Loop Closure Transformation Estimation and Verification Using 2D LiDAR Scanners

Nicholas Corso

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-73
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-73.html

May 16, 2013

Acknowledgement

I would like to acknowledge my research advisor for her continued guidance and support in all of my research endeavors.

Abstract

In many simultaneous localization and mapping (SLAM) systems, it is desirable to exploit the fact that the system is traversing though a previously visited environment. Once these locations, commonly known as loop closures, have been detected the system must be able to both compute and verify the relative transformation between proposed locations. In this thesis we present two independent algorithms, using 2D LiDAR scanners, for robustly computing the transformation between arbitrary locations with overlapping geometry and validating the resulting transforms. First, a scan matching algorithm based on a genetic search and a fractional distance metric is presented. Secondly, two metrics are proposed to verify the recovered transforms. Through experimental results the proposed algorithms are shown to robustly estimate and validate loop closure transformations for both manually and automatically defined candidates. Finally, the proposed algorithms are demonstrated in an end-to-end SLAM framework.

Contents

1 BACKGROUND

1.1 Introduction

In recent years there has been great interest in the mapping of indoor environments. Driven by applications such as historical preservation, emergency response, and energy efficiency simulation, it is often desirable to rapidly position oneself in an environment even when no prior knowledge of the environment is available. Known as simultaneous localization and mapping (SLAM), the task is to build a geometrically consistent map while simultaneously positioning oneself within it. Despite extensive research, large scale operation of SLAM systems remains difficult due to issues such as unmodeled sensor bias, calibration errors, or incorrect data association.

An important class of data association problems is known as loop closure. As a mobile robot traverses around an unknown environment, small incremental errors can compound and cause the system's position estimate to diverge. However, if the robot is able to recognize previously visited locations, a new position estimate can be derived that incorporates this constraint in a topologically consistent manner. Previous research, such as [1], [2], and [3], has formalized the SLAM problem so that loop closure constraints can be utilized to generate consistent robot trajectories. Figure 1 shows an example of loop closure whereby small errors in the blue trajectory compound so that when an area is revisited, the result is inconsistent. However, enforcing the constraint that the start and end of the loop are the same location allows for the recovery of an optimized and consistent trajectory.

Regardless of which formalization of the SLAM problem is chosen, loop closure detection, estimation, and verification is an essential task. With the prevalence of both vision and range based robotics system, many methods for loop closure detection have been presented. For example, [4] presents a vision based method for identifying previously visited locations from streams of image data. In addition to image based loop closure algorithms, LiDAR range data has been used to generate loop closure candidates. Granstrom *et al.* present an offline supervised learning approach that utilizes 2D range data by reframing the loop closure identification task in a machine learning context [5]. In contrast, Brunskill *et al.* presents an online method for generating maps using a graph partitioning technique known as spectral clustering [6]. A common limitation of detection

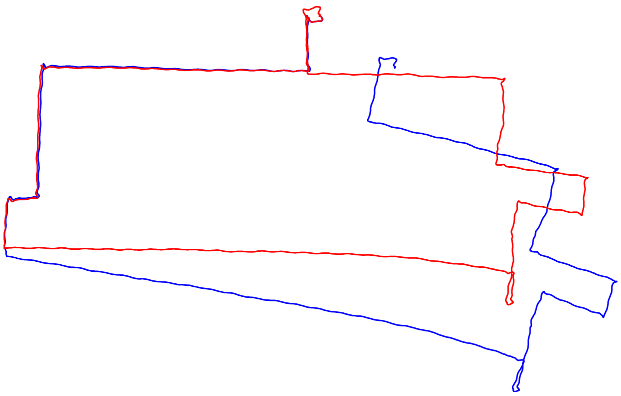


Figure 1: An example of loop closure. The blue path denotes a typical path without utilizing any loop closure constraints. The red path shows the result of enforcing the start and end positions to be in the same location.

algorithms is that false positives can cause erroneous loop closures which degrade the performance of the mapping system and in some cases cause the recovered map to be topologically inconsistent.

Once loop closure constraints have been identified, the metric transformation between locations must be computed before the constraints can be incorporated into a SLAM framework. Many techniques, such as Iterative Closest Point (ICP), utilize range data available from LiDAR scanners to estimate the transformation [7]. Due to the nonlinear nature of the objective function, ICP relies on an accurate initial guess of the transformation to succeed. As seen in Figure 1, the accumulated error between loop closure locations can be large and an adequate initial condition is not always available. Inclusion of erroneous loop closure transformations can lead to undesirable trajectories and geometrically inconstant maps. Additionally, systems that utilize the 2.5D assumption, such as [8], [9], and [10], can suffer from ground or ceiling strikes which may cause iterative optimization procedures, such as ICP, to converge to a local minima.

While the majority of previous research is focused on loop closure identification, this thesis focuses on the transformation estimation and verification problems. The main contribution of this thesis is twofold. We first present an automated algorithm to estimate the transformation between loop closure constraints using 2D LiDAR scanners. Based on a genetic search technique and a distance metric that specifically models outliers, the algorithm is able to provide robust transformation estimation even when an approximate initial condition is not known. Secondly, we propose an independent method for vetting possible loop closure candidates. Decoupled from the detection and transformation estimation tasks, the technique can be applied to a wide verity of 2D

LiDAR based loop closure systems for verification of candidate loop closures.

1.2 Related Work

The problem of matching range data when a good initial condition is not available has been addressed in previous works. Bosse and Zlot present a novel method based on projection histograms and entropy sequences to directly compute an initial condition for iterative scan matching [9]. Their procedure decouples the rotation and translation estimates into a series of histogram correlations. While their method is global, in that it does not depend on the initial condition, it does not allow recovery from erroneous results. Another global approach, presented by Checchin *et al.*, frames the task as an image registration problem [11]. By transforming the range data recovered from a microwave radar scanner using the Fourier-Mellin Transform, they compute an automated and global estimate for the rigid transformation between adjacent radar scans through correlative techniques. Their method is able to incorporate not only the range but also the power information of the radar image.

In contrast to the global strategies of [9] and [11], several authors have proposed methods based on genetic algorithms to match scans with poor initial conditions. Martinez *et al.* suggest using a genetic search of the solution space to obtain an initial condition [12]. Using ICP's error metric as a fitness function, they evaluate and mutate a set of initial estimates until a single dominant transformation remains. The authors argue that the genetic search provides a rough estimate that can be refined through the standard ICP algorithm. Lenac *et al.* also suggest using a genetic search to find a rough initial condition followed by the ICP algorithm to optimize the result of the genetic search [13]. The main distinction made in [13] is to use a correlative fitness function so the fitness metrics can be evaluated using a simple lookup table.

In this thesis we introduce a genetic scan matching algorithm that differs from the previous works in a few fundamental ways. First, the fitness function is augmented to include explicit modeling of outlier points. Using the metric presented in [14], the algorithm stochastically explores the solution space for an optimal transformation while also simultaneously developing a labeling of the points as inliers or outliers. Secondly, we apply a transformation function to each chromosome that forces it into the nearest local minima of the fitness function. Lastly, we dramatically speed up convergence of the proposed algorithm by exploiting the nature of the objective function.

The second problem we address in this thesis, namely the verification of loop closure transformations, has only been addressed in a few previous works. In [15], verification metrics based on the sum of the maximal correlation between projection histograms and entropy sequences are used to validate the estimated loop closure transformation. The authors lump together detection, transformation estimation, and verification to report performance metrics of 48% true positive rate for 1% false positive rate. The result however is later improved to 51% in [9]. The work presented in [5] suggests a verification criteria based the final alignment of the scans after the transformation has been applied. Tested on unstructured outdoor environments, a true positive rate of 89% is reported while maintaining an 1% false positive rate for their combined loop closure detection, estimation, and verification system.

In contrast to the previous works, this thesis deals strictly with loop closure verification in indoor environments. Loop closures can be identified from a wide verity of sources and therefore verification algorithms must be independent of detection method. In this thesis we present an algorithm agnostic to detection source which allows it to be applied to a wide range of loop closure systems. Furthermore, indoor environments often contain areas where geometry is ill-conditioned. Areas, such as long straight hallways with low ceilings, can prove challenging due to the lack of distinguishable features present in LiDAR range data. Through experimental results, we show that a combination of two separate metrics can provide robust validation of candidate transforms in indoor environments.

2 TRANSFORMATION ESTIMATION

2.1 Fractional Iterative Closest Point

The classical ICP problem is framed in the following manner. Given two dimensional points sets, P and Q, ICP iteratively attempts to find transformation $T(\cdot,\mu)$ that minimizes the following objective function:

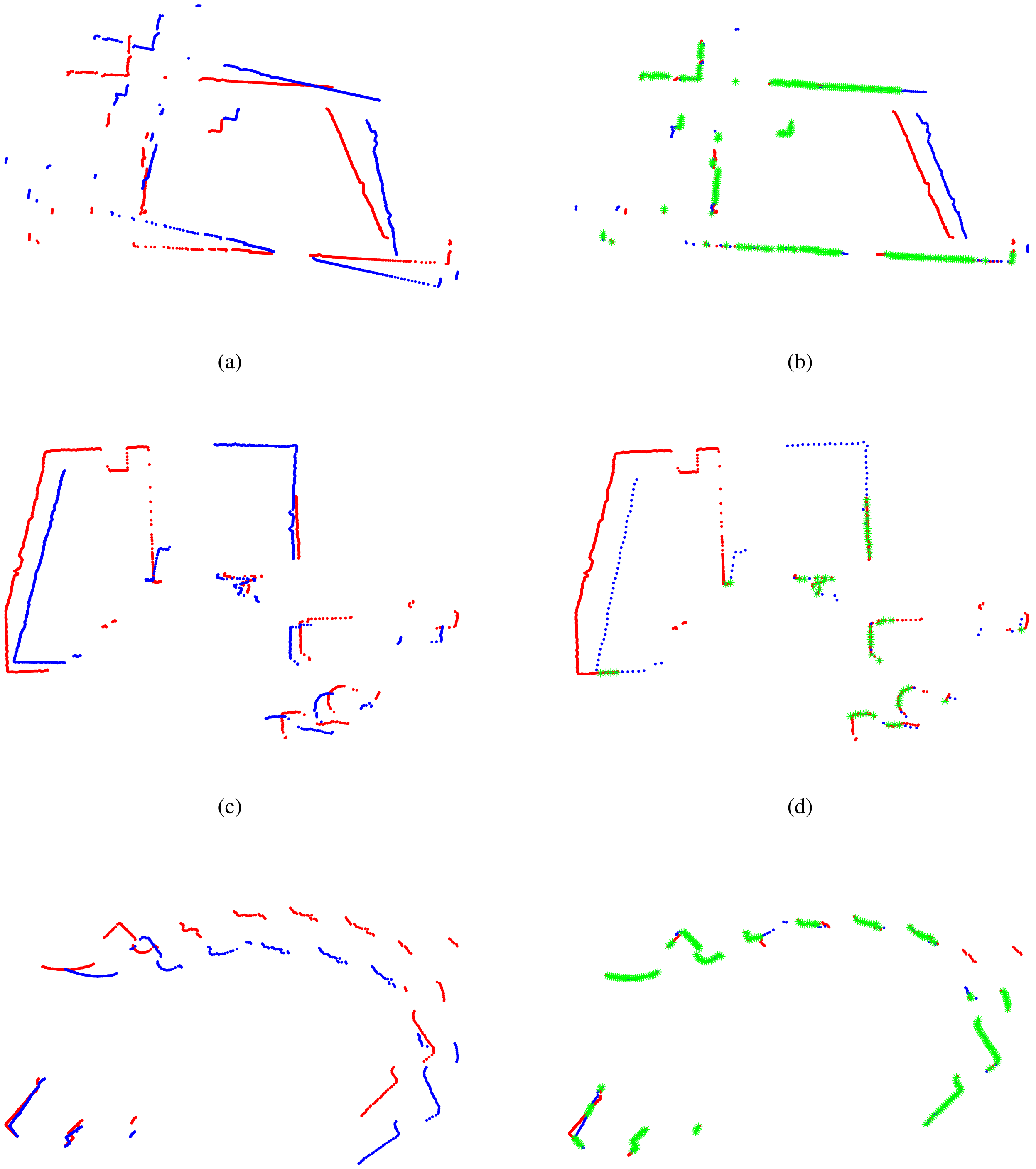
$$e = \sum_{\mathbf{q}_i \in Q} \|\mathbf{p}_i - T(\mathbf{q}_i,\mu)\|^2 \tag{1}$$

where $\mathbf{p}_i \in P$ and $\mathbf{q}_i \in Q$ are matched elements and $T(\cdot,\mu)$ is the transformation operator that rotates and translates point $\mathbf{q}_i$ into the reference frame of P using transformation parameters μ. In this thesis we are primarily concerned with the specific case of 2D point matching. Variants of the metric of (1) have been suggested for improving convergence accuracy and speed. In this work, we use the following metric which is a combination of the point-to-line metric [16] and the explicit modeling of outliers [14]:

$$e = \frac{1}{f^{\lambda}} \sqrt{\frac{1}{|D_f|} \sum_{\mathbf{q}_i \in D_f} \|\mathbf{n}_i^T (\mathbf{p}_i - T(\mathbf{q}_i,\mu))\|^2} \tag{2}$$

where f is the fraction of points that are considered inliers, D_f is the set of inlier points, $\mathbf{n}_i$ is the normal vector of the surface at point $\mathbf{p}_i$, and λ is a free parameter that controls how aggressively points are labeled inliers. The fractional iterative closest point (FICP) algorithm attempts to minimize the objective function (2), known as the fractional root mean square distance (FRMSD), for the optimal transform parameters $T(\cdot,\mu)$ and inlier set $D_f \subseteq Q$ by iterating the following steps [14]:

- Given an initial transform $T(\cdot,\mu_0)$, points in Q are matched to their nearest neighbors in set P.
- Equation (2) is minimized to obtain an inlier set D_f assuming a fixed $T(\cdot,\mu_0)$.
- Using inlier set D_f, the transform parameters μ are recovered using a first-order Taylor expansion of (2) and solving for the optimal linear estimate of μ [9].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2: Examples of performing the FICP algorithm on various scans. (a), (c), and (e) The individual scans, depicted in red and blue, aligned using the initial estimate of the transformation. (b), (d), and (f) The scan pairs after the FICP algorithm has been performed. The portion of geometry declared as the inlier points are denoted in green.

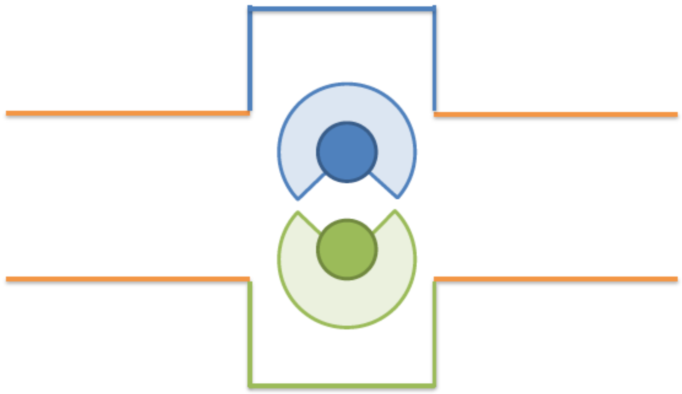


Figure 3: An example of ill-conditioned scan matching arising from two scanners with minimal overlapping of fields of view. Each scanner can see their respective colored geometry while the geometry seen by both scanners is shown in orange.

Figure 2 shows example results of the FICP algorithm. Figures 2(a), 2(c), and 2(e) depict pairs of LiDAR scans, shown in red and blue, aligned using an initial estimate of the transformation based upon a priori information. Figures 2(b), 2(d), and 2(f) depict the alignment of scan pairs from 2(a), 2(c), and 2(e) respectively after the FICP algorithm has been performed. The portions of geometry that was found to be part of the inlier set D_f are shown in green.

By solving for both the set of inliers and the optimal transformation parameters, the objective function models the presence of outliers in the data without any prior knowledge of their distribution [14]. The process of iteratively segmenting geometry and recovering the transformation serves the basis for the algorithms presented in this section.

2.2 Local Submaps

Loop closure constraints arising from revisiting a previously visited location can often result in situations where the orientation offset can be as large as 180° [17]. Due to occlusions and limited fields of view, scan matching could become an ill-conditioned problem. Figure 3 shows an example of how limited field of view can lead to ill-conditioned geometry. Shown in orange, the minimal overlapping fields of view between the two scanners allows only two parallel lines to be matched.

To overcome this limitation, we aggregate temporally adjacent scanner readings into a single

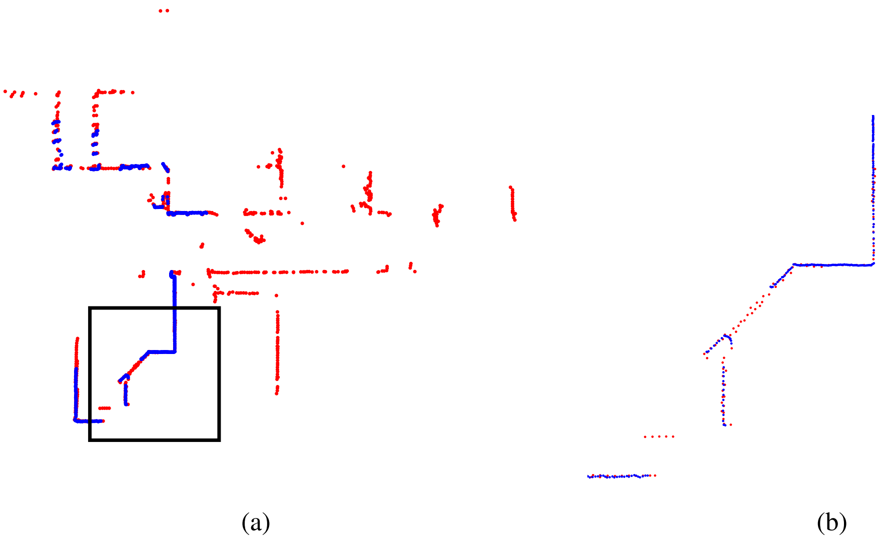


Figure 4: A typical result of RBPF based submapping. The original sensor's readings are shown in blue while the resulting submap is shown in red. (a) The amount of visible geometry has been expanded well beyond the original sensor's readings. (b) A close up of the section of the submap denoted by the black square shown in (a).

local submap to be used both in our proposed transformation estimation of Section III-B and verification step of Section IV. Using a Rao-Blackwellized Particle Filter (RBPF), sequential scans are fused into a single geometrically consistent map [18]. Although typically formalized for large scale mapping, we utilize the RBPF to generate accurate maps of subsections of the environment. The approach we take here follows the classical grid mapping framework with two distinctions.

First we generate odometry by pairwise scan matching of temporally adjacent scans. Using Eqn. (2), both the relative change in position and inlier set D_f is computed. Only the inlier points are considered when merging points into the map or computing the weighting factor. Secondly, rather than using a strict discretization to create the grid map, we compute the average point in each grid cell. By using the average point as the representative sample for each grid, quantization errors can be mitigated.

Figures 4 and 5 show typical results of RBPF based submapping. The blue points represent the sensor's original readings while the red points represent geometry that is added by the submapping procedure. By aggregating temporally adjacent scans, the amount of visible geometry for scan matching has been substantially increased. Additionally, as seen in Fig. 5(b), tracking the average point of each grid cell mitigates quantization effects that result from the grid mapping approach.

