Remote Sens. **2013**, *5*, 6611-6646; doi:10.3390/rs5126611

OPEN ACCESS

Remote Sensing

ISSN 2072-4292

www.mdpi.com/journal/remotesensing

Article

Indoor Localization Algorithms for an Ambulatory Human Operated 3D Mobile Mapping System

Nicholas Corso and Avideh Zakhor

Video and Image Processing Lab, University of California, Berkeley, CA 94720, USA;
E-Mails: ncorso@eecs.berkeley.edu (N.C.); avz@eecs.berkeley.edu (A.Z.); Tel.: +1-510-642-3214

Received: 18 October 2013; in revised form: 25 November 2013 / Accepted: 28 November 2013 / Published: 3 December 2013

Abstract: Indoor localization and mapping is an important problem with many applications such as emergency response, architectural modeling, and historical preservation. In this paper, we develop an automatic, off-line pipeline for metrically accurate, GPS-denied, indoor 3D mobile mapping using a human-mounted backpack system consisting of a variety of sensors. There are three novel contributions in our proposed mapping approach. First, we present an algorithm which automatically detects loop closure constraints from an occupancy grid map. In doing so, we ensure that constraints are detected only in locations that are well conditioned for scan matching. Secondly, we address the problem of scan matching with poor initial condition by presenting an outlier-resistant, genetic scan matching algorithm that accurately matches scans despite a poor initial condition. Third, we present two metrics based on the amount and complexity of overlapping geometry in order to vet the estimated loop closure constraints. By doing so, we automatically prevent erroneous loop closures from degrading the accuracy of the reconstructed trajectory. The proposed algorithms are experimentally verified using both controlled and real-world data. The end-to-end system performance is evaluated using 100 surveyed control points in an office environment and obtains a mean accuracy of 10 cm. Experimental results are also shown on three additional datasets from real world environments including a 1500 meter trajectory in a warehouse sized retail shopping center.

Keywords: backpack; mobile mapping; SLAM; indoor localization

1. Introduction

In recent years, there has been great interest in the modeling and analysis of interior building structures using laser range finders (LRFs). Utilizing many range measurements and the location of the sensor during each measurement, a dense view of the environment can be reconstructed. Applications for the reconstructed point clouds, such as energy efficiency simulation, augmented reality, entertainment, architectural modeling, and emergency response, yield significant motivation to increase the accuracy of the modeling process and decrease the time to acquire the data.

For applications where dense, millimeter level accuracy is needed, the standard practice is to use a static scanning approach. During static scanning, a 3D scanning station is placed on a tripod and a small section of the environment is captured with high detail. The tripod is then moved around and the process is repeated until the entire area has been captured. The small 3D point clouds can then be stitched together to build a single, unified representation of the environment. Stitching is typically achieved either by placing small markers throughout the environment or a combination of manual intervention and point matching techniques. While this process is accurate and reliable, it is both slow and invasive.

To improve acquisition efficiency, the scanning equipment is often mounted on a mobile platform such as a wheeled platform or human operator. Mobile mapping is generally faster than static scanning, but reconstructing the point cloud from the captured range measurements can be more complex [1,2]. In contrast to static scanning approaches, it is typically not feasible to utilize markers or manual alignment to stitch sensor readings and thus, mobile mapping systems must track the system's position during data collection. In environments where GPS is available, high-end GPS/INS systems are provide accurate tracking of the mobile platform [3]. However, in environments such as building interiors or "urban canyons" GPS information is not reliable due to poor signal strength or multi-path interference. In these environments, mobile mapping systems typically rely on simultaneous localization and mapping (SLAM) algorithms for positioning. Common to all SLAM algorithms, the mobile mapping system must both create a map of the environment while simultaneously tracking its position within that map.

GPS-denied mobile mapping has been an active area of research for many years. The system presented by Holenstein *et al.* [4] consists of a 3D scanner constructed from a rotating 2D LRF. It is mounted on a wheeled platform and has to be manually pushed through the environment. The system accurately maps cave-like environments while producing a water-tight mesh of the observed space. In order to overcome this challenge of scanning while simultaneously moving, the authors use a novel algorithm that allows the scanner to be in motion while the scanner is rotating [5]. Called "sweep matching," the algorithm considers entire periods of the nodding scanner's motion as a single scan and attempts to recover the motion that takes place between successive sweeps while simultaneously correcting the sweep's distortion caused by the continuous motion of the system. The main drawback of this approach is that a wheeled platform is unable to easily navigate uneven terrain.

In order to handle complex indoor environments, such as staircases, human-mounted systems have shown significant promise [6,7]. The authors Hesch *et al.* rigidly mounted a single 2D scanner and an inertial measurement unit (IMU) on to a human operator [6]. Intended for visually impaired operators, the system aims to provide real-time tracking through complex 3D indoor environments. The main drawback of this approach is that it assumes all vertical walls to be orthogonal, to one another.

This assumption is rather restrictive and does not allow for the system to be utilized in many indoor environments. Using the same sensors, the authors Bosse *et al.* mount an IMU and LRF are mounted on the end of a passive spring [7]. Using the natural motion of the human operator, the spring system oscillates and sweep-like scans are collected. Once the sweep matching algorithm of Bosse *et al.* is applied, the scans are used to build a model of the environment. By adding a rigidly mounted camera to the system it was extended to provide rich, colorized point clouds of the environment [8].

Another human-mounted mapping system combines a 2D LRF, an IMU, a barometer, and an RGBD camera to provide real-time mapping for rapid response missions [9]. By utilizing the IMU and barometer information, the system is able to produce a collection 2D maps from a multistory building by detecting floor transitions. Although the system is lightweight and relatively inexpensive, the limited field of view of the RGBD camera and orientation of the LRF do not allow for rapid capture of the full detail of a complex 3D environment.

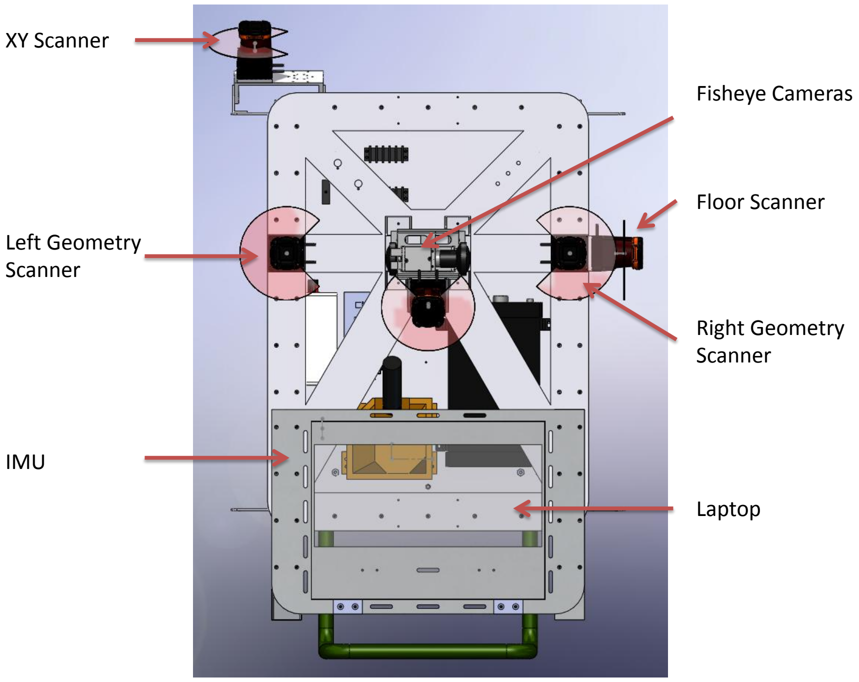
Foot mounted systems have also been considered for indoor navigation [10–12]. In [10], a single IMU is mounted to a shoe and used to localize the operator without any additional external sensors. The system utilizes the constraint that the shoe is stationary when it is in contact with the ground to estimate sensor bias and reduce accumulated drift. Other systems, including the works of Angermann *et al.* [11] and Moafipoor *et al.* [12], extend the idea by incorporating a barometer, magnetometer, and GPS sensor information to obtain an accurate position estimate that is globally referenced by occasional GPS fixes. While these types of systems are lightweight and inexpensive, the complex motion of the human gait makes it difficult to fuse the position estimate with data from sensors not mounted to the operator's shoe.

Human-mounted mobile mapping systems have also been successfully applied to outdoor environments [13,14]. Fusing the data from a GPS unit, cameras, and LRFs, the authors Bok *et al.* demonstrated large scale mapping of outdoor cultural heritage sites [13]. By relying GPS information to limit accumulated errors, the authors' proposed system is able to accurately map trajectories of many kilometers with little to no manual intervention. The authors note however that when GPS is unavailable or unreliable, the system must fall back to a combination of manual intervention and point cloud matching techniques to limit accumulated error. Due to this constraint, this approach may not be suitable for automatic mobile mapping of large scale, GPS-denied environments.

Unmanned Aerial Vehicles (UAVs) have also been employed for mobile mapping applications [15–17]. In [15], Shen *et al.* present a quadrotor system that is equipped with a 2D LRF and is used to build a collection of 2D maps representing the environment. To build full 3D point clouds, both Bachrach *et al.* [16] and Shen *et al.* [17] present UAV systems equipped with an RGBD camera for indoor mappings. While these systems are able to autonomously explore complex terrain, short battery life ultimately limits the size of the area that can be mapped.

Our previous works on mobile mapping include mapping 3D indoor environments using the ambulatory backpack system [18–20] shown in Figure 1. The system in Figure 1, is equipped with 5 LRFs, an IMU, and 2 fish-eye cameras. When worn upright, the XY scanner is approximately level and when combined with the IMU readings, it is possible to recover the 2D trajectory of the system [18]. The floor scanner is pointed downward, parallel to the direction of travel, and is used to detect planar floor environments and to recover the height of the system. Although we enumerate all sensors on the system for completeness, only the IMU and 2 LRFs are used for localizing the system. The remaining three LRFs and cameras are used for other applications such as generating a dense, colorized 3D point cloud or building a watertight, texture mapped 3D model.

Figure 1. A CAD model of the ambulatory human-mounted backpack system used for data collection.

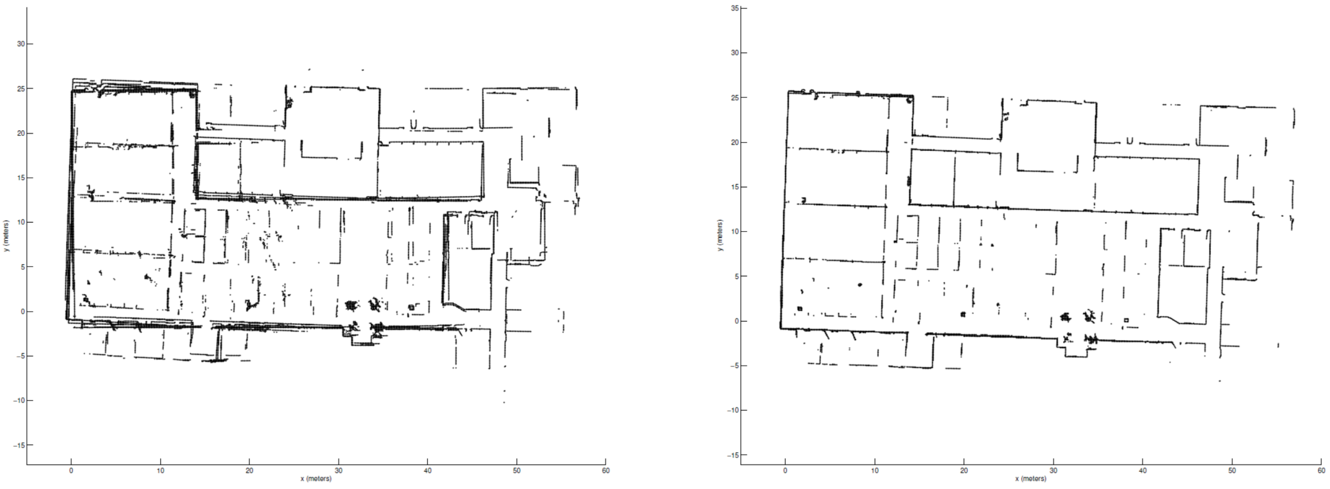


By assuming that the environment consists of vertical planes, we solved the 3D localization problem of the system in Figure 1 by first breaking down into a 2D scan matching problem followed by a 1D height estimation problem [18]. Due to this constraint, the system is inherently limited to man-made environments with vertical walls where the 2.5D assumption holds true. During acquisition, the operator was required to return to the starting position and orientation in order to enforce a "begin to end" constraint so as to mitigate accumulated errors. The approach in [18] does not scale well to large environments in that the accumulated error is often too high to overcome with the single start-to-end constraint. Figure 2 shows an example of using only the start-to-end constraint for a 350 meter long trajectory. As seen in Figure 2a, despite the single constraint on the trajectory, many misalignments remain. Figure 2b shows an accurate map created by our proposed method for comparison.

Using sequential images from the cameras in Figure 1, we previously presented a method for fusing camera imagery and laser data into a single trajectory [19]. By comparing features of interest in temporally adjacent images, additional constraints on the trajectory were derived. While this method improves visual alignment in the texture mapped models, the algorithm fails to scale to large datasets. Since the constraints are derived from temporally adjacent images, they suffer from the same accumulated drift and geometric misalignments for large environments.

In [20] we presented algorithms which addressed the geometric alignment of the system in Figure 1. We detect when the operator revisits a previously visited location by applying the FAB-MAP algorithm [21] to the optical imagery captured by the system. Known as a loop closure, this type of constraint is used to enforce geometric consistency in the reconstructed map. Despite the ability to automatically detect loop closure locations, this method is not able to robustly compute the metric transformation between image locations. For large trajectories, the dead reckoning trajectory contains significant drift and is unsuitable as an initial condition for scan matching between temporally distant sensor readings. Furthermore, since the loop closures are detected in the image domain, loop closures were often detected in locations that were ill-conditioned for scan matching, such as long narrow hallways. Inaccurate or misidentified loop closure constraints cause misalignments in the reconstructed geometry and require tedious manual intervention to correct.

