J Intell Robot Syst (2015) 80:231–254
DOI 10.1007/s10846-014-0158-y

Perimeter-Based Polar Scan Matching (PB-PSM) for 2D Laser Odometry

Chen Friedman · Inderjit Chopra · Omri Rand

Received: 30 April 2014 / Accepted: 12 November 2014 / Published online: 23 November 2014

Abstract The paper presents Perimeter-Based Polar Scan Matching (PB-PSM), a new 2D scan matching algorithm. The algorithm favors matches with a larger perimeter overlap between the two input scans, while using a robust cost minimization process (using an adaptive direct search method, made possible due to a linear complexity data association technique). PB-PSM is benchmarked against the previously published PSM and PSM-C algorithms, and numerous realizations of the ICP algorithm. Results for convergence, accuracy, and computational speed are discussed. PB-PSM is employed on several laser scan datasets, both existing and in-house. Quantitative comparison of resulting maps is done using a new metric for evaluating occupancy grid maps accuracy, by calculating the average cell distance from the walls of the true map. The relative importance of each novel contribution is quantified using the new metric. Additional qualitative analysis is provided for previously published and relatively large datasets.

Keywords Scan matching · Laser range scanner · SLAM · Mapping

C. Friedman (✉) · I. Chopra
Alfred Gessow Rotorcraft Center, Department of Aerospace Engineering, University of Maryland, College Park, MD 20742, USA
e-mail: chen.friedman1@gmail.com

I. Chopra
e-mail: chopra@umd.edu

O. Rand
Faculty of Aerospace Engineering, Technion, Israel Institute of Technology, Haifa, Israel
e-mail: aeromri@aerodyne.technion.ac.il

Present Address:
C. Friedman
Department of Mechanical and Aerospace Engineering, The George Washington University, 801 22nd St, NW Washington, DC 20052, USA

List of Symbols

n	Total number of points.
n_c	Number of cost-contributing points.
r	Range [mm].
R_{min}, R_{max}	Minimum/maximum laser range, respectively [mm].
T_E	Threshold for eliminating matching anomalies.
T_M	Threshold for consideration of matching success.
T_F	Threshold for final cost function accepted value.
x, y	Cartesian laser point coordinates [mm].
α	Threshold for shallow angle definition.
θ	Beam angle.
ψ	Azimuth angle [rad].
ε	Error with respect to ground truth.
$\Delta()$	Property shift.
$()'$	Quantity after roto-translation.
$()''$	Interpolated radii value.

$()_C$	Current scan related property.
$()_R$	Reference scan related property.

List of Abbreviations

EKF	Extended Kalman Filter.
FOV	Field of View.
GPS	Global Positioning System.
ICP	Iterative Closest Point.
IDC	Iterative Dual Correspondence.
IMRP	Iterative Matching Range Point.
OG	Occupancy Grid.
PB-PSM	Perimeter-Based Polar Scan Matching.
PM	Perimeter Matching.
PSM	Polar Scan Matching.
PSM-C	Polar Scan Matching - Cartesian.
SLAM	Simultaneous Localization And Mapping.

1 Introduction

Operation in GPS denied environments is imperative for a variety of mission scenarios including surveillance, search and rescue, and biological chemical agent detection using unmanned vehicles. These mission scenarios require accurate position information and mapping capability of the operational area. In cases where no a priori map of the environment is known, and no external position information can be received, the problem becomes a coupled estimation process known as Simultaneous Localization and Mapping (SLAM) [12, 40].

Some SLAM algorithms use a dynamic model of the platform to aid the estimation process, which is usually based on some form of a Bayesian filter [40]. The most common estimation algorithms for SLAM are known to be Extended Kalman Filters [3, 11, 12], and Particle Filters [1, 4, 20, 36].

Other algorithms rely on scan matching techniques alone to generate both map and position estimates [7, 25, 30, 32, 35]. The process of scan-matching between two environment scans results in the roto-translation values required to match one scan on top of the other. Many types of scan matching algorithms exist [2, 8, 10, 19, 24, 26, 31, 42], with different strengths and weaknesses.

Scan matching algorithms do not require a dynamic model for the platform's motion, and thus may, in principle, be applied to any platform regardless of its dynamic behavior. The majority of the work in this area makes use of 2D laser scanners [7, 25, 30, 32]. However, some research efforts use 3D laser scanners [27, 38] (providing 6D SLAM capabilities). Scan matching algorithms are used either for loop closure [10] or for providing odometry information and generating the evolving map [35].

A scan matching algorithm can be used for pose estimation and mapping in a sequential fashion. Scan matching gives the estimate for the new position, based on which the latest scan can be used to update the map, and the process repeats with the next laser scan. This can be performed using sequential laser scans or between a newly acquired laser scan and a scan of the evolving map, made from an approximate position (by means of ray casting). Scan matching performed on subsequent laser scans is likely to result in relatively larger accumulation of errors as demonstrated by Bailey and Nebot [2].

This paper limits the discussion to local scan matching which is performed between two subsequent scans of the environment (unlike global scan matching which matches between a laser scan and a complete map [41]). The scan of the environment may be carried out using laser scanners, sonar range sensors, or cameras. In this work, we focus on representing the environment using a 2D laser scan. The result of a single scan is a set of range measurements given over a set of azimuthal angles in the scanner plane. For this type of data, three categories of scan matching techniques exist:

1. Feature-to-Feature [24]: this technique extract features (such as lines, corners, etc.) from both the current and the reference scans. The features are then matched using some algorithm to get the translation and rotation between the two scans. Correspondence between the features needs to be established correctly in order to assure accuracy, speed, and convergence. These techniques are relatively fast as they reduce the amount of data from the number of laser points, to a much smaller number of extracted features.
2. Point-to-Feature [8, 9, 19]: in this technique, features are extracted only from the reference scan, while the points from the current scan are associated with those features to establish the correct solution. Here too, establishing the right

correspondence between points and features is key to the success of this technique.

3. Point-to-Point [10, 24]: this technique is considered to be more robust as the scanned environment does not have to be comprised of geometric features. The technique makes use of the point sets themselves in the computation of the scan matching solution. However, the number of points dominates the complexity of the algorithm. The correspondence between the matched points can be based on inter-point distance such as in the Iterative Closest Point (ICP) algorithm [6], range from the origin as in Iterative Matching Range to Point (IMRP) [24], or other variants.

As these techniques all have strengths and weaknesses - a number of works attempted a synthesis of two techniques that would complement one another and yield an overall better solution. Examples include the Iterative Dual Correspondence (IDC) algorithm which combines ICP and IMRP [24], and combining IDC with a line-based algorithm [9] to form a hybrid algorithm that works well in either a polygonal or non-polygonal environments [19]. These attempts also aim to take advantage of the computational complexity advantages of each algorithm. There are several common ingredients between these methods:

(a) start with an initial guess (if any).
(b) project the Current Scan onto the Reference Scan's coordinate system.
(c) eliminate points that are either measurement outliers or occluded.
(d) define correspondence between the points or features in the two scans.
(e) calculate a cost function to evaluate the match.
(f) employ a minimization algorithm to minimize the cost.

Polar Scan Matching (PSM) is an algorithm that exploits the natural data structure of a laser scanner readings in polar coordinates (r, θ). This paper presents an alternative version of using polar coordinates for scan matching, which was first utilized by Diosi and Kleeman [10] for scan matching 2D laser scans. The key feature in scan matching using polar coordinates, is that establishing correspondence between the laser points is carried out using the matching bearing rule, which makes use of the point's azimuth angles.

Diosi and Kleeman present two PSM algorithms [10]. The first relies on regression fit for the planar matching, and on an optimization of a cost function (total distance), for azimuth matching. The second algorithm, named PSM-C, estimates the translation in cartesian coordinates by direct solution of the minimization problem for the planar translation. PSM-C showed a certain relative speed advantage over PSM, while maintaining relatively the same accuracy performance.

The algorithm presented herein is dubbed Perimeter-Based Polar Scan Matching (PB-PSM). In addition to using a form of direct search (for both rotation and translation) to overcome the very common local minima phenomenon [10], PB-PSM favors matches with larger overlap between the scans, thus increasing robustness. The advantages of both features are shown by comparing PB-PSM to several other algorithms on a complete laser scan dataset benchmark. The algorithm is proven to be quite accurate across both in-house and previously published datasets. The complete PB-PSM cost function results in a linear $O(n)$ complexity (where n is the number of points), which is an improvement as compared to previous cost functions with $O(n^2)$, $O(kn)$ (using a limited search window), or $O(n \log n)$ complexity. This makes direct search feasible in real time on current processors.

The paper presents the PB-PSM scan matching algorithm, followed by a validation and performance comparison against Diosi and Kleeman's original PSM, PSM-C, and several different realizations of the ICP method, employed on multiple datasets. Convergence and computational resources requirements are also discussed. Resulting maps from laser odometry are compared quantitatively by employing a new proposed metric which ranks the resulting maps against a true map of the scenario. Qualitative analysis is also presented. The algorithm shows robustness to different sensors, FOV, and scan frequency (number of scans per second). Finally, conclusions are drawn about the performance of the suggested PB-PSM algorithm.

2 Algorithm Description

The PB-PSM algorithm detailed below takes two scans as input, and outputs the roto-translation between them. No initial guess is required as input. Both scans are first passed through a series of point filters such as range, field of view, mixed pixels, and occlusion. The filtered scans are then searched for data association, and a cost function is built from the associated point pairs. The cost is rewarded for larger overlap between the scans. The best solution is found by minimizing the cost using a form of exhaustive search. The minimization process is separated for rotation first followed by planar translation, and the process is iterated until adequate convergence is achieved.

Although no initial guess is used (or required), if an initial guess is available (such as from wheel encoder based odometry, IMU, etc.) it may be used to reduce the number of iterations required for convergence. A good initial guess will also increase robustness, and reduce computational requirements.

2.1 Point Filters

The acquired points are passed through a series of filters designed to leave only valid laser points for the scan matching process. These filters include: range limits filter, field of view filter, outlier/mixed pixels filter, and an occluded points filter, which are described below. The filters design depends on the laser scanner's capabilities (range, resolution, FOV, accuracy, and sensitivity).

2.1.1 Range Limits Filter

Most laser scanners output either zero or their respective maximum detection range when no valid range reading is picked up. Therefore, the range limits filter simply excludes validity of points where $r < r_{min}$ or $r > r_{max}$. Values of 400 mm and 29000 mm respectively, which are attributed to the UTM-30LX Hokuyo sensor, are used in this work. These values are updated for different laser scanners (when using 3^{rd} party datasets collected with different laser scanners).

2.1.2 Field of View Filter

After the rotation and translation of the Current Scan, some of the points may lie outside of the FOV of the Reference Scan. Therefore, it is necessary to identify the right FOV overlap between the two scans so that the scan matching process will only be carried out on potentially matching points. The FOV filter is designed to exclude points where either $\theta_C < \theta_{R_{min}}$ or $\theta_C > \theta_{R_{max}}$. This can also occur with the Reference Scan where points do not have any overlap with the Current Scan's FOV, and so this filter also excludes points where $\theta_R < \theta_{C_{min}}$ or $\theta_R > \theta_{C_{max}}$. Note that laser points that are eliminated by this filter from the Current Scan are still being updated into the Occupancy Grid (OG), after the scan matching process is completed (as these are simply new points that may not be observed from the Reference Scan's origin).