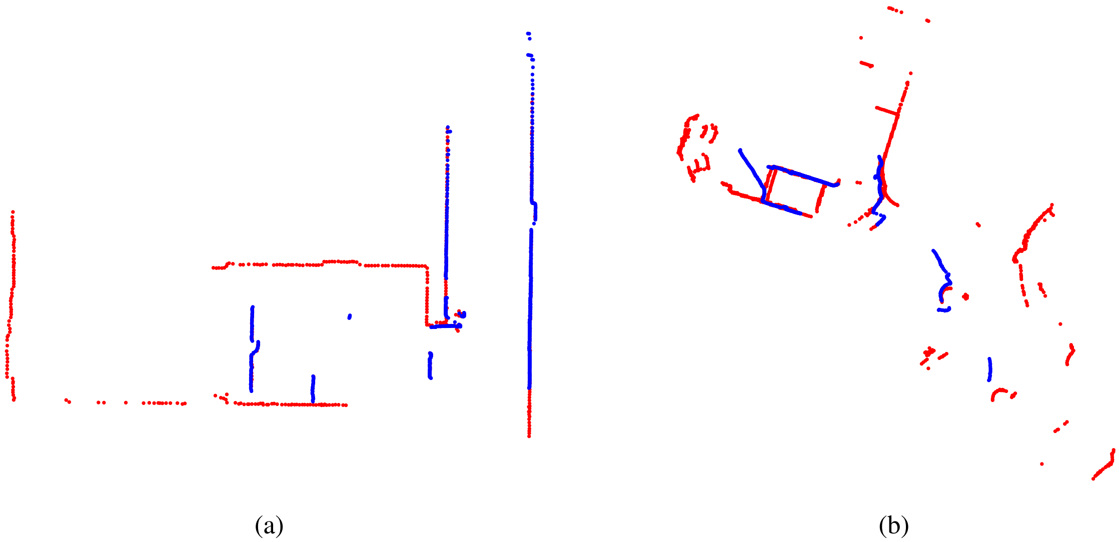


Figure 5: Two example results of RBPF based submap generation. The original sensor readings, shown blue, are superimposed on the generated submap, shown in red. In each example the amount of visible geometry has been expanded.

The original corner, denoted by the black square in Fig. 5(a), matches very closely to the corner reconstructed during submap construction.

The number of temporally adjacent scans used in submap construction impacts both the construction time and the amount of geometry in the reconstructed RBPF based submapping algorithm. To limit computation time, we choose local submap sizes based on the following heuristic. Given a location of interest, we collect scans from neighboring locations until either the estimated cumulative translation has exceeded N meters or the estimated rotation has exceeded M degrees. For the ambulatory backpack system we use during the experiments in the thesis, we empirically determined values of $N = 2$ and $M = 30^\circ$ to work well [10, 17, 19].

2.3 Fractional Genetic Scan Matching

A widely known limitation of non-linear optimization techniques is that they can fall into local minima of the objective function. To overcome this limitation, the algorithm is typically supplied with an initial condition $T(\cdot,\mu_0)$. The resulting solution is often highly dependent on choice of $T(\cdot,\mu_0)$ and poor choice of initial condition can lead to an undesirable solution. A class of

Algorithm 1 The FGSM Algorithm

```
$\mu_0 \leftarrow$ initial estimate
$S_0 \leftarrow random\_chromosomes(\mu_0)$
while $S_{n+1} \neq S_n$ do
    $R_n \leftarrow \emptyset$
    for $\mathbf{s}_i \in S_n$ do
        $\mathbf{r}_i \leftarrow ICP(\mathbf{s}_i)$
        $R_n \leftarrow R_n \cup \{\mathbf{r}_i\}$
    end for
    $R_n \leftarrow best\_of(R_n)$
    $S_{n+1} \leftarrow R_n \cup make\_children(R_n)$
end while
return $\mu$ and $D_f$ from best chromosome.
```

stochastic optimization techniques, known as genetic algorithms, can overcome this problem by providing a derivative-free approach for exploring the solution space. Each solution, or chromosome, is evaluated according to a cost function and assigned a fitness value. Thus, genetic searches are able to better cope with the presence of local minima or poor initial conditions [20].

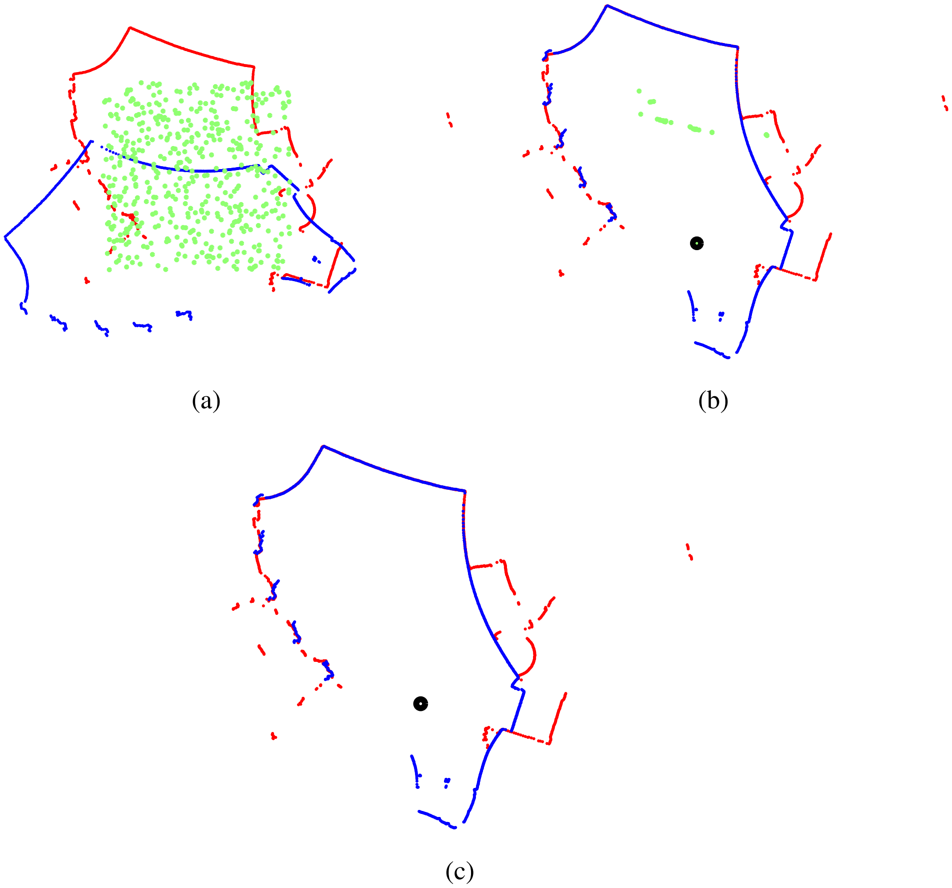
Genetic search (GS) algorithms operate by first considering a randomly distributed set of solution chromosomes. Using an appropriate fitness metric, the population is evaluated and each chromosome is assigned a fitness value. The most fit chromosomes are retained and the rest are discarded. The population is then replenished by creating new chromosomes whose parameters are chosen randomly from the surviving chromosomes. The generated solutions are mutated by adding random variation to the chromosomes' parameters and the whole process is iterated until the population reaches a stable configuration. In this manner, GS algorithms avoid local minima while exploring solutions that were not in the original sampling of the solution space.

Similar to [12] and [13], we parameterize the solution space using a three element chromosome. Specifically, each individual in the population consists of two translational and one rotational parameter.

$$\mu = \begin{bmatrix} \Delta x, & \Delta y, & \Delta\theta \end{bmatrix}^T \tag{3}$$

In this parametrization Δx and Δy represent incremental translations in the x and y directions respectively while $\Delta\theta$ represents a counter-clockwise rotation by angle θ.

In contrast to the existing approaches of [12] and [13], our proposed algorithm, Fractional Ge-



(a)

(b)

(c)

Figure 6: An example of the FGSM process. The scans are shown in red and blue while the chromosomes are shown in green. A black circle denotes the location of the best chromosome in the population. (a) The initial chromosome sampling. (b) After a few generations the chromosomes have converged to a subset of the local minima. (c) After enough generations have passed the population has converged to a single dominant location.

netic Scan Matching (FGSM), introduces a transformation function into the lifetime of each chromosome. Specifically, FGSM starts by considering a random set of chromosomes, S_0, sampled uniformly from a rough initial condition, μ_0. The initial condition allows for a priori information, such as odometry, to be incorporated into the FGSM algorithm. Denoting each individual chromosome $\mathbf{s}_i$, we transform the initial population by applying the ICP optimization procedure to obtain the optimal transformation $\mathbf{r}_i$. The fitness value for each chromosome $\mathbf{s}_i$ is set to the residue of the objective function (2) evaluated at $\mathbf{r}_i$. The chromosomes with the best fitness scores are replaced by their optimal transform and the rest are discarded. Because each chromosome is replaced by its optimal transformation $\mathbf{r}_i$ it is appropriate to think of the FGSM algorithm as a combination of a stochastic and gradient based search over a subset of the local minima of the objective function.

To simulate the mutation between generations, random i.i.d. Gaussian noise is added to the newly generated chromosomes. The variance of the added Gaussian noise is obtained by considering the variance of the remaining chromosomes' parameters. Specifically, if the population's n parameter has variance σ_n^2 then a random sample drawn from $\mathcal{N}(0, \sigma_n^2)$ is added to each of the generated chromosome's n-th parameter. The process of transformation, fitness evaluation, selection, and mutation is iterated until the population converges to a single dominant trait. This procedure is summarized in Algorithm 1.

An example of the FGSM algorithm is shown in Figure 6. Fig. 6(a) shows the initial chromosome population sampled uniformly around the supplied initial condition μ_0. Shown in Fig. 6(b) is the distribution of chromosomes after a few iterations. Notice that the chromosomes only inhabit the a few locations corresponding to the local minima of the objective function (2). After enough iterations have passed, the population, as shown in Fig. 6(c), has converged to a single dominant chromosome representing the correct transformation parameters.

Figure 7 shows a comparison of the FICP and FGSM algorithms when the initial condition is far from the optimal transformation. Figures 7(a) and 7(b) show the scans at the location specified by the initial condition. The results of FICP only are shown in 7(c) and 7(d). Notice that the FICP algorithm converges to the incorrect transformation due to local minima in the objective function. Figures 7(e) and 7(f) show that the FGSM algorithm was able correctly align the scans despite the erroneous initial condition.

2.4 Efficient Implementation

As seen in line 6 of Algorithm 1, each chromosome is used as the initial condition to the ICP procedure and is replaced by the resulting locally optimal transform. While this forces each chromosome into a local minima of the objective function, it is also computationally expensive. Furthermore, as the population nears convergence many chromosomes reside in approximately the same location. This causes redundant computation that should be avoided.