Figure 2. (**a**) An example map reconstructed using only dead reckoning and the start-to-end constraint. Many inaccuracies are visible in the reconstructed geometry. (**b**) An accurate map of the same environment created using the proposed method.



(**a**) (**b**)

In order to address the shortcomings of [18–20], we present a shift in paradigm from our previous works. Previously we relied on appearance based loop closure detection algorithms to reduce accumulated errors in the dead reckoning trajectory. Instead, we first use a Rao-Blackwellized particle filter to obtain a coarse grid map of the environment so that we can robustly detect loop closures. The detected loop closures are then fused with the dead reckoning trajectory to alleviate the spatial and temporal quantization effects inherent to grid mapping. To this end, we propose three novel contributions related to the system shown in Figure 1.

First, rather than relying on loop closure constraints detected from optical imagery, we derive them from an occupancy grid map created via Rao-Blackwellized particle filtering. Previous methods for loop closure detection, such as those relying on keypoints [22], features learned from a machine learning framework [23], or correlative map matching [24] all aim to discover loop closures constraints between arbitrary locations in the environment. Although these methods have successfully been applied to indoor environments, they often result in erroneous detections when repeated patterns appear in the environment. In indoor environments repeated structures exist in both the optical imagery, such as a repeated wallpaper pattern, or in the LRF data. Furthermore, because such methods do not incorporate prior geometric information, they are unable to prune false detections using line-of-sight or other geometric constraints. To address these issues, we propose to utilize the information contained in the occupancy grid map as a geometric prior and only detect loop closures in locations that are suitable for scan matching. By detecting loop closures based on a grid map representation of the environment, we leverage the information contained in the occupancy grid map and ensure that constraints are detected only in locations that are well conditioned for scan matching.

Our second contribution is our proposed outlier-resistant, genetic scan matching algorithm that accurately matches scans despite a poor initial condition. Previous genetic scan matching algorithms, such as the Hybrid Genetic Scan Matcher [25], propose to use a genetic search using a correlation based metric followed by a round of ICP to refine the solution. We propose two extensions to this algorithm. First, as our system's LRFs often scan the ceiling, a large number of outliers can be present in the scan data and therefore a correlation based metric is not appropriate. To that end, we use the fractional root mean square distance metric [26] to provide robustness against outlier points. Secondly, we apply scan matching to force each chromosome into a local minima of the objective function. Under this formulation, the genetic scan matching algorithm efficiently searches only the local minima of the objective function.

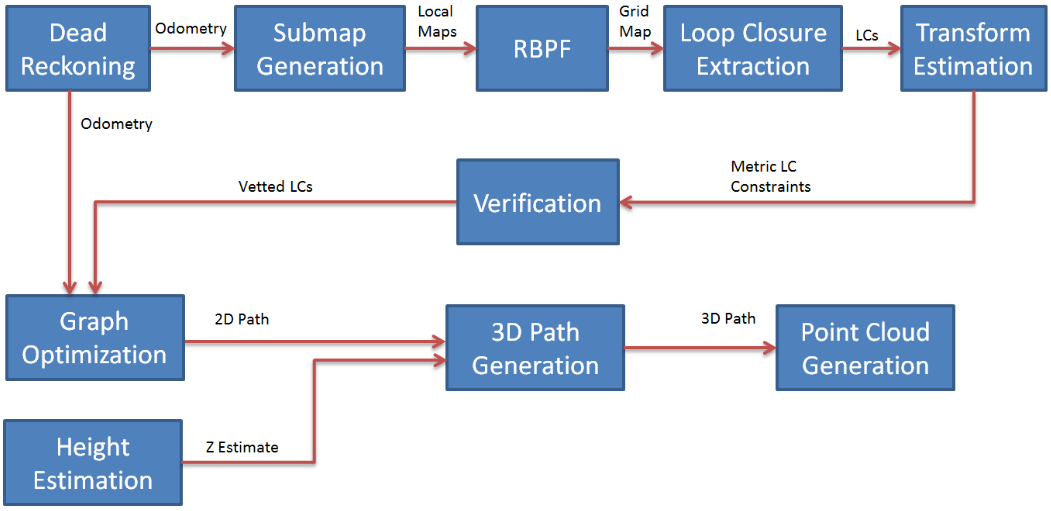
Lastly, we present two metrics based on the amount and complexity of overlapping geometry in order to vet the estimated loop closure constraints. Indoor environments contain many locations, such as long narrow hallways, where scan matching is ill-conditioned. By examining the quantity and complexity of overlapping geometry we automatically prevent erroneous loop closures from degrading the accuracy of the reconstructed trajectory.

Figure 3 shows a block diagram of our proposed off-line algorithmic pipeline for the human-mounted backpack system of Figure 1. Unlike wheeled systems, an ambulatory system is unable to wheel encoders for dead reckoning and thus we first use the XY scanner and IMU to generate odometry measurements. Assuming that the environment is composed of vertically oriented planes, we are able to undistort scan distortion that arises from the pitch and roll motion of the operators natural gait. Furthermore, due to the ambulatory nature of the system, outlier points are often detected when the LRF scans the ceiling or ground planes. We overcome this problem by applying a scan matching algorithm which explicitly models the presence of outliers. By segmenting the scan into a set of inlier points, we automatically detect points which violate the vertical wall assumption without a priori knowledge of the outlier distribution. The scan matching odometry results are then concatenated to form a dead reckoning trajectory.

We then use the dead reckoning trajectory to aggregate temporally adjacent LRF readings into local submaps. Since the XY scanner is subjected to pitch and roll motion, it contains many points, such as ceiling or ground strikes, which do not fit the vertical wall assumption. These points must be eliminated before any 2D SLAM approach can be effectively utilized. We accomplish this by employing a Rao-Blackwellized particle filter to merge sequential scans into geometrically consistent local submaps while simultaneously eliminating points that do not fit the vertical wall assumption. Then we fuse the submaps into a single topographically correct, occupancy grid map by applying a separate particle filtering to the generated submaps.

In contrast to [20], we detect loop closure constraints using the occupancy grid map representation of the environment. Unlike our previous approaches [18–20], we utilize a geometrically consistent occupancy grid map to only construct loop closures only in regions that are inherently well conditioned for scan matching. Since the particle filter also computes a rough estimation of the position for each submap, we are able to derive an initial condition for the loop closure constraints and compute the metric transformation between locations by applying a genetic scan matching algorithm. We then vet the detected loop closure constraints using a combination of metrics that quantify the amount and complexity of overlapping geometry. This allows us to both detect and vet loop closure constraints without any tedious manual intervention.

Figure 3. Block diagram of the algorithms used for localizing an ambulatory backpack system.



We then fuse the dead reckoning trajectory with the verified loop closure constraints via graph optimization. Specifically, we form an edge directed graph using the odometry to create pairwise connections between temporally adjacent nodes. We then insert the vetted loop closure constraints between the detected loop closure locations. We apply a graph optimization procedure such as TORO [27] or SAM [28] to generate a single 2D optimized trajectory.

The 2D mapping results are extended to a full 6DOF pose by combining the pitch and roll from the IMU with a height estimate at each locations. We use the adaptive height estimator of Kua *et al.* [20] directly to obtain an estimate of the operator's height. We concatenate the 2D location and height to obtain the 3D position and combine pitch, roll, and heading estimates to recover the orientation. The 6DOF path is then combined with the cameras' and geometric scanners' data to produce a dense, colorized 3D point cloud.

The rest of the paper is organized as follows: Section 2 describes algorithms used in computing the 3D trajectory of the system. Experimental results for the proposed algorithms are included in Section 3. Finally, applications for the data produced by our approach are presented in Section 4 and conclusions are drawn in Section 5.

2. Localization Algorithms

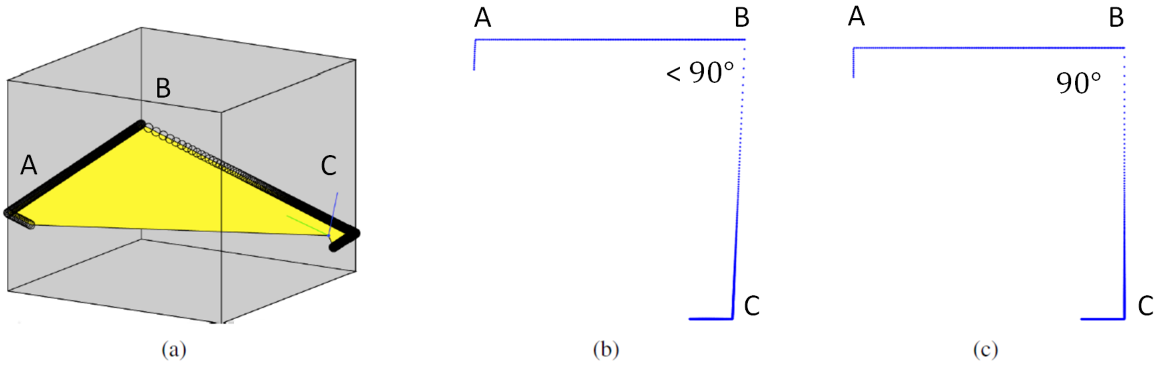
This section provides a detailed description of the off-line algorithmic pipeline shown in Figure 3 which is used for localizing the human-mounted, ambulatory backpack system. Section 2.1 reviews how we derive odometry measurements using the XY scanner and IMU. Sections 2.2 and 2.3 describe how the individual scans are combined into a single geometrically consistent occupancy grid map using a Rao-Blackwellized particle filter. We then present the proposed loop closure detection and transformation estimation algorithms in Section 2.4 and Section 2.5. The loop closures are then vetted using the proposed metrics in Section 2.6 to prevent any erroneous constraints before being combined via graph optimization. The height estimation and 3D path generation are briefly reviewed in Sections 2.7 and 2.8. Finally, the point cloud generation is summarized in Section 2.9.

2.1. 2D Dead Reckoning

Unlike wheeled systems where wheel encoders provide dead reckoning, human-mounted mobile mapping systems, such as the one shown in Figure 1, have to derive odometry from other sources such as scan matching or IMU measurements [5,6,9,18]. We utilize scan matching to align temporally adjacent sensor readings from the XY scanner to estimate incremental motion. The incremental motion is then concatenated to produce a dead reckoning trajectory. Since the system is carried by a human operator, the XY scanner is not always perfectly level. Specifically, pitch and roll introduced by the operator's gait causes significant warping of the sensor readings and direct scan matching leads to large errors in the reconstructed trajectory.

Assuming that the walls are perfectly vertical, we project the scans along the direction of gravity to undo the warping in the XY scanner's data. By projecting the scan points along the direction of the gravity vector, we correct for the warping introduced by non-zero pitch and roll. Figure 4 shows an an example of this procedure. Figure 4a shows a simulated scan inside a box-like environment. The raw sensor readings, shown in Figure 4b, contain warping caused by the non-zero pitch and roll resulting in the angle between lines to be less than 90°. Figure 4c shows the result of scan projection which results in the lines' angles of intersection to be 90°.

Figure 4. An example of scan projection. (**a**) An off-axis scanner in a cubic environment. (**b**) The raw readings of the laser scanner. Note that the angle ABC is less than 90° due to scan warping. (**c**) After projection the angle ABC has been corrected to 90°.



Assuming that the environment remains static between scans, we use successive readings from the LRFs to estimate the incremental motion between scans. Many approaches have been suggested to solve the scan matching problem including global approaches [29] and iterative approaches [24,30,31].

Of particular interest is a class of iterative algorithms known as the Iterative Closest Point algorithms, or ICP [32].

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. Since our system is equipped with 2D LRFs, we are concerned only with the specific case of 2D point matching.

Variants of the metric of Equation (1) have been suggested for improving accuracy and robustness to outliers. The point-to-line metric of [30] uses point-to-surface matching to reduce inaccuracies caused by the LRF sampling the surfaces at different locations. Additionally, because an ambulatory backpack system undergoes significant roll and pitch motion due to the operators natural gait, a large number of outliers may be present from dynamic objects or ceiling strikes. Proposed originally for arbitrary point cloud matching, the fractional iterative closest point (FICP) algorithm introduces a metric known as the fractional root mean square distance (FRMSD). The optimal transform parameters $T(\cdot, \mu)$ and inlier set $D_f \subseteq Q$ are obtained by iterating the following steps [26]:

- Given an initial transform $T(\cdot, \mu_0)$, points in Q are matched to their nearest neighbors in set P.
- Assuming a fixed $T(\cdot, \mu_0)$, an optimal set of inlier points D_f is identified.
- Using inlier set D_f, the transform parameters μ are recovered using a first-order Taylor expansion and solving for the optimal linear estimate of μ [24].

Although the vertical wall assumption corrects for the natural gait of the human operator, scan points that originate from the ceiling, floor, or dynamic objects in the environment are not well modeled by a vertical plane and must be handled separately. To this end, we use the the point-to-line and fractional metrics in the following objective function:

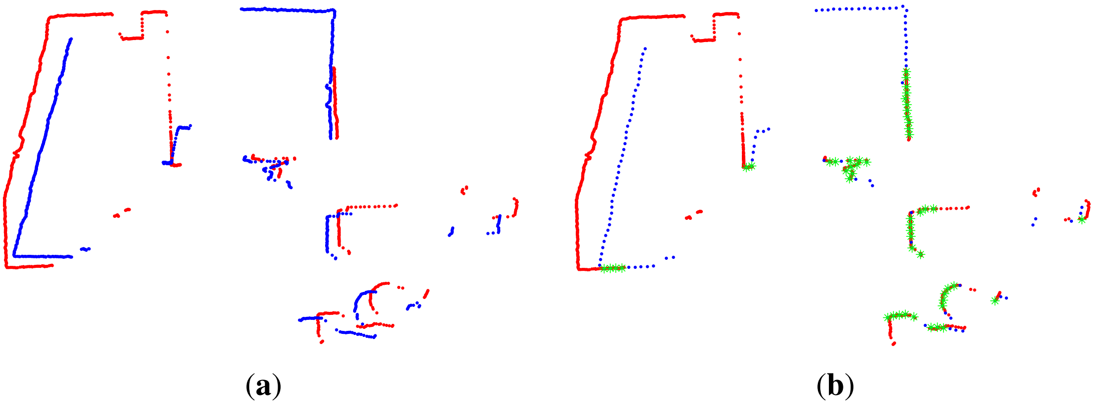
$$e = \frac{1}{f^{\lambda}} \sqrt{\frac{1}{|D_f|} \sum_{\mathbf{q}_i \in D_f} \|\mathbf{n}_i^T(\mathbf{p}_i - T(\mathbf{q}_i, \mu))\|^2} \tag{2}$$

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 objective function is then minimized using the FICP framework [26].