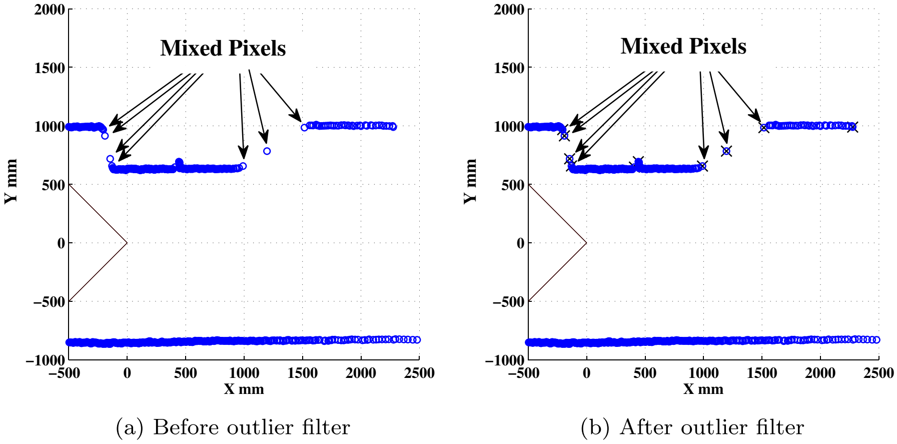
2.1.3 Mixed Pixels Filter

A rather common and inherent laser scan error occurs when the scenario contains surface discontinuities. This is quite typical in any environment that contains edges. Wherever a range discontinuity occurs, the laser beam may capture two surfaces with a depth difference, producing an outlier point that can take on any range value between the two different surfaces' ranges. An example of a laser scan that produced several outliers is presented in Fig. 1a. Such points are also known as mixed pixels [10].

Mixed pixels typically lie in the free space, and thus are not representative of any real object. Since the nature of outliers would change from one scan to another (as they are considered to be an anomaly of the laser scanner), one may not use outlier points when performing scan matching, as those would undoubtedly introduce errors into the process. Therefore, outlier points should be ignored altogether.

Conventional approaches to eliminating such mixed pixels involve evaluating the distance between the mixed pixel and the preceding point [10]. The suspected mixed pixels would be accepted if the range is smaller than a set threshold, typically set after some experiments with the laser scanner. An improvement for the above approach could be to consider the range from the mixed pixel to both the preceding point and to the succeeding point. However, for any set threshold, some mixed pixels may avoid being detected. Observing Fig. 1, it can easily be claimed that every set threshold will only eliminate some of the mixed pixels, but not all of them.

The mixed point filter presented below is specifically designed for the PB-PSM scan matching algorithms in two dimensions. A comprehensive review and testing of mixed pixel filters is given in the work



(a) Before outlier filter

(b) After outlier filter

Fig. 1 A laser scan (*circles*) before and after applying the outlier filter. All the points that may be mixed pixels are eliminated (*crossed out*). The *two lines* show the field of view boundaries (scanning counter clockwise)

by Tang et al. [37]. The outlier filter used in this work relies on examining the angle between successive points. As mixed pixels lie on range discontinuities, the angle between a mixed pixel and its preceding point would typically be relatively shallow. Thus, we can identify outlier candidates using the relative angle between the beam and the line connecting the candidate point to its succeeding point.

Figure 2 presents a schematic description of how outlier points are detected. Laser measurements are marked as red stars, three representative laser scanner rays are shown as red lines, and the laser origin is marked by a red circle. The filter checks for the angle between every two neighboring laser points relative to a line perpendicular to the laser beam angle (denoted as α in Fig. 2). If α is close to 90° – the point is discarded. A value of $\alpha = 85°$ is used throughout this work.

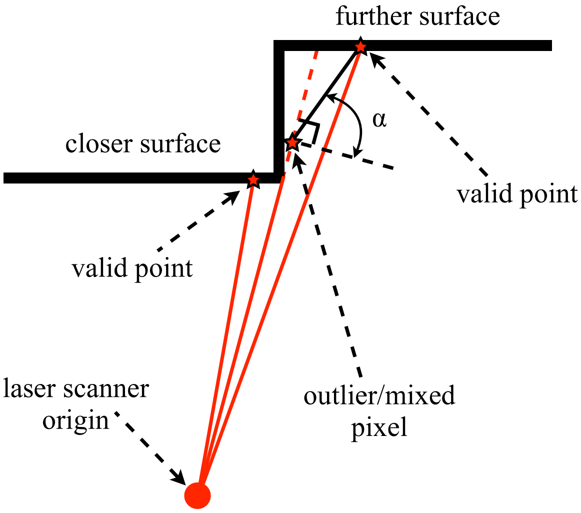


Fig. 2 Example of identifying an outlier (mixed pixel) laser measurement

Figure 1b presents the result after applying the outlier filter described above. The eliminated points are crossed out with a black ‘x’ mark. In some rare cases points can be wrongfully eliminated (due to laser noise creating the same conditions between two neighboring laser points). Several such points may be seen on the right hand side of the upper wall in Fig. 1b. However, the large amount of laser points in each scan still provides sufficient information for the algorithm to perform well.

A positive byproduct of using the angle-based outlier filter is that it also eliminates laser points where the angle of the beam with the measured surface is relatively shallow. Measuring walls at a shallow angle is likely to produce larger errors [28] and thus such points are less desired when performing scan matching.

2.1.4 Occlusion Filter

After the laser scan is roto-translated (using Eq. 1 below), some scanned objects may become occluded by others. A simple example is a scan that is performed right when the laser has passed around a corner, as in Fig. 3a. The points that are picked up by the laser after passing the corner cannot be viewed when the scan is transformed to the new origin (marked by an ‘x’).

Roto-translating the scan to any point that’s located before the corner would result in an occluded wall, since the same wall could not have been observed from that point. The geometric result of this transformation is a change in the order of the laser scan angles, as depicted in Fig. 3b.

The filter is therefore built to identify such switch backs in the previously ordered laser points' angles, and remove only the ones that are occluded. Note that in some cases due to the inherent laser sensor noise, some points are wrongfully filtered out, however, the number of wrongfully eliminated points is minuscule (on the order of 10 points), and so it does not degrade the overall performance in any way. The Occlusion filter is employed only on points that are still "active", and as such, it must follow the min/max filter immediately so it would not be affected by the operation of other filters.

The filter is in the form of a simple sweep algorithm, starting from the smallest angle (from left to right). Figure 4 shows two scenarios of a series of laser measurements (circles, connected by a line for clarity), with their respective angles θ on the 'x' axis, and range on the 'y' axis (after a roto-translation). Each new laser measurement is tested for an angle switch back event, and when a switchback is discovered the algorithm acts according to one of the scenarios depicted in Fig. 4, eliminating only the points that are in fact occluded (marked with an 'x').

Algorithm 1 Occlusion filter

```
for all $r_C$ do
    if $\theta_i < \theta_{i-1}$ then    ▷ If an angle order reversal is detected
        if $r_i > r_{i-1}$ then    ▷ Scenario (a) - the following points are occluded
            March until $\theta_j > \theta_{i-1}$
            Eliminate points $i$ through $j-1$
        else    ▷ Scenario (b) - the previous points are occluded
            March until $\theta_j > \theta_{j-1}$
            Eliminate previous points $k$ through $j-2$, where $\theta_k > \theta_{j-1}$
        end if
    end if
end for
```

The filter was designed to have $O(n)$ complexity, where n is the number of points in the laser scan. Since the laser measurements are supplied from the scanner sorted by their respective angle - there's no need to employ a sorting algorithm, or a window-based search for correspondence (avoiding a complexity of $O(n \log(n))$ or $O(kn)$, respectively). Every point is examined for angle switchback, and at most examined again for being part of an occluded range, and possibly eliminated. Therefore, the complexity is upper bounded by $O(3n)$ which represents a linear complexity of $O(n)$. An example of the elimination process is given in Fig. 5, where the radii values are plotted against their respective angle, after transformation. The laser points that cannot be observed are marked with an 'x'.

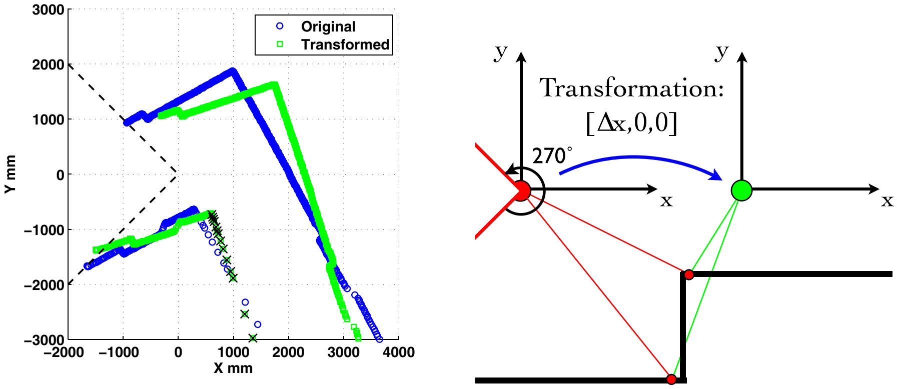
2.2 Linear Complexity Correspondence Search

Finding point correspondence (data association) between the Reference and the Current scans is described using the following pseudo-code:

Algorithm 2 Point correspondence search

```
$k \leftarrow b$    ▷ b: first Current Scan point within the FOV of the Reference Scan
for $i = a \rightarrow n_R - 1$ do    ▷ a: first Reference Scan point within the FOV of the Current Scan
    if Reference Scan point $i$ is active then
        while $k < n_C$ do    ▷ k: running index on the Current Scan
            if (k) and (k-1) are both valid laser points then
                if $\theta_{C_k} \geq \theta_{R_i} \geq \theta_{C_{k-1}}$ then
                    Establish correspondence between point $i$ in the Reference Scan and points $[k-1, k]$, in the Current Scan
                else
                    $k \leftarrow k + 1$
                end if
            else
                $k \leftarrow k + 1$
            end if
        end while
    end if
end for
```

This algorithm resembles that of Diosi and Kleeman [10]. For each active point in the Reference scan, this algorithm searches for two corresponding



(a) Roto-translated Laser scan example

(b) Schematics

Fig. 3 An example and schematics for object occlusion. **a** Roto-translated laser scan example. *Blue circles* represent the originally acquired laser scan, with the FOV marked by *dashed black lines*. *Green squares* represent the roto-translated scan points, where the occluded points are marked with black 'x' marks. **b** Schematics of an occlusion situation showing the angle order reversal which implies an occluded object

active points in the Current scan with azimuth angles that bound the Reference point's azimuth value. Note that association is unique for Current scan points, but may not be unique for Reference scan points (this does not pose a problem for the algorithm). We claim a linear complexity of $O(n_R)$ (assuming $n_R \geq n_C$) for the correspondence search, as each point in the Reference Scan is matched only once to a single pair of points, and the k-index does not consider any point in the Current Scan more than once.