Examining (2), we notice that if both the point matches and the inlier set are given, then the

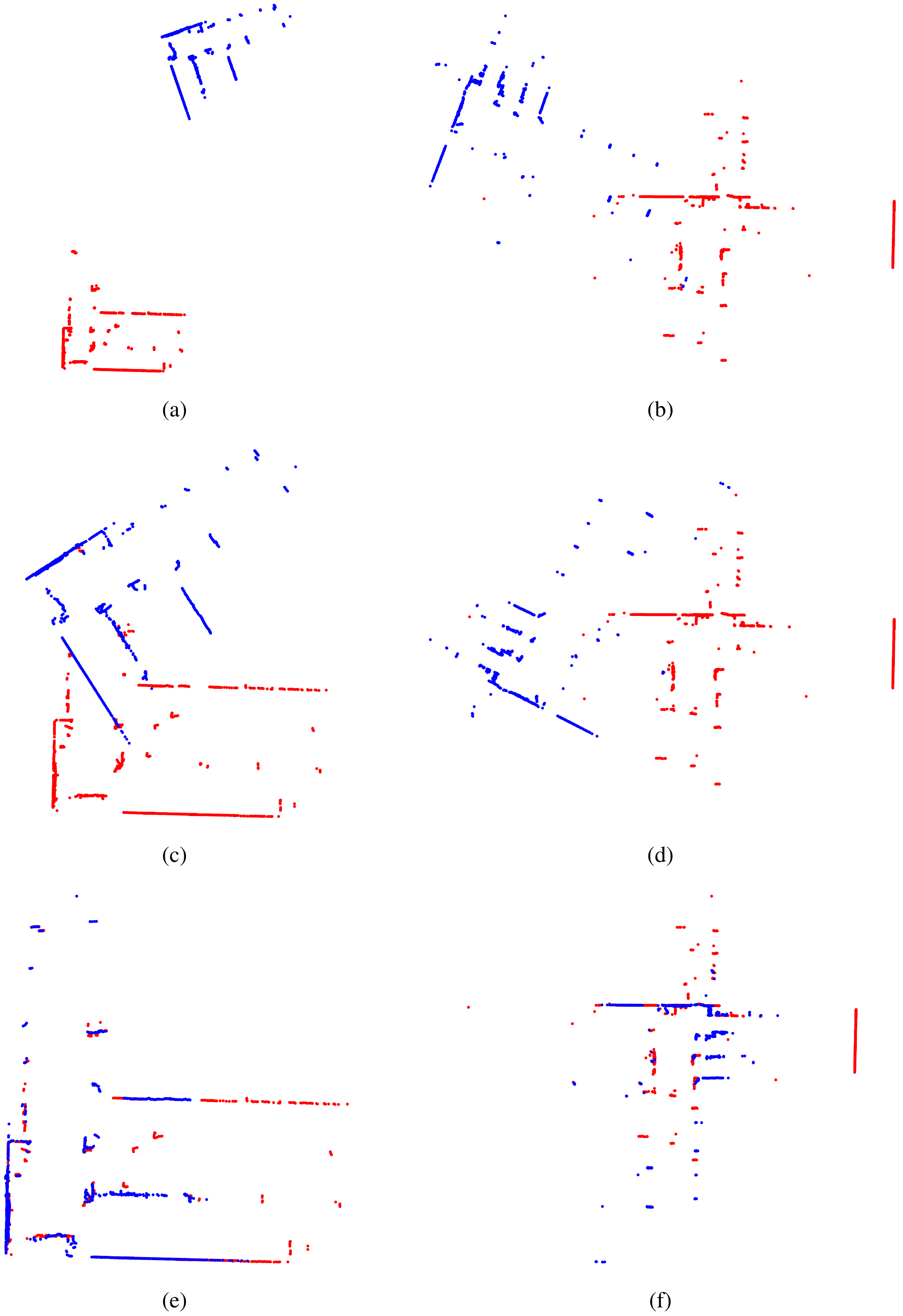


Figure 7: Example results of the FGSM algorithm. (a)-(b) The LiDAR scans, depicted in red and blue respectively, are shown in the initial condition to both the FICP and FSGM algorithms. (c)-(d) The result of applying only FICP. (e)-(f) The scans correctly aligned using the result of the FGSM algorithm.

ICP problem reduces to minimizing the following objective function:

$$e = \sum_{\mathbf{q}_i \in D_f} \|\mathbf{n}_i^T(\mathbf{p}_i - T(\mathbf{q}_i, \mu))\|^2 \tag{4}$$

Following [9], Equation (4) can be rewritten as:

$$e = \mathbf{e}(\mu,\, D_f)^T \mathbf{e}(\mu,\, D_f) \tag{5}$$

where i-th element of vector $\mathbf{e}(\cdot)$ corresponds the error contributed from the i-th element of inlier set D_f. Using the Taylor expansion with respect to the transformation parameters, a linearized version of $\mathbf{e}(\cdot)$ can be written as:

$$\mathbf{e}(\mu_0,\, D_f) \approx \mathbf{e}(\mu_0,\, D_f) + \mathbf{J}\mu_0 \tag{6}$$

where μ_0 is the linearization point and $\mathbf{J} \in \mathbb{R}^{|D_f| \times 3}$ is the Jacobian of $\mathbf{e}(\cdot)$ with respect to μ. The i-th row of $\mathbf{J}$ is computed via

$$\mathbf{J}_i = \begin{bmatrix} -\mathbf{n}_i \\ \mathbf{n}_i^T \begin{bmatrix} \sin(\Delta\theta) & \cos(\Delta\theta) \\ -\cos(\Delta\theta) & \sin(\Delta\theta) \end{bmatrix} \mathbf{p}_i \end{bmatrix}^T \tag{7}$$

and the optimal transform can be found using

$$\mu = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathbf{e}(\mu_0,\, D_f) \tag{8}$$

Equations (7) and (8) reveal that when the Jacobian is evaluated around small angles, $\theta \approx 0$, deviations $\delta\mu$ from the linearization point result in insignificant changes to the resulting transformation provided they are small enough. This analysis relies on the fact that the objective function is continuous, which in this case is not true. However, (4) is locally continuous provided the deviations $\delta\mu$ are small enough to not alter the nearest neighbor point matches.

In order to take advantage of the near-linearity of the Jacobian, we take the following strategy. We first discretize the solution space into small grid cells and bin chromosomes according to their parameters. Each time a chromosome is used as the initial condition for ICP, a record is kept of the resulting transformation. In this way, redundant computation can be eliminated by building a lookup table between initial conditions and the resulting transforms.

The amount of discretization in the solution space represents a trade-off between speed and adherence to the underlying assumptions. The discretization must represent small enough changes to not violate the small-angle approximation or the continuity of the objective function, yet be large enough to effectively speed up the genetic search. We have empirically found a spacing of 10 centimeters and 1 degree to provide the best trade-off between computation and accuracy for our experimental setup. Section 4.2 provides empirical justification for these values.

3 VERIFICATION CRITERIA

Even though the FGSM algorithm is a robust way of matching scans with unknown initial conditions, care must still be taken to eliminate erroneous transformations. Poor scan matches can arise when a loop closure has been falsely identified or when geometry is ill-conditioned. In these situations it is best to recognize that the algorithm has failed and avoid using the loop closure constraint. We now propose two metrics, independent of the detection and estimation tasks, to robustly reject erroneous loop closure transformations while maintaining a high true positive rate. The proposed metrics are designed to encompass two important factors for characterizing a loop closure transformation: the amount and complexity of the shared geometry.

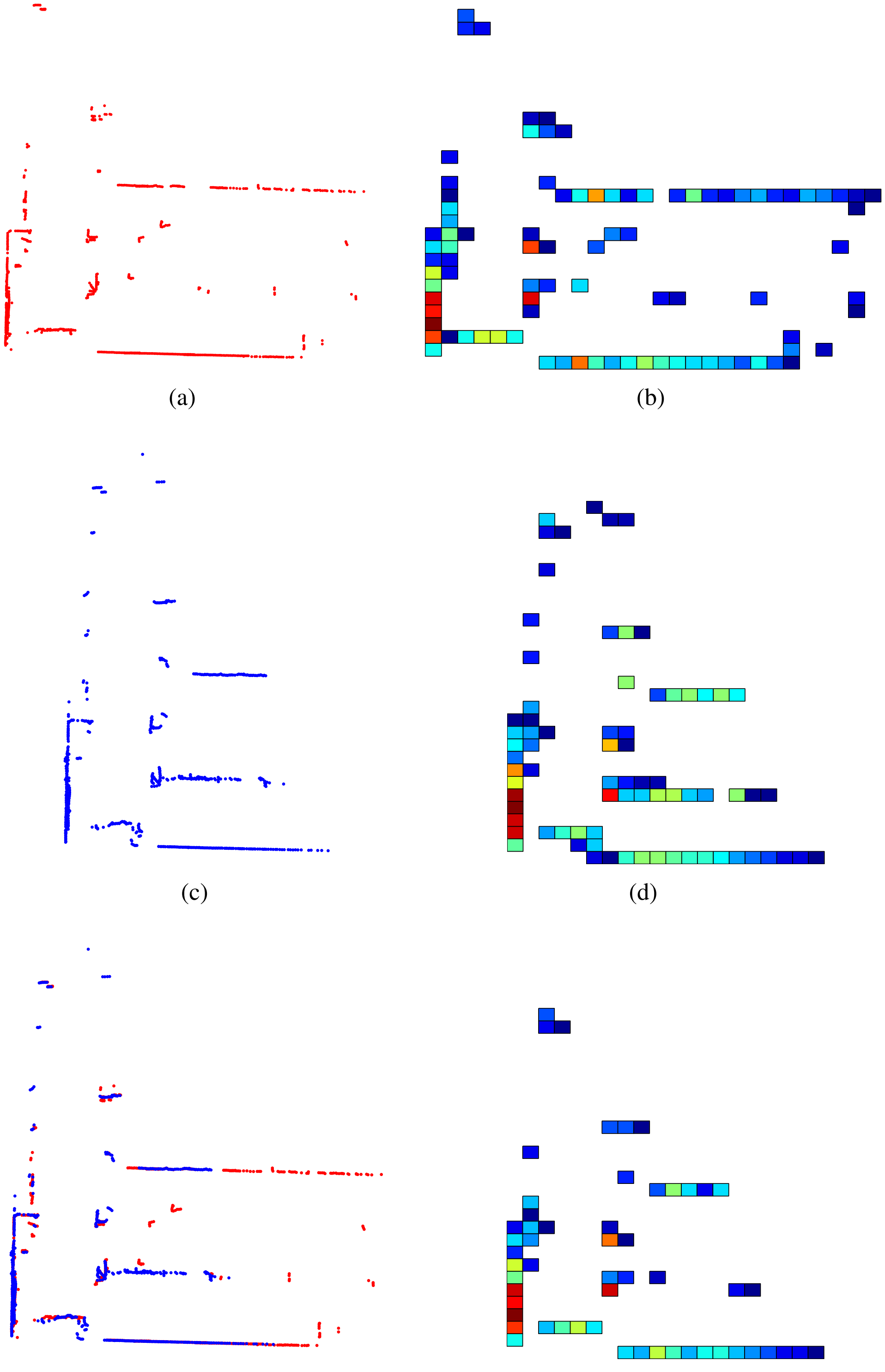
3.1 Correlation Metric

The first metric characterizes the amount of shared geometry between matched scans. Using the proposed transformation, the scans are rotated and translated into alignment. Next, a pair of normalized two-dimensional histograms are built by binning the scans' points into equally sized spatial bins. Then the histograms are correlated using the intersection kernel [21]:

$$c = \sum_{i,j} \min(h_1(i,j), h_2(i,j)) \tag{9}$$

where $h_1(i,j)$ and $h_2(i,j)$ corresponds to the i,j-th bin of the respective histograms. This measure evaluates to $c = 1$ if the histograms are identical and $c < 1$ if they differ.

Figure 8 shows an example computation of the correlation metric. Figures 8(a) and 8(c) show the individual scans of a candidate loop closure. The normalized 2D histograms, shown in 8(b) and 8(d) respectively, are generated by spatially binning the scan points. The proposed loop closure transformation is shown in Fig. 8(e). The intersection histogram, shown in Fig. 8(f), is obtained via the $\min(\cdot,\cdot)$ operator in (9). The resulting correlation score of $c = 0.6634$ is obtained by summing the intersection histogram.



(a) (b) (c) (d) (e) (f)

Figure 8: Scan correlation example. (a) and (c) A candidate pair of scans for correlation. (b) and (d) The 2D histograms generated from the respective scans. (e) The proposed alignment between scans. (f) The histogram generated using the intersection kernel. The resulting correlation score: $c = 0.6634$.

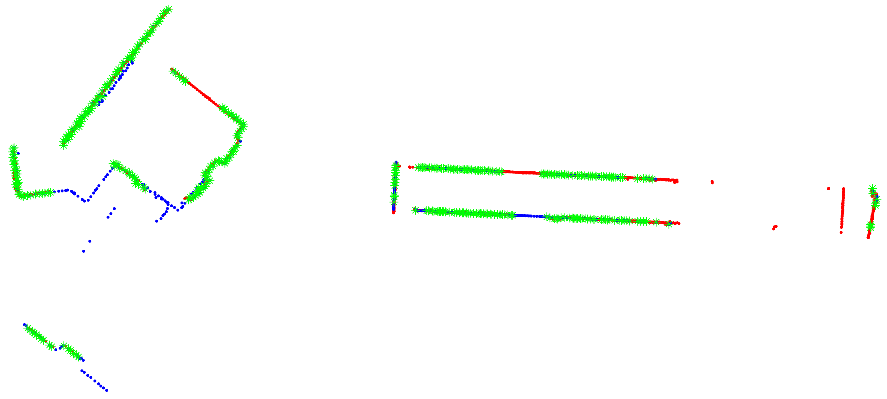

(a) (b)

Figure 9: Example scan matches in environments with differing geometric complexities. The individual scans, shown in red and blue, share the portion of geometry highlighted in green. (a) A complex environment with many corners yields a complexity score of $r = 0.4979$. (b) A long narrow hallway, which is ill-conditioned for scan matching, has a complexity score of $r = 0.1114$.

3.2 Complexity Metric

Our second metric measures the complexity of the overlapping geometry. Using the proposed transform, (2) can be minimized to obtain an inlier set D_f. Considering only the points in D_f, the portion of geometry that is shared between grid maps is identified. Next, the normal vectors of the inlier points, $\mathbf{n}_i$ are collected into the matrix

$$\mathbf{N} = \Big[\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_n\Big]^T \tag{10}$$

with which the correlation matrix:

$$\mathbf{R} = \mathbf{N}^T \mathbf{N} \tag{11}$$

is formed. The eigenvalues of $\mathbf{R}$, denoted by λ_1 and λ_2, are used to form the ratio

$$r = \frac{\lambda_1}{\lambda_2} \tag{12}$$

assuming $\lambda_1 \leq \lambda_2$. When geometry is ill-defined for scan matching, such as in a long featureless hallway, the normal vectors of the points lie almost entirely along the line normal to the walls. This observation implies that λ_1, and by extension r, is close to zero. On the other hand, in situations where geometry is well defined, such as in a complex environment with many corners, $\lambda_1 \approx \lambda_2$

and thus r is close to 1. While the normal vector for each point is not detected using a range sensor, in practice techniques such as SVD analysis of the point's local neighborhood can provide a reasonable estimate.

Figure 9 shows examples of candidate loop closures in environments with differing geometric complexities. A complex environment, shown in Fig. 9(a), has a many distinguishable features that aid in scan matching. The eigenvalues of the matrix $\mathbf{R}$, formed from the inlier set D_f, are approximately equal which yields a complexity score of $r = 0.4979$. Figure 9(b) shows an example of geometry that is ill-conditioned for scan matching. The portion of overlapping geometry is dominated by long parallel lines which have normal vectors that lie dominantly along a single axis. The resulting complexity score of $r = 0.1114$ is much smaller than that of the configuration shown in Fig. 9(a).

4 RESULTS

In this section we present results for both the FGSM algorithm and the loop closure transformation verification criteria. While the algorithms introduced in the previous sections can be applied to both submaps and individual scans, the results presented here utilize submaps generated with the method described in Section 2.2.

4.1 FGSM Performance Evaluation

To evaluate the performance of the FGSM algorithm we performed the following experiments. Data was taken using an ambulatory backpack system and a Hokuyo UTM-30LX [10]. The UTM-30LX is a LiDAR based scanner that collects 1080 range measurements equally spaced over a 270° field of view at a rate of 40 Hz. Data was collected from various indoor environments including two hotels, two construction site, four office buildings, and a mall.

The FGSM algorithm was evaluated by first hand-constructing a set of one hundred ground truth scan matching results. Then, random initial conditions were created by selecting values from a collection i.i.d. Gaussian random variables with increasing variance centered around the ground truth. Ten trials were conducted for each scan pair which resulted in one thousand total trials for each level of noise variance. Assuming a discretization of 10 centimeters and 1 degree, the FSGM algorithm was run and the resulting transform was recorded. If the resulting transform was within one grid cell and 0.5° of the ground truth location and orientation, then we consider it to have converged to the correct solution. For all experiments, the number of initial chromosomes was set to 100 and the initial search range was set proportional to the variance of the added noise.

Table 1 summarizes the results of the conducted experiments. In the first test we add pure translational offset to the initial condition. For small translational offsets the algorithm is able to recover the correct transformation 100% of the time. As the variance increased to 10 m^2 the accuracy follows only a minor downward trend. This shows that the FGSM algorithm is extremely robust to translational offset in the initial conditions even when the variance is as high as 10 m^2.

In the second experiment we repeat the same procedure, but add only rotational offset using

Table 1: Fractional Genetic Scan Matching Accuracy

Variances used:					
Translation (m^2):	0.1	0.3	1	5	10
Rotation (rad^2):	$\pi/10$	$2\pi/10$	$3\pi/10$	$4\pi/10$	$5\pi/10$
Translation Only:					
Accuracy (%)	100	99	97	97	98
Rotation Only:					
Accuracy (%)	92	85	80	76	78
Both:					
Accuracy (%)	92	85	76	82	81

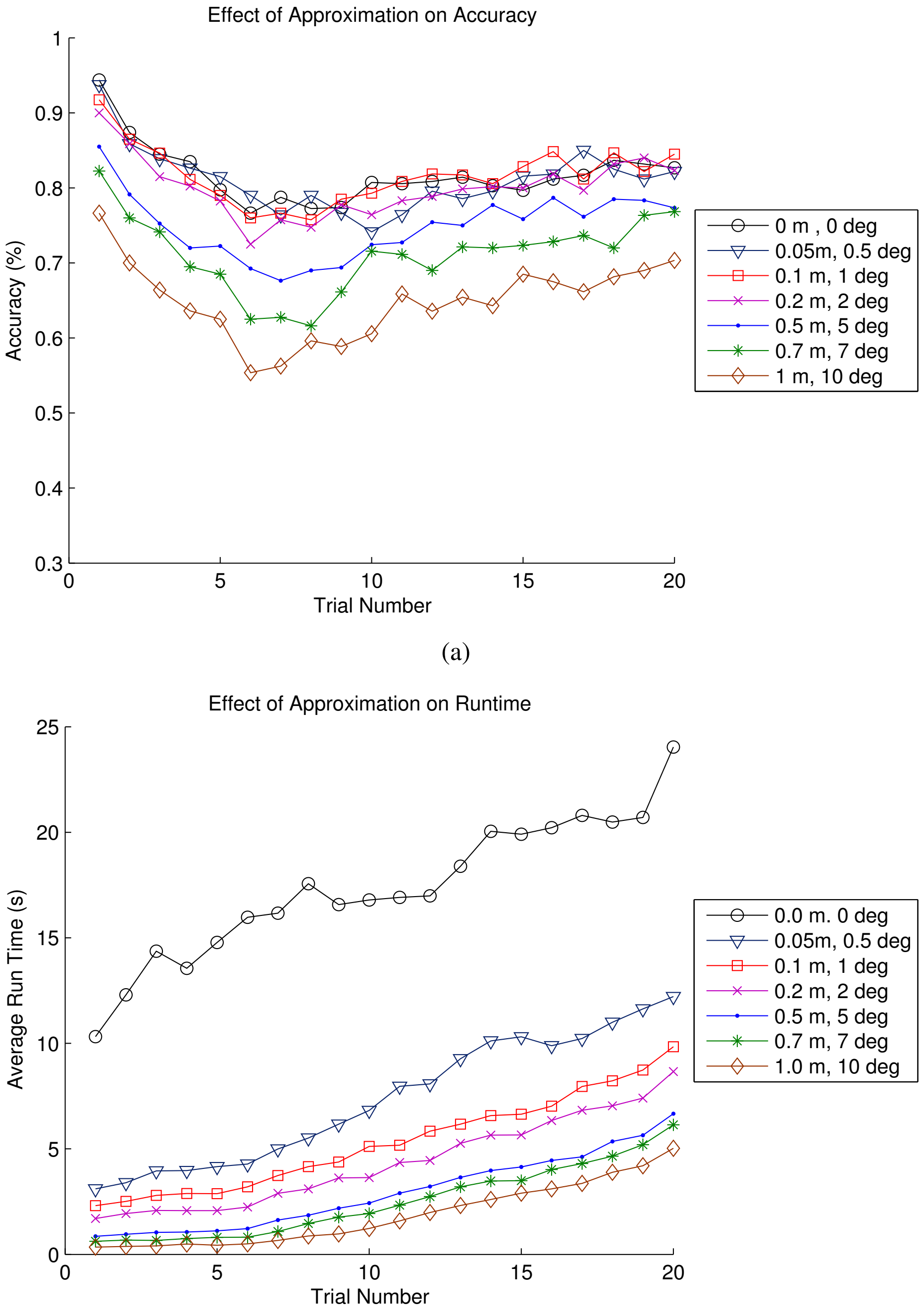
variances ranging from $\pi/10$ rad^2 to $5\pi/10$ rad^2 in increments of $\pi/10$. When the variance is $\pi/10$ the algorithm is able to recover the correct transformation 92% of the time. However, as the variance approaches $\pi/2$ the accuracy falls to 78%. This result is not surprising because the nonlinearity of (7) is in the rotational parameter. The nonlinear relationship in the rotational component makes it harder for the genetic search to arrive at the correct local minima.