By solving for both the set of inliers and the optimal transformation parameters, the metric in Equation (2) identifies outliers in the data without any prior knowledge of their distribution. The process of iteratively segmenting geometry and recovering the transformation allows for accurate recovery of incremental motion even in the presence of a large number of outlier points.

Figure 5 shows an example of our scan matching algorithm. Figure 5a depicts an example pair of LiDAR scans, shown in red and blue, aligned using an initial estimate of the transformation based upon a priori information. Figure 5b depicts the alignment of scan pair from 5a after the FICP algorithm has been performed. The portion of the geometry that has been found to be part of the inlier set D_f is shown in green. By segmenting the inlier points, we detect and ignore points that do not meet the vertical wall assumption.

Figure 5. An example of performing the FICP algorithm. (**a**) The individual scans, depicted in red and blue, aligned using the initial estimate of the transformation. (**b**) The scan pairs after the FICP algorithm has been performed. The portion of geometry declared as the inlier points are denoted in green.



We take the incremental changes in position and integrate them to recover the dead reckoning trajectory. Since the path is built recursively, any errors in the transformations are compounded and lead to drift in the reconstructed trajectory. Inaccuracies in the recovered path lead to a geometrically inconsistent map and thus are not suitable for most mapping applications. In order to overcome the accumulated drift in the dead reckoning trajectory, we reformulate the problem to obtain a solution that optimizes both the path and the environment simultaneously. Classical solutions such as those involving Kalman filters, particle filters, and graph based approaches are discussed in [33].

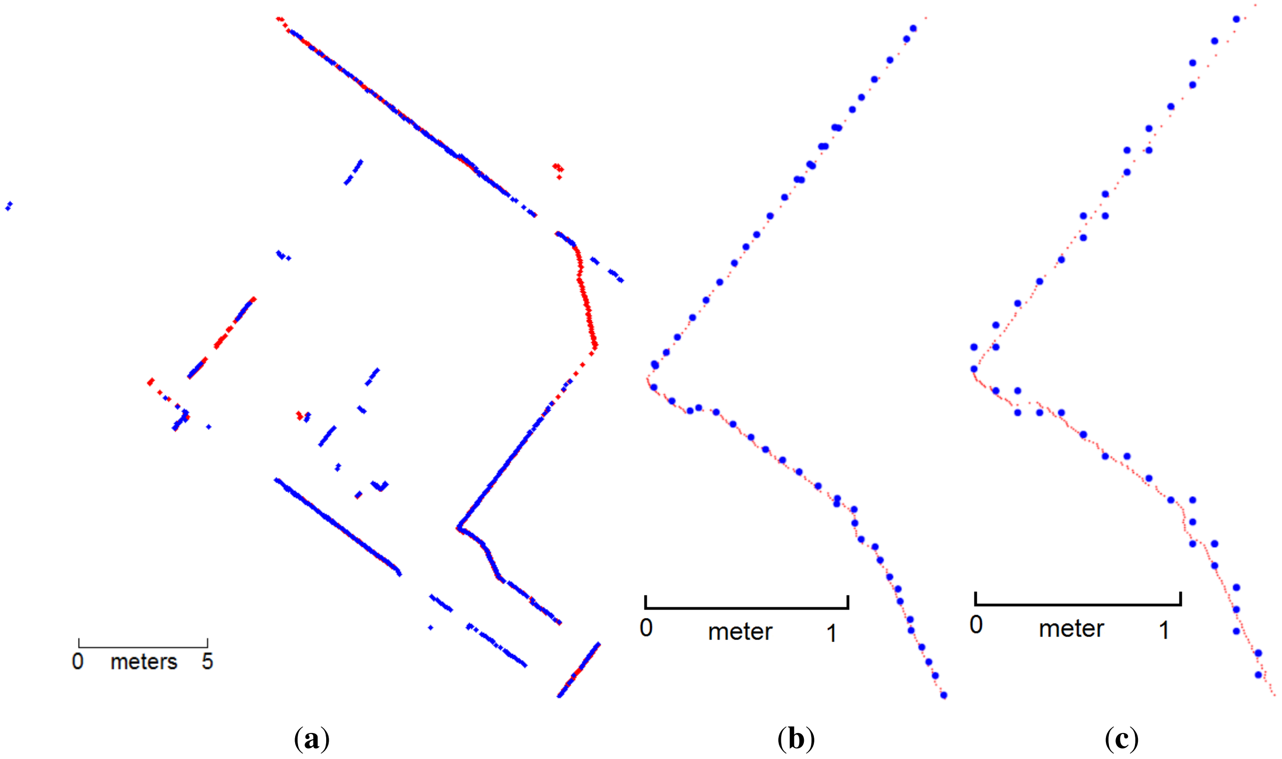
2.2. Submap Generation

Since our system is mounted on a human operator, the sensor readings can contain a large number of outlier points due to clutter, ceiling strikes, or dynamic objects in the environment. In order to apply classical 2D SLAM algorithms, we must first eliminate the outlier points that result from the roll and pitch motion of the system. We aggregate temporally adjacent scanner readings into local submaps to eliminate outlier points and enhance the sensor's field of view before applying the standard 2D Rao-Blackwellized particle filtering algorithm as described in Section 2.3. This section discusses the submap generation procedure.

In order to fuse sequential scans into a geometrically consistent local submap, we use a particle filter that only merges scans from a small, temporally close, region of the environment. Although typically formalized for large scale mapping, we utilize the RBPF to generate accurate maps of subsections of the environment [34]. The approach we take here follows the classical grid mapping framework with three distinctions. First, we use the dead reckoning results of Section 2.1 as the odometry and consider only the inlier points of the scan matching result when merging points into the map or computing the weighting factor. By considering only the inlier points, we ignore any outlier points that originated from dynamic objects in the environment. Secondly, rather than using a strict discretization to create the grid map, we compute the average position of all points that lie in a grid cell. By using the average position as the representative sample for each grid, quantization errors can be mitigated. Lastly, we only consider the temporally closest observations when creating each submap as we are only interested in creating a map for a small subsection of the environment.

Figure 6 show typical results of RBPF based submapping. The red points represent the sensor's original readings while the blue 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 and ceiling points have been removed. Additionally, as seen in Figure 6b, tracking the average point of each grid cell mitigates quantization effects that result from the grid mapping approach. The original corner, denoted by the black square in Figure 6a, matches very closely to the corner built during submap construction. For comparison, the same corner generated with a strict discretization is shown in Figure 6c.

Figure 6. A typical result of RBPF based submapping. The original sensor's readings are shown in red while the resulting submap is shown in blue. (**a**) The amount of visible geometry has been expanded beyond the original sensor's readings and ceiling points have been removed. (**b**) A close up of the section of the submap denoted by the black square shown in (a) shows how spatial averaging mitigates some quantization errors. (**c**) The same corner using a strict discretization.



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. We have empirically determined values of $N = 2$ and $M = 30°$ to work well for our experiments.

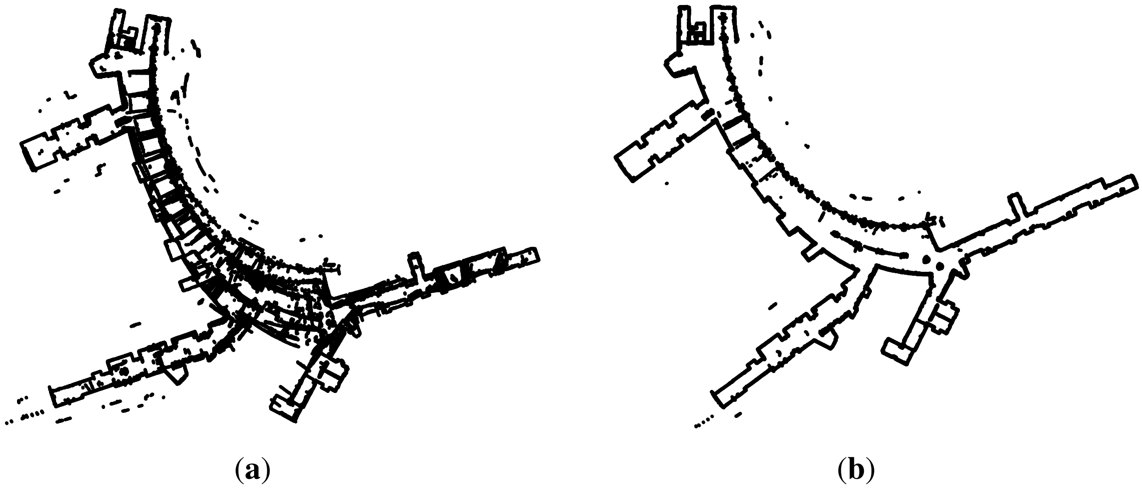
2.3. Rao-Blackwellized Particle Filtering

In this section we discuss the process for fusing the local submaps into a single geometrically consistent occupancy grid map.

Once the submaps have been created, we combine them into a single geometrically consistent occupancy grid map using another round of particle filtering. This time, we match sequential submaps to obtain new odometry estimates. We then fuse the submaps by utilizing the Sampling Importance (SIR) filter and adaptive proposal distribution [35] with one main distinction. Our submaps are not directly generated from a LRF and thus this model is not appropriate. Rather, we use a simple approach that approximates the importance weight of each particle by using the spatial correlation of the new sensor readings with the particle's current map.

Figure 7 shows an example of the RBPF algorithm. A trajectory of approximately 740 m was traversed over a period of 15 min in a hotel. The map formed by using dead reckoning alone is shown in Figure 7a while the map created by the RBPF is shown in Figure 7b. The occupancy grid map generated by the RBPF is geometrically consistent even when traversing previously visited locations.

Figure 7. An example result of applying the Rao-Blackwellized particle filtering algorithm to the generated submaps. (**a**) The map generated using only odometry. (**b**) The map resulting from the RBPF using 100 particles at a resolution of 10 cm.



While the resulting grid map is geometrically consistent, the accuracy is still fundamentally limited by the size of the grid cells used. Despite tracking the average of the contributions to each grid cell, quantization errors still exist. Furthermore, because temporally adjacent scans are first aggregated into local submaps, the system trajectory resulting from the RBPF contains poses for only the locations of the submaps, not the original sensor reading locations. In order to compute poses for all sensor readings we must interpolate between the locations of the submaps. The interpolation is carried out by formulating a graph optimization problem where the poses are the nodes of the graph, the odometry readings serve as the edges between temporally adjacent poses, and loop closures extracted from the occupancy grid map provide global constraints on the graph. By fusing the full-rate odometry and RBPF localization results into a single full-rate trajectory we obtain a geometrically consistent trajectory with no temporal or spatial quantization.

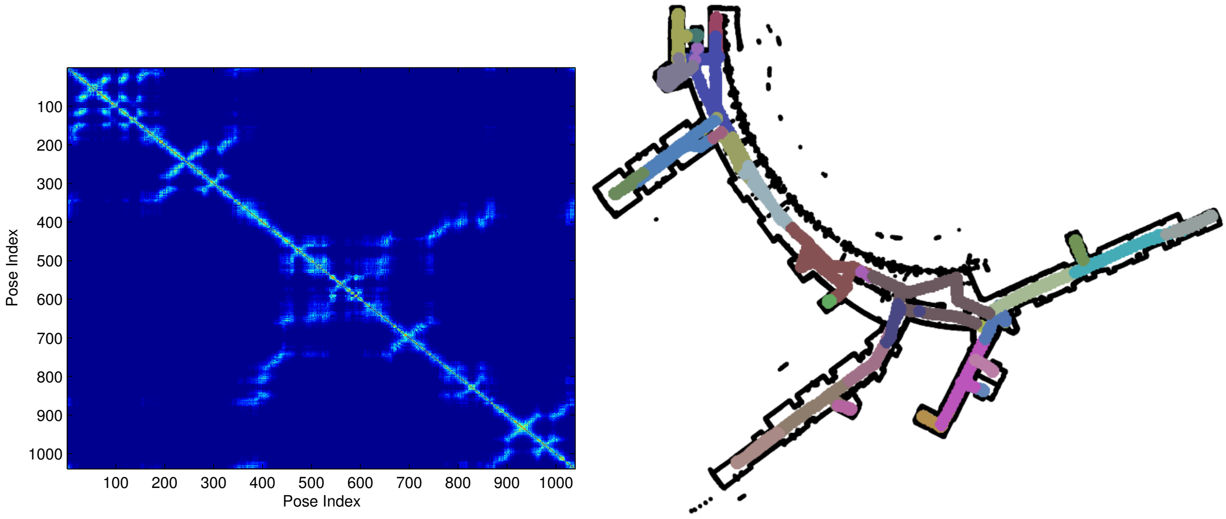
2.4. Loop Closure Extraction

This section describes the proposed methodology for extracting loop closure constraints from the occupancy grid maps created in Section 2.3. To detect loop closures we first convert the map into a representation that defines a measure of similarity between poses in the trajectory. Using both the grid map and accompanying trajectory, we explicitly recover which poses have a point fall in each of the cells from the grid map. We define the correlation $C(i, j)$ between each pose i and j using the following function:

$$C(i, j) = \frac{|Z_i \bigcap Z_j|}{\min(|Z_i|, |Z_j|)} \tag{3}$$

where Z_i and Z_j are the set of occupancy grid cells observed by pose i and j respectively and $|Z|$ is the number of elements in set Z. We repeat this for all pairs of poses and collect the resulting coefficients into a correlation matrix C. Figure 8a shows the correlation matrix formed from the grid map of Figure 7b. Note that regions of high correlation that are located off diagonal correspond to temporally distant pose pairs that contain a large number of overlapping grid cells.

