

The LiDAR Compass: Extremely Lightweight Heading Estimation with Axis Maps

Marc J. Gallant^{a,*}, Joshua A. Marshall^a

^a*Mining Systems Laboratory, Queen's University
25 Union Street, Kingston, Ontario, Canada K7L 3N6*

Abstract

This paper introduces the *LiDAR compass*, a bounded and extremely lightweight heading estimation technique that combines a two-dimensional laser scanner and *axis maps*, which represent the orientations of flat surfaces in the environment. Although suitable for a variety of indoor and outdoor environments, the LiDAR compass is especially useful for embedded and real-time applications requiring low computational overhead. For example, when combined with a sensor that can measure translation (e.g., wheel encoders) the LiDAR compass can be used to yield accurate, lightweight, and very easily implementable localization that requires no prior mapping phase. The utility of using the LiDAR compass as part of a localization algorithm was tested on a widely-available open-source data set, an indoor environment, and a larger-scale outdoor environment. In all cases, it was shown that the growth in heading error was bounded, which significantly reduced the position error to less than 1% of the distance travelled.

Keywords: heading estimation, LiDAR, compass, localization

1. Introduction

Localization of mobile robots or other ground vehicles is an active area of research that has important applications in mapping, planning, and control. In the absence of an absolute positioning system (e.g., GPS), localization is traditionally performed by measuring the internal state of the vehicle with interoceptive sensors (e.g., accelerometers, gyroscopes, wheel encoders), and/or measuring the local environment surrounding the vehicle with exteroceptive sensors (e.g., cameras, LiDAR). Accurately estimating the robot's heading is particularly important when the localization algorithm involves dead reckoning. In this common scenario, the positional components of the motion model are often tightly coupled with the heading component, causing inaccuracies in heading estimation to be quickly propagated into substantial position errors. As a

*Corresponding author

Email addresses: m.gallant@queensu.ca (Marc J. Gallant),
joshua.marshall@queensu.ca (Joshua A. Marshall)

result, accurate heading estimation can be essential for localization. Although sensors exist that can directly or indirectly be used for heading estimation (e.g.,
15 a compass or gyroscope), environmental limitations and/or dead reckoning errors can render these estimates unreliable for many applications. To this end, this paper introduces the *LiDAR compass* (LC), which transforms data from a horizontally-oriented 2D scanning LiDAR into an absolute heading estimate that can be used in a variety of indoor and outdoor environments.

20 The LC uses *axis maps* (AMs), which are a minimal representation of the orientations of surfaces in the environment. To emphasize the fact that the orientation of a flat two-dimensional surface (i.e., a line) is invariant to rotations of $\pm\pi$, it is called the *axis* of the surface. This naming convention is consistent with other types of axial data in directional statistics [1]. Lines are extracted
25 from 2D LiDAR scans and compared to both an *a priori* and local AMs, which provide information about the absolute or relative heading of the robot. As a result, it is assumed that the operating environment contains some approximately straight surfaces. This assumption is met in a great many common scenarios, including both indoor (e.g., walls, furniture) and outdoor (e.g., buildings, cars)
30 environments. In these environments, the LC is essentially a virtual heading sensor that—when combined with a means to measure translation—can be used to aid localization. Using the LC in this way is analogous to augmenting wheel odometry with a compass and gyroscope. Much like how a compass provides an absolute heading reference by measuring the direction of the Earth’s magnetic
35 field, the LC uses the dominant axes of the surfaces in the environment as the absolute reference. And similar to how a gyroscope provides a dead reckoning heading estimate by integrating angular velocity, the LC also provides relative heading estimates by tracking the axes of local surfaces not in the *a priori* AM. The result is an easy-to-implement, very lightweight, virtual heading sensor
40 that (unlike a gyroscope) provides an absolute heading reference, and (unlike a compass) can be used in any environment from which an AM can be derived.

There are several applications where the advantages of using an LC would be especially useful. For example, the state of the art in autonomous mapping is graph-based simultaneous localization and mapping (SLAM), which is formulated as a nonlinear least-squares problem that maximizes the likelihood
45 of sensor measurements. This formulation requires an initial estimate of the robot trajectory, and the quality of this initial estimate can greatly affect the accuracy of the resulting map [2]. The LC can be combined with odometry to provide an easily-implemented solution, while not requiring any additional sensors. For instance, Figure 1 illustrates the trajectories estimated by raw odometry, an
50 LC with odometry, and SLAM [3] for the MIT Killian Court data set [4]. The detailed results using this dataset are provided in Section 5.1.

Another application for the LC is in small, consumer robots, where the embedded computing environment would benefit from a lightweight, real-time localization algorithm (e.g., the Neato Robotics robot vacuums, which are equipped
55 with a 2D LiDAR). These products tend to operate in semi-structured environments from which an AM could be easily derived. Finally, vehicles equipped with an integrated GPS and inertial navigation system (GPS/INS) in urban

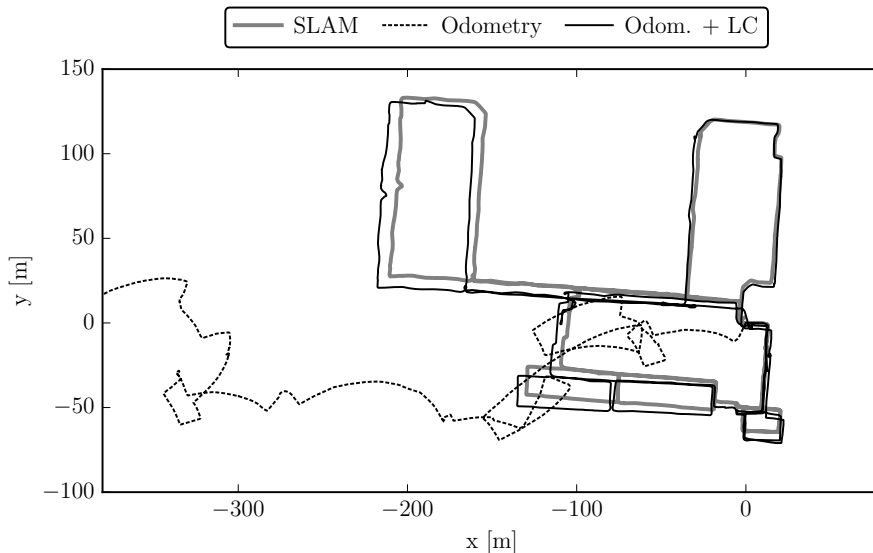


Figure 1: The trajectory of the MIT Killian Court data set as estimated by odometry, odometry with a LiDAR compass (LC), and the full SLAM solution [3]. Much of the odometry-estimated path is outside of the scale of this figure. The LC estimate provides a good initial guess for graph-based SLAM. A major source of error of the LC trajectory is from translational error in the odometry estimate, which may be partially mitigated with proper calibration.

environments rely on dead-reckoning from accelerometers and gyroscopes when
 60 GPS signals are unavailable (e.g., in tunnels or when surrounded by buildings).
 Augmenting such a system with an LC could easily and drastically improve
 localization by using the obstructions themselves to populate the AM.

1.1. Related Work

The LC requires re-occurring, line-extractable surfaces in the environment.
 65 Although its map consists only of the axes of lines extracted from these environments,
 line-based map representations for localization and mapping have considerable heritage
 in mobile robotics research. Before relatively low-cost LiDAR became available,
 sonar was used to observe *a priori* line-based maps of the environment [5]. Here,
 the idea of improving localization by incorporating geometric constraints in the
 70 environment was exploited. One of the first implementations using LiDAR provided
 online localization given an *a priori* line-based map [6], where it was noted that
 many indoor environments are suitable for this type of map representation. An
 early implementation that actually constructed line-based maps [7] predates modern
 SLAM and actually decouples mapping and localization. However, even with a
 75 rudimentary 2D LiDAR, line-based maps of less structured environments such as
 underground mines were shown to be effective representations.

More recent efforts have used line-based maps in SLAM implementations by

employing the position and axis of line segments to update the pose estimate
80 of the robot. One EKF SLAM implementation [8] demonstrated the accuracy
of using line segments in the SLAM state, whose compactness also reduces the
burden of the computational complexity of SLAM. However, as line segments
have four degrees of freedom, effective data association among line segments
requires several correspondence tests. This issue is non-existent in AMs because
85 AM entries have only a single degree of freedom.