In practical scenarios the error in initial condition is typically a mix of both rotation and translation. To characterize the effects of both error sources simultaneously, we repeat the experiment but this time simultaneously add variation to both the rotational and translational components of the initial condition. The algorithm performs similarly to the case of rotation only error. For small variations, the accuracy results are nearly identical to the rotation only case, while at larger variations it performed marginally better.

4.2 Effect of Discretization on the FGSM Algorithm

The following test was conducted to determine what level of solution space discretization is acceptable for efficient and robust implementation of the FGSM. The final experiment from the previous section was reran, but this time the amount of discretization of the solution space was varied along with the level of noise added. Figure 10 shows the results of this experiment. The baseline, shown in black, corresponds to no approximation.

As the solution space discretization is increased, both the average runtime and accuracy decreases. For specific discretization parameters of 0.1 meters and 1 degree, the runtime is decreased



(a)

(b)

Figure 10: The effect of solution space discretization on the FGSM algorithm. (a) Accuracy computed for different amounts of solution space discretization. (b) The average runtime of the FGSM algorithm for different levels of discretization.

by a factor of 3.25 without sacrificing a significant amount of accuracy compared to the baseline algorithm.

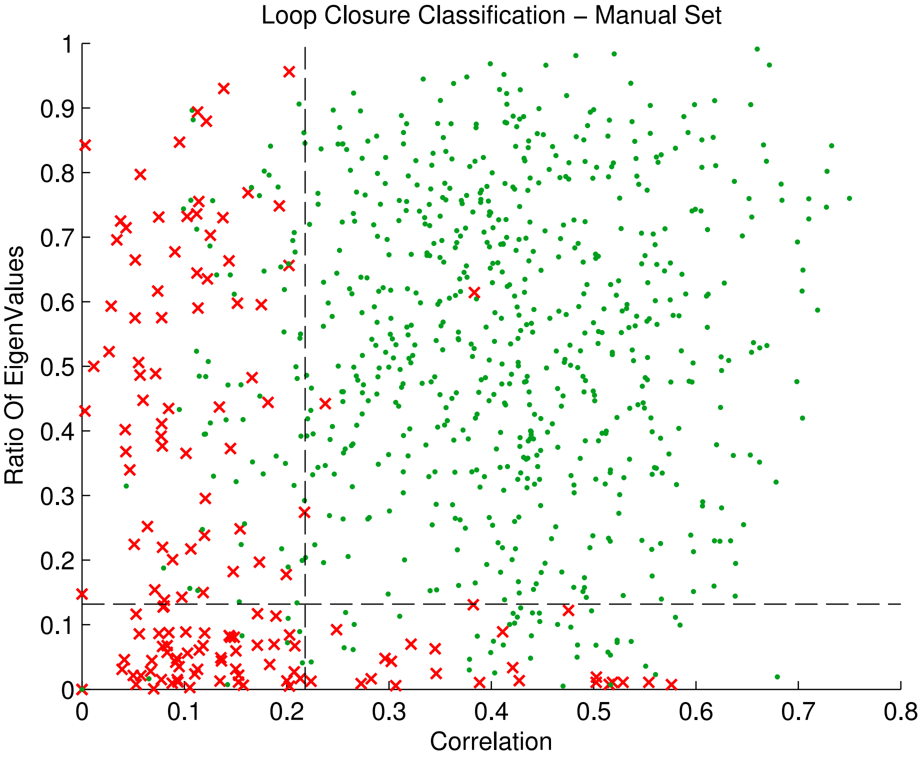


Figure 11: A scatter plot showing the distribution of loop closure candidates using the proposed metrics. A red × corresponds to a failed loop closure transformation, while a green ○ represents a successful candidate pair. The optimal thresholds are denoted using the dashed black lines.

4.3 Loop Closure Verification

In order to characterize the effectiveness of the proposed metrics for loop closure verification, a database of 20 datasets were collected from 13 unique environments. The classification metrics were evaluated in the following two scenarios. First, between 40 and 70 true loop closures were manually defined for each dataset resulting in 675 positive examples. To simulate the presence of falsely identified loop closures, 10 loop closures were randomly defined for each dataset yielding 200 potentially erroneous examples. Using an initial condition derived from dead reckoning, the FGSM algorithm was run on each loop closure candidate. The resulting transformations were then manually inspected and labeled.

Figure 11 shows a scatter plot of the results. The candidates for which the FGSM algorithm correctly identifies the transformation is shown using a green ○, while the incorrect transforms are shown via a red ×. The two regions form a near disjoint partition of the graph suggesting the use of

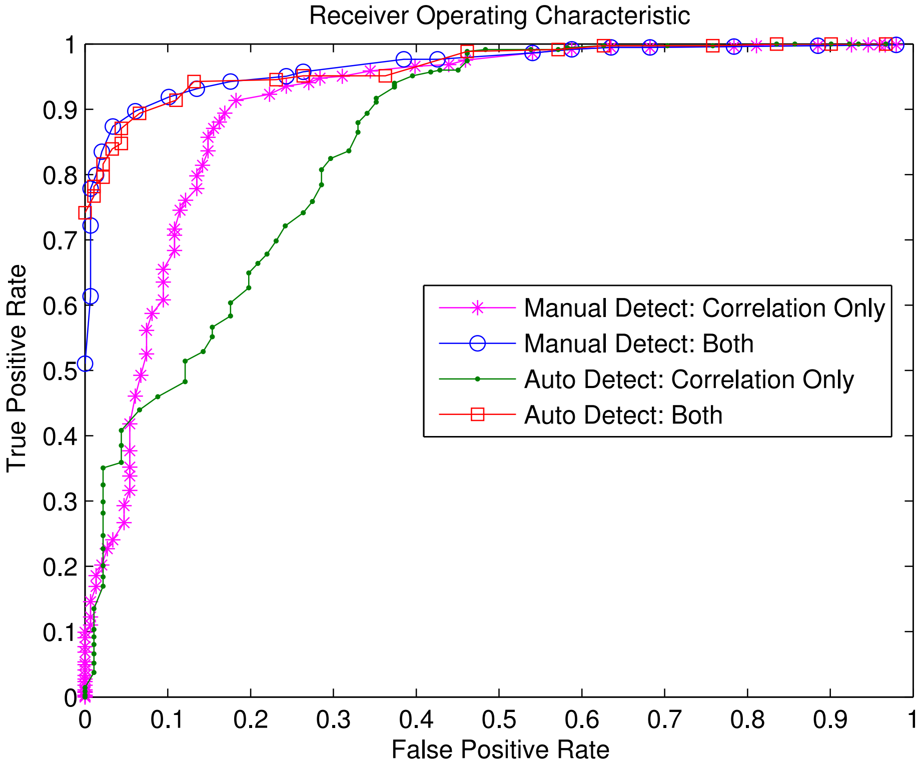


Figure 12: Comparison of receiver operating characteristic curves when using both metrics versus only correlation.

a threshold based rule. Specifically, transformations are declared correct if the ratio of eigenvalues exceeded threshold r and the correlation metric exceeded threshold c.

Statistics for various thresholds are computed and the resulting receiver operating characteristic (ROC) curve is shown in Figure 12. When the false positive rate is required to be $\leq 1\%$, the corresponding true positive rate is 84.5% for thresholds of $r > 0.132$ and $c > 0.218$. The computed area under the ROC curve was found to be 93.2%.

The same test was repeated, but this time using automatically detected loop closures. Camera imagery, along with the FAB-MAP algorithm [4], are used to automatically identify locations of possible loop closure. Across the 20 collected datasets, 439 locations are identified. Due to the repetitive nature of indoor environments, such as repeating wallpaper patterns, many erroneous locations are identified. Again using an initial condition derived from dead reckoning, the FGSM is run on each of the candidate loop closure pairs.