Figure 8. (**a**) Correlation matrix formed from the occupancy grid map of Figure 7b. (**b**) The results of clustering the correlation matrix from (a).



(**a**) (**b**)

Next we apply hierarchical clustering to group the poses into smaller sets that contain similar geometry in order to reduce the search space of possible loop closure locations. The clustering algorithm begins with each pose of the trajectory as its own cluster, and iteratively joins the most similar clusters until either the required number of clusters is obtained or all remaining clusters have a similarity metric of 0 [36]. The chosen definition of similarity can dramatically affect the clustering results. We define the similarity between two clusters X and Y using the arithmetic mean between cluster elements [36]:

$$s = \frac{1}{|X||Y|} \sum_{i \in X} \sum_{j \in Y} C(i,j) \tag{4}$$

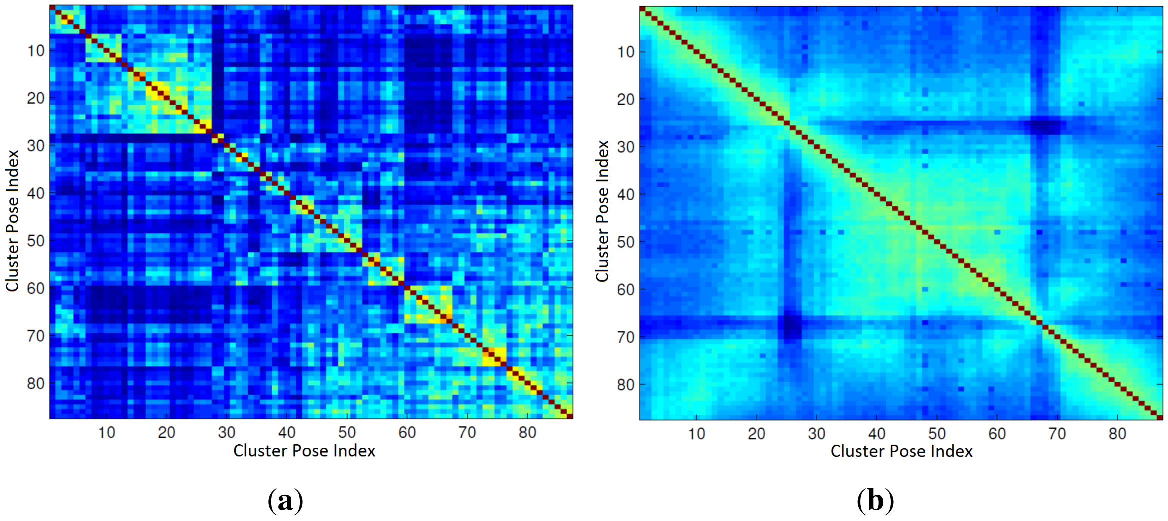
Additionally, for a hierarchical clustering algorithm it is to necessary to first determine the number of desired clusters. We compute this by inspecting the eigenvalues of the correlation matrix C. Given the eigenvalues $\lambda_1^C < \lambda_2^C < \ldots, < \lambda_{\max}^C$ we choose the number of clusters by finding the first eigenvalue where

$$\frac{\lambda_{n_c}^C}{\lambda_{\max}^C} \geq r^C \tag{5}$$

and denote the number of clusters n_c. We have empirically found that $r^C = 0.3$ provides a sufficient number of clusters for our experiments.

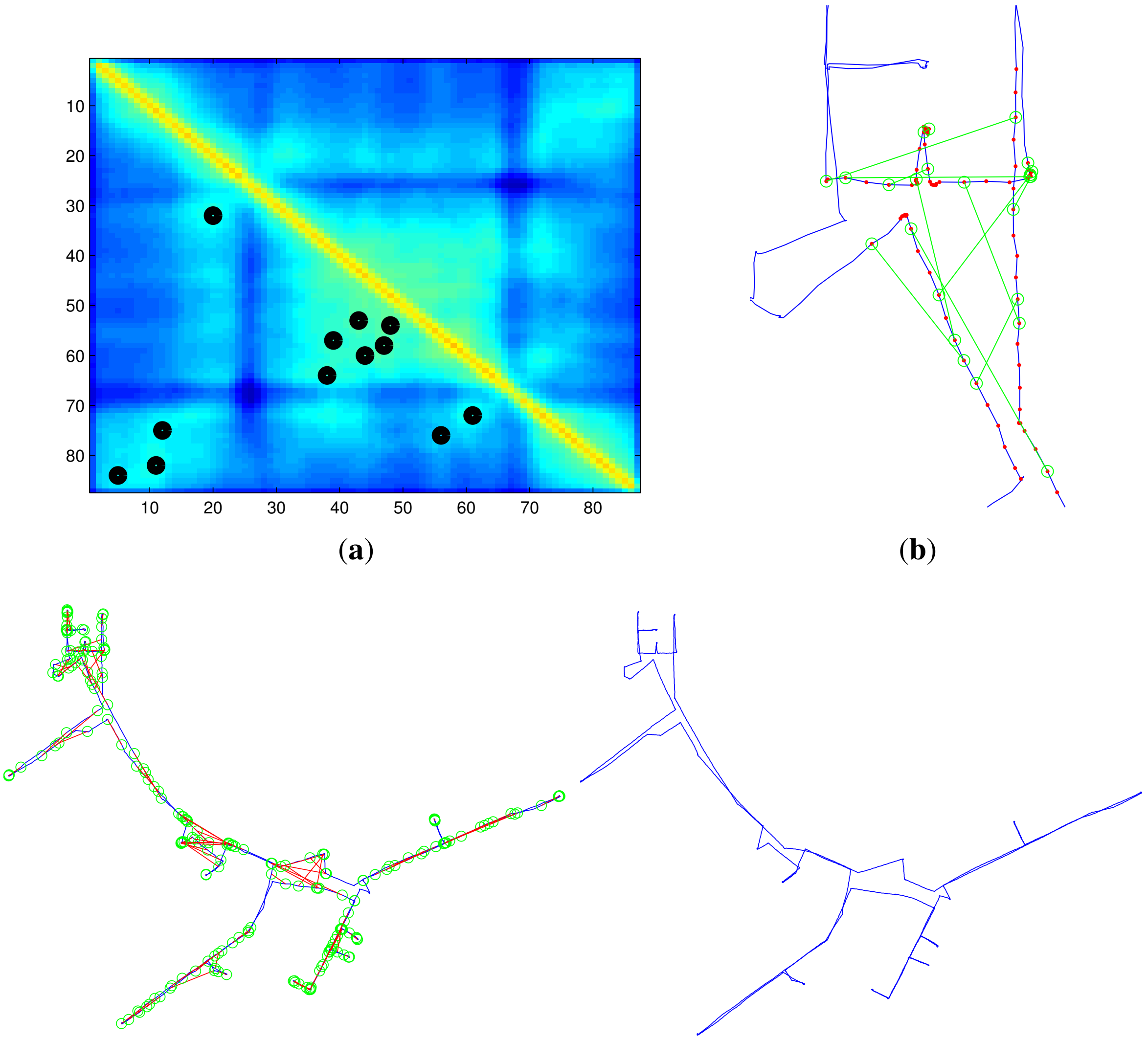
Figure 8b shows the results of the hierarchical clustering algorithm on the correlation matrix of Figure 8a. Each cluster's poses are shown using a different color overlain on the occupancy grid map. The clustering algorithm correctly groups the poses spatially even when the poses are temporally distant. This means we only must search for loop closure indices inside a single cluster which greatly reduces the search space and speeds up the subsequent cluster correlation matrix optimization.

Figure 9. (**a**) The correlation matrix of a cluster before optimization of pairwise constraints. (**b**) The correlation matrix after FICP has been run on the pairwise constraints.



Once the clusters have been identified, we extract the portion of the correlation matrix that corresponds to each cluster. Figure 9a shows an example of a cluster's correlation matrix. However, because the occupancy grid map may still have small errors in the poses, we optimize the alignment between each pair of poses within a cluster to further refine each cluster's correlation matrix. To do so, we run a round of FICP on each pair of scans to optimize alignment before computing the correlation using Equation (3). Figure 9b shows the result of applying the optimization procedure to the similarity matrix from Figure 9a. The resulting matrix contains smoother gradients between regions which adds local consistency. Additionally, a greater number of strong correlation peaks are present which reduces the rate of missed loop closure candidates. We then select candidate loop closures using the local maxima of the clusters' correlation matrices. The intuition for this is that local maxima located away from the matrix diagonal correspond to revisited locations with a large amount of overlapping geometry. To avoid grouping loop closures candidates too closely, we smooth the clusters' correlation matrices using a 3-by-3 averaging filter before selecting candidate pairs and enforce the restriction that a pose can only be part of a single loop closure candidate.

Figure 10. (**a**) The detected maxima of the cluster correlation matrix of Figure 9b. The maxima from the lower triangular section of the matrix are shown using black dots. (**b**) The loop closure pairs corresponding to the detected maxima from (a). (**c**) The result of detecting maxima from all clusters is shown by green circles connected by red lines. (**d**) The fully optimized trajectory.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 10a shows the result of detecting correlation maxima on the cluster correlation matrix of Figure 9b. The maxima from the lower triangular section of the matrix are shown using black dots. Figure 10b shows a close up of the upper portion of the trajectory. The candidate pairs corresponding to the detected maxima are connected via green lines and the red dots indicate poses in the cluster. The result of maxima detection for all clusters is shown in Figure 10c. As shown, the algorithm correctly extracts loop closure constraints at revisited locations and between poses that contain a significant portion of overlapping geometry.

2.5. Loop Closure Transform Estimation

Once the candidate pairs have been extracted, the transformation and covariance matrices must be computed before they can be used in a graph optimizer. Since the submaps from Section 2.2 are spatially quantized, we take the original pair of scans corresponding to a given loop closure, apply the scan projection algorithm of Section 2.1, and match them via a genetic scan matching algorithm. The result is a set of metric transformations between loop closure pose indices.

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 a poor choice of initial condition can lead to an undesirable solution. A class of 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 [37].

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 process is iterated until the population reaches a stable configuration. In this manner, GS algorithms avoid local minima while exploring solutions that are not in the original sampling of the solution space.

Similar to the previous works of Martinez *et al.* [38] and Lenac *et al.* [25], we parameterize the solution space using a three element chromosome. Specifically, each individual in the population consists of two translational and one rotational parameter.

$$\mu = \begin{bmatrix} \Delta x, & \Delta y, & \Delta\theta \end{bmatrix}^T \tag{6}$$

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

In contrast to the existing approaches of [25,38], we propose a new algorithm, Fractional Genetic Scan Matching (FGSM), which 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 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 residual of the Equation (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 $\mathbf{r}_i$ the FGSM algorithm can be interpreted 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 determined by considering the variance of the remaining chromosomes' parameters. Specifically, if the population's n^{th} 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.

Algorithm 1 The FGSM Algorithm

```
1:  $\mu_0 \leftarrow$ initial estimate
2:  $S_0 \leftarrow random\_chromosomes(\mu_0)$
3:  **while** $S_{n+1} \neq S_n$ **do**
4:      $R_n \leftarrow \emptyset$
5:      **for** $\mathbf{s}_i \in S_n$ **do**
6:          $\mathbf{r}_i \leftarrow ICP(\mathbf{s}_i)$
7:          $R_n \leftarrow R_n \cup \{\mathbf{r}_i\}$
8:      **end for**
9:      $R_n \leftarrow best\_subset\_of(R_n)$
10:     $S_{n+1} \leftarrow R_n \cup make\_children(R_n)$
11: **end while**
12: **return** $\mu$ and $D_f$ from best chromosome.
```

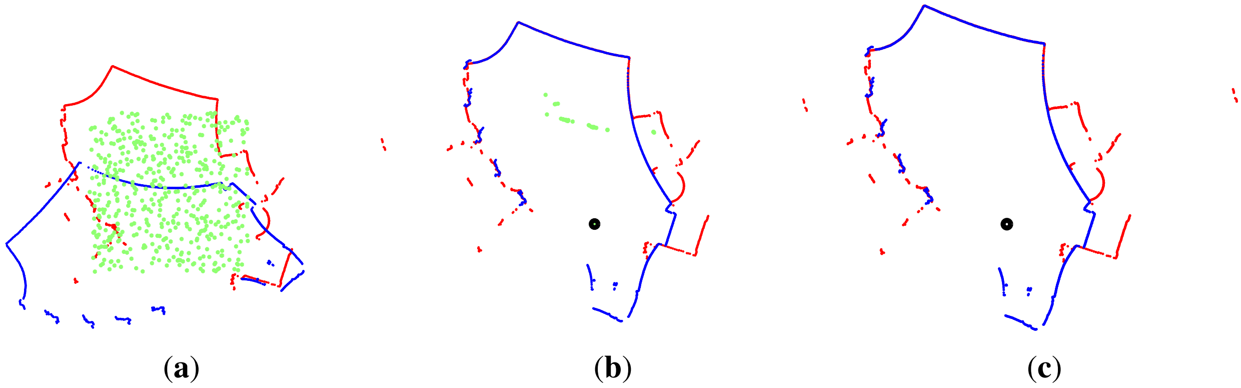
An example of the FGSM algorithm is shown in Figure 11. Figure 11a shows the initial chromosome population sampled uniformly around the supplied initial condition μ_0. Shown in Figure 11b is the distribution of chromosomes after a few iterations. Notice that the chromosomes only inhabit a few locations corresponding to the local minima of the Equation (2). After enough iterations have passed, the population, shown in Figure 11c, has converged to a single dominant chromosome representing the correct transformation parameters.