The popular line-based Orthogonal SLAM [9] takes advantage of an orthog-
onality assumption of the surfaces in the environment (e.g., the perpendicular
walls common in most indoor areas). This approach was later extended to cre-
ate a lightweight Rao-Blackwellized particle filter Orthogonal SLAM [10]. Here,
90 only orthogonal lines extracted from the environment are used to update the
SLAM state. By fixing the possible axes of lines to an absolute reference (and
discarding lines that do not meet this criterion), remarkable mapping and lo-
calization accuracy is achieved by limiting the growth of heading errors. Recent
work [11] has explored automatically identifying additional types of structure
95 (e.g., point-to-point distances, circles) and incorporating these constraints into
the graph-based optimization of the map. The LC presented in this paper shares
some of the benefits of these structure-sensitive SLAM algorithms with its *a pri-*
ori axis map. Unlike the aforementioned approaches, the LC is intended to use
the structure of the environment for real-time, lightweight, and easily imple-
100 mented heading estimation. It is effective in environments whose surfaces are
not orthogonal or not even in the *a priori* map.

A different approach of heading estimation called a *visual compass* (VC) is
currently an active research topic in mobile robot localization. Although many
implementations of VCs exist, most implementations use a camera to track
105 changes or features in the image frame to infer rotational information. VCs dif-
fer from visual odometry because they usually discard all positional information
in the sensor data, not unlike the LC. A common approach is to unwrap sequen-
tial omnidirectional images and observe how simple extracted features appear
to be displaced as the robot rotates [12, 13]. An overview of this method con-
110 cluded that its dead reckoning heading estimation performed similarly or better
than inertial sensors in appropriate environments [12]. Other forms of VCs do
not necessarily require omnidirectional cameras and instead track the motion of
specific vanishing points at far distances [14, 15], which work well in large, open
environments. Finally, a recent VC implementation addresses the restrictions of
115 many other VCs (e.g., computation, prior environmental knowledge, or calibra-
tion requirements) with an algorithm that tracks circles and lines detected using
edge detection, RANSAC, and Hough transforms [16]. However, this algorithm
was only tested on short traverses (< 30 m) and suffered from problems caused
by changes in illumination or sharp turns. Although the LC shares some of
120 the same principles as VCs, it is far simpler to implement, has environmental
restrictions that are less restrictive (e.g., lighting conditions, closed environ-
ments), and may be more easily used as an absolute heading sensor (due to the
ease of producing AMs).

As mentioned in Section 1, one application of the LC is as an initial guess for

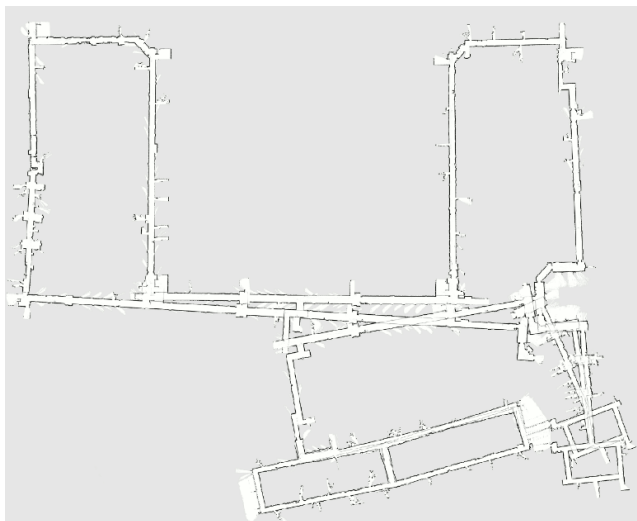


Figure 2: A map of MIT Killian Court produced by the scan matching algorithm described in [18] and extended in [19]. This figure is taken from [17], where the estimated path of the robot that generated this map was used as an initial guess in a comparison of graph-based SLAM algorithms.

125 graph-based SLAM algorithms. When equipped with a two-dimensional laser scanner and (optionally) wheel encoders and/or a gyroscope, the initial guess of the path travelled by the robot is commonly provided by using scan matching [17]. Simply described, scan matching compares overlapped scans (or local maps generated by the scans) between two poses and estimates their relative rotation and translation. Incrementally performing scan matching (where motion sensors are often used to provide a first guess to the scan matching algorithm) is a viable form of localization. A map of MIT Killian Court generated using this type of algorithm is shown in Figure 2. Because scan matching is a form of dead reckoning, the correctness of the map degrades severely as the robot's trajectory lengthens (e.g., its heading estimate has particularly degraded in Figure 2). When compared to the LC path plotted in Figure 1, and considering the simplicity of the LC implementation described in Section 3, using the LC in place of scan matching for the initial guess provided to graph-based SLAM becomes an attractive alternative.

140 *1.2. About This Paper*

The aim of the LC is to provide accurate, lightweight, and absolute heading estimation in any environment that contains line-extractable surfaces. The increasing ubiquity of 2D LiDAR on autonomous vehicles allows the LC to provide vast improvements in heading estimation without the need for an additional sensor and with low computational overhead. This paper extends previous preliminary work on the LC [20] and includes significant improvements, as well as field testing results in a variety of environments. The work is presented in

three main parts. Section 2 provides background information on axis maps, line extraction, and the observation model that joins these two concepts. Section 3
 150 presents the theoretical background of the LC, while Section 4 describes how it may be used in localization. Finally, Section 5 presents field results from three test environments: an established and widely available data set, a challenging indoor environment, and a large outdoor environment.

2. Preliminaries

155 The LC uses an axis-only representation of the environment called an *axis map*; see Section 2.1. Unlike conventional metric or topological maps, axis maps are extremely succinct because they contain no positional information of landmarks, features or surfaces in the environment. Entries of axis maps are observed by extracting lines from 2D LiDAR scans. The implementation
 160 of line extraction used in this paper is outlined in Section 2.2; however, other algorithms—especially those that consider polar-based noise models of laser scanners—could be used with the LC. Finally, a lightweight observation model that generates predicted axes from the entries of an axis map is described in Section 2.3.

165 2.1. Axis Maps

Axis maps (AMs) represent the axes of extractable lines segmented from 2D laser scans of the environment. Lines are extractable if the surfaces generating them are observable by one or more 2D laser scanners. An AM is a set of elements

$$\Phi = \{\phi^{(i)} : \phi^{(i)} \in [0, \pi), i = 1, \dots, N\}, \quad (1)$$

where $\phi^{(i)}$ represent the axes coincident with the normal vectors of lines in the environment. The absence of positional information means that a many-to-one relationship between lines extracted from the environment and axes in the AM is possible. For example, parallel walls in a hallway are represented by one entry
 170 in an AM. This also facilitates mounting the laser scanner on the robot because only its orientation relative to the robot coordinate frame matters.

The LC uses two different AMs: a static *a priori* AM $\bar{\Phi}$ with little or no uncertainty, and a dynamic local AM $\hat{\Phi}$ with estimated axes that are frequently added or removed. The axes of the dominant surfaces in the environment (e.g.,
 175 walls, sides of buildings) are determined beforehand and are added to $\bar{\Phi}$. For example, $\bar{\Phi} = \{0, \frac{\pi}{2}\}$ is likely sufficient for many indoor environments. Although $\bar{\Phi}$ does not need (and should not be) an exhaustive map of all axes, consistently re-occurring surfaces should be added if possible for the best results. For example, $\bar{\Phi} = \{0, \frac{\pi}{2}\}$ can be easily determined beforehand for the environments
 180 in Figure 3. However, it is likely that there will be local areas in these environments where this AM is insufficient. For example, diagonal hallways that do not follow the general layout of a building, or an area populated by objects (e.g., furniture, vehicles) with unknown orientations. As a result, the axes of extracted lines that cannot be associated with the entries of $\bar{\Phi}$ are added to $\hat{\Phi}$.