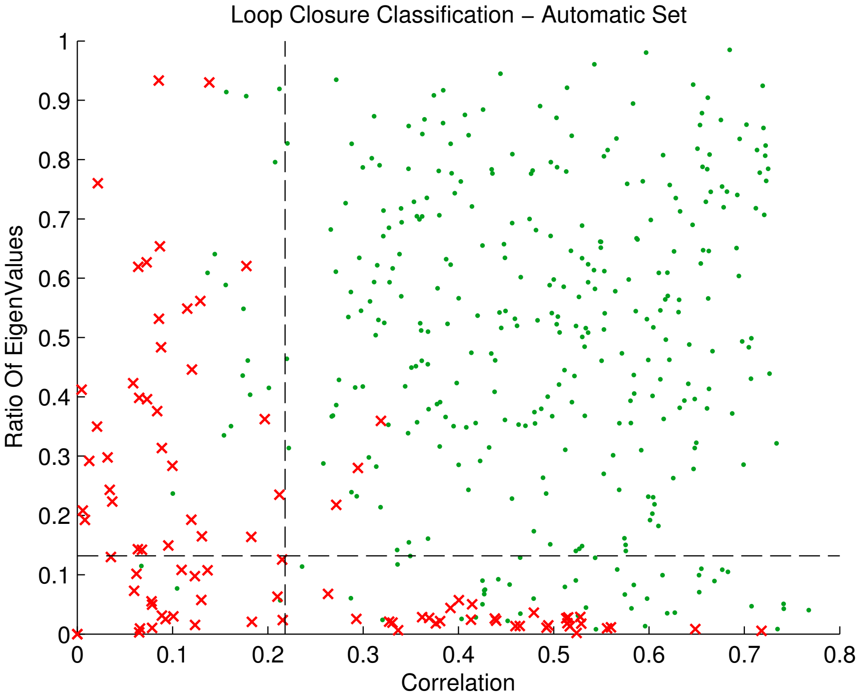
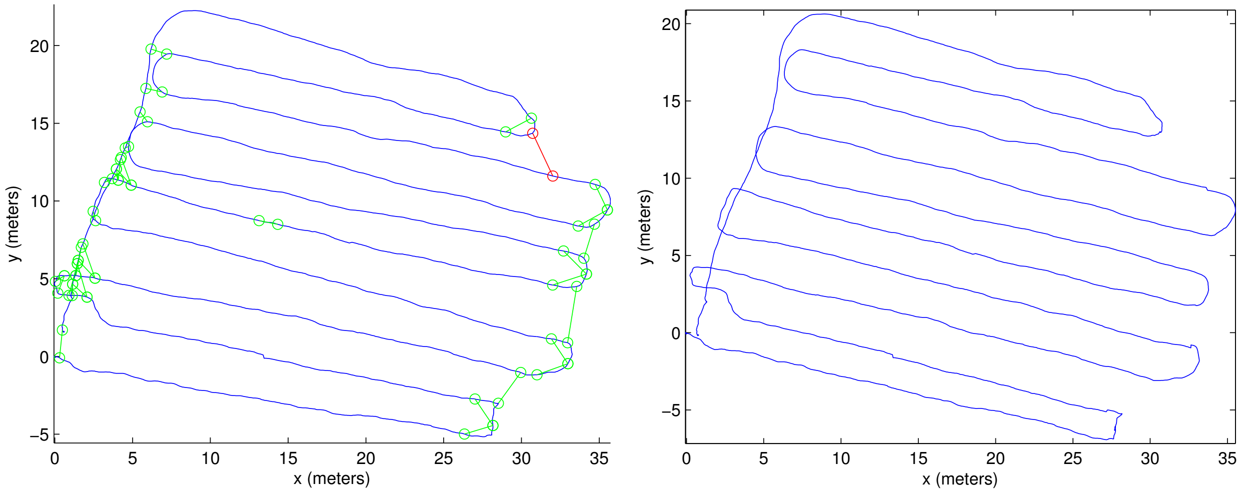


Figure 13: Scatter plot showing the distribution for the automatically detected loop closure candidates. A red × corresponds to a failed loop closure transformation, while a green ○ represents a successful candidate pair. The thresholds used are denoted with a dashed black line.

Figure 13 presents the results of using the proposed classifier on the automatically detected image based loop closures. The distribution of points in the plot supports the use of a threshold based rule. Using the thresholds derived in the previous experiment, a true positive rate of 82.5% and a false positive rate of 3.3% is obtained. This analysis empirically shows that the choice of thresholds is only loosely dependent on the source of loop closure candidates.

Since the previous works of [5] and [9] only report statistics for detection, estimation, and validation combined, a one-to-one comparison is not possible. However, both previous works suggest using only correlation-like techniques to validate loop closure transformations. To demonstrate the performance of a correlation only scheme in indoor environments, we perform the loop closure vetting again using only the correlation metric c of Eq. (9). Figure 12 shows the resulting ROC curves. When the false alarm rate is required to be $\leq 1\%$, we obtain true positive rates of 16.9% and 13.5% for the manual and automatic loop closure sets respectively. This empirically shows that both the correlation metric c and the environment complexity metric r are required for robust operation in indoor environments.



(a)

(b)

Figure 14: Application of the proposed algorithms in an indoor office environment. (a) The dead reckoning path with detected loop closures overlaid. The loop closures that were accepted by the verification metrics are shown using green lines while the loop closures shown in red represent rejected candidates. (b) The resulting path after loop closure has been applied.

4.4 SLAM Examples

To demonstrate the FGSM algorithm and verification criteria in a SLAM context we have performed the following experiments. Data is collected using an ambulatory backpack system equipped with cameras and a LiDAR scanner [10]. Pairwise scan matching of temporally adjacent scans is used to generate a dead reckoning trajectory. Loop closure candidates are then identified and the FGSM algorithm is run on each candidate pair and the proposed verification metrics are computed using the resulting transformations. Using the thresholds derived in the previous section, specifically $r > 0.132$ and $c > 0.218$, the candidates are classified and the erroneous transformations are discarded. The remaining candidates are used to generate an optimized trajectory [2].

The first example, a 330 meter trajectory, was collected from a small retail shopping center. Figure 14(a) shows the 35 manually defined loop closure candidates superimposed on the trajectory estimated by dead reckoning. In this dataset, the dead reckoning trajectory is fairly accurate due to the small scale of the environment. The loop closures that are classified as correct are shown in green while the candidates that are rejected are shown in red. The verification metrics correctly accept 34 loop closures and falsely reject a single candidate. The optimized trajectory, shown in Fig. 14(b), only slightly differs from the dead reckoning result.

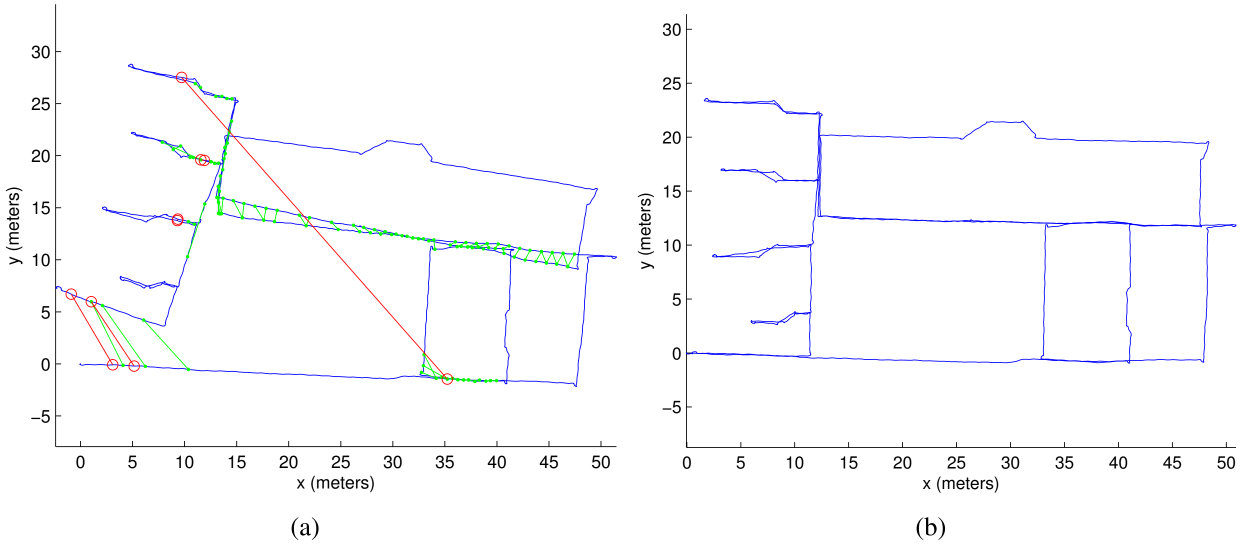


Figure 15: Application of the proposed algorithms in an indoor office environment. (a) The dead reckoning path with detected loop closures overlaid. The loop closures that were accepted by the verification metrics are shown using green lines while the loop closures shown in red represent rejected candidates. (b) The resulting path after loop closure has been applied.

The second example was collected along a 300 meter path from an indoor office environment. Figure 14 shows the result of this experiment. Fig. 15(a) shows 82 loop closure candidates, detected automatically from camera data, overlaid on the dead reckoning trajectory. The path walked during data collection was more complicated than the first example and as a result, the recovered trajectory via dead reckoning has accumulated significant error. The algorithm correctly rejects the one erroneous loop closure, correctly accepts 75 loop closures as correct, and falsely rejects five candidates. Figure 15(b) shows the resulting path after optimization.

The final dataset, a trajectory of 1400 meters, was collected from a warehouse-sized retail store. The data collection lasted a period of 31 minutes and covered an area of approximately 5600 m^2. Figure 16 shows the 107 manually defined loop closure candidates overlaid on the initial dead reckoning trajectory. Because of the large scale of the dataset, the accumulated error has grown to be substantial. The proposed algorithm correctly accepts 66 of the loop closure candidates. Unfortunately, the algorithm was unable to reject 3 erroneous loop closures. Figure 17(a) shows the path after optimization has been applied. Notice that the incorrectly accepted loop closures cause the subsection of the path denoted by a black square to be globally inconsistent. Figure 17(b) shows a magnified view of the erroneous subsection of the path.

