recall with respect to the proposed algorithm. The drop in recall indicates that max-mixtures is incorrectly rejecting a large fraction of the correct landmark measurements.

precision and recall values of 1, so ISCC is near-optimal for most of the Victoria Park Monte Carlo trials. Results for the CityTrees10000 dataset were similar.

C. Loop Closure Dataset Results

Results for the loop closure datasets are shown in Tables I and II. Results for l_1-SLAM were omitted for Sphere2500 because l_1-SLAM can only be applied to 2D datasets. Additionally, while processing the city10000 dataset with l_1-SLAM, the computer ran out of memory and as a result a solution could not be generated.

There are several notable findings in the results. First, ISCC's performance is comparable to the best results from the other robust SLAM algorithms on each dataset. Moreover, ISCC significantly outperforms RRR on every dataset except for Bicocca where the results are comparable. The low recall scores for RRR indicate that the difference in RMSE performance between ISCC and RRR can be attributed to RRR rejecting a larger number of loop closures. In effect, RRR is finding a consistent solution that is not metrically accurate because it does not explicitly attempt to maximize the number of measurements included in the factor graph. Overall, these results demonstrate that ISCC's outlier rejection strategy can provide comparable performance to existing robust SLAM techniques when loop closure errors occur.

VI. Conclusions

This paper demonstrated that state-of-the-art robust SLAM algorithms can not provide robust solutions for datasets with incorrect landmark measurements. The root cause of these issues is that current robust algorithms focus on ensuring that the SLAM solution is locally consistent but do not require the solution to be globally consistent.

To address this issue, we developed a new formulation of the robust SLAM problem that requires a globally consistent solution. Motivated by the new robust SLAM problem, we presented a novel incremental SLAM algorithm, ISCC, that can provide robust solutions when incorrect landmark measurements occur. Simulated and experimental results demonstrated that the new algorithm provides significantly better solutions than current robust SLAM algorithms when incorrect landmark measurements occur.

Acknowledgment

This work was funded by the C.S. Draper Laboratory Internal Research and Development Program. The authors would like to thank Luca Carlone for providing an implementation of the l_1-SLAM algorithm that was used in the evaluations.

References

[1] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, pp. 1365–1378, 2008.

[2] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3281–3288, 2011.

[3] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3607–3613, 2011.

[4] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[5] N. Sünderhauf and P. Protzel, "Towards a robust back-end for pose graph slam," in *2012 IEEE International Conference on Robotics and Automation (ICRA),*, pp. 1254–1261, IEEE, 2012.

[6] E. Olson and P. Agarwal, "Inference on networks of mixtures for robust robot mapping," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[7] Y. Latif, C. C. Lerma, and J. Neira, "Robust loop closing over time," in *Proceedings of Robotics: Science and Systems*, (Sydney, Australia), July 2012.

[8] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 62–69, 2013.

[9] G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Robust pose-graph loop-closures with expectation-maximization," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[10] M. Graham and J. How, "Robust simultaneous localization and mapping via information matrix estimation," in *ION/IEEE Position, Location and Navigation Symposium*, pp. 937–944, May 2014.

[11] L. Carlone, A. Censi, and F. Dellaert, "Selecting good measurements via l_1 relaxation: a convex approach for robust estimation over graphs," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[12] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 99–108, 2006.

[13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, pp. 167–193, Springer, 1990.

[14] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 475–492, 2014.

[15] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

TABLE I: RMSE Results for Loop Closure Datasets

Dataset	DCS mean, median, max	Max-Mixtures mean, median, max	RRR mean, median, max	l_1-SLAM mean, median, max	ISCC mean, median, max
Manhattan3500	**0.80**, **0.79**, **0.87**	0.81, 0.79, 2.01	6.21, 2.45, 50.97	1.70, 0.83, 18.23	0.85, 0.80, 2.55
Intel	0.006, 0.006, **0.006**	**0.001**, **0.001**, 0.024	0.716, 0.074, 8.42	0.010, 0.005, 0.044	0.009, 0.006, 0.209
Bicocca	1.83, 2.12, 3.14	4.17, 4.15, 6.07	1.59, 1.08, 2.96	1.73, 2.12, **2.96**	**1.56**, **0.97**, **2.96**
City10000	**1.00**, **1.00**, 1.41	1.08, 1.00, 6.56	2.75, 2.33, 5.60	N/A, N/A, N/A	**1.00**, **1.00**, **1.02**
Sphere2500	**0.34**, **0.34**, **0.34**	4.67, 4.68, 4.68	5.88e6, 27.92, 5.88e7	N/A, N/A, N/A	0.35, 0.35, 0.36

Note: Results for l_1-SLAM were omitted for Sphere2500 because the algorithm can not be applied to 3D datasets. Results for l_1-SLAM were omitted for City10000 because the algorithm ran out of memory while attempting to perform the optimization.

TABLE II: Average Precision/Recall Results for Loop Closure Datasets

Dataset	Max-Mixtures Avg. Precision/Recall	RRR Avg. Precision/Recall	l_1-SLAM Avg. Precision/Recall	ISCC Avg. Precision/Recall
Manhattan3500	0.999/1.00	0.997/0.894	0.999/0.994	0.999/0.984
Intel	1.00/0.991	1.00/0.770	1.00/0.994	1.00/0.988
City10000	1.00/0.997	0.998/0.994	N/A	1.00/0.999
Sphere2500	0.958/1.00	1.00/0.595	N/A	1.00/0.993

Note: Results for l_1-SLAM were omitted for Sphere2500 because the algorithm can not be applied to 3D datasets. Results for l_1-SLAM were omitted for City10000 because the algorithm ran out of memory while attempting to perform the optimization.

[16] N. Sünderhauf, *Robust Optimization for Simultaneous Localization and Mapping*. PhD thesis, Chemnitz University of Technology, 2012.

[17] Y. Latif, C. Cadena, and J. Neira, "Realizing, reversing, recovering: Incremental robust loop closing over time using the iRRR algorithm," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4217, 2012.

[18] N. Sünderhauf and P. Protzel, "Switchable constraints vs. max-mixture models vs. RRR-a comparison of three approaches to robust pose graph SLAM," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5198–5203, IEEE, 2013.

[19] R. Kannan and C. Monma, "On the computational complexity of integer programming problems," in *Optimization and Operations Research* (R. Henn, B. Korte, and W. Oettli, eds.), vol. 157 of *Lecture Notes in Economics and Mathematical Systems*, pp. 161–172, Springer Berlin Heidelberg, 1978.

[20] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization." `https://github.com/RainerKuemmerle/g2o`, 2014.

[21] Y. Latif, C. C. Lerma, and J. Neira, "Rrr." `https://github.com/ylatif/rrr`, 2014.

[22] P. Agarwal, E. Olson, and W. Burgard, "Max-mixture - open source implementation with g2o." `https://github.com/agpratik/max-mixture`, 2012.