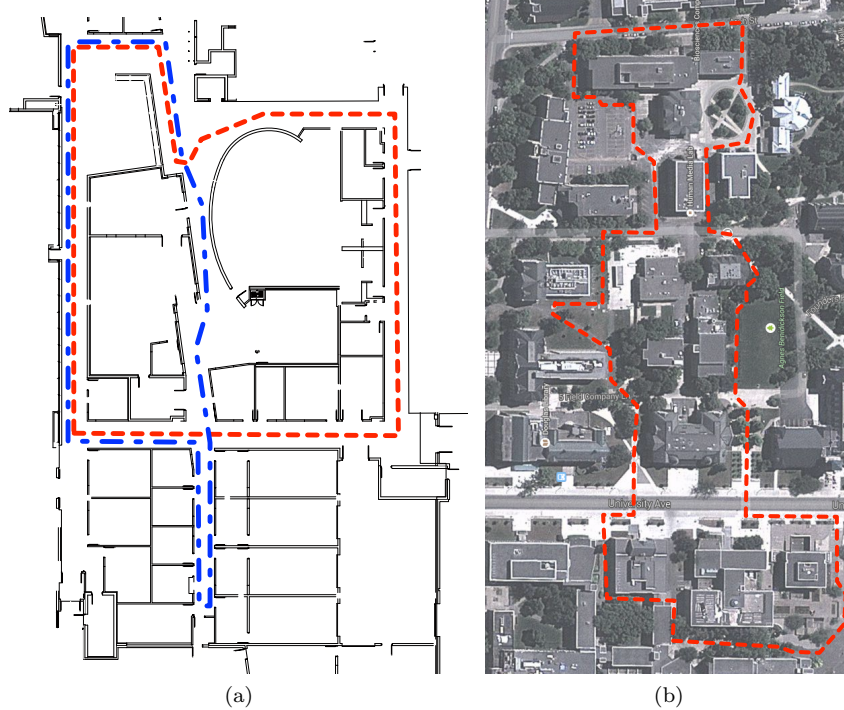


Figure 3: Examples of two different environments from which $\bar{\Phi}$ can be easily extracted. (a) A blueprint of the ground floor of Beamish-Munro Hall at Queen's University (approximately $35 \text{ m} \times 55 \text{ m}$). Despite some non-orthogonal and curved areas, the AM $\bar{\Phi} = \{0, \frac{\pi}{2}\}$ describes the dominant, re-occurring walls. The dashed and dash-dotted paths are the first and second routes used in the experiments described in Section 5.2, respectively. (b) Satellite imagery (©2014 DigitalGlobe) of a section of the Queen's University campus (approximately $200 \text{ m} \times 400 \text{ m}$). The outer walls of most of the buildings share the same axes. The AM $\bar{\Phi} = \{0, \frac{\pi}{2}\}$ would also be appropriate for this environment. The dashed path is the route used in the experiments described in Section 5.3.

185 The estimated mean and uncertainty of these locally added axes are updated as they continue to be observed. Local axes that are no longer being observed are removed from $\hat{\Phi}$. This maintains low dimensionality, which decreases the risk of erroneous data association and improves robustness to outliers (e.g., lines extracted from noise).

190 Because the entries of $\hat{\Phi}$ are observed from a robot heading $\theta \in [-\pi, \pi)$, which itself is uncertain, the entries of $\hat{\Phi}$ are correlated to θ and to each other. As a result, the LC actually estimates the joint heading/AM state vector $\Theta = (\theta, \hat{\Phi})$ and its dense covariance matrix \mathbf{P} . The details of adding, removing, and updating entries of $\hat{\Phi}$ are described in Section 3.4.

195 *2.2. Line Extraction*

Line extraction algorithms can generally be divided into two steps: *line segmentation* and *line fitting*. Line segmentation determines the groups of points that form lines, and line fitting determines the coefficients of the equation describing each line and their uncertainties. A thorough comparison of line segmentation algorithms for 2D LiDAR [21] concluded that split-and-merge techniques are the preferred choice for real-time SLAM applications. After removing outlying points based on the proximity to their neighbours, split-and-merge is used for line segmentation in this study. Line fitting is performed using a weighted least-squares approach that has been optimized for LiDAR data [22].

The input to the line extraction algorithm is the range and bearing data (and their uncertainties) generated by 2D LiDAR. However, unlike most applications of line extraction, the output is simply the estimated axes $z^{(j)}$ of the extracted lines expressed in the LiDAR frame, as well as their corresponding scalar variances $\sigma_{z^{(j)}}^2$; i.e.,

$$\mathbf{Z} = \{z^{(i)} : z^{(i)} \in [0, \pi), i = 1, \dots, M\}, \quad (2)$$

205 where each $z^{(j)}$ has additive noise $v^{(j)} \sim \mathcal{N}(0, \sigma_{z^{(j)}}^2)$.

2.3. Observation Model

The observation model produces predicted axes in the LiDAR frame given an AM and the current heading of the robot. The predicted observation of the i -th map entry at time k in the LiDAR frame is then

$$\hat{z}_k^{(i)} = \phi^{(i)} - (\theta_k + \beta), \quad (3)$$

where $z^{(j)} \in [0, \pi)$ and β is the (constant) orientation of the sensor in the robot coordinate frame. The axes in the map, sensor orientation, robot heading, and coordinate frames are illustrated in Figure 4. Without loss of generality, $\beta = 0$ is assumed for the remainder of this paper. Recall that $\phi^{(i)}$ could be an entry from either $\bar{\Phi}$ or $\hat{\Phi}$. Note that the observation space is independent of the positions of the robot, laser scanner (relative to the robot), and lines in the environment. Furthermore, note that (3) consists of only scalar addition and subtraction, which is part of the reason why the LC is computationally simple.

215

3. The LiDAR Compass

The LiDAR compass (LC) provides heading estimates to a vehicle navigating within an environment containing line-extractable surfaces. These surfaces could be walls (indoors and outdoors), furniture, cars, rock faces, etc. For each laser scan, line extraction is performed (e.g., see Section 2.2), followed by a heading estimate by the LC. Like the observation model (Section 2.3), much of the calculation described in the following sections is performed on scalars,

220

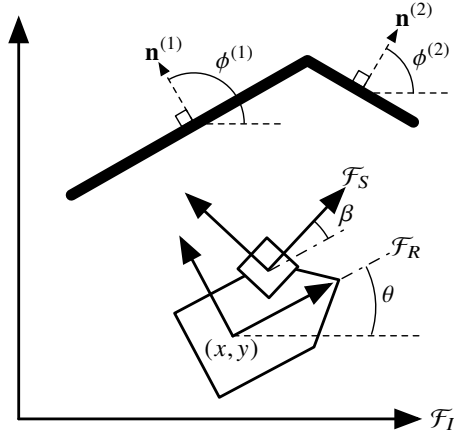


Figure 4: An illustration of the robot heading, entries of an AM, and the inertial (\mathcal{F}_I), robot (\mathcal{F}_R), and sensor (\mathcal{F}_S) coordinate frames. The entries of an AM are the axes coincident with their normal vectors $\mathbf{n}^{(i)}$. Each predicted observation $\hat{z}_k^{(i)}$ is an AM entry $\phi^{(i)}$ expressed in \mathcal{F}_S .

resulting in lightweight computational requirements and a straightforward implementation.

225 Each time the axis of an extracted line is observed, the LC performs (up to) four discrete steps described in the following sections. First, an attempt is made to associate the observation with an entry of $\bar{\Phi}$ (Section 3.1). If successful, Θ is updated relative to the difference (innovation) between the observation and associated AM entry. If no correspondence is found, the observation is compared to the entries of $\hat{\Phi}$ (Section 3.2). As before, Θ is updated for successful associations. When Θ is updated, entries of $\hat{\Phi}$ are merged if they become sufficiently similar, and observations not associated with either AM are added to $\hat{\Phi}$ (Section 3.3). Finally, infrequently observed entries of $\hat{\Phi}$ are removed by tracking how persistently they are extracted in recent laser scans (Section 3.4).
 230
 235 An overview illustrating the fundamental steps of the LC algorithm is presented in Figure 5.