2.3 Cost Function

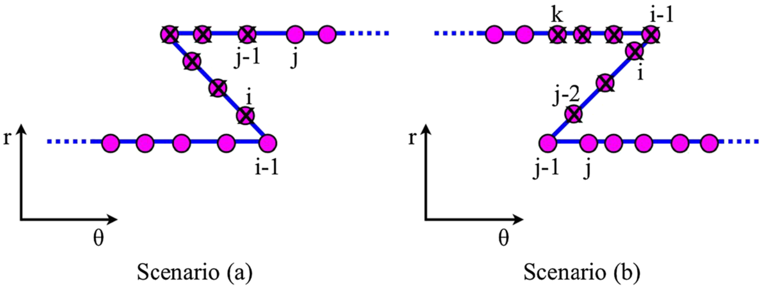
For any given triplet of Δx, Δy, $\Delta\psi$, the cost function to be minimized comprises of the absolute differences in radii between the Reference Scan and the roto-translated Current Scan (interpolated values, see details below). Note that only valid points are considered for contributions to the cost. The cost is then rewarded based on the amount of perimeter overlap achieved between the two scans for the attempted triplet.

2.3.1 Interpolated Radii Values

As described above, each valid Reference scan range value was paired with two range values from the Current scan. The contribution to the cost function is determined using the difference between the Reference scan range value and an interpolated value from the current scan, using both range points, their corresponding angles, and the Reference scan angle. A schematics of the angles and range values involved in a single cost function contribution is shown in Fig. 6.

The cost function calculation for any given pair of scans, is constructed by the following steps (after

Fig. 4 The two possible occlusion scenarios



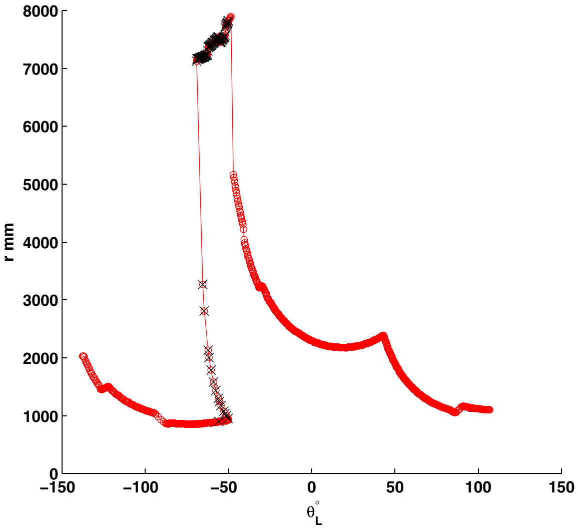


Fig. 5 Example of a case that introduces object occlusion. Range readings (*circles*), plotted against their respective angle, with the occluded points marked with an ‘x’

employing the range limits filter, and the Outlier filter at the beginning):

i. Roto-translate the Current Scan with proposed motion. The roto-translation equations are given in (1):

$$\begin{aligned} x'_C &= \Delta x + r \cdot \cos(\theta_C + \Delta\psi) \\ y'_C &= \Delta y + r \cdot \sin(\theta_C + \Delta\psi) \\ r'_C &= \sqrt{x'^2_C + y'^2_C} \\ \theta' &= tan^{-1}(y'_C, x'_C) \end{aligned} \tag{1}$$

The results x'_C and y'_C are the Current Scan point’s roto-translated cartesian coordinates.

ii. Occlusion filter: discard laser points that become occluded by other surfaces after the roto-translation (as detailed above).

iii FOV filter: discard points in both scans that do not fall within the overlapped FOV.

iv. Establish point correspondence (as detailed above).

v. Calculate the interpolated value r''_C using:

$$r''_C = r'_{C_{left}} + \frac{r'_{C_{right}} - r'_{C_{left}}}{\theta'_{C_{right}} - \theta'_{C_{left}}} \left(\theta_R - \theta_{C_{left}}\right). \tag{2}$$

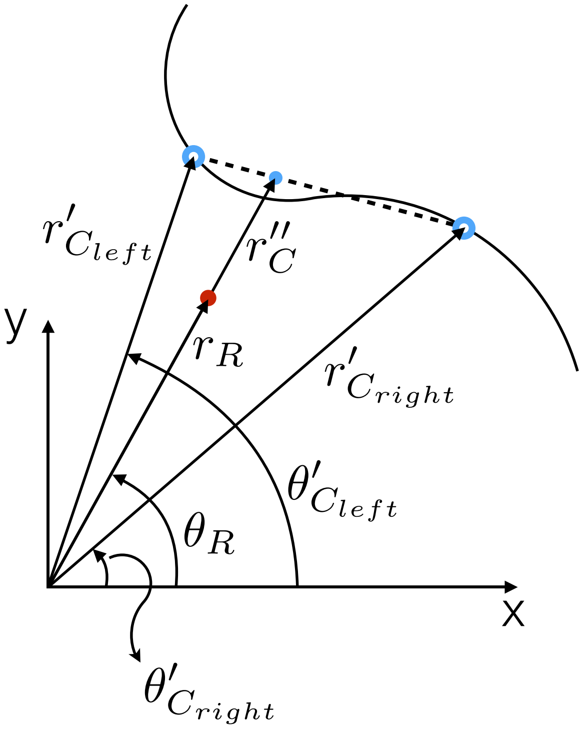


Fig. 6 Schematics for building interpolated radii values for the cost function calculation. The Reference range value r_R is paired with two Current scan range values $r'_{C_{left}}$ and $r'_{C_{right}}$. Using the corresponding angles, the interpolated radii value r''_C is computed. The cost contribution is then $F = |r''_C - r_R|$ (see (3))

vi. Calculate the current point’s cost contribution:

$$F_i = |r''_{C_i} - r_{R_i}|. \tag{3}$$

vii. Discard contributions where $F_i > T_E$, where T_E is the elimination threshold (see details below).

viii. Calculate P - the length of the perimeter created by connecting all the points where $F_i < T_M$ i.e., where T_M is a threshold for successfully matched points (see details below).

ix. Calculate the total cost function:

$$f = \left(\frac{1}{n_c}\sum_{i=1}^{n_c} F_i\right)\left(1 - \frac{P}{P_0}\right). \tag{4}$$

The cost f is normalized by n_c – the number of points that contribute to the total cost. This normalizes the total cost values between different matching attempts. Rewarding the cost by reducing it for matching attempts with larger matched perimeters, results in favoring matchings where the maximum portion of the scenario is considered. Otherwise, small surfaces may sometimes be inadvertently discarded. Other rewarding function may be used, however the linear rewarding function used here was found to be adequate.

Unlike the PSM algorithm, PB-PSM does not make use of least squares for the translation solution. In fact a valid point only contributes the absolute distance between itself and its matched counterpart. This has a relative advantage as some wrongful matches, which may be the result of errors in one of the scans, will not contribute the square of their respective distance from their matched counterpart. Thus, the effect of a singular contribution is less likely to significantly sway the solution.

2.4 Cost Minimization

To find the best scan matching solution, an adaptive direct search method was constructed, where the best rotation angle between the scans is found first, followed by the best pure translation. This process is repeated in an iterative manner while continuously reducing the range of the search grid in a geometric rate after each iteration (thus refining the search grid, in both the plane and azimuth, while the number of grid points is kept constant). In the plane, the search grid shape is in the form of a circle, and the points are evenly distributed along the radial direction and about the azimuthal direction.

The algorithm described above requires a large and fine enough search grid (as described above). In the current work, 50 points for the azimuthal grid were used, and the minimal planar grid was a 7 × 7 grid, so each iteration required 100 function evaluations. It was found that for most scenarios, the above search grid resolution appears sufficient. A mesh refinement scheme may be employed for cases where the resulting cost function is too high. However, this was not required in the current work.

Convergence is defined when the maximum change between two subsequent iterations is less than 1 *mm* in translation and 0.01° in azimuth. On average, it was found that a total of approximately 8 iterations were required for each scan matching process. Cases of scan matching failure are identified by $f > T_F$ where f is the final cost.

An example of a typical scan matching result is given in Fig. 7, where circles represent the Current Scan's laser readings taken from the origin at $[0, 0]$ (all points are shown in order to accurately represent the geometry captured by the laser scanner), while the laser FOV is marked with dashed lines. The Reference Scan is represented by dots, and the matched roto-translated Current Scan points are represented by squares (here, not all points are shown, for clarity). Points that were eliminated by the various filters described above are crossed with 'x' (some points are outside the field of view, some are possible outliers, and some are occluded as the laser picked up points from around the corner while the Reference Scan was taken from the previous location). The current algorithm is implemented in two dimensions and assumes that both planar and rotational platform motions are slow compared to laser scanner speed. The algorithm also assumes a relatively small laser sensor pitch and roll attitudes (ensuring a 2D environment).

2.5 Threshold Settings

The various thresholds used in the algorithm are detailed below, including justifications for the chosen

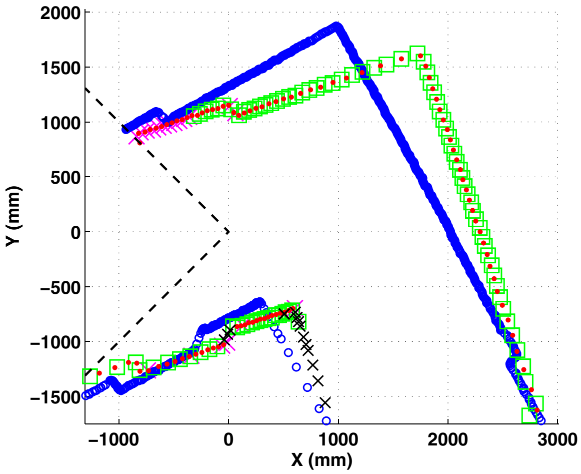


Fig. 7 Scan matching on a simple corner-like geometry. *Circles* - Current Scan points, *squares* - roto-translated laser points, *dots* - Reference Scan points. *Eliminated points* are marked with an 'x' (*small black* - Current Scan points, *large purple* - Reference Scan points). The *dashed lines* shows the laser's 270° field of view

values. Sensitivity of the algorithm to theses values was found to be minimal (as long as the same order of magnitude is maintained).

1. The "matching anomaly" threshold T_E, is a user defined threshold, and is typically set to be two orders of magnitude larger than the typical cell size. This additional point filtering is designed to eliminate the possibility of wrongful correspondence of points from different surfaces. In this work, this number is set to $T_E = 1000\, mm$.
2. Typically, the "matched point" threshold T_M is set to be a number of the same order of magnitude of an occupancy grid cell, and with a certain respect to the laser range accuracy. In the current work, a value of $T_M = 50\, mm$ was used for short range readings, which is increased to the laser's accuracy of approximately 0.5 % for long range readings.
3. The final cost threshold was set at $T_F = 10$. This threshold represents a mean contribution of the same size as a single cell length in the OG (10 mm). This value was found to be a good threshold across multiple experiments.

2.6 Complexity Analysis

We claim that the cost function has $O(n)$ linear complexity, where n is the number of Reference Scan points. The proof may be constructed as follows:

1. The range filter requires a single $O(1)$ examination for each point, which forms an $O(n)$ overall.
2. The FOV filter also requires a single examination for each point (at the most), and so it has an $O(n)$ complexity as well.
3. The outlier filter has $O(n)$ complexity, as each point requires an $O(1)$ calculation and examination of the angle formed with the succeeding point.
4. The occlusion filter was proven above to have an $O(n)$ complexity.
5. The roto-translation has $O(n)$ complexity, which can be seen directly from (1).
6. The process of finding matching candidates has $O(n)$ complexity as explained above. This was also discussed by Diosi and Kleeman [10], and is considered an advantage of scan matching using polar coordinates.
7. Calculating the perimeter is also an $O(1)$ calculation for each participating point, and so it introduces another $O(n)$ to the total cost function complexity.