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 step 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 resulting in redundant computation that should be avoided.

Examining Equation (2), we notice that if both the point matches and the inlier set are given, then the ICP problem reduces to minimizing the following objective function:

$$e = \sum_{\mathbf{q}_i \in D_f} \|\mathbf{n}_i^T(\mathbf{p}_i - T(\mathbf{q}_i, \mu))\|^2 \tag{7}$$

Figure 11. 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.



Following [24], Equation (7) can be rewritten as:

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

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:

$$\mathbf{e}(\mu_0,\ D_f) \approx \mathbf{e}(\mu_0,\ D_f) + \mathbf{J}\mu_0 \tag{9}$$

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*

$$\mathbf{J}_i = \begin{bmatrix} -\mathbf{n}_i \\ \mathbf{n}_i^T \begin{bmatrix} \sin(\Delta\theta) & \cos(\Delta\theta) \\ -\cos(\Delta\theta) & \sin(\Delta\theta) \end{bmatrix} \mathbf{p}_i \end{bmatrix}^T \tag{10}$$

and the optimal transform can be found using

$$\mu = -(\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T\mathbf{e}(\mu_0,\ D_f) \tag{11}$$

Equations (10) and (11) reveal that when the Jacobian is evaluated around small angles, $\delta\theta \approx 0$, small deviations $\delta\mu$ from the linearization point result in insignificant changes to the resulting transformation. This analysis relies on the fact that the objective function is continuous, which in this case is not true. However, Equation (7) 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 so as not to 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 3.2 provides empirical justification for these values.

2.6. Loop Closure Transformation Verification

Even though the FGSM algorithm is a robust way of matching loop closure 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. In this section 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.

The first metric characterizes the amount of shared geometry between matched scans. Using the proposed transformation from Section 2.5, 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 [39]:

$$c = \sum_{i,j} \min(h_1(i,j), h_2(i,j)) \tag{12}$$

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.

Our second metric measures the complexity of the overlapping geometry. Using the proposed transform from Section 2.5, Equation (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

$$\mathbf{N} = \Big[\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_n\Big]^T \tag{13}$$

with which the correlation matrix:

$$\mathbf{R} = \mathbf{N}^T\mathbf{N} \tag{14}$$

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

$$r^R = \frac{\lambda_1^R}{\lambda_2^R} \tag{15}$$

assuming $\lambda_1^R \leq \lambda_2^R$. 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^R, and by extension r^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, $\lambda_1^R \approx \lambda_2^R$ and thus r^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.

2.7. Height Estimation

After the optimized 2D path has been obtained, the height of the system must be estimated at each pose before a full 6DOF path can be recovered. We recover height using the adaptive method presented in [20] whereby the floor scanner's data is classified as either planar or non-planar. If the floor is detected planar, then a direct estimate of the system's height to the floor can be made. However, if the system is not on flat terrain, sequential scan data from the floor scanner can be matched to recover the incremental change in height [18]. The incremental and absolute measurements are combined to estimate the height of the system in the global frame of reference.

2.8. 3D Path Generation

The height information along with the IMU readings and the optimized 2D path are combined to form a full six degree of freedom trajectory. The six degree of freedom pose at time t, $\mathbf{x}_t$, is comprised of

$$\mathbf{x}_t = [x_t, y_t, z_t, \phi_t, \psi_t, \theta_t]^T \tag{16}$$

where x_t, y_t, and z_t are the x, y, and z positions and ϕ_t, ψ_t and θ_t are the roll, pitch and yaw orientations. Our system uses the 3-2-1 Euler angle convention to describe orientations in 3D space. In the 3-2-1 Euler convention roll, corresponds to a rotation about the x-axis, pitch corresponds to a rotation about the y-axis, and yaw corresponds to a rotation about the z-axis. Specifically, if $R_a(b)$ is a rotation about axis a by angle b, then the rotation matrix that transforms from the local coordinate frame to the global coordinate frame, R, is given by:

$$R = R_z(\theta)R_y(\psi)R_x(\phi) \tag{17}$$

Since the 2D SLAM and height estimation algorithms provide values in the global coordinate system, the position component of $\mathbf{x}_t$ is built simply by concatenating the previously obtained values. The rotational components however must be formed more carefully. The 2D path reconstruction recovers the heading angles α_t which represent the projection of the operator's forward direction into the global xy-plane. Since the operator has non-zero pitch and roll, the heading and yaw angle are not identical in the general case. However, we can derive the correct yaw angle by combining the pitch, roll, and heading into the correct rotation matrix via:

$$R_{L\rightarrow W} = R_z(\alpha_t - \beta_t)R_y(\psi_t)R_x(\phi_t) \tag{18}$$

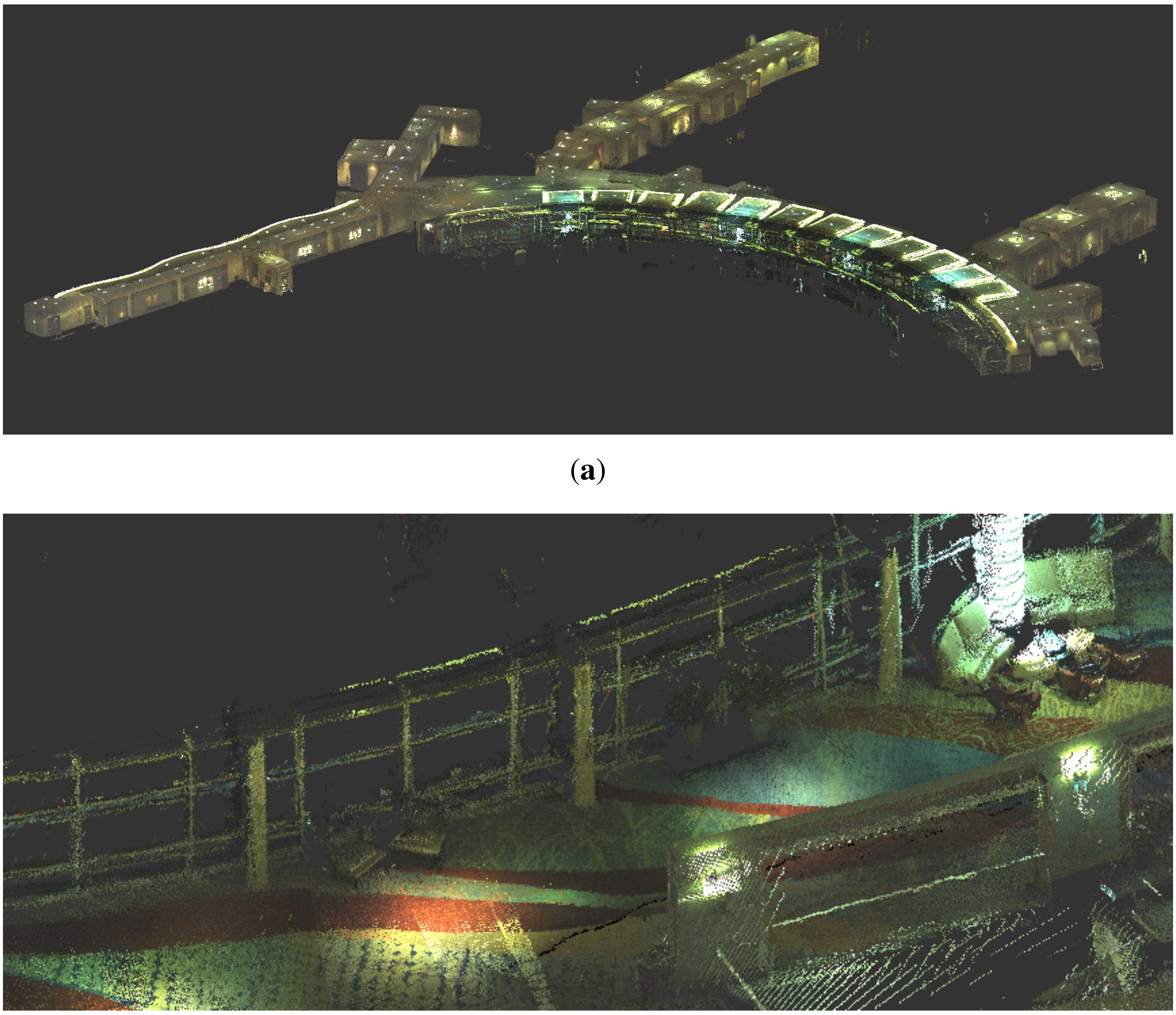
The correction factor β_t is the calculated heading if only the roll and pitch components of the orientation were applied. By accounting for β_t we ensure that even after the pitch and roll is applied the heading angle remains α_t.

The rotation matrix in Equation (18) is then decomposed into the correct roll, pitch, and yaw values. Concatenating the position and orientation values yields the full six degree of freedom pose. Figure 10d shows an example of an optimized path using the example data shown in Figure 10a–c.

2.9. Point Cloud Generation

Using the data collected by the left and right geometry scanners and the recovered trajectory we are able to generate a dense, 3D point cloud by placing the sensor readings at the correct pose location. Since the system is also equipped with 2 fish-eye cameras, the sensor readings can be colorized by projecting them into the temporally closest image. Figure 12a shows an example point clouded generated using the trajectory from Figure 10. A close up shot of some detailed furniture is shown in Figure 12b. The point cloud contains over 106 million points each with a time stamp, position, color, and unique identifier. When represented in a space delimited ASCII file, the point cloud takes up approximately 4 GB on disk.

Figure 12. (**a**) The exterior view of a dense, colorized, 3D point cloud generated by the backpack system. (**b**) A close up view of an interior section illustrates the point clouds high level of detail.



(**a**)

(**b**)

3. Results

In this section we present results for the proposed algorithms. Section 3.1 presents performance results of the FGSM algorithm using a variety of initial conditions. Section 3.2 provides empirical justification for the FGSM's discretization parameters used throughout. A thorough analysis of the vetting criteria is presented in Section **??** using both manual and automatically selected loop closure candidates. Finally, end-to-end system performance and examples are presented in Section 3.3.

3.1. FGSM Performance Evaluation

In this section we compare the proposed FGSM algorithm and a few state of the art alternatives. First, we compare against the open source libpointmatcher library [40] because it contains several implementations of state of the art ICP variants. In particular, we use the library's FICP implementation as it provided the best performance on our data. Secondly, we compare our proposed FGSM algorithm with the original hybrid genetic algorithm from which it was extended [25].

We construct the following experimental testbed to quantify the performance of each of the three chosen algorithms. We first hand-construct a set of 177 ground truth scan matching results taken from 10 different environments. Then, random initial conditions are created by selecting values from a collection of i.i.d. zero-mean Gaussian random variables with increasing standard deviation. Ten unique trials are conducted for each scan pair resulting in 1,770 attempts for each standard deviation. Each algorithm is run and the resulting transforms are recorded. We consider any transform within 5 cm or 1 degree to have converged to the correct solution.

The first experiment assesses the algorithms' performance in the presence of translation only error. Using increasing levels of translation offset, ranging from 0.25 m to 10 m in standard deviation, we ran the experimental setup and computed the percentage of successful scan matches. Table 1 summarizes the results of this experiment. In situations where the error is small, the algorithms perform nearly identical. However, as the level of errors is increased, the accuracy rates of libpointmatcher and the hybrid genetic scam matching (hybrid-GSM) algorithms decay at a much faster rate than the proposed method.

Table 1. A comparison of the proposed FGSM algorithm to previous methods in the case of translation only error. The percentage of trials that succeeded for various standard deviations in the translational component of the initial condition.

σ_T (m)	Libpointmatcher [40]	Hybrid-GSM [25]	FGSM (proposed)
0.25	93%	90%	94%
0.5	89%	91%	95%
1.0	82%	89%	94%
2.0	68%	73%	91%
3.0	58%	59%	85%
5.0	45%	42%	75%
8.0	30%	36%	66%
10.0	25%	30%	62%

In the second experiment we repeat the same procedure, but add only rotational offset using standard deviations ranging from 18° to 180°. Table 2 presents the results of this experiment. In contrast to the translation only scenario, the proposed algorithm outperforms both competitors by a wide margin. Surprisingly, the hybrid-GSM algorithm lags behind libpointmatcher by a considerable amount in this test. We suspect that this is because the hybrid-GSM algorithm does not model the presence of outliers data points.

Table 2. A comparison of the proposed FGSM algorithm to previous methods in the case of rotational only error. The percentage of trials that succeeded for various standard deviations in the rotational component of the initial condition.

σ_θ (degrees)	Libpointmatcher [40]	Hybrid-GSM [25]	FGSM (proposed)
18	74%	60%	92%
30	59%	44%	84%
45	45%	31%	73%
60	34%	24%	63%
90	25%	17%	49%
180	17%	11%	37%

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 add variations to both the rotational and translational components of the initial condition. As seen in Table 3, the algorithms performs similarly to the case of rotation only error. This result is not surprising because only the rotational variable appears as a non-linear in the object functions Jacobian. The results of these experiments show that the FGSM algorithm clearly outperforms comparable algorithms for data typical of our system.