3.1. Heading Estimation with the A Priori Axis Map ($\bar{\Phi}$)

The joint heading/AM estimate Θ_{k+1} at time $k+1$ is estimated given $\bar{\Phi}$, Θ_k , and the observed axes \mathbf{Z}_k . This step is illustrated in Figure 5b. It is required that the most likely correspondences among $\bar{\Phi}$ and \mathbf{Z}_k are established. This step begins by generating a set of predicted observations $\{\hat{z}^{(1)}, \dots, \hat{z}^{(N)}\}$ with (3); one for each of the N entries in $\bar{\Phi}$. Next, the total variance $s^{(j)}$ in the sensor frame of each observation $z^{(j)}$ is calculated by combining the variances of the robot heading and the observation; i.e.,

$$s^{(j)} = \sigma_{\theta_k}^2 + \sigma_{z^{(j)}}^2, \quad (4)$$

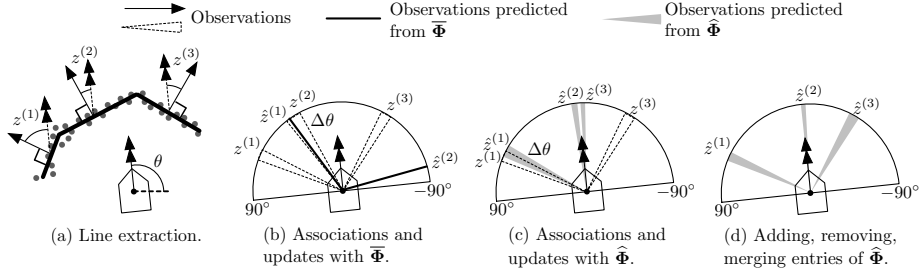


Figure 5: An overview of the LiDAR compass algorithm. (a) The axes of lines ($z^{(1)}$, $z^{(2)}$, and $z^{(3)}$) are extracted from LiDAR data (Section 2.2). (b) Predicted observations generated from $\bar{\Phi}$ ($\hat{z}^{(1)}$ and $\hat{z}^{(2)}$) are compared to the extracted axes. Here, $z^{(2)}$ is associated with $\hat{z}^{(1)}$, which produces a heading update $\Delta\theta$ and removes $z^{(2)}$ from further associations (Section 3.1). (c) Predicted observations generated from $\hat{\Phi}$ ($\hat{z}^{(1)}$, $\hat{z}^{(2)}$, and $\hat{z}^{(3)}$) are compared to the extracted axes. Here, $z^{(1)}$ is associated with $\hat{z}^{(1)}$, which produces a heading update $\Delta\theta$ (Section 3.2). (d) The remaining unassociated observation $z^{(3)}$ is added to $\hat{\Phi}$. Additionally, the old $\hat{z}^{(2)}$ is removed from $\hat{\Phi}$ because its brightness reached zero (see Section 3.4 for an explanation of brightnesses). In all parts, the semi-circles represent the observation space $[-\frac{\pi}{2}, \frac{\pi}{2})$ and the width of the wedges represent axial uncertainty.

where $\sigma_{z^{(j)}}^2$ is the variance of the axis observation, as described in Section 2.2. Data association is performed by selecting the predicted observation $\hat{z}^{(i)}$ with the minimum Mahalanobis distance $r^{(j,i)}$ from the observation $z^{(j)}$; i.e.,

$$r^{(j,i)} = \min \left(\frac{[z^{(j)} - \hat{z}^{(i)}]^2}{s^{(j)}}, \frac{[z^{(j)} - (\hat{z}^{(i)} + \pi)]^2}{s^{(j)}} \right), \quad (5)$$

$$i^* = \underset{i}{\operatorname{argmin}} r^{(j,i)}, \quad \text{for } i = 1, \dots, N, \quad (6)$$

where i^* is the index of $\bar{\Phi}$ that most likely corresponds with the j -th observation, and both $z^{(j)}$ and $\hat{z}^{(i)}$ are bound to $[0, \pi)$, as noted in (2) and (3). Note that the smallest of the two terms in (5) is taken to account for the bounds on entries of AMs; each predicted observation has an equivalent prediction on the opposite side of the unit circle. If the Mahalanobis distance $r^{(j,i^*)}$ is below a threshold, the j -th observation is used to update Θ , much like the observation update of a Kalman filter; i.e.,

$$\mathbf{C} = [-1 \quad 0^{(1)} \quad \dots \quad 0^{(N)}], \quad (7)$$

$$\mathbf{K}^{(j)} = \frac{\mathbf{P}_k \mathbf{C}^\top}{s^{(j)}}, \quad (8)$$

$$\Theta_{k+1} = \Theta_k + \mathbf{K}^{(j)} (z^{(j)} - \hat{z}^{(i^*)}), \quad (9)$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}^{(j)} \mathbf{C}) \mathbf{P}_k, \quad (10)$$

where $\hat{z}^{(i^*)}$ may require a rotation by π depending on the result of (5), and \mathbf{C} is the (constant) Jacobian of (3) with respect to the state. In the example

240 of Figure 5b, only one observation has a Mahalanobis distance meeting the threshold, generating the update $\Delta\theta$, which represents (7–10). Observations with a Mahalanobis distance $r^{(j,i^*)}$ greater than a threshold are compared to $\hat{\Phi}$, as described in Section 3.2 and illustrated in Figure 5c.

3.2. Heading Estimation with the Local Axis Map ($\hat{\Phi}$)

Estimating an update to the joint AM/heading state Θ with $\hat{\Phi}$ requires only minor amendments to the procedure described in Section 3.1 and is illustrated in Figure 5c. First, the joint variance $s^{(j,i)}$ between the j -th observation and i -th entry of $\hat{\Phi}$ replaces $s^{(j)}$ in (4); i.e.,

$$\mathbf{C}^{(i)} = [-1 \quad 0^{(1)} \quad \dots \quad 1^{(i)} \quad \dots \quad 0^{(N)}], \quad (11)$$

$$s^{(j,i)} = \mathbf{C}^{(i)} \mathbf{P}_k^- [\mathbf{C}^{(i)}]^\top + \sigma_{z^{(j)}}^2. \quad (12)$$

245 This change is made to account for the correlation between θ and $\hat{\Phi}$. Once again, the Mahalanobis distance determines the maximum likelihood correspondences among \mathbf{Z}_k and $\hat{\Phi}$, and is calculated using (5) and (6) as before with $s^{(j,i)}$ substituted for $s^{(j)}$. Successful correspondences (i.e., the Mahalanobis distance is below a threshold) are used to update Θ in a fashion similar to (7–10); one only
 250 needs to replace \mathbf{C} with $\mathbf{C}^{(i^*)}$ and $s^{(j)}$ with $s^{(j,i^*)}$, where i^* is the index of the successfully associated entry of $\hat{\Phi}$.

3.3. Merging and Adding Entries to the Local Axis Map ($\hat{\Phi}$)

It is possible that the estimated axes of two entries in $\hat{\Phi}$ converge as Θ is updated, meaning they represent the same line axis in the environment. These occurrences are detected by calculating the Mahalanobis distance $r^{(m,n)}$ between the expected distance (which is zero) and the estimated distance between each pair (m, n) of the N entries in $\hat{\Phi}$; i.e.,

$$\begin{aligned} \mathbf{C}^{(m,n)} &= [0, 0^{(1)}, \dots, 1^{(m)}, \dots, -1^{(n)}, \dots, 0^{(N)}], \\ s^{(m,n)} &= \mathbf{C}^{(m,n)} \mathbf{P}_{k+1}^- [\mathbf{C}^{(m,n)}]^\top, \\ r^{(m,n)} &= \frac{[0 - (\phi^{(m)} - \phi^{(n)})]^2}{s^{(m,n)}}. \end{aligned} \quad (13)$$

If $r^{(m,n)} < r_{\text{thresh}}$, entries m and n must be merged. First, Θ and \mathbf{P} are updated using (7–10) with $z^{(j)} = 0$, $\hat{z}^{(i^*)} = \phi^{(m)} - \phi^{(n)}$, and $\mathbf{C}^{(m,n)}$ in place of \mathbf{C} . The
 255 n -th entry of $\hat{\Phi}$ is then removed from Θ and \mathbf{P} . The small size of $\hat{\Phi}$ makes a comparison of each pair of entries computationally tractable.