Hence, building a single cost function requires $O(6n)$ which represents an overall complexity of $O(n)$. Due to the highly effective point filters presented here - there is no need to take the top 80 % of all matches as suggested by Gutmann [18]. This eliminates the need for a sorting process and thus reduces the complexity from $O(nlogn)$ to a linear $O(n)$ complexity.

2.6.1 Computational Requirement

The average time required for a complete scan matching operation on a laptop equipped with a 2.4 GHz processor was approximately 0.3 seconds, which for two scans per second is considered a real time capable SLAM system.

2.7 Map Representation

We use an occupancy grid [13, 33, 39] to store the laser scans data, with a relatively fine resolution of 10 mm by 10 mm. The occupancy grid stores the laser hits, and thus the occupied cells represent the resulting map. Updating the map is a constant time operation as it involves the same number of laser points (N_C) at every step.

2.8 Failure Modes

The failure modes associated with the PB-PSM algorithm are listed below:

1. Convergence to local minima is quite rare with PB-PSM since it's using a form of exhaustive search for the cost minimization. However, the algorithm does depend on the minimization search grid density. After investigating multiple cases, scenarios, and algorithm parameters, it was empirically determined that a 10 × 10 search grid is sufficient for all cases using the PB-PSM algorithm. Reducing the grid resolution under a 7 × 7 grid may result in local minima failure mode, as the cost function behavior is not accurately captured (either for rotation or translation).

2. An inadequate size of the minimization search grid may lead to failure to find the cost minima point. The search grid size is required to be larger than a typical traveled distance between two steps (a similar requirement for the rotation minimization process).
3. As in other algorithms, lack of features in a scene may lead to failure. In particular, obstacles such as long parallel corridor walls, where no features exist in the cross direction may lead to a translation estimation error.
4. Shapeless clusters of point such as those received from scanning bushes or moving objects may also lead to failure in cases where the majority of the scan is consisted of these objects. if only parts of the scan are affected, these points may be eliminated by the T_E threshold detailed above.

3 Algorithm Comparison

We present comparison between the performance of the PB-PSM algorithm, and that of several other algorithms, including PSM, PSM-C, as well as a several different realizations of ICP implementations. The comparison against the various ICP realizations is important as ICP is currently the most common scan matching algorithm [8, 29, 34, 35]. We present examples for matching two individual scenes, and concentrate on the more challenging comparison of entire maps, built using sequential scan matching by different algorithms (without the use of any SLAM algorithm).

3.1 Scene Matching Comparison

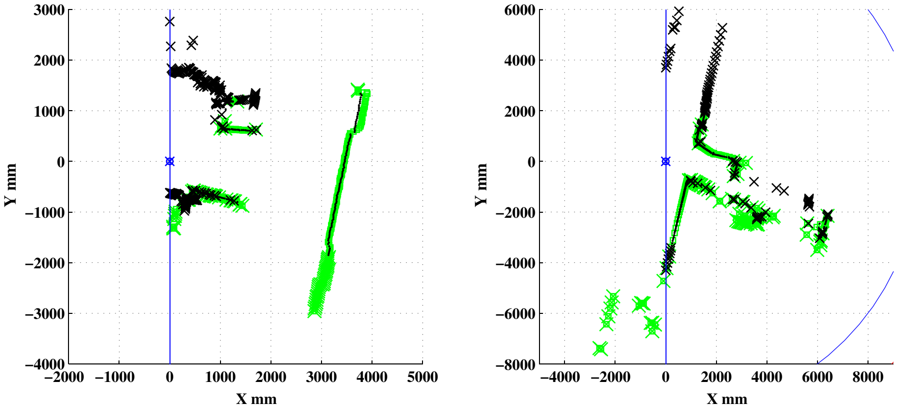
Laser scans from the experiments by Diosi and Kleeman [10] were used for initial benchmarking and testing of the PB-PSM algorithm. The experiments were performed in several scenarios where in each scenario scans were taken from four distinct known locations and points of view using a rigid plastic frame. Some scenarios were mentioned by Diosi and Kleeman to be of higher quality (where all algorithms compared were able to show satisfactory solutions), of which two were chosen for the current comparison: scenes 2 and 7. The most challenging roto-translation scan matching attempted is reported to be match number 3, with the largest total translation distance of 717 mm, and the largest rotation of 27°. The ground truth translation and rotation between the two scans for match number 3 is given by the following triplet: $[\Delta x, \Delta y, \Delta\psi] = [219.4\,mm, 683.3\,mm, -27°]$. Therefore, match number 3 was also selected as the benchmark for the PB-PSM algorithm.

Both scan matched scenes are presented in Fig. 8. Note that all points are presented, and points that are eliminated by one of the point filters are simply crossed out. The scenes require all the point filters discussed earlier, as they contain out-of-range points, occluded points, mixed pixels, and points that become outside of the field of view after the roto-translation. It is quite evident that the scan matching algorithm relies mostly on well defined features such as walls, corners, and objects of significant size (i.e. represented by more than one or two laser point). The relatively shapeless areas with random point clusters, are highly cluttered areas, as stated by Diosi and Kleeman [10].

3.1.1 Convergence Pattern

The highly accurate convergence criterion that was set for the PB-PSM scan matching algorithm requires $|x^n - x^{n-1}| < 1\,mm$, $|y^n - y^{n-1}| < 1\,mm$, and $|\psi^n - \psi^{n-1}| < 0.01°$. The extremely strict ψ convergence ensures better overall estimation since a rotational error causes an immediate map inconsistency, while a translation error only gradually degrades the map. Justification for the relatively tight convergence criterion is shown below.

Figure 9 presents convergence plots for both scenes using the above mentioned convergence criteria. Both scenes converged after approximately 9-12 iterations. The number of iterations is significantly reduced when less challenging scenes are scan matched, and typically the algorithm converges within 5-10 iterations. Both rotation and translation convergence patterns appear quite rapid. The planar convergence process may, in some cases, show a slight overshoot, which is immediately corrected in the following iteration. In all cases, the algorithm recovers from any increase in the error within a single iteration, which is an important characteristic of the algorithm's convergence, and attests for its robustness.



Scene 2

Scene 7

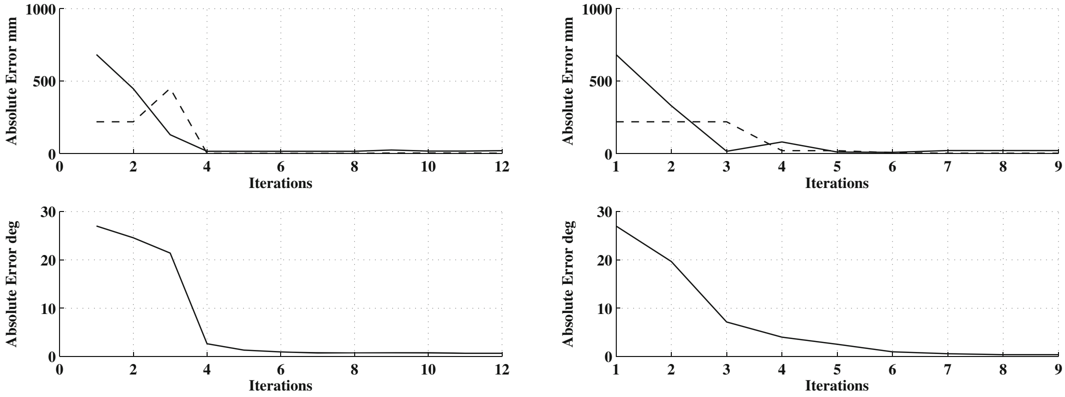
Fig. 8 Two representative scan matching scenes. *Green squares*-Current Scan points (after roto-translation), *black dots*-Reference Scan, 'x' marks-eliminated points. The field of view considered was 180°, with a range of 10 *m* (marked with a *blue line*, where visible)

3.1.2 Estimation Error

Comparison between total translation and rotation error was carried out on the above two scenes between the following algorithms:

1. PSM [10].
2. PSM-C [10].
3. PB-PSM (current study).
4. PB-PSM without the Perimeter Matching term (current study). Since the PM term is excluded, this essentially represents PSM with exhaustive search for minimizing the cost function.
5. ICP [10].
6. ICP (current study, implementation by Bergstrom [5]), with 10 % of the points allowed as outliers). This was done in order to compare another implementation against that used by Diosi and Kleeman [10], shown above.
7. ICP with exhaustive search, rather than least squares. This allows using the perimeter matching term with the ICP algorithm. The cost function minimization process is identical to that of PB-PSM, but the data association follows the closest point metric, rather than the matching bearing metric.
8. ICP with exhaustive search, but without employing the perimeter matching term.

In general, for scene 7, the translation was estimated with similar accuracy of approximately 20 *mm* − 30 *mm* by all algorithms. Rotation estimation was also similar across all methods, showing approximately 0.3° error with the exception of the ICP



Scene 2

Scene 7

Fig. 9 Convergence plots for both scenes. *Upper figures*: Translation error (in *mm*, *solid lines* - 'x' error, *dashed lines* - 'y' error). *Lower figures* - Rotation error (in degrees). Convergence criterion: 1 *mm* for translation (for each direction separately), and 0.01° for rotation

implementation of Diosi and Kleeman [10], which presented with a 0.6° error, and the current study's ICP which performed noticeably worse with a 2.3° error. For scene 2, translation estimation was again generally similar across all methods ($10\ mm - 20\ mm$) except the ICP by Diosi and Kleeman which had twice the translation error. Rotation for scene 2 showed close resemblance between PSM and PB-PSM methods (approximately $0.3° - 0.5°$ error), with all ICP methods having significantly higher rotational errors ($1.4° - 6.1°$). The perimeter matching term had a significant effect on the rotation estimation in scene 2, but was not as significant for scene 7. Its effect is much more pronounced over multiple scan matching operations, spanning longer time periods, as shown below.

3.2 Computational Requirements

Both PB-PSM and PSM (or PSM-C) rely on discrete search for the azimuth estimation. However, for the pose estimation in the plane, PSM relies on solving a least squares problem, while PB-PSM relies on adaptive direct discrete search. Therefore a complete iteration for PB-PSM requires more cost function evaluations than a complete iteration of PSM. Hence, when comparing computational requirements between the two methods, one may consider the total number of matching function calls that are made in both algorithms (on average). The ICP algorithm also has an embedded cost function (data association based on finding the closest point), and so the total number of calls can be used as basis for comparison between all four methods.

According to Diosi and Kleeman [10], the PSM algorithm required 40 matching function calls for the azimuth estimation, and 3 calls for the planar pose estimation, for a total of 43 matching function calls per iteration. The results shown in this paper using PB-PSM were conducted with a search grid of 50 points for the azimuthal grid, and 49 points in the planar grid (a 7 × 7 search grid was used). Therefore, each complete iteration requires 99 matching function calls.