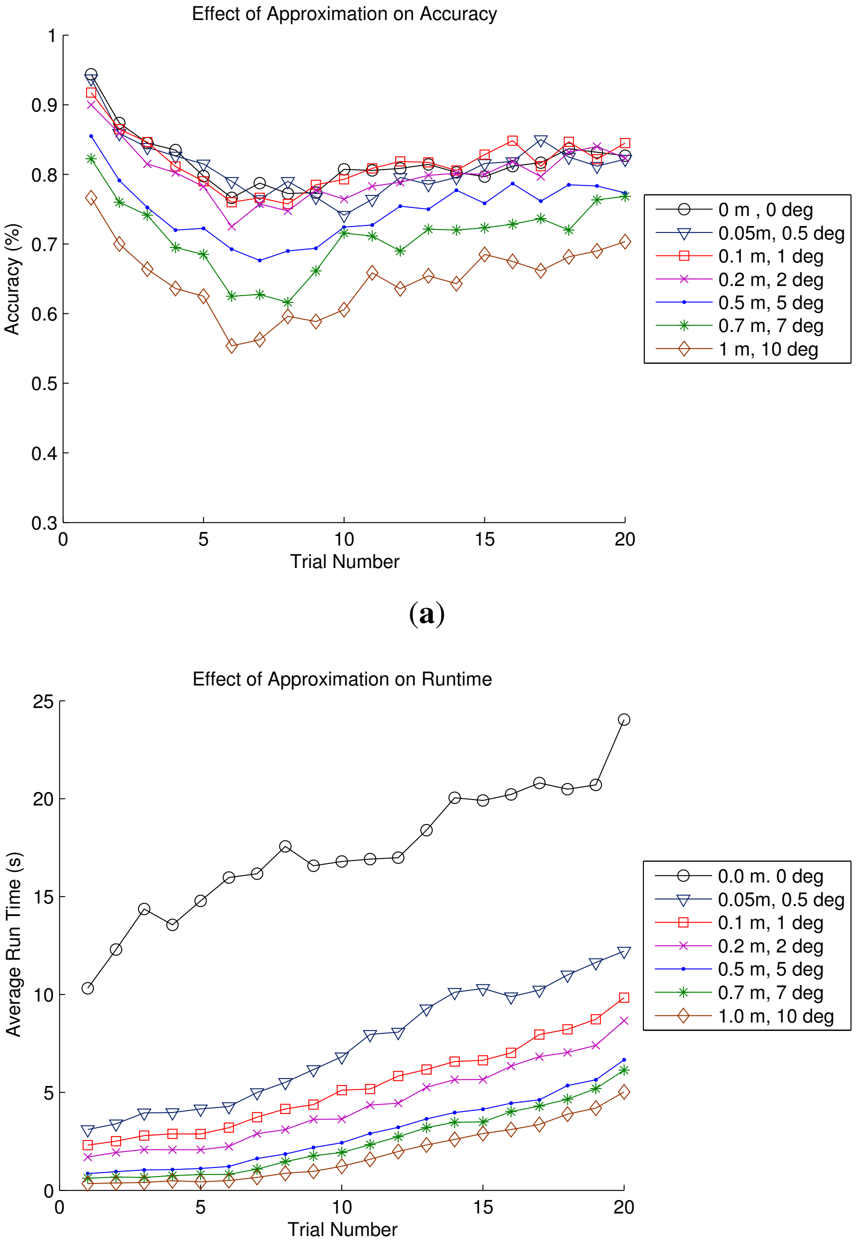
Table 3. A comparison of the proposed FGSM algorithm to previous methods in the case of both rotational and translational error. The percentage of trials that succeeded for various standard deviations in the rotational and translational components of the initial condition.

$(\sigma_T, \sigma_\theta)$ (meters,degrees)	Libpointmatcher [40]	Hybrid-GSM [25]	FGSM (proposed)
(0.25, 18)	74%	60%	91%
(0.5, 30)	55%	43%	84%
(1.0, 45)	42%	33%	74%
(2.0, 60)	28%	21%	61%
(3.0, 90)	18%	11%	45%
(5.0, 180)	9%	6%	33%

3.2. Effect of Discretization on the FGSM Algorithm

As stated in Section 2.5, the amount of solution space discretization in the FGSM algorithm represents a trade-off between computational efficiency and accuracy. In order to determine an appropriate level of discretization, we conducted the following test. We selected seven levels of discretization, ranging from no discretization to 1 m and 10 degrees, and evaluated the performance of the FGSM at increasing amounts of error in the algorithm's initial condition. For each chosen level of discretization and error in initial condition, we conducted a trial of one thousand scan matches drawn from a database of one hundred scan pairs. For each scan match we used two hundred initial chromosomes and computed the accuracy and running time by comparing the FGSM results to the manually defined ground truth alignment.

Figure 13. The effect of solution space discretization on the FGSM algorithm for increasing levels of error in initial condition. (**a**) Accuracy computed for different amounts of solution space discretization. (**b**) The average runtime of the FGSM algorithm for different levels of discretization.



(**a**)

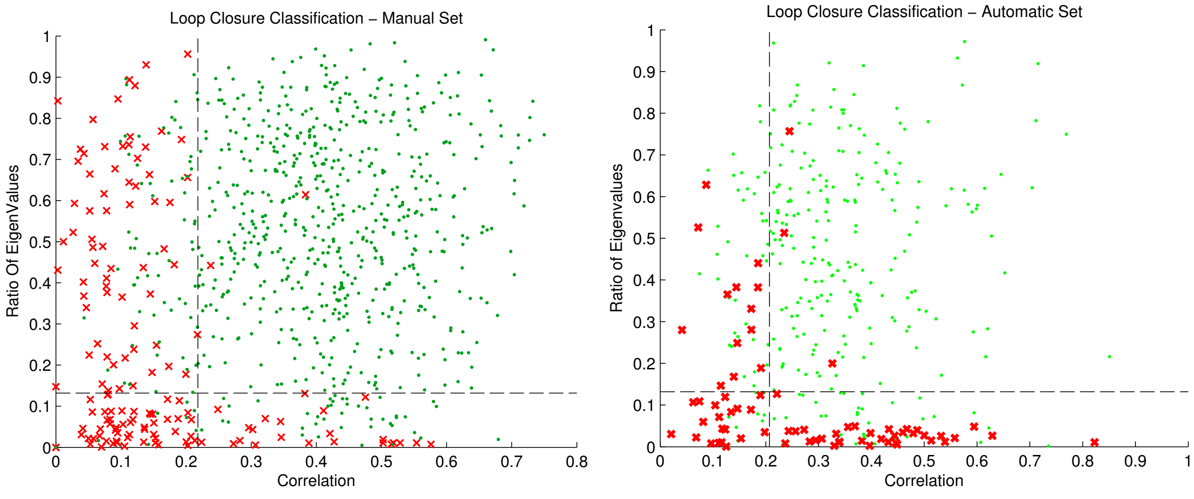
(**b**)

Figure 13 shows the results of this experiment. Figure 13a shows the effect of discretization on accuracy while Figure 13b shows the corresponding effect on run time. 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 by a factor of 3.25 without sacrificing a significant amount of accuracy compared to the baseline algorithm. Furthermore, Figure 13b shows that the FGSM algorithm's running time is correlated with the amount of error in initial condition. This result is expected because as the error in initial condition increases, the algorithm must explore a larger solution space and thus it converges at a slower rate.

In order to characterize the effectiveness of the proposed metrics for loop closure verification, a database of 15 datasets has been collected from 10 unique environments. The classification metrics are evaluated in the following two scenarios. First, between 50–100 true loop closures are manually defined for each dataset resulting in 590 positive examples. To simulate the presence of falsely identified loop closures, 10 loop closures are randomly defined for each dataset yielding 150 potentially erroneous examples. The FGSM algorithm is then run on each loop closure candidate and the resulting transformations were then manually inspected and labeled.

Figure 14a 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 a threshold based rule. Specifically, transformations are defined to be correct if the ratio of eigenvalues exceeds threshold r^R and the correlation metric exceeds threshold c.

Figure 14. 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 chosen thresholds are denoted using the dashed black lines. (**a**) The results for the manually defined loop closure candidates. (**b**) The results for the automatically detect set of loop closures.

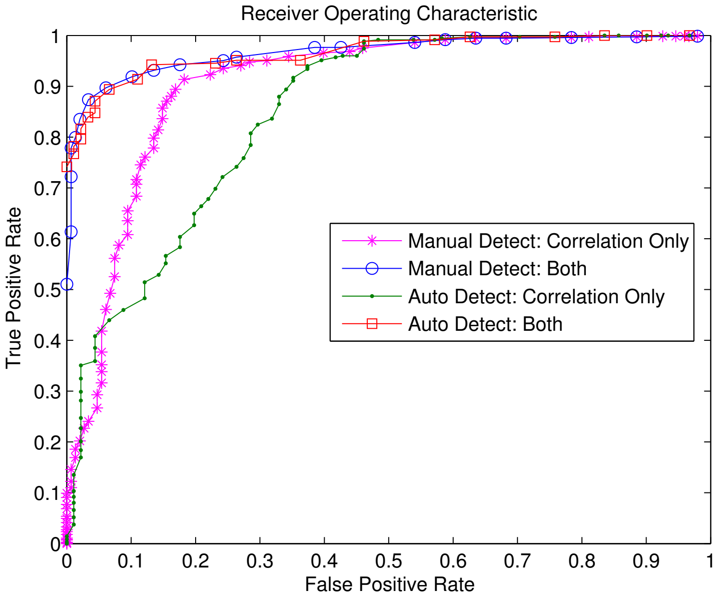


(**a**) (**b**)

Statistics for various thresholds are computed and the resulting receiver operating characteristic (ROC) curve is shown in Figure 15. When the false positive rate is required to be $\leq 1\%$, the corresponding true positive rate is 84.7% for thresholds of $r^R > 0.132$ and $c > 0.207$. The computed area under the ROC curve was found to be 94.2%.

Since the existing work on loop closure verification [24] only report statistics for detection, estimation, and validation combined, a one-to-one comparison is not possible. While a direct comparison can-not be made, the authors rely on a correlation-like technique 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 Equation (12). Figure 15 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 a correlation metric is insufficient for robust operation in indoor environments.

Figure 15. Comparison of receiver operating characteristic curves when using both metrics *versus* only correlation.

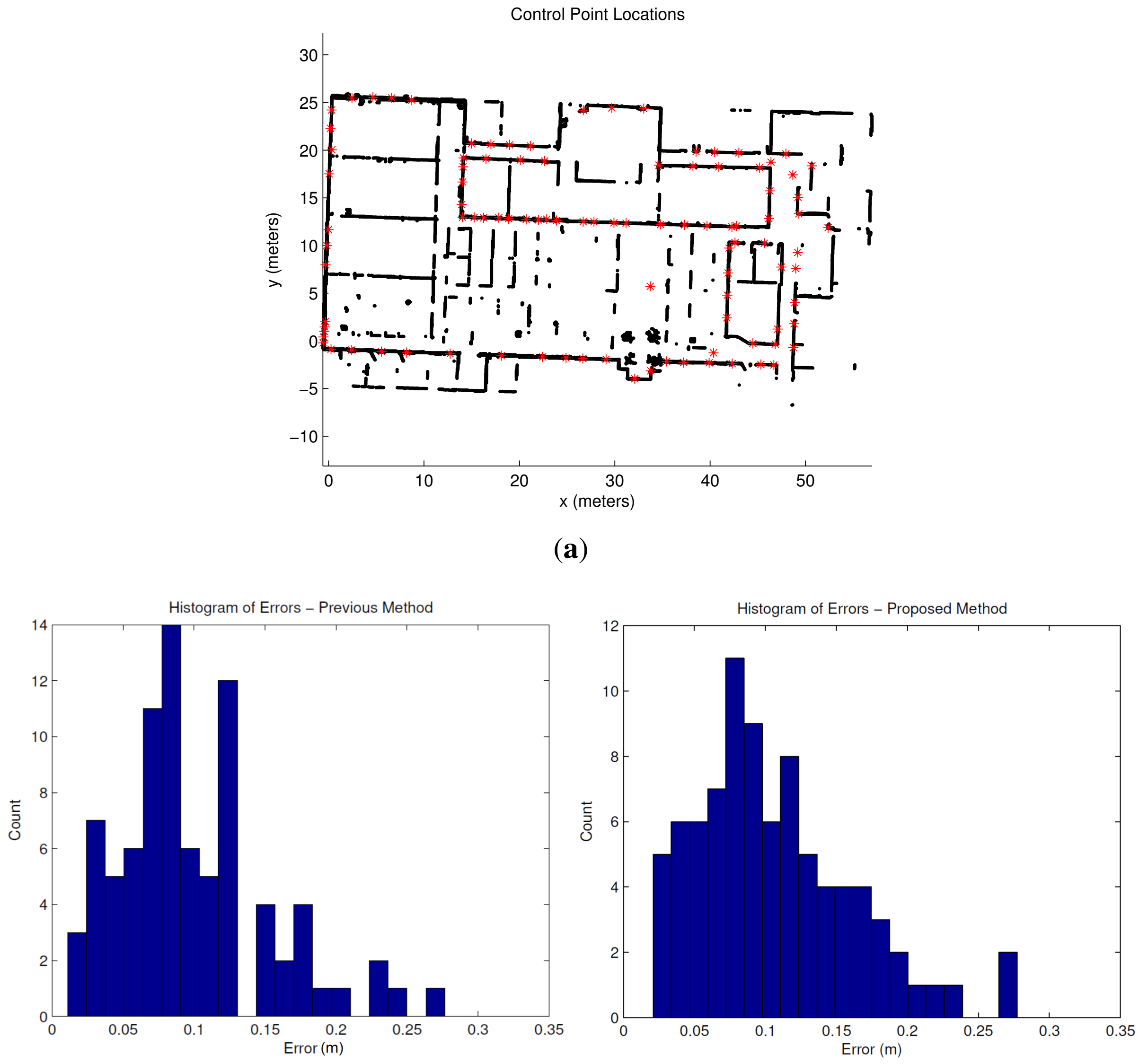


The same test is repeated, but this time using automatically detected loop closures via the method of Section 2.4. Using 5 new datasets, 361 examples are automatically detected from grid map data. The FGSM algorithm is run and the results are manually inspected and labeled. Figure 14b presents the results of using the proposed classifier on the automatically detected loop closures. The distribution of points in the plot supports the use of a threshold based rule. Using the thresholds derived from the manually defined set of loop closures, a true positive rate of 72.3% and a false positive rate of 4.3% is obtained. This analysis empirically shows that the false alarm rate and optimal thresholds are only loosely dependent on the source of loop closure candidates.