Observations that failed to be associated with either $\bar{\Phi}$ or $\hat{\Phi}$ (such as $z^{(3)}$ in the example of Figure 5c–d) are used to add new entries to $\hat{\Phi}$ with the inverse

observation model; i.e.,

$$\mathbf{F} = [1 \quad 0^{(1)} \quad \dots \quad 0^{(N)}], \quad (14)$$

$$\Theta_{k+1} = \begin{bmatrix} \theta_k \\ \widehat{\Phi}_k \\ z_k^{(j)} + \theta_k \end{bmatrix}, \quad (15)$$

$$\mathbf{P}_{k+1} = \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_k \mathbf{F}^\top \\ \mathbf{F} \mathbf{P}_k & \sigma_{\theta_k}^2 + \sigma_{z^{(j)}}^2 \end{bmatrix}. \quad (16)$$

3.4. Brightnesses and Removing Entries from $\widehat{\Phi}$

To maintain low-dimensionality and mitigate erroneous data associations, only frequently observed axes are kept in $\widehat{\Phi}$. To choose which entries are kept in $\widehat{\Phi}$, each entry is assigned a *brightness* $b \in [0, 1]$. Observing map entries causes them to *brighten*, while unobserved map entries *fade*. When a new entry is added to $\widehat{\Phi}$, it is assigned an initial brightness $b_0 \in (0, 1)$. For each observation in \mathbf{Z}_k that is associated with $\phi^{(i)}$, $b^{(i)}$ increases by $\Delta b \in (0, 1)$ (typically $\Delta b < 0.1$), up to a maximum brightness of 1. If $\phi^{(i)}$ is not observed at time k , $b^{(i)}$ decreases by Δb . Map entries fading to zero are removed from $\widehat{\Phi}$, resulting in a succinct AM that is comprised solely of map entries actively being observed. For example, in Figure 5c, the prediction $\hat{z}^{(3)}$ is not associated with any observation; thus, its brightness is decreased. Supposing its brightness was low at the start of this time step, this decrease reduces its brightness to zero and it is removed from $\widehat{\Phi}$. As a result, it no longer appears in Figure 5d.

By selecting b_0 and Δb in conjunction with the frequency of the observations, one can specify the typical amount of time it takes for map entries to reach maximum brightness or fade to zero. Erroneously-added entries (such as lines extracted from noise) are quickly removed from the map because they tend not to be re-observed. For the experiments in this paper, $b_0 = 0.2$ was used and Δb was selected such that a consistently observed axis would reach maximum brightness after 4–5 seconds. Additionally, the brightness of map entries can be used to adjust the heading update by inserting a scaling factor to the gain calculation; e.g., changing (8) to

$$\mathbf{K}^{(j)} = b^{(i^*)} \frac{\mathbf{P}_k [\mathbf{C}^{(i^*)}]^\top}{s^{(j, i^*)}} \quad (17)$$

when an entry of $\widehat{\Phi}$ is observed (i.e., the procedure in Section 3.2). This results in heading updates associated with bright map entries (e.g., the axes of the walls of a room) having a greater effect on the heading estimate than faded map entries.

3.5. Heading Initialization

The previous sections state that the *a priori* AM provides an absolute reference frame from which the absolute heading estimates can be derived. These

statements imply that the estimated heading is properly initialized. Proper initialization is also required to rectify the $\pm\pi$ ambiguity of axes. Once initialized, absolute heading estimates are observable as long as the heading uncertainty is not permitted to grow to the point where observations are incorrectly associated with entries of $\bar{\Phi}$. This condition is easily met if axes are regularly observed and entries of $\bar{\Phi}$ are sufficiently distinct. Maintaining this condition bounds the growth of heading error by regularly observing surfaces with known axes in the global coordinate frame.

Proper initialization consists of setting an initial heading in the same coordinate frame as $\bar{\Phi}$. The initial heading uncertainty must be selected such that early axis observations will correctly associate with the corresponding entries in $\bar{\Phi}$. This process is usually trivial. For example, in the experiments where $\bar{\Phi} = \{0, \frac{\pi}{2}\}$, the vehicle was parked roughly parallel to a surface whose axis was in $\bar{\Phi}$. The initial heading was set to the entry in $\bar{\Phi}$ corresponding to the parallel surface, and the initial uncertainty was set to account for not being exactly parallel to the surface. When started, the algorithm would quickly adjust the initial heading by a few degrees (i.e., the initial misalignment) to align it with $\bar{\Phi}$.

4. Localization with a LiDAR Compass

This section describes how the LC could be used as part of a localization algorithm for a mobile robot or ground vehicle. The experimental results presented in Section 5 demonstrate the effectiveness of using an LC for this application. When used to aid localization, an LC is analogous to providing the robot with both an absolute and relative heading sensor. Much like the magnetic field measured by a standard compass, the *a priori* AM $\bar{\Phi}$ provides a static, absolute reference with which the heading can be corrected. Additionally, the local AM $\hat{\Phi}$ estimates local axes and provides incremental updates to the heading. Like a gyroscope (however, without the need for integration), associating observations with $\hat{\Phi}$ provides dead-reckoning heading estimates. The combination of line extraction with $\bar{\Phi}$ and $\hat{\Phi}$ makes the LC act as a virtual heading sensor. By combining the LC with interoceptive sensor(s) that accurately measure translation (e.g., wheel encoders), a simple filter can be derived that localizes a vehicle with a pose represented by $\mathbf{q} = (x, y, \theta) \in \mathbb{R}^2 \times \mathbb{S}^1$. To account for the estimation of the full pose, $\Theta = (\mathbf{q}, \hat{\Phi})$ replaces the heading-only estimator presented in Section 3.

4.1. Motion Update

This section provides background about how the measurements of an interoceptive sensor are used to update the state estimate. This procedure is provided for completeness, but is not dissimilar from well-established methods [23, 24]. The motion update estimates Θ_{k+1} given the sensor measurements $\mathbf{d}_k = (d_{f,k}, d_{\theta,k})$ and the latest state estimate Θ_k . The measurement $\mathbf{d}_k \in \mathbb{R} \times \mathbb{S}^1$ represents the displacement ($d_{f,k}$) and the change in heading direction ($d_{\theta,k}$). It

is assumed to be normally distributed with covariance matrix \mathbf{Q} . The motion at each time step can be modelled as the traversal of an arc with constant radius (circular motion); i.e.,

$$\begin{aligned}\boldsymbol{\Theta}_{k+1} &= f(\boldsymbol{\Theta}_k, \mathbf{d}_k) \\ &= \begin{bmatrix} x_k + \frac{d_{f,k}}{d_{\theta,k}} [\sin(\theta_k + d_{\theta,k}) - \sin(\theta_k)] \\ y_k - \frac{d_{f,k}}{d_{\theta,k}} [\cos(\theta_k + d_{\theta,k}) - \cos(\theta_k)] \\ \theta_k + d_{\theta,k} \\ \hat{\boldsymbol{\Phi}}_k \end{bmatrix}. \end{aligned} \quad (18)$$

Note that (18) requires a non-zero turning radius (i.e., $d_{\theta,k} \neq 0$). When this is not the case (in practice, when $d_{\theta,k}$ is less than a threshold), $f(\boldsymbol{\Theta}_k, \mathbf{d}_k)$ is simplified to

$$\boldsymbol{\Theta}_{k+1} = \boldsymbol{\Theta}_k + \begin{bmatrix} d_{f,k} \cos \theta_k \\ d_{f,k} \sin \theta_k \\ d_{\theta,k} \\ \mathbf{0} \end{bmatrix}. \quad (19)$$

A similar arc-traversing motion model for velocity commands (in place of interoceptive measurements) is presented in [23]. Because $f(\boldsymbol{\Theta}_k, \mathbf{d}_k)$ is nonlinear, an unscented transformation [25] is used to reconstruct the mean and covariance of the state after (18) or (19) is applied. First, the state is augmented to include the zero-mean noise vector $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$; i.e.,

$$\boldsymbol{\Theta}_k^a = \begin{bmatrix} \boldsymbol{\Theta}_k \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{P}_k^a = \begin{bmatrix} \mathbf{P}_k & \mathbf{0} \\ \mathbf{0}^\top & \mathbf{Q}_k \end{bmatrix}. \quad (20)$$