Since the ICP algorithm has a significantly different structure compared to the PSM algorithm, we present computational cost comparison only between the polar scan matching based algorithms. The primary reason for this is the $O(n^2)$ complexity involved in the ICP algorithm cost function, which makes it impossible to compare the number of cost function calls. Diosi and Kleeman [10] report computational time of the order of 5–8 times larger for the ICP as compared with either PSM or PSM-C.

The total number of matching function calls for all 3 scan matching methods can be observed in Table 1 (data taken from Diosi and Kleeman [10]). The PB-PSM algorithm is shown to be approximately 3 times slower as compared to the PSM algorithm, and approximately 4 times slower as compared to the PSM-C. However, as shown above, although PB-PSM is slower, it compensates with a much improved performance, in terms of accuracy.

3.3 Map Based Evaluations

Comparing scan matching algorithms based on individual cases as presented above may provide useful information. However, a given algorithm's overall performance may vary substantially between different scan scenes, different scenarios, and different laser sensors (varying number of points, laser sensor noise, FOV, etc.). For this reason, we present quantitative as well as qualitative comparison of maps created by laser odometry, using several algorithms employed on different scenes. The maps are comprised of matching subsequent laser scans, and therefore present a more challenging objective for performance evaluation.

A quantitative as well as qualitative comparison is brought herein, using a unique metric described below. Note that in order to allow proper comparison, the planar search grid for the function minimization was refined to allow proper convergence for all the different algorithms that use a search grid cost minimization (grid resolutions of 20 × 20 for the planar search). This is important as a large error in one scan matching solution immediately causes a catastrophic shift in the map, and subsequently the map's quality decreases significantly.

Table 1 Number of cost function calls made by each algorithm

Scene	2	7
PB-PSM	1485	1089
PSM	430	387
PSM-C	239	343

Important notes on the presented results:

- All presented mapping results were obtained by scan matching alone. No SLAM algorithm was used in this work. Results using SLAM methods are presented from previous work for comparison purposes.
- Scan matching using PB-PSM was not given an initial guess. The matches are always between the previous position and the current scan position. This further supports the robustness claimed for PB-PSM.
- The PB-PSM may be used for loop closure purposes as well. Naturally, the search area for the cost function minimization process should be made large enough to capture the loop closure opportunity.

3.3.1 Current Experimental Setup

For this part, we used a 2D laser scanner made by Hokuyo [21], with a maximum usable range of approximately 30 m, and azimuthal field of view of 270°. The angular resolution is 0.25°, which produces 1081 laser measurements per scan with a maximum scan frequency of 40 Hz.

Several tests were carried out to establish the capabilities of the Hokuyo laser sensor, including distance-accuracy across the detection range, effects of surface inclination, and the types of outliers that may appear (mixed pixels, low reflectivity surface readings, and out of range readings). Distance accuracy was found to be below 1 % of the measured distance, at the worst case. The laser had no problem to accurately measure distance to surfaces at inclination angles of up to 60° relative to the measuring beam.

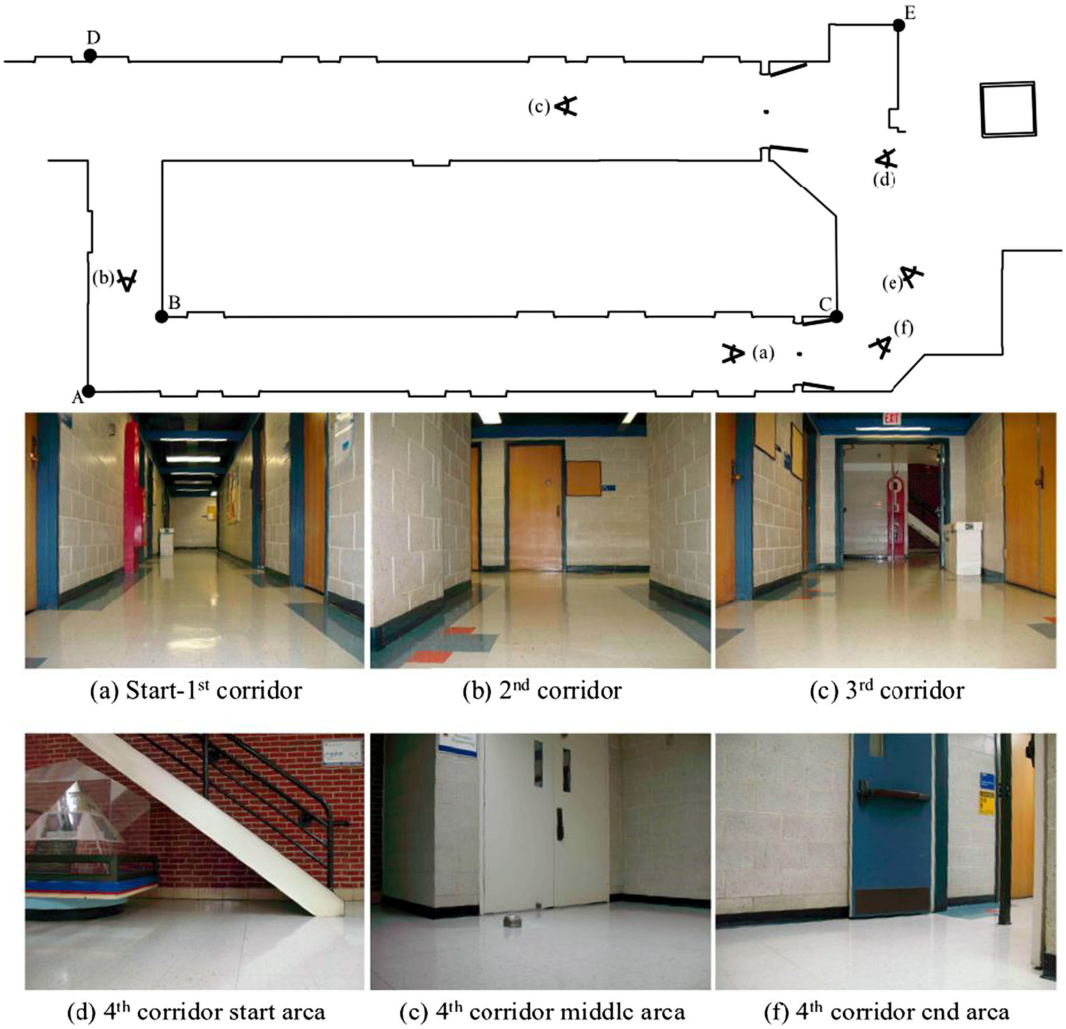
The laser was mounted on a wheeled cart (approximately 50 cm by 40 cm in length and width, and 55 cm in height), which was manually driven through the scenario while the laser records scans at given time intervals (scan rate). Note that the laser in this case was kept in motion while taking the scans, so the presented algorithm is examined for application on a moving platform.

For this metric benchmarking, a section of the 3^{rd} floor in Martin Hall, at the University of Maryland was hand measured in detail, with high accuracy (any feature that is larger than 1 cm was mapped). The environment is presented in a sequence of pictures presented in Fig. 10, along with its 2D layout (the hand measured map), also showing the locations from which the pictures were taken (marked by lower case letters), and five representative points (marked by dots and capital letters) that will later be used for comparing selected measurements. The course includes corridors of different width, with several doorsteps, rectangular trash cans, two thin poles, and several access doors; some were kept closed (Fig. 10e), and some were kept wide open at some angle to the surrounding walls (Fig. 10f). The overall hand measured map accuracy is estimated to be approximately 2 cm. The hand measured mapping accuracy is estimated based on the largest missing gap when drawing the map from two directions (clockwise and counter clockwise). The hand measured map is represented by over 350 straight segments, while the map from the scan matching process is represented by an occupancy grid. This accuracy metric is explained below in detail.

3.3.2 Proposed Metric

Most large environments prohibit the setup of a motion capture system, while a quantitative comparison against a true map is still highly desired. Quantitative map comparison across different algorithms is therefore achieved using a newly developed metric, providing a single number that represents a given map's match to the true measured map (hence, a true measured scenario map is required). Using a metric also allows to evaluate individual effects of the algorithm's ingredients. In addition, we compare several known measured distances between key points in the map.

For the calculation of the proposed metric, we match each occupied cell with a segment from the hand measured map, by calculating the distance from its center to the matched wall (see point p_1 in Fig. 11). The distance d_1 in Fig. 11 is calculated for each segment where the cell center coordinates fall between the segment's end point coordinates in either the 'x' or the 'y' axes. The matched segment is that which yields the smallest d_1 value, which becomes the point p_1's contribution to the map cost. For occupied cells that fall outside of all segments' boundaries we calculate the distance to the nearest corner (for example p_2 in Fig. 11). The distance is weighted by the cell occupancy level (normalized to unity),



(a) Start-1st corridor (b) 2nd corridor (c) 3rd corridor

(d) 4th corridor start arca (c) 4th corridor middlc arca (f) 4th corridor cnd arca

Fig. 10 Martin Hall environment layout. Points of view for all pictures are marked with their respective letters (lower case, in parentheses). Selected points are marked with *dots* and respective *capital letters*, to be used later for measurements comparison

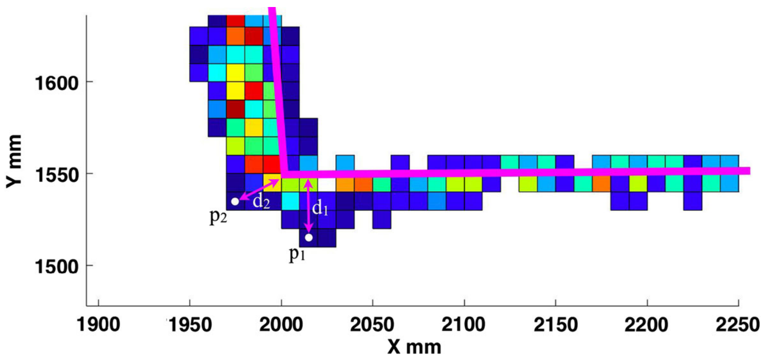


Fig. 11 Proposed map matching metric schematics. Occupied cells are *color coded* by occupancy level (*red-high*, *blue-low*), and the *two purple lines* represent two segments of the hand measured true map. Representative cell centers are shown as white points p_1 and p_2, with their respective metric distance contributions d_1 and d_2, shown in *purple arrows*

and all contributions are then added to a total cost, given by (5):

$$C_W = \left(\frac{1}{n_o} \sum_{i=1}^{n_o} D_i W_i \right) \tag{5}$$

where C_W is the weighted cost, n_o is the total number of occupied cells, D_i and W_i are the distance and occupancy weight of the i^{th} cell, respectively. Note that we later present the value without the cell's occupancy weight, for it to represent the average distance of all occupied cells from their respective associated walls. The minimizing process is done via exhaustive search in x, y, and ψ, to assure a global minimum is achieved.

Important notes about this metric:

1. This metric evaluates scan matching algorithms over multiple scenes. This improves the overall algorithm evaluation as multiple scenes are less prone to occasional success or failure. Algorithm evaluation based on maps instead of individual scenes poses a more stringent test for the algorithm's capabilities. In this way, many different scenes are included with different obstacle surfaces, shapes, and possible failure modes are tested.
2. The map metric score is prone to producing a significantly high score (low quality map) in cases of single large errors which may occur in isolated cases of algorithm failure. This requires minimal post-scoring naked eye examination of the evaluated map to ensure the high score is consistent with multiple errors, rather than the result of a single error without which the map would have been highly accurate.