3.3. End-To-End System Results

We evaluated the end-to-end system performance using the following experiment. We collected data along a 350 m trajectory from an office building with approximately 1,500 m^2 of floor space. The office environment contained 100 pre-surveyed control points each denoted using a paper target containing a checkerboard-like pattern of known size. Using a Leica HDS6000 and C10 Scanstation, 17 high-density static scans were automatically stiched together to provide millimeter level accuracy for each survey point. We then localized the system using both the proposed algorithm and our previous approach [20]. We estimated the location of each control point by projecting the temporally closest LRF readings into the optical imagery and then manually selecting all laser points that landed on the paper target. We then fit a plane to these points using Principal Component Analysis [41] to determine the precise (x, y, z) location of the control point. The estimated locations were then compared the the ground truth locations in order to characterize the accuracy of each method.

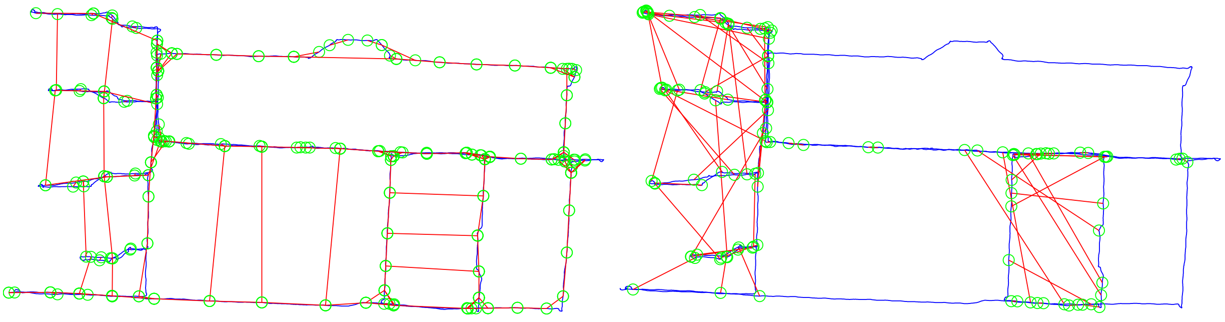
Figure 16. (**a**) The locations of the control points, shown in red, overlain on the reconstructed map. (**b**) Histogram of the errors in estimating the control points for the previous method. (**c**) Histogram of the errors in estimating the control points for proposed method. (**d**) The locations of the manually selected loop closure constraints, shown by green circles connected by red lines, used when applying the previous method. (**e**) The loop closure locations automatically selected via the proposed method.



(**a**)

(**b**) (**c**)

Figure 16. *Cont.*



(**b**) (**c**)

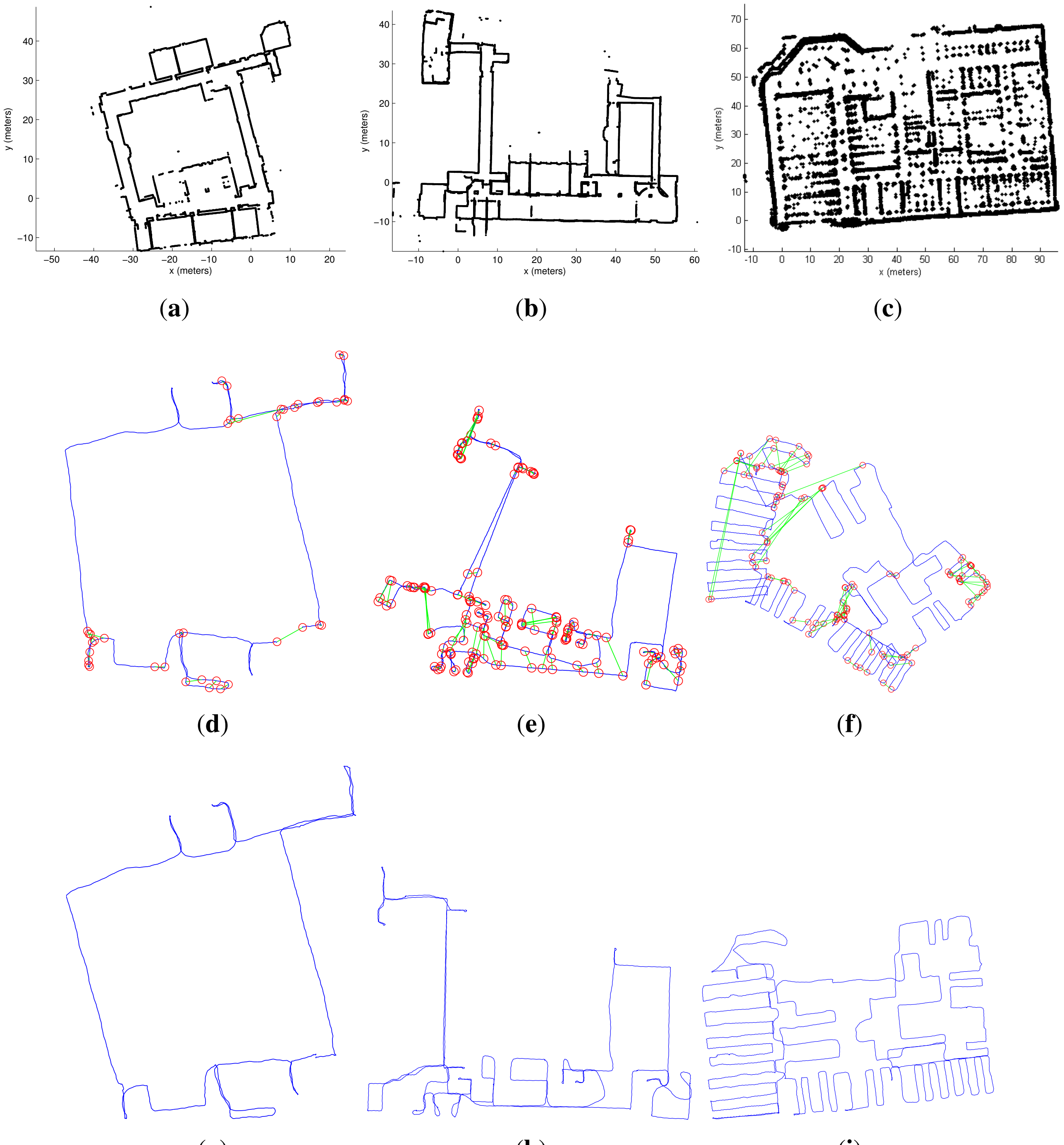
Figure 16 shows the results of the control point experiment. Figure 16a shows the locations of the control points, shown in red, overlain on the reconstructed geometry. A well trained human engineer tediously spent approximately 10 h manually selecting, estimating, and vetting 150 loop closure constraints in order to apply the method of [20]. Figure 16b shows the resulting histogram of errors for the previous method with a maximum reported error of 27.66 cm a mean error of 9.91 cm, and standard deviation of 5.30 cm amongst all control points. The histogram of errors for the proposed method is shown in Figure 16c. The proposed method utilizes only 63 loop closures, yet still attains a mean error of 10.66 cm, standard deviation of 5.49 cm, and a maximum error of 27.73 cm. Figure 16d,e show the locations of the loop closure constraints used for both the previous and proposed methods. Despite automatically generating fewer loop closure constraints, the new method is able to match the performance of the previous method which required manual intervention.

To further demonstrate the end-to-end performance of the backpack system, we have applied the proposed algorithms to three more datasets of increasing complexity. Figure 17a,d,g show a simple dataset taken from a 250 m single loop on one floor of an academic building. The resulting occupancy grid map is shown in Figure 17a. Global constraints have been extracted from the grid map and the FGSM algorithm is run to compute the transformation between loop closure locations. The trajectory after optimization is shown in Figure 17g.

The second set, shown in Figure 17b,e,h is a longer dataset taken from an entire floor of an academic building containing a few inner loops. Totaling 22 min, the operator walked for 750 m through the environment. Figure 17e shows the dead reckoning trajectory with the 80 accepted loop closures overlain. The optimized, geometrically consistent trajectory is shown in Figure 17h.

The final dataset is the most challenging because the trajectory contained many interior loops and a large portion of the surrounding geometry did not fit the vertical wall assumption made in Section 2.1. Taken from a warehouse-sized retail shopping center, the scanned area exceeded 5,820 m^2. The operator traversed the environment for 31 min and covered a total distance of 1,500 m. Despite the large amount of accumulated bias present in Figure 17f, the 59 accepted loop closure constraints extracted from the occupancy grid map enforce geometric consistency in the final optimized trajectory. This result shows that our proposed algorithms are able to scale to buildings of considerable size.

Figure 17. Results of applying the end-to-end system. (**a**), (**b**), (**c**) The occupancy grid maps that result from the RBPF algorithm. (**d**), (**e**), (**f**) The dead reckoning trajectories with validated loop closure constraints overlain. (**g**), (**h**), (**i**) The 3D paths viewed from the top down after optimization has been applied.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

Although the proposed algorithms were designed to be run off-line, it is important to highlight the approximate running time for the various stages of the algorithms. Table 4 lists the approximate running time for the various subtasks using unoptimized code on a single core of an 2.4 GHz Intel Core i7 laptop. For the 40 min data collection taken in the warehouse-sized retail shopping center, the approximate running time of all stages was 139 min. Since the localizations algorithms are run off-line, we typically use an overabundance of particles in the RBPFs. As such, the running time is generally dominated by the submap generation and grid mapping stages.

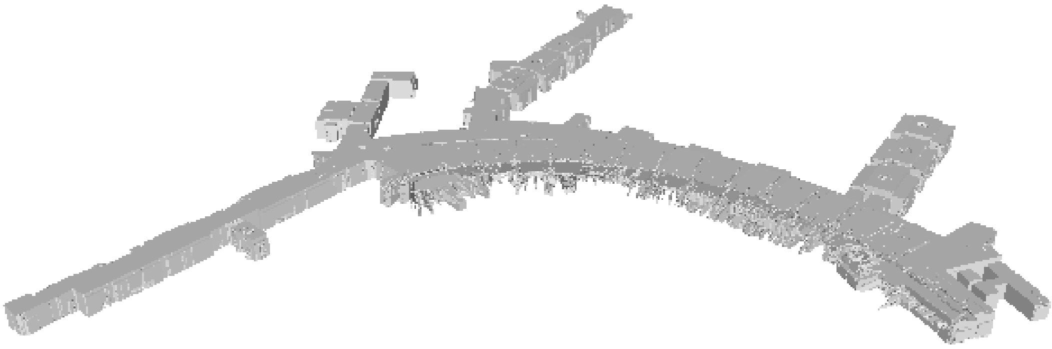
Table 4. A rough estimate of the running time for the various stages of the proposed algorithm for a 40 min data collection.

Subtask	Running Time (min)
Dead Reckoning	3
Submap Generation	50
RBPF Grid Mapping	60
Loop Closure Detection	10
Transform Estimation	2
Verification	1
Graph Optimization	2
Height Estimation	10
3D Path Generation	1

4. Applications

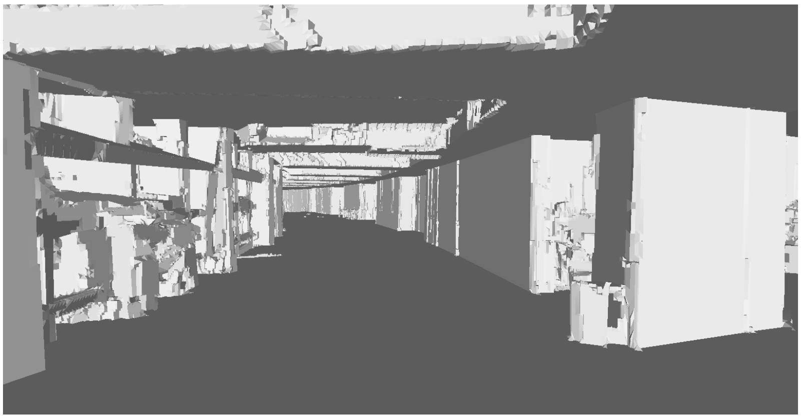
The 3D trajectories and point clouds generated using the backpack system can be utilized in many applications such as augmented reality, modeling, or architectural planning. However, other applications, such as entertainment and texture mapped models, require a lightweight triangulated mesh. Many methods exist that can convert a point cloud to a triangulated mesh. Using the method presented by Turner and Zakhor [42], we obtain a watertight mesh that represents the interior space of the environment. The meshing result of the point cloud from Figure 12a contains 4 million faces and takes up only 91MB on disk. Figure 18 shows an exterior and interior views of the mesh.

Figure 18. (**a**) A watertight mesh generated using a dense 3D point cloud. (**b**) An interior view of the mesh.