Next, $2P$ sigma points $\mathcal{X}_k^{(i)}$ (where $P = \dim \boldsymbol{\Theta}_k^{(a)}$) are sampled from the augmented state; i.e.,

$$\mathcal{X}_k^{(i)} = \begin{cases} \boldsymbol{\Theta}_k^a + (\sqrt{P\mathbf{P}_k^a})_i & \text{for } i = 1, \dots, P \\ \boldsymbol{\Theta}_k^a - (\sqrt{P\mathbf{P}_k^a})_{i-P} & \text{for } i = P+1, \dots, 2P. \end{cases} \quad (21)$$

The state $\mathcal{X}_{\boldsymbol{\Theta},k}^{(i)}$ and noise $\mathcal{X}_{\mathbf{v},k}^{(i)}$ are extracted from each sigma point and propagated through the motion model by applying the noise to the interoceptive measurement; i.e.,

$$\mathcal{X}_{k+1}^{(i)} = f\left(\mathcal{X}_{\boldsymbol{\Theta},k}^{(i)}, \mathbf{d}_k + \mathcal{X}_{\mathbf{v},k}^{(i)}\right). \quad (22)$$

The mean and covariance of the state is reconstructed with the transformed sigma points, yielding the estimate

$$\begin{aligned}\boldsymbol{\Theta}_{k+1} &= \frac{1}{2P} \sum_{i=1}^{2P} \mathcal{X}_{k+1}^{(i)}, \\ \mathbf{P}_{k+1} &= \frac{1}{2P} \sum_{i=1}^{2P} \left(\mathcal{X}_{k+1}^{(i)} - \boldsymbol{\Theta}_{k+1}\right) \left(\mathcal{X}_{k+1}^{(i)} - \boldsymbol{\Theta}_{k+1}\right)^\top. \end{aligned} \quad (23)$$

4.2. Observation Update

315 The observation update estimates the joint pose/AM state Θ_{k+1} given the observed axes \mathbf{Z}_k and the latest state estimate Θ_k . This process requires only minor modifications of the heading estimator presented in Section 3. The primary difference is simply prepending the various \mathbf{C} or \mathbf{F} matrices in (7), (11), (13), and (14) with $\begin{bmatrix} 0 & 0 \end{bmatrix}$ to account for the replacement of θ in Θ with \mathbf{q} .

320 5. Experimental Results

This paper presents results from testing the LC using data collected from three distinct environments. These include (i) the MIT Killian Court data set, which is openly available [4] and is commonly used to test SLAM algorithms [26]; (ii) data collected in Beamish-Munro Hall, a modern building at Queen's University that contains a challenging layout for the LC; and (iii) data collected outdoors on the Queen's University campus. This section provides the details of these experiments and discusses the effectiveness of the LC in each scenario. In all of the experiments, the line extraction algorithm was implemented in C++, while the LC was implemented in Python 2.7.6. All algorithms were executed on a laptop with an Intel i5 2.53 GHz processor and 4 GB of RAM. Despite having been implemented in a high-level language, there were no computational issues that prevented the LC from running in real-time.

5.1. MIT Killian Court

335 The odometry and LiDAR data from this data set were used to perform localization with the LC, primarily to test its functionality as a provider of an initial guess for graph-based SLAM algorithms. The resulting paths as estimated by odometry, odometry with an LC, and the SLAM solution by [3] are illustrated in Figure 1. Compared to the map generated by scan matching (Figure 2), this shows how the LC can provide an excellent initial guess in appropriate environments, without requiring any additional sensors. Because the LC only arrests the growth of heading error, translation error due to wheel slip or poor encoder calibration still accumulate over time. This is the major source of the discrepancy between the LC and SLAM solutions in Figure 1. In this plot, the robot started at the origin, and arrived at the small loop in the vicinity of (10 m, -50 m) after a relatively short traverse. The misalignment of the odometry/LC path around this loop suggests either poor calibration of wheel encoders or that a considerable amount of slip has occurred (unlikely, due to the indoor environment). This suggests that ensuring proper calibration of the wheel encoders for translation is important for odometry/LC localization.

350 5.2. Indoor Experiment at Beamish-Munro Hall

Beamish-Munro Hall (BMH) is a modern building on the campus of Queen's University in Kingston, ON, Canada. Experiments were performed indoors on the ground floor of the building, whose blueprints are shown in Figure 3a. The routes through the building used in the experiments are highlighted in the figure.

355 BMH was selected because it has areas where the walls are oriented differently
 from the main layout of the building, and it also has a large curved wall to add
 difficulty for the LC. To perform the experiments, a Clearpath Robotics Husky
 A200 mobile robot was equipped with a SICK LMS111 2D laser scanner oriented
 to scan in the horizontal plane. This laser scanner has a field of view of 270° ,
 360 a resolution of 0.5° , and a range of 20 m. The robot was also equipped with a
 Microstrain 3DM-GX3-25 IMU, used solely for one of its gyroscopes to compare
 its performance with the LC. At the time of the experiments, the environment
 was fairly dynamic with a considerable amount of pedestrian traffic.

Ground truth poses at 10 locations (A, B, . . . , J) were determined by using
 365 hand measurements (see the appendix for the methodology). Two routes were
 selected of different difficulty for the LC, and an *a priori* map $\bar{\Phi} = \{0, \frac{\pi}{2}\}$ was
 selected. The first route (ABCDEFGA) consistently had observable surfaces in
 $\bar{\Phi}$. The second route (AHIJEFGA) included areas where no entries of $\bar{\Phi}$ were
 observable, had surfaces that were oriented slightly differently than those in $\bar{\Phi}$
 370 (i.e., $< 8^\circ$ difference), and a large, curved surface. Five trials were performed on
 each route, each of which used odometry to estimate translation. The heading
 was estimated using four different methods: odometry (i.e., comparing left and
 right wheel encoders), integration of a gyroscope, scan matching, and the LC.
 An open source implementation [27] of the popular point-to-line scan matching
 375 algorithm by Censi [28] was used. The robot was briefly parked at the ground
 truth poses to record the pose estimate of each algorithm, and was then driven
 (at varying speeds of 0.1–1.0 m/s and not necessarily in a straight line) to the
 next ground truth pose. The estimated paths of each algorithm for a single trial
 of each route are shown in Figure 6. To more closely observe the effectiveness and
 380 consistency of the LC for heading estimation, its comparison to the gyroscope
 and scan matching estimates for a trial from each route is shown in Figure 7 and
 a summary of the results over the course of five trials of each route is provided
 in Table 1.

As is apparent in Figure 6, when odometry, the gyroscope, or scan match-
 385 ing was used for heading estimation, the localization estimates suffered from
 the accumulation of dead reckoning error. In particular, the error accumulated
 using scan matching varied significantly among the trials (see Table 1). It is
 possible that the somewhat dynamic environment (pedestrians) contributed to
 this inconsistency. Although the LC does not prevent accumulating transla-
 390 tional errors, it bounds heading errors if entries of $\bar{\Phi}$ are observed often enough.
 Adding and observing entries of $\hat{\Phi}$ limits the growth rate of heading error in
 areas void of surfaces corresponding to entries of $\bar{\Phi}$. However, as the position
 estimate is tightly coupled to the quality of the heading estimate (i.e., see (18)
 and (19)), the full pose estimation was drastically improved when using an LC
 395 as compared to the other methods. Table 1 demonstrates the consistency of the
 LC over multiple trials.