3.3.3 Effect of Perimeter Matching Term

Evaluation of the perimeter matching term effect was carried out using 12 similar experiments, conducted in the scenario presented in Fig. 10. In each experiment, a total of 100 laser scans were collected over approximately 50 m of traveled path starting at viewpoint (a) and traveling in a clockwise loop until reaching the area of viewpoint (a) again (approximately). Scan frequency was 2 Hz (i.e. 2 scans recored from the laser scanner per second), which implies an average velocity of 1 m/s.

Since the overall error over a complete traveled path is cumulative by nature, evaluating an algorithm over a series of interdependent scan matching scenes is considered to be far more challenging and rigorous, than evaluations based on isolated scenes. A large set of scenes contains a far greater variety of different shaped objects, scanned from multiple different viewing angles and ranges.

The PB-PSM algorithm was employed with and without the perimeter matching rewarding term, on subsequent laser scans, performing laser odometry for all 12 experiments, while maintaing all the other parameters unchanged. No SLAM algorithm is used, no information is extracted from the evolving map, and subsequently no loop closure algorithm is employed on the resulting map, neither during, nor after the path was completed. Hence, the resulting map accuracy is solely the outcome of the scan matching algorithm's accuracy.

The 12 resulting maps were evaluated using the proposed new metric, and the root mean square value of all 12 evaluations was calculated. The results show that using the perimeter matching term reduced the map cost from $C_{(no\ PM)} = 49.2\ mm$ to $C_{(with\ PM)} = 41.3\ mm$, which represents an improvement of 16 %, and almost a whole centimeter of average distance from the wall. Since the proposed metric accounts for tens of thousands of occupied cells (in this conducted experiment), this improvement is considered to be very significant. This experiment was repeated with coarser exhaustive search grids, where the perimeter matching rewarding term was found to be even more effective.

3.3.4 Length Measurements Comparison

In addition to comparing maps using the proposed metric, we compared several distance measurements between selected points (marked by capital letters in Fig. 10). The effectiveness of the perimeter matching term is shown using length comparisons from the same 12 identical experiments, with and without the rewarding term, for segment lengths: $|BC|$, $|AD|$, $|AE|$, $|BE|$, $|DE|$, and $|CD|$ (see Fig. 10). Segment lengths were manually extracted from the hand measured maps and compared against the equivalent measurements taken from the resulting occupancy grids.

Figure 12 presents the root mean square value of the error value in all 12 experiments, for each length,

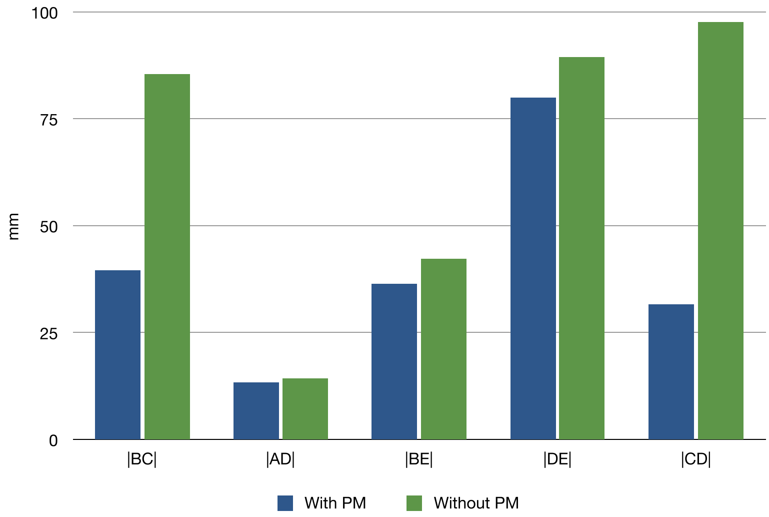


Fig. 12 Root mean square of the error for several segment lengths in 12 identical experiments (comparing true measured lengths with lengths that were manually extracted from the occupancy grids). *Bars* on the left are the results when using the perimeter matching term, showing significant advantage as compared to not using the PM rewarding term

comparing the results with and without the perimeter matching term. Using the perimeter matching term clearly shows a significant advantage for all the measured distances (only the $|AD|$ segment shows a minor effect, as its a measurement of a relatively short corridor length and thus it is less affected).

3.3.5 Effect of Convergence Criterion

The effect of the convergence criterion was examined by employing the map metric described above on a set of maps created with varying convergence requirements (using laser odometry only). Figure 13 shows the resulting map cost with the translation convergence criterion varied from 1 mm, to 50 mm, and the rotation convergence varied from 0.01°, to 0.5°. In this case, the experiment used contained 200 laser scans along the trajectory. The map cost, which represents the average distance of all occupied cells from their associated walls grows significantly, as the convergence criterion is loosened, showing the importance of using relatively tight convergence requirements. The criterions are both gradually increased, as a tight requirement for one criterion may compensate for a relatively loose criterion for the other.

3.3.6 Comparing to Other Algorithms

The map cost metric also allows for quantitative comparison of scan matching algorithm performance. The experiment chosen had 200 scans along the same traveled path, and was successfully completed without any scan matching failures by all algorithms (and their derivatives). Scan frequency was maintained on 2 Hz, and so the velocity was approximately 0.5 m/s, which is half the speed of the previously presented 12

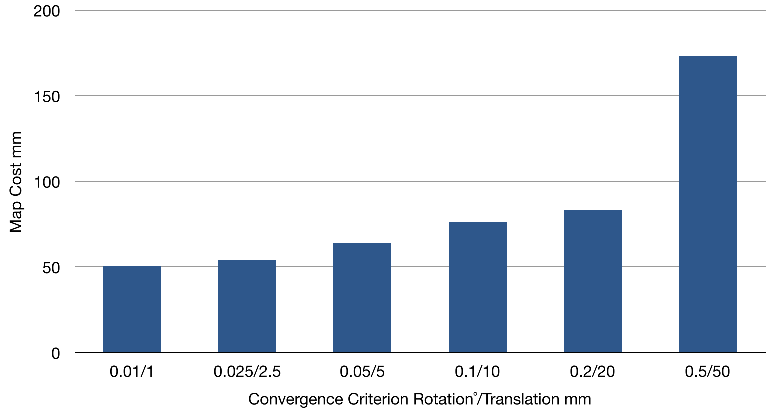
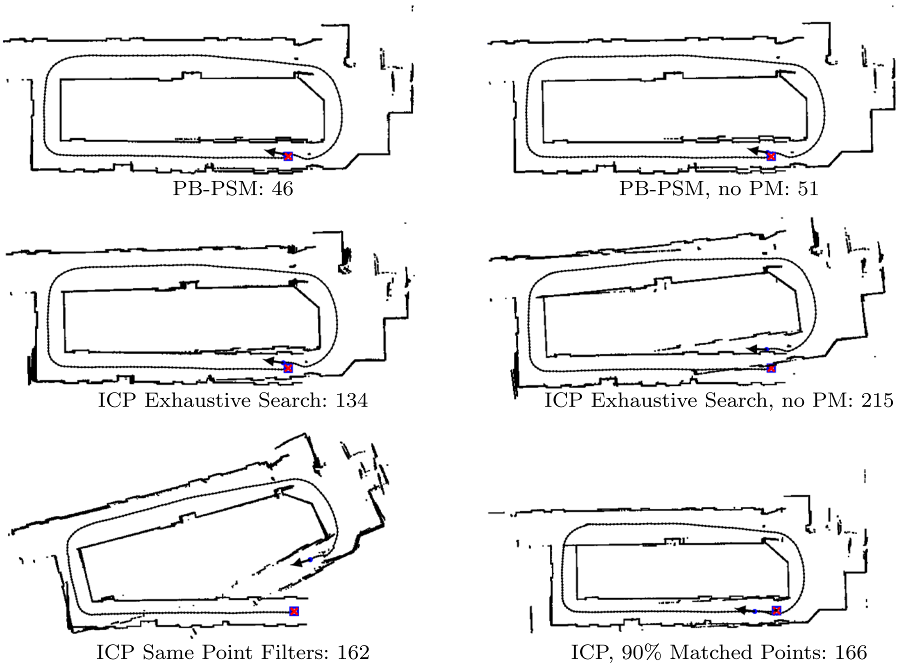


Fig. 13 Effect of convergence criterions on the final map cost. The rotation convergence criterion is presented in degrees (on the *left*), while translation convergence criterion is presented in millimeters



PB-PSM: 46

PB-PSM, no PM: 51

ICP Exhaustive Search: 134

ICP Exhaustive Search, no PM: 215

ICP Same Point Filters: 162

ICP, 90% Matched Points: 166

Fig. 14 Maps and traveled paths of the comparison scenario, created by different algorithms. The scenario is comprised of 200 scans. Map scores show the advantage in using the perimeter matching term, and relative advantage over ICP methods

experiments. The higher number of scans (and respectively slower velocity) results in higher similarities between the subsequent scans.

Figure 14 presents six maps, employing different algorithms on this dataset. All the maps were evaluated using the proposed map metric, showing the final value next to each map. PB-PSM was employed with and without the perimeter matching term, yielding map costs of 46, and 51, respectively, showing again the advantage of using it to reward the cost function.

The scan matching cost function detailed above, was also calculated using the "closest point" rule for data association (rather than PB-PSM's "matching bearing" rule), making it a derivative of the classic ICP algorithm. This allowed for using the perimeter matching term, while using the same algorithm for the function minimization by the adaptive direct search. The resulting maps with and without employing the perimeter matching term are presented in Fig. 14 with map scores of 134 and 215, respectively, showing the advantage of using the rewarding term in this case as well. Note that since the "closest point" data association rule has a higher complexity than $O(n)$, the realization in this case is considerably slower than PB-PSM, and therefore did not allow real time capability.

A comparison is also made against a classic implementation of the ICP algorithm, where the solution is obtained using least squares. Two options were tested, the first was using the same point filters as in described for PB-PSM, and the second was that the ICP will consider only the best 90 % of the matched points for the least squares solution (similar to the practice in Diosi and Kleeman [10]). The second option allows the ICP algorithm to eliminate outliers based on their matching contribution. The resulting maps, presented in Fig. 14, yielded a map score of 162 when using the same filters as in PB-PSM, and 166 when using 90 % of the best matched points, both well above the score achieved by PB-PSM. Although certain parts of the maps obtained by the ICP algorithms appear accurate, a close examination reveals several failure points and multiple relatively small errors which accumulated to account for this relatively poor score.

Figure 14 also allows a basic qualitative assessment of the resulting maps, showing that the map created by PB-PSM appears to be the best result. Moreover, the use of the perimeter matching cost rewarding term in either of the two data association rules, is shown to significantly benefit the map quality. Lastly, the two results using both ICP algorithm variants result in relatively poorer maps, as compared to PB-PSM.