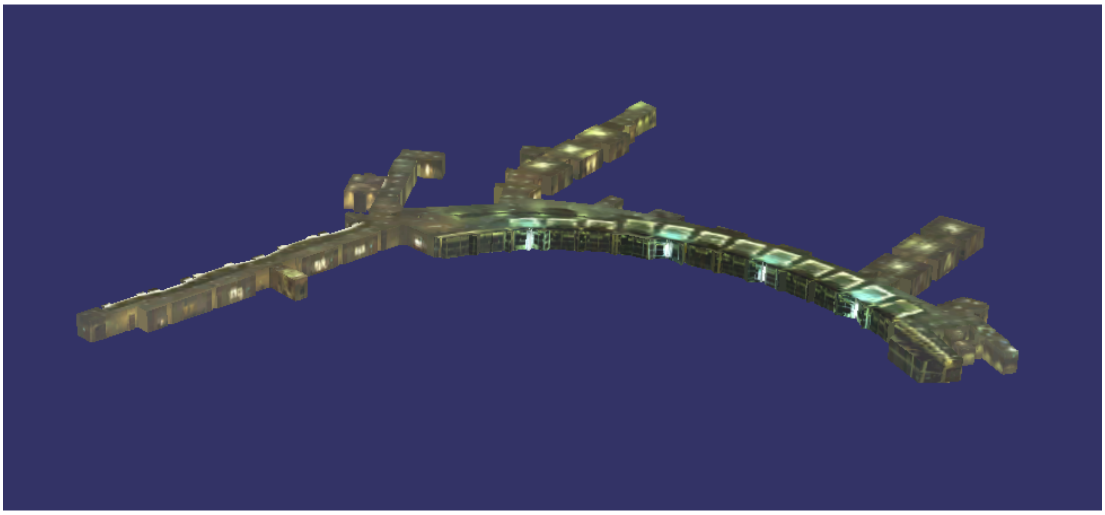
(**a**)

Figure 18. *Cont.*

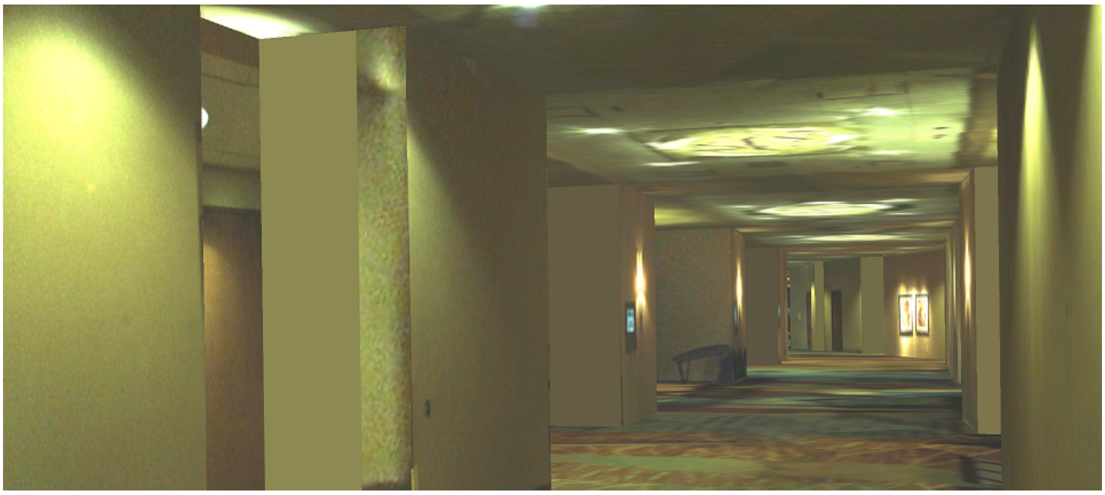


(**b**)

Figure 19. (**a**) A photo-realistic texture mapping of a watertight mesh. (**b**) An interior view from a hallway section of the texture mapped mesh.



(**a**)



(**b**)

Textures from the collected images can then be projected onto the mesh elements resulting in a photo-realistic rendering of the captured environment [43]. Figure 19 shows the result of texturing mapping the watertight mesh. The texture mapped mesh can be used used for virtual walkthroughs or augmented reality applications.

5. Conclusions and Future Work

In this paper, we highlight three novel contributions in our off-line, end-to-end pipeline for automatically mapping indoor environments using an ambulatory, human-mounted backpack system. First, we present an algorithm which automatically detects loop closure constraints from an occupancy grid map. In doing so, we ensure that constraints are detected only in locations that are well conditioned for scan matching. Secondly, we address the problem of scan matching with poor initial condition by presenting an outlier-resistant, genetic scan matching algorithm that accurately matches scans despite a poor initial condition. Third, we present two metrics based on the amount and complexity of overlapping geometry in order to vet the estimated loop closure constraints. By doing so, we automatically prevent erroneous loop closures from degrading the accuracy of the reconstructed trajectory.

The proposed algorithms were tested on a verity of datasets. The fractional genetic scan matching algorithm was tested using a hand labeled set of scan pairs disturbed by a random initial condition. The algorithm is shown to be extremely robust to translational initial condition and moderately robust against rotation. Two metrics were also proposed to verify proposed transformations. Tested on a set of 1,500 manually defined examples, the verification metrics are able to maintain a true positive rate of 84.7% when a false alarm rate of 1% is required. Finally, a metric evaluation of the end-to-end system was presented using approximately 100 pre-surveyed control points and resulted in a mean error of 10 cm. Further empirical results were presented on three datasets including a 1,500 m trajectory in a warehouse sized shopping center.

Based on our experimental results, the accuracy of the proposed method makes the system applicable for low accuracy GPS-denied mobile mapping tasks such as architectural pre-design and as-built floor plan recovery. The main advantage of the presented ambulatory system is that the operator can easily map uneven terrain such as staircases or thick carpeting that would be difficult for rolling cart systems such as the TIMMS unit [44]. However, because rolling cart systems utilize wheel odometery and do not undergo significant roll and pitch motion during data acquisition, rolling systems are likely result in a more accurate 3D reconstruction of the environment.

Future work is needed to extend the algorithms beyond man-made indoor environments dominated by vertical walls. Since the 2D trajectory is created using a 2D occupancy grid map, we implicitly make the assumption that the environment can be accurately modeled using a single occupancy grid map. For multistory buildings this is not true. We plan to extend the straightforward 2D particle filter to recognize when the operator changes floors and keep track of a separate grid map for each building level. In order to map environments that have little artificial structure, future work is needed to relax the vertical wall assumption while still ensuring accurate and robust operation.

Additionally, a significant amount of data captured by our system is not utilized to improve the performance. The side facing cameras provide rich optical imagery that can be utilized to improve the location estimates. A promising avenue for future research could be to implement an algorithm which fuses both the laser readings and the optical imagery into a single, accurate trajectory.

Acknowledgments

The authors would like to thank ARPA-E for supporting this research.

Conflict of Interest

The authors declare no conflict on interest.

References

1. Li, R. Mobile mapping: An emerging technology for spatial data acquisition. *Photogramm. Eng. Remote Sens.* **1997**, *63*, 1085–1092.
2. Karimi, H.; Khattak, A.; Hummer, J. Evaluation of mobile mapping systems for roadway data collection. *J. Comput. Civil Eng.* **2000**, *14*, 168–173.
3. Ellum, C.; El-Sheimy, N. Land-based mobile mapping systems. *Photogramm. Eng. Remote Sens.* **2002**, *68*, 13–28.
4. Holenstein, C.; Zlot, R.; Bosse, M. Watertight Surface Reconstruction of Caves from 3D Laser Data. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems San Francisco, CA, USA, 25–30 September 2011; pp. 3830–3837.
5. Bosse, M.; Zlot, R. Continuous 3D Scan-Matching with a Spinning 2D Laser. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 4312–4319.
6. Hesch, J.; Mirzaei, F.; Mariottini, G.; Roumeliotis, S. A Laser-Aided Inertial Navigation System (L-INS) for Human Localization in Unknown Indoor Environments. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 5376–5382.
7. Bosse, M.; Zlot, R.; Flick, P.; Zebedee: Design of a spring-mounter 3-D range sensor with application to mobile mapping. *IEEE Trans. Robot.* **2012**, *28*, 1104–1119.
8. Moghadam, P.; Bosse, M.; Zlot, R. Line-Based Extrinsic Calibration of Range and Image Sensors. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013.
9. Fallon, M.; Johannsson, H.; Brookshire, J.; Teller, S.; Leonard, J. Sensor Fusion for Flexible Human-Portable Building-Scale Mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura , Portugal, 7–12 October 2012; pp. 4405–4411.
10. Foxlin, E. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Comput. Graph. Appl.* **2005**, *25*, 38–46.

11. Angermann, M.; Robertson, P. Footslam: Pedestrian simultaneous localization and mapping without exteroceptive sensors—Hitchhiking on human perception and cognition. *Proc. IEEE* **2012**, *100*, 1840–1848.
12. Moafipoor, S.; Grejner-Brzezinska, D.; Toth, C. Multisensor Personal Navigator Supported by Adaptive Knowledge Based System: Performance Assessment. In Proceedings of the IEEE/ION Position, Location, and Navigation Symposium, Monterey, CA, USA, 5–8 May 2008.
13. Bok, Y.; Jeong, Y.; Choi, D. Capturing village-level heritages with a hand-held camera-laser fusion sensor. *Int. J. Comput. Vis.* **2011**, *94*, 36–52.
14. Google Trekker. Available online: http://www.google.com/maps/about/partners/streetview/trekker/ (accessed on 20 November 2013).
15. Shen, S.; Michael, N.; Kumar, V. Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 20–25.
16. Bachrach, A.; Prentice, S.; He, R.; Henry, P.; Huang, A.; Krainin, M.; Maturana, D.; Fox, D.; Roy, N. Estimation, planning, and mapping for autonomous flight using an RGB-D camera in GPS-denied environments. *Int. J. Robot. Res.* **2012**, *31*, 1320–1343.
17. Shen, S.; Michael, N.; Kumar, V. Autonomous Indoor 3D Exploration with a Micro-Aerial Vehicle. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 9–15.
18. Chen, G.; Kua, J.; Naikal, N.; Carlberg, M.; Zakhor, A. Indoor Localization Algorithms for a Human-Operated Backpack System. In Proceedings of the 3D Data Processing, Visualization, and Transmission, Paris, France, 17–20 May 2010.
19. Liu, T.; Carlberg, M.; Chen, G.; Chen, J.; Kua, J.; Zakhor, A. Indoor Localization and Visualization Using a Human-Operated Backpack System. In Proceedings of the International Conference on Indoor Positioning and Indoor Navigation, Zurich , Switzerland, 15–17 September 2010.
20. Kua, J.; Corso, N.; Zakhor, A. Automatic loop closure detection using multiple cameras for 3D indoor localization. *IS&T/SPIE Electron. Imag.* **2012**, doi: 10.1117/12.916639.
21. Newman, P.; Cummins, M. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665.
22. Bosse, M.; Zlot, R. Keypoint design and evaluation for place recognition in 2D lidar maps. *Robot. Auton. Syst.* **2009**, *57*, 1211–1224.
23. Granstrom, K.; Callmer, J.; Nieto, J.; Ramos, F. Learning to Detect Loop Closure from Range Data. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 15–22.
24. Bosse, M.; Zlot, R. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *Int. J. Robot. Res.* **2008**, *26*, 667–691.
25. Lenac, K.; Mumolo, E.; Nolich, M. Robust and Accurate Genetic Scan Matching Algorithm for Robotic Navigation. In Proceedings of the International Conference on Intelligent Robotics and Applications, Aachen, Germany, 6–8 December 2011; pp. 584–593.

26. Phillips, J.; Liu, R.; Tomasi, C. Outlier Robust ICP for Minimizing Fractional RMSD. In Proceedings of the 3-D Digital Imaging and Modeling, Montreal, QC, Canada, 21–23 August 2007; pp. 427–434.
27. Grisetti, G.; Stachniss, C.; Burgard, W. Non-linear constraint network optimization for efficient map learning. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 428–439.
28. Dellaert, F.; Kaess, M. Square root SAM: Simultaneous localization and mapping *via* square root information smoothing. *Int. J. Robot. Res.* **2006**, *25*, 1181–1203.
29. Fitzgibbon, A. Robust Registration of 2D and 3D Point Sets. In Proceedings of the British Machine Vision Conference, Manchester, UK, 10–13 September 2001; pp. 662–670.
30. Censi, A. An ICP Variant Using a Point-to-Line Metric. In Proceedings of the IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 19–25.
31. Chetzerikov, D.; Svirko, D.; Stepanov, D.; Kresk, P. The Trimmed Iterative Closest Point Algorithm. In Proceedings of the International Conference on Pattern Recognition, Quebec, Canada, 11–15 August 2002.
32. Besl, P.; McKay, N. A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256.
33. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; The MIT Press: Cambridge, MA, USA, 2005.
34. Murphy, K. Bayesian Map Learning in Dynamic Environments. In Proceedings of the Neural Information Processing Systems (NIPS), Denver, CO, USA, 30 November 1999.
35. Grisetti, G.; Stachniss, C.; Burgard, W. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters By Adaptive Proposals and Selective Resampling. In Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 18–22 April 2005; pp. 2432–2437.
36. Jain, A.; Murty, M.; Flynn, P. Data clustering: A review. *ACM Comput. Surv.* **1999**, *31*, 264–323.
37. Man, K.; Tang, K.; Kwong, S. *Genetic Algorithms, Concepts and Designs*; Springer: London, UK, 1999.
38. Martinez, J.; Gonzalez, J.; Morales, J.; Mandow, A.; Garcia-Cerezo, J. Mobile robot motion estimation by 2D scan matching with genetic and itertive closest point algorithms. *J. Field Robot.* **2006**, *23*, 21–34.
39. Barla, A.; Odone, F.; Verri, A. Histogram Intersection Kernel for Image Classification. In Proceedings of the International Conference on Image Processing Barcelona, Spain, 14–17 September 2003; Volume 3, pp. 513–516.
40. Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing ICP variants on real-world data sets. *Auton. Robot.* **2013**, *34*, 133–148.
41. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *1*, 37–52.
42. Turner, E.; Zakhor, A. Watertight Planar Surface Meshing of Indoor Point-Clouds with Voxel Carving. In Proceedings of the International Conference on 3D Vision, Seattle, WA, USA, 29 June–1 July 2013.

43. Cheng, P.; Anderson, M.; He, S.; Zakhor, A. Texture Mapping 3D Planar Models of Indoor Environments with Noisy Camera Poses. To Appear In Proceedings of the SPIE Electronic Imaging Conference, San Francisco, CA, USA, 2–6 February 2014.
44. Trimble TIMMS. Available online: http://www.trimble.com/Indoor-Mobile-Mapping-Solution/ (accessed on 20 November 2013).