All trials for each route showed very similar behavior. One exception oc-
 curred at ground truth pose H in the more difficult route, where the LC consis-
 tently provided an incorrect heading estimate (see Figure 7b). The cause of this
 400 discrepancy revealed a short-coming of the LC when there exists surfaces whose

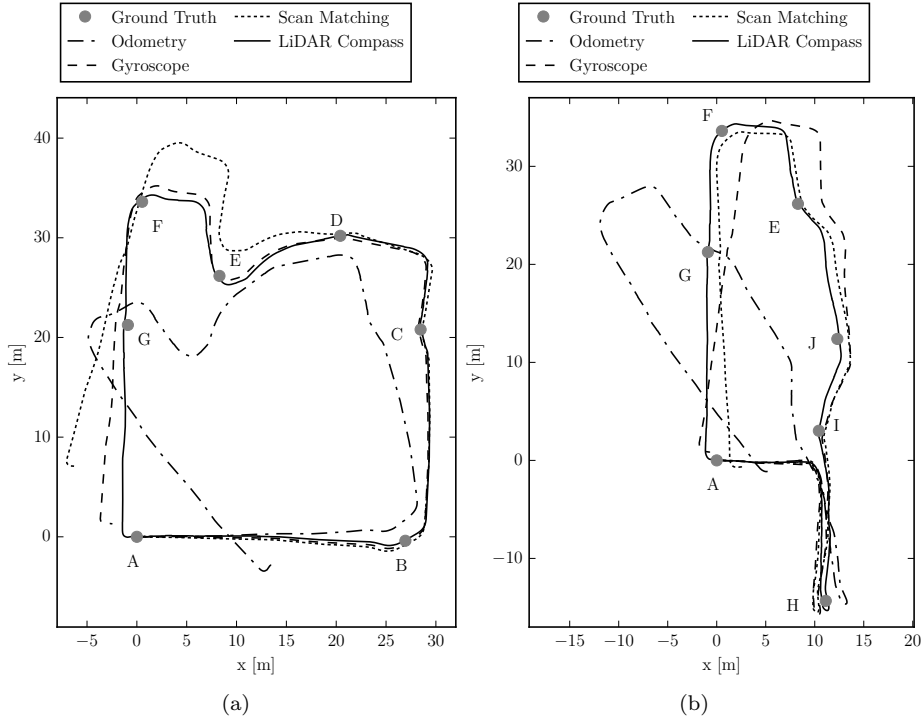
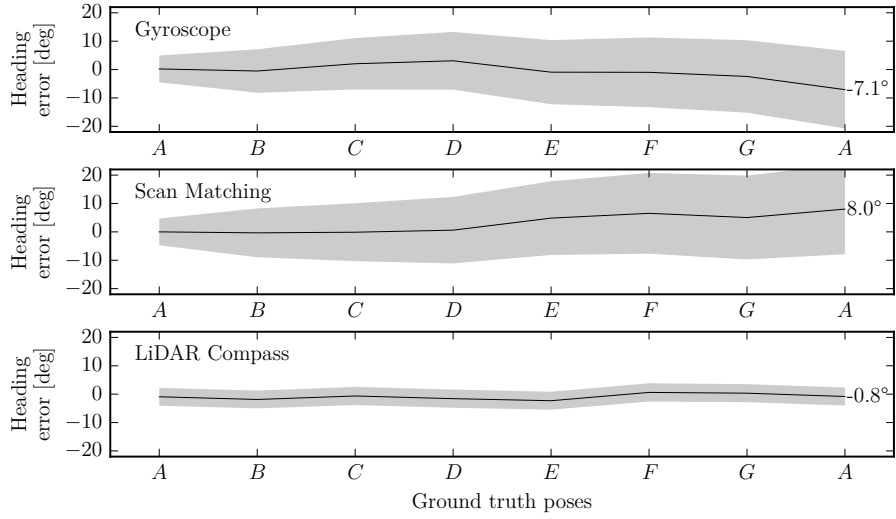


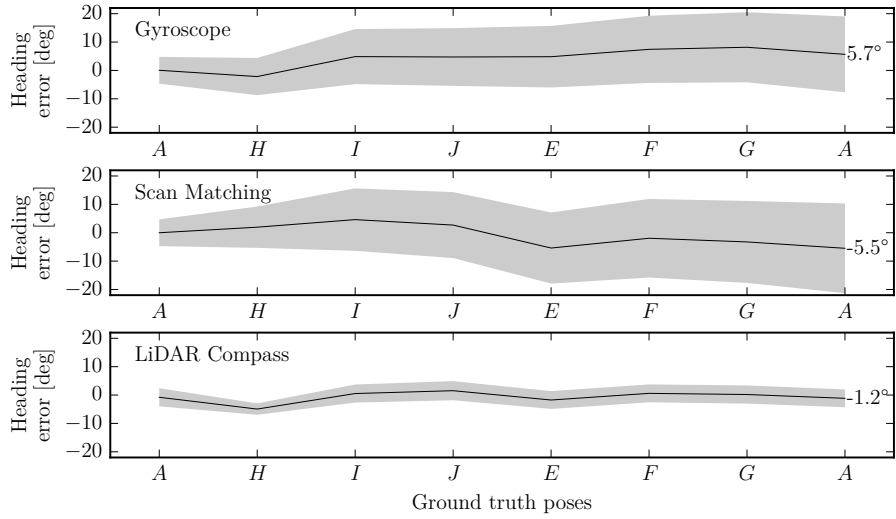
Figure 6: The paths estimated by odometry using four different methods of heading estimation (odometry, gyroscope, scan matching, and the LC). The results from routes ABCDEFGA and AHLJEFGA of the BMH experiments are plotted in (a) and (b), respectively. In both cases, the trial in which scan matching produced the best result is shown. The robot was briefly parked at each of the ground truth poses, whose positions (A, B, ...) are labelled in the plots.

Table 1: A summary of the results from all trials in BMH. The root mean square error is the average difference between the estimated heading and the heading at the ground truth poses.

Route	Trial	Time [s]	Root Mean Square Error [deg]		
			Gyroscope	Scan Matching	LC
ABCDEFGA	1	430	3.00	4.42	1.31
	2	417	3.34	10.99	1.22
	3	404	4.68	6.28	1.28
	4	400	0.79	4.55	1.43
	5	283	3.34	12.87	1.34
AHLJEFGA	1	311	5.12	14.80	2.42
	2	306	3.52	16.55	2.01
	3	309	3.26	8.99	2.05
	4	329	4.39	13.28	1.96
	5	349	5.34	3.64	2.01



(a)



(b)

Figure 7: A comparison of the heading errors of the gyroscope, scan matching, and LC estimates for the BMH data. (a) The results of the first trial on the first route (ABCDEFGA). (b) The results of the fifth trial on the second route (AHIJEFGA). These trials were the best results (i.e., smallest root-mean-square error) achieved by scan matching. The shaded areas illustrate the 3σ uncertainty of the errors, and include the (relatively small) uncertainty of the ground truth.

axes are slightly different than entries of $\overline{\Phi}$. In this case, a wall was frequently observed with an axis only a few degrees different from an entry of $\overline{\Phi}$. There was sufficient combined uncertainty from the observation (from line extraction) and the current heading estimate that incorrect data association occurred. This mild failure highlights the sensitivity of the LC on selecting an appropriate Mahalanobis distance threshold for this type of scenario. The threshold must be large enough to capture the expected uncertainty of the observations, which may result in slight misalignments (e.g., a shelf not quite aligned with a wall) being incorrectly associated. However, the LC easily recovered once sufficient subsequent observations of true entries of $\overline{\Phi}$ were made. All the trials concluded with nearly zero heading error when the LC was used.

5.3. Outdoor Experiment on the Queen’s University Campus

Data was collected by driving a small electric vehicle outdoors around the grounds of the Queen’s University campus. The electric vehicle (pictured in Figure 8) was equipped with the same SICK LMS111 LiDAR and Microstrain 3DM-GX3-25 IMU used in the BMH experiments as well as US Digital A2 encoders on the drive shaft and steering column. A Novatel GPS/INS system was also installed to record ground truth (see the appendix for details). The area on campus where data was collected (pictured in Figure 3b) includes a variety of building, trees, pathways, roads, and parking lots, and is typically populated by many cars, cyclists, and pedestrians. Five trials were performed of a loop that is approximately 1.3 km in length (illustrated in Figure 3b), that covers several types of environments (e.g., narrow paths between buildings, parking lots, open roads, areas with many trees, etc.). The vehicle was driven at speeds between 1.4–4.4 m/s (5–16 km/h). Like the experiments in BMH, odometry was used to estimate translation and the same four methods of heading estimation (odometry with the steering encoder, gyroscope integration, scan matching, and the LC) were compared. A simple *a priori* AM $\overline{\Phi} = \{0, \frac{\pi}{2}\}$ was used, and the pose estimates were recorded and synchronized with the ground truth provided by the GPS/INS system. The estimated paths of each algorithm for a single trial are shown in Figure 9. The errors of the heading estimation algorithms for one of the trials is shown in Figure 10, and a summary of the heading errors for all trials is provided in Table 2.