3.3.7 Qualitative Map Comparison Using Existing Datasets

An additional qualitative comparison using laser odometry is presented using Diosi and Kleeman's [10] results for mapping the first room from their dataset, using only their PSM algorithm without their SLAM algorithm (i.e. laser odometry only, no loop closure employed). Figure 15a shows their results, where they conclude that all scan matching based laser

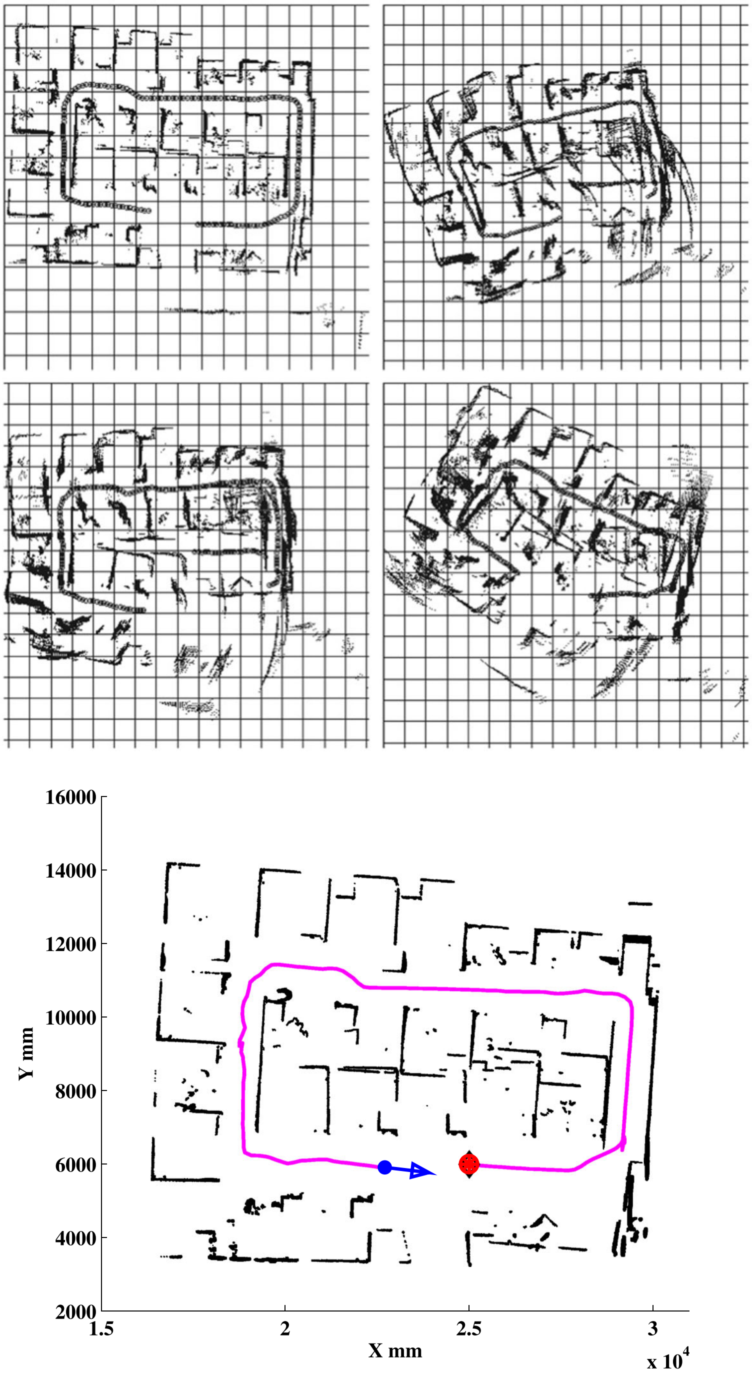
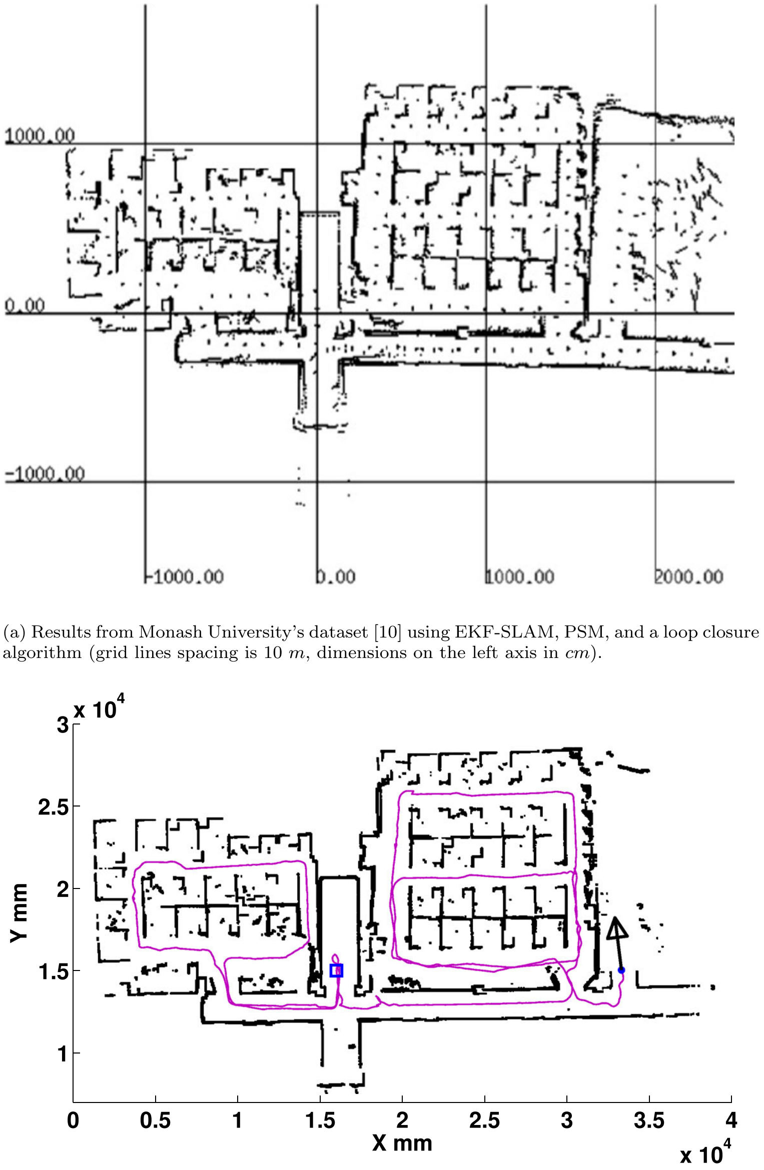


Fig. 15 Diosi and Keelam's dataset, first mapped room. All scan matching algorithms were applied for laser odometry (no SLAM algorithm involved). Comparing results for sequential laser-to-laser scan matching. *Top left*: Odometry only, *Top right*: PSM, *Bottom left*: PSM-C, *Bottom right*: ICP (Results by Diosi and Kleeman). *Bottom*: PB-PSM. Start point is marked by a *large asterisk* and *a red circle*, end point is marked by a *smaller blue* point, with an arrow pointing towards the final azimuth

odometry techniques were outperformed by simply using the wheel encoders odometry for aligning sequential scans.

However, when aligning the scans using results from PB-PSM, the map shown in Fig. 15b appears to be better than both Diosi and Kleeman's laser odometry maps and wheel encoder based odometry map. Some of the walls appear to be more crisp, which implies a more accurate match (less spread of the laser measurements). This demonstrates the higher-accuracy that can be achieved using PB-PSM as compared with the other three methods presented by Diosi and Kleeman, or raw wheel odometry alone.

Fig. 16 Scenario exploration by Diosi and Kleeman



(a) Results from Monash University's dataset [10] using EKF-SLAM, PSM, and a loop closure algorithm (grid lines spacing is 10 m, dimensions on the left axis in cm).

(b) Results using PB-PSM scan matching only (units in mm). Start point is marked by a large blue square, end point is marked by a blue dot, with the final heading arrow in black. The traversed path is drawn as a purple solid line (dimensions in mm). Algorithm fails in the 3^{rd} room due to sparse laser scans with little information available for scan matching. Map generated with PB-PSM appear crisp, with more straight corridors and fewer "double walls" occurrences.

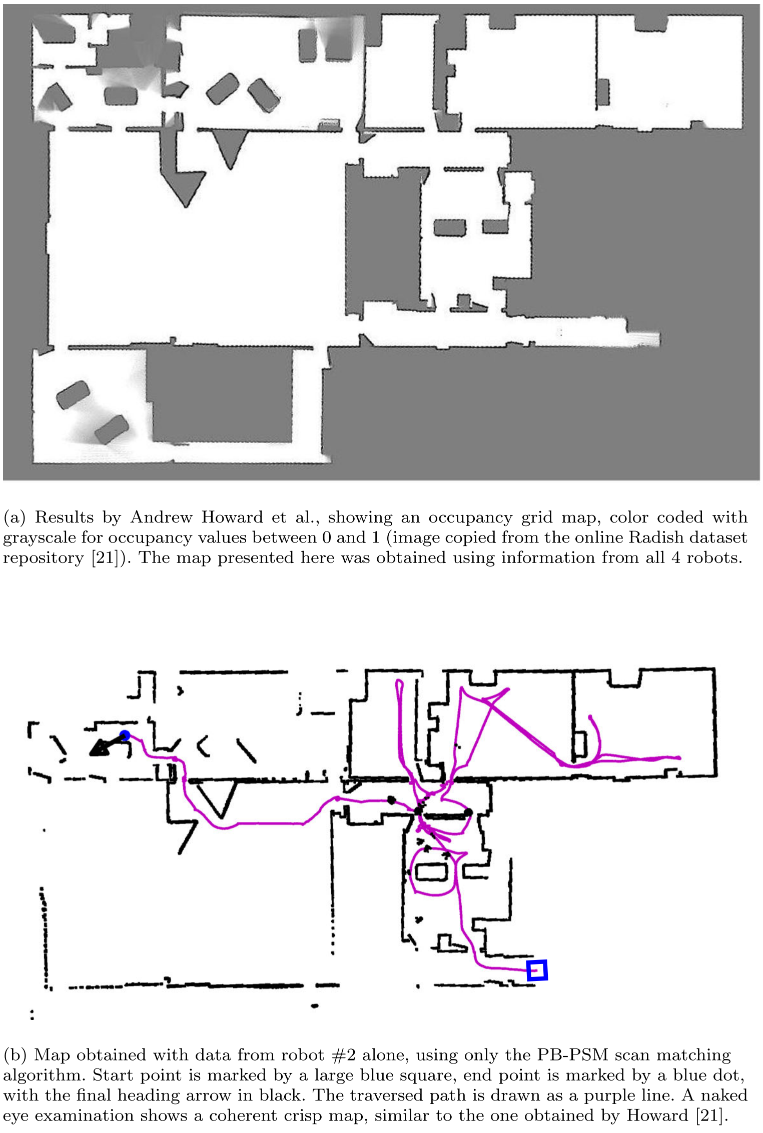
4 Additional Results Using Virtual Scans

The performance of the algorithm can be further improved by scan matching new incoming laser scans against scans of the evolving occupancy grid, called "Virtual Scans". This method improves the accuracy, as new laser scans are matched with a scan that contains information from all the previously collected scans (stored in the occupancy grid map thus far). The virtual scan is performed by means of ray casting operations. Additional technical information about how virtual scans are preformed can be found in the works by Friedman et al. [15–17] and Friedman [14]. Results are presented for previously published experimental data by both Diosi and Kleeman [10], and Howard et al. [23]. These results further support the capabilities of the PB-PSM scan matching algorithm.