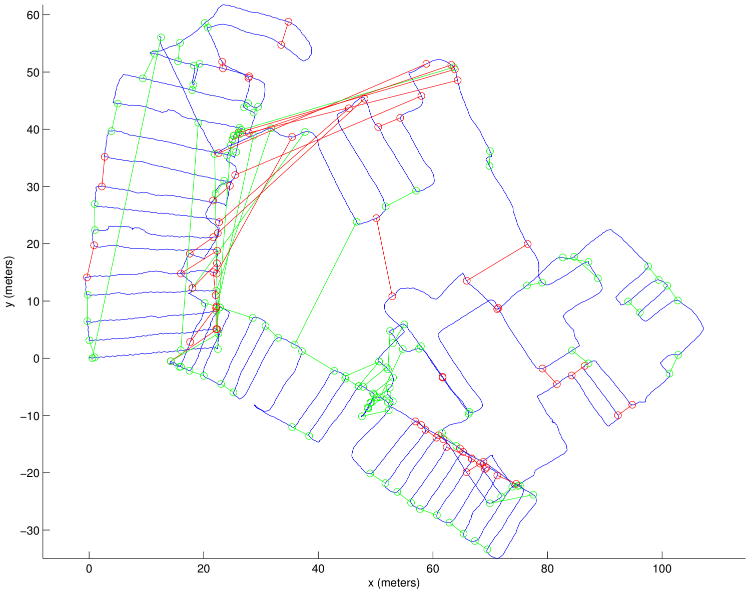


Figure 16: The dead reckoning trajectory for the warehouse-sized retail store. The operators estimated path is shown in blue. The loop closures that were accepted by the verification metrics are shown using green lines while the loop closures shown in red represent rejected candidates.

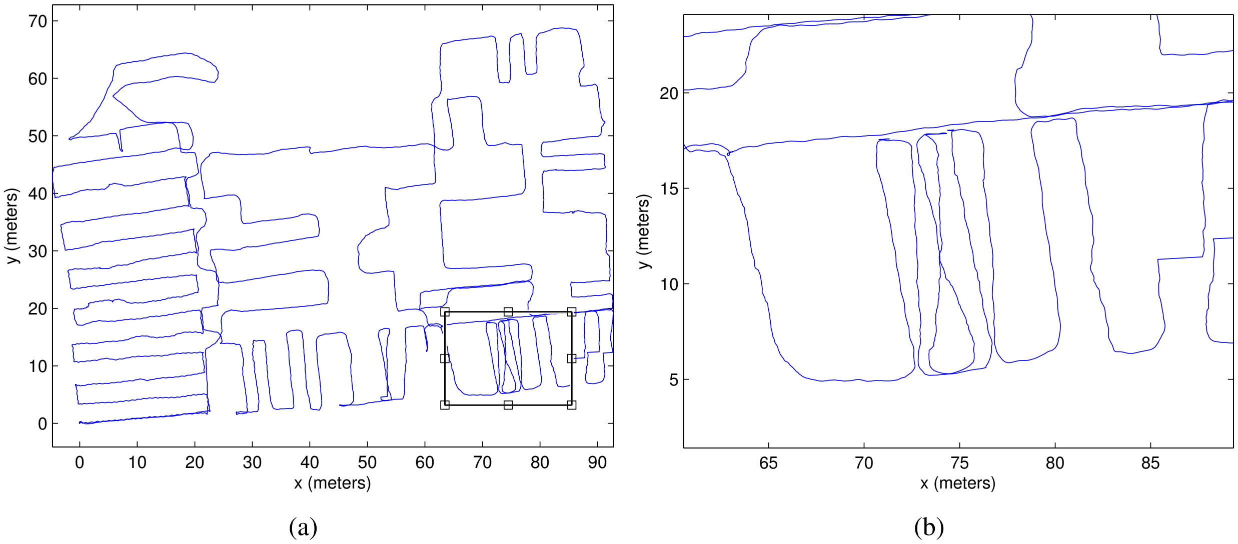


Figure 17: The optimized trajectory for the warehouse-sized dataset after the proposed algorithm has been applied. (a) The optimized trajectory. (b) A magnified view of the erroneous section, denoted in (a) by a black square, caused by incorrectly accepted loop closures.

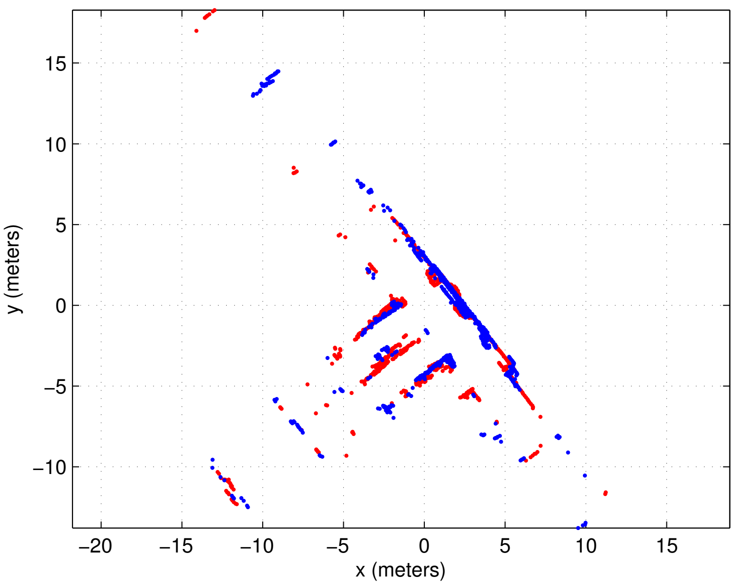


Figure 18: Erroneously accepted loop closure. The proposed metrics are unable to discern between correct and erroneous transformations in repeated complex geometric environments.

The previous experiment highlights a shortcoming of the proposed algorithm. Figure 18 shows one of the loop closures that from the warehouse-sized retail store that was falsely accepted. The algorithm attempts to match the wrong aisles of the retail store and, as a result, the translation was estimated incorrectly. This type of ill-conditioned geometry is not modeled by the verification metrics. As a result, when a large amount of the overlapping geometry is repeating and complex, the metrics are unable to classify the transformation as erroneous and it is falsely validated.

5 CONCLUSION AND FUTURE WORK

In this thesis we presented and evaluated two algorithms related to loop closure estimation and verification. Through empirical data, we have shown the FGSM algorithm to provide robust transformation estimation even for poor initial conditions. Secondly, we presented two metrics that characterized both the scene complexity and the geometric alignment. Using a simple thresholding scheme, the two metrics are able to robustly reject bad loop closures 99% of the time while still accepting greater than 84% of true positives. The algorithms were tested in an end-to-end and empirically shown to provide robust loop closure transformation estimation and verification in most scenarios.

Future work is needed to address the remaining algorithmic issues. When large, repeating, complex geometry is present scan matching is an ill-conditioned problem and the transformation parameters can be incorrectly computed. Furthermore, this type of error is not modeled by the verification metrics and, as a result, they are unable to correctly reject poor matches.

In addition to algorithmic issues, further investigation is needed to generalize the proposed algorithms to different data sources. Scanning systems, such as time of flight or structured light cameras, produce orders of magnitude more points per reading and thus algorithms such as the FGSM may become computationally unattractive.

References

[1] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficent fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," *IEEE International Conference on Intelligent Robots and System*, pp. 206–211, 2003.

[2] G. Grisetti, C. Stachniss, and W. Burgard, "Non-linear constraint network optimization for efficient map learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, pp. 428–439, 2009.

[3] F. Dellaert and M. Kaess, "Square root sam: Simultaneous location and mapping via square root information smoothing," *International Journal of Robotics Research*, 2006.

[4] P. Newman and M. Cummins, "Fab-map: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27(6), p. 647665, 2008.

[5] K. Granstrom, J. Callmer, J. I. Nieto, and F. T. Ramos, "Learning to detect loop closure from range data," *IEEE International Conference on Robotics and Automation*, pp. 15–22, 2009.

[6] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," *IEEE International Conference on Intelligent Robots and System*, pp. 3491–3496, 2007.

[7] P. Besl and N. McKay, "A method for registration of 3d shapes," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[8] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained mav," *IEEE International Conference on Robotics and Automation*, pp. 20–25, 2011.

[9] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based slam," *The International Journal of Robotics Research*, vol. 27, pp. 667–690, 2008.

[10] G. Chen, J. Kua, N. Naikal, M. Carlberg, and A. Zakhor, "Indoor localization algorithms for a human-operated backpack system," *3D Data Processing, Visualization, and Transmission*, 2010.

[11] P. Checchin, F. Gerossier, C. Blanc, R. Chapuis, and L. Trassoudaine, “Radar scan matching slam using the fourier-mellin transform,” *Springer Tracts in Advanced Robotics, Field and Service Robotics*, vol. 62, pp. 151–161, 2010.

[12] J. L. Martinez, J. Gonzalez, J. Morales, A. Mandow, and A. J. Garcia-Cerezo., “Mobile robot motion estimation by 2d scan matching with genetic and iterative closest point algorithms,” *Journal of Field Robotics*, vol. 23(1), pp. 21–34, 2006.

[13] K. Lenac, E. Mumolo, and M. Nolich, “Robust and accurate genetic scan matching algorithm for robotic navigation,” *International Conference on Intelligent Robotics and Applications*, pp. 584–593, 2011.

[14] J. M. Phillips, R. Liu, and C. Tomasi, “Outlier robust icp for minimizing fractional rmsd,” *International Conference on 3-D Digital Imaging and Modeling*, 2007.

[15] M. Bosse and J. Roberts, “Histogram matching and global initialization for laser-only slam in large unstructured environements,” *IEEE International Conference on Robotics and Automation*, 2007.

[16] A. Censi, “An icp variant using a point-to-line metric,” *IEEE International Conference on Robotics and Automation*, pp. 19–25, 2008.

[17] J. Kua, N. Corso, and A. Zakhor, “Automatic loop closure detection using multiple cameras for 3d indoor localization,” *IS&T/SPIE Electronic Imaging*, 2012.

[18] G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling.” *IEEE International Conference on Intelligent Robots and Systems*, pp. 2432–2437, 2005.

[19] T. Liu, M. Carlberg, G. Chen, J. Chen, J. Kua, and A. Zakhor, “Indoor localization and visualization using a human-operated backpack system,” *International Conference on Indoor Positioning and Indoor Navigation*, 2010.

[20] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms, concepts and designs.* UK: Springer, 1999.

[21] A. Barla, F. Odone, and A. Verri, “Histogram intersection kernel for image classification,” *International Conference on Image Processing*, vol. 3, pp. 513–516, 2003.