4.1 Monash University Database

The algorithm was employed on a dataset collected by Diosi and Kleeman [10], and the results are presented in Fig. 16. The experiment was conducted in an office environment, using a ground platform and a 2D laser range scanner producing 361 points with a FOV of 180°, recording data at 30 Hz. This dataset

Fig. 17 Scenario exploration. Andrew Howard's database



(a) Results by Andrew Howard et al., showing an occupancy grid map, color coded with grayscale for occupancy values between 0 and 1 (image copied from the online Radish dataset repository [21]). The map presented here was obtained using information from all 4 robots.

(b) Map obtained with data from robot #2 alone, using only the PB-PSM scan matching algorithm. Start point is marked by a large blue square, end point is marked by a blue dot, with the final heading arrow in black. The traversed path is drawn as a purple line. A naked eye examination shows a coherent crisp map, similar to the one obtained by Howard [21].

is described by Diosi and Kleeman to be challenging since the laser sensor picked up more than 10 people walking in its FOV and several doors were opened and closed while traversing environment.

The results show a relatively crisp map where the corridors appear to be straighter with significantly fewer occurrences of "double walls" (examined by naked eye). However, the algorithm failed without recovery when the 3$^{\text{rd}}$ room was reached as the laser scans in that room were quite sparse with very few points that can be successfully used for scan matching. Although the results by Diosi and Kleeman also show very poor mapping quality in the 3rd room, their use of a full SLAM algorithm and wheel encoders allowed further mapping of the environment (with some bending to the walls).

4.2 University of Southern California Database

The second dataset was collected by Andrew Howard [22] using four autonomous robots individually exploring a closed area (with some human supervision). Several moving objects (people and robots) were captured in the dataset, thus rendering it challenging to obtain good scan matching results. Additional details about the experiment and the algorithms used to obtain the dataset and the map can be found in the work by Howard et al. [23]. In this dataset, a laser sensor with 180 points was used, with a FOV of 180° (significantly less than the 1081 points per scan and 270° FOV in the authors' datasets).

The map obtained by Andrew Howard (fusing data from all four robots) is compared with that obtained by PB-PSM in Fig. 17 (using data from robot #2 only). The mapping results appear to be in excellent agreement, judging by a naked eye comparison. The course traveled by robot #2 is quite complex with multiple sharp turns, complete 180° turns and occasions where the robot revisits previously mapped areas. Nevertheless, PB-PSM was able to construct an accurate and crisp map even without the use of any loop closure algorithms.

5 Conclusions and Future Work

This work proposes PB-PSM, an algorithm for scan matching 2D laser scans. The algorithm exploits the polar coordinate nature of a set of laser range measurement, as they are output from a 2D laser range scanner. The algorithm is shown to perform well with relatively high accuracy, using various laser sensors, and different environments.

Several key factors are shown to contribute to the performance of the PB-PSM algorithm:

1. Perimeter matching cost-rewarding term, favoring matches with higher overlap between the two scans.
2. Adaptive direct search for the cost function minimization process, which overcomes the common phenomenon of local minima.
3. Tight convergence requirements of 0.01° for rotation and 1 mm for translation, which are shown to have a strong effect on the overall performance.

Performance repeatability and robustness are shown through 12 identical experiments conducted at the University of Maryland, and mapping accuracy is analyzed by employing root mean square values of the errors obtained in all 12 experiments. The comparison metrics included a newly developed metric which ranks occupancy grid maps similarity to the true map of a benchmark scenario, as well as measured errors in multiple distance measurements across that benchmark scenario. The new metric ranks occupancy grid maps based on calculating the average cell distance from its associated true map wall. This metric may easily be used by others to express the mapping accuracy of other algorithms.

When applied to sequential laser scan matching (laser odometry), PB-PSM is shown to yield significantly improved results, as compared to PSM, PSM-C, and several realizations of the ICP algorithm. This was shown across both in-house and previously published datasets.

Although slower as compared with PSM and PSM-C, the algorithm is computationally lightweight enough to be executed in real time on modern onboard hardware. The scan matching process was employed on large offsets between the matched scans (over 0.5 m). In cases where a better initial guess is available, it may be easily used to reduce the cost minimization search grid area, which in turn will greatly reduce computational time.

Acknowledgments This work was supported by the Israeli Ministry of Defense [grant number GOI-MOD 1000118123].

References

1. Achtelik, M., Bachrach, A., He, R., Prentice, S., Roy, N.: Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments. In IARC First Symposium on Indoor Flight Issues, July 21st. University of Puerto Rico, Mayagüez, Puerto Rico (2009)
2. Bailey, T., Nebot, E.: Localisation in large-scale environments. Robot. Auton. Syst. **37**, 261–281 (2001)
3. Bailey, T., Nieto, J., Guivant, J., Stevens, M., Nebot, E.: Consistency of the EKF- SLAM algorithm. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3562–3568. Beijing (2006)
4. Bailey, T., Nieto, J., Nebot, E.: Consistency of the FastSLAM algorithm. In: IEEE International Conference on Intelligent Robotics and Automation, pp. 424–427. Beijing (2006)
5. Bergström, P.: ICP implementation for Matlab. http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=12627&objectType=FILE (2007)
6. Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)
7. Brenneke, C., Wagner, B.: A scan based navigation system for autonomous operation of mobile robots in man-made environments. In: International Conference of systems engineering (ICSE). Coventry (2003)
8. Censi, A.: An ICP variant using a point-to-line metric. In: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on Robotics and Automation (ICRA), pp. 19–25. Pasadena (2008)
9. Cox, I.J.: Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. IEEE Trans. Robot. Autom. **7**(2), 193–204 (1991)
10. Diosi, A., Kleeman, L.: Fast laser scan matching using polar coordinates. Int. J. Robot. Res. **26**(10), 1125–1153 (2007)
11. Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. IEEE Trans. Robot. Autom. **17**(3), 229–241 (2001)
12. Durrant-Whyte, H., Bailey, T.: Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. IEEE Robot. Autom. Mag. **13**(2), 99–110 (2006)
13. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. Computer **22**(6), 46–57 (1989)
14. Friedman, C.: Accurate slam with application for aerial path planning, Ph.D. thesis, Clark School of Engineering (2013)
15. Friedman, C., Chopra, I., Potyagaylo, S., Rand, O., Kanza, Y.: Towards model-free SLAM using a single laser range scanner for helicopter MAV. In: AIAA Guidance Navigation and Controls Conference. Portland (2011)
16. Friedman, C., Chopra, I., Potyagaylo, S., Rand, O., Kanza, Y.: Towards obstacle avoidance and autonomous uav operation. In: AHS Specialists' Meeting on Unmanned Rotorcraft and Network Centric Operations. Tempe, AZ (2011)
17. Friedman, C., Chopra, I., Rand, O.: Highly accurate simultaneous localization and mapping for rotary wing UAVs. In: American Helicopter Society 68th Annual Forum. Fort Worth (2012)
18. Gutmann, J.S.: Robuste navigation autonomer mobiler systeme. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg (2000)
19. Gutmann, J.S., Schlegel, C.: Amos: Comparison of scan matching approaches for self-localization in indoor environments. In: Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots, pp. 61–67. Kaiserslautern (1996)
20. Hähnel, D., Burgard, W., Fox, D., Thrun, S.: An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Las Vegas (2003)
21. Hokuyo: UTM-30LX. Tech. rep. http://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html (2009)
22. Howard, A.: Laser scans dataset collected at fort AP hill, as part of the DARPA/IPTO sdr project. Tech. rep. http://cres.usc.edu/radishrepository/view-one.php?name=ap_hill_07b (2004)
23. Howard, A., Parker, L.E., Sukhatme, G.S.: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. Int. J. Robot. Res. **25**(5–6), 431–447 (2006)
24. Lu, F., Milios, E.: Robot pose estimation in unknown environments by matching 2D range scans. J. Intell. Robot. Syst. **18**(3), 249–275 (1997)
25. Nguyen, V., Harati, A., Siegwart, R.: A lightweight SLAM algorithm using orthogonal planes for indoor mobile robotics. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007, 29 October - 2 November. San Diego (2007)
26. Núñez, P., Vázquez-Martin, R., Bandera, A., Sandoval, F.: Fast laser scan matching approach based on adaptive curvature estimation for mobile robots. Robotica **27**, 469–479 (2009)
27. Nüchter, A., Surmann, H., Lingemann, K., Hertzberg, J., Thrun, S.: 6D SLAM with an application in autonomous mine mapping. Robot. Autom. **2**, 1998–2003 (2004)
28. Okubo, Y., Ye, C., Borenstein, J.: Characterization of the hokuyo URG-04LX laser rangefinder for mobile robot obstacle negotiation. SPIE Defense, Security + Sensing, Unmanned Systems Technology XI (2010)
29. Olson, E.B.: Real-time correlative scan matching. In: IEEE International Conference on Robotics and Automation, 2009. ICRA '09 (2009)
30. Park, S., Park, S.K.: Spectral scan matching and its application to global localization for mobile robots. In: IEEE International Conference on Robotics and Automation Anchorage Convention District. Anchorage (2010)
31. Röfer, T.: Using histogram correlation to create consistent laser scan maps. In: Proceedings of the IEEE International Conference on Robotics Systems (IROS-2002), pp. 625–630. EPFL, Luasanne, Switzerland (2002)
32. Saarinen, J., Mazl, R., Kulich, M., Suomela, J., Preucil, L., Halme, A.: Methods for personal localisation and mapping. In: 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles Instituto Superior Tecnico. Lisboa (2004)
33. Schiele, B., Crowley, J.L.: A comparison of position estimation techniques using occupancy grids. Robot. Auton. Syst. **12**(3–4), 163–161 (1994)

34. Segal, A., Hähnel, D., Thrun, S.: Generalized-ICP. In: Proceedings of Robotics: Science and Systems (RSS) (2009)
35. Shen, S., Michael, N., Kumar, V.: Autonomous multi-floor indoor navigation with a computationally constrained MAV. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 20–25 (2011)
36. Stachniss, C., Grisetti, G., Hähnel, D., Burgard, W.: Improved rao-blackwellized mapping by adaptive sampling and active loop-closure. In: Proceedings of the Workshop on Self-Organization of AdaptiVE behavior (SOAVE) (2004)
37. Tang, P., Huber, D., Akinci, B.: A comparative analysis of depth-discontinuity and mixed-pixel detection algorithms. In: Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM), pp. 29–38 (2007)
38. Thrun, S.M., Hähnel, D.: Scan alignment and 3-D surface modeling with a helicopter platform. In: The 4th International Conference on Field and Service Robotics (2003)
39. Thrun, S.: Learning occupancy grid maps with forward sensor models. Auton. Robots **15**(2), 111–127 (2003)
40. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. 1. The MIT Press (2005)
41. Tomono, M.: A scan matching method using euclidean invariant signature for global localization and map building. In: ICRA, pp. 886–871. Piscataway (2004)
42. Yuan, X., Zhao, C.X., Tang, Z.M.: Lidar scan-matching for mobile robot localization. Inf. Technol. J. **9**(1), 27–33 (2010)