The heading estimated by scan matching sometimes suffered singular points of failure due to the experiments taking place in a three-dimensional environment. When the vehicle transitioned from a downward slope to a flat surface, its rear-facing LiDAR would sometimes suddenly measure the ground behind the vehicle. This situation is illustrated in Figure 11. When this occurred near a turn, the heading estimate provided by scan matching would sometimes incorrectly shift by up to tens of degrees. The LC was unaffected by these abrupt changes in the scans. As a result, to fairly compare the performance of scan matching, these erroneous occurrences were manually removed from the resulting estimates. The number of manual corrections required for each trial is shown in Table 2. Figures 9 and 10 and Table 2 report the scan matching results after the corrections were made.



Figure 8: The electric vehicle (Taylor-Dunn SS-534) used for the Queen’s University campus experiments. It has a maximum speed of approximately 4.4 m/s (16 km/h). Digital encoders were installed on the drive shaft and steering column. The (rear-facing) LiDAR was mounted at the back.

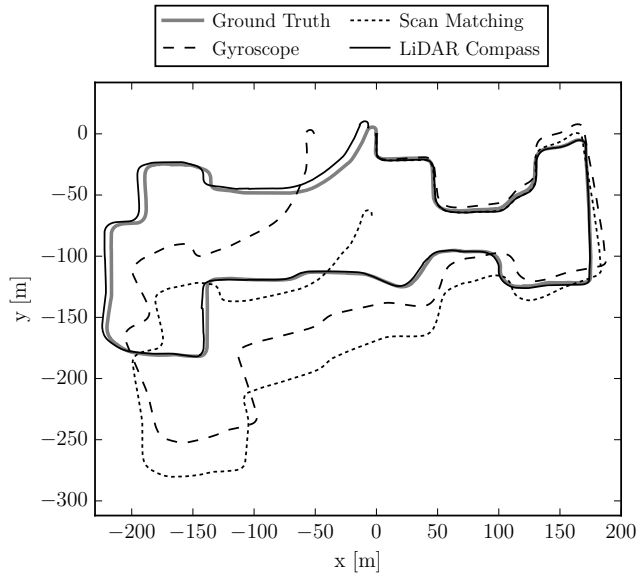


Figure 9: The paths estimated by odometry using three different methods of heading estimation (gyroscope integration, scan matching, and the LC) for the first trial of the Queen’s University campus data set. The path using odometry for the heading estimate is not shown; it is the least accurate and cannot be contained in the scale of this plot. This trial is illustrated because it represents the best result (i.e., smallest root-mean-square error) achieved by scan matching. The vehicle both started and finished at the origin of the plot (driving clockwise).

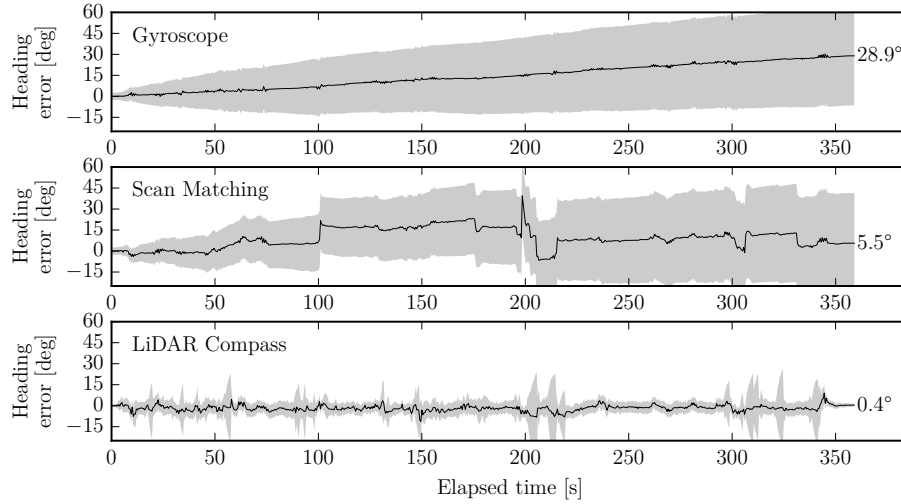


Figure 10: A comparison of the heading errors of the gyroscope, scan matching, and LC estimates for the first trial on the Queen’s University campus. This trial is illustrated because it represents the best result (i.e., smallest root-mean-square error) achieved by scan matching. To improve the readability of the plot, only every tenth point is plotted. Similarly to Figure 7, the shaded areas illustrate the 3σ uncertainty of the errors, and include the (relatively small) uncertainty of the ground truth.

Table 2: A summary of the results from all trials on the Queen’s University campus. The number in parentheses next to each scan matching result is the number of singular points of failure that were manually corrected.

Trial	Time [s]	Root Mean Square Error [deg]		
		Gyroscope	Scan Matching	LC
1	365	16.64	11.58 (2)	2.79
2	358	15.57	36.49 (3)	4.06
3	365	15.76	31.66 (3)	3.03
4	338	16.47	14.89 (2)	2.81
5	346	19.76	30.58 (3)	3.61

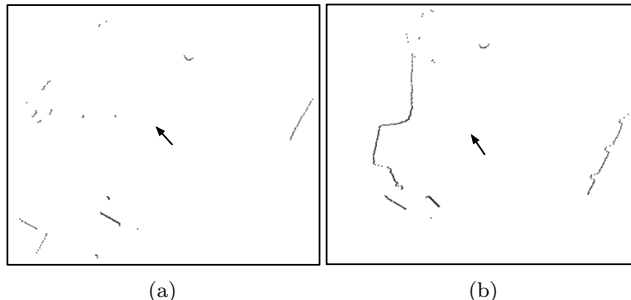


Figure 11: A laser scan (a) before, and (b) shortly after the vehicle transitioned from a downward slope to a flat surface. The arrow indicates the approximate pose of the LiDAR. The time between these scans is approximately 0.25 s. This sharp transition often caused singular points of failure in the heading estimate by scan matching.

Comparing the estimated paths in Figure 9 illustrates how very simple *a priori* information (in this case, $\bar{\Phi} = \{0, \frac{\pi}{2}\}$) can vastly improve heading estimation, and by extension, localization. Although the vehicle was often driven in areas where surfaces corresponding to entries of $\bar{\Phi}$ were not present, tracking local surfaces in $\hat{\Phi}$ sufficiently limited the growth of heading errors until entries of $\bar{\Phi}$ could be re-observed. The result is an estimated path that differs from ground truth mostly in scale, and not in rotation. This scale difference can mostly be attributed to two sources: (i) the translational error in the encoders; and (ii) changes in elevation, which are reflected in the ground truth but not the two-dimensional pose estimate provided by odometry/LC. One application previously mentioned for the LC is the generation of a good initial guess for pose-graph SLAM algorithms. The (mostly) scale-only difference between the odometry/LC estimate and the ground truth presents the interesting possibility of developing a lightweight pose-graph implementation that uses this constraint. In other words, an acceptable map could be produced by assuming the orientation of nodes in the pose-graph are correct, and optimizing only over position space, which makes the optimization completely linear. Although not as accurate as a full pose-graph SLAM implementation, this simplification may be sufficient for some practical applications.

The heading error plots illustrated in Figure 10 show that the heading error of the LC estimate typically stayed near zero for the duration of the trial, with occasional spikes in its uncertainty. These spikes (e.g., at 55 s, 205 s, and 315 s) were consistent in all of the trials. They tended to occur while the vehicle underwent a rapid change in heading while few local surfaces were available for axis extraction. As a result, heading estimation briefly relied on the highly uncertain prior provided by the steering encoder on the vehicle; i.e., the assumption that line-extractable surfaces are frequently observable failed. A temporary lack of surfaces when the vehicle is moving relatively straight is not a problem if the sensor providing the prior estimate has minimal bias (i.e., it reports near-zero heading change when driving straight). In all cases, recovery

was rapid once line-extractable surfaces were re-observed.

The importance of modelling the growth of heading uncertainty (i.e., the \mathbf{Q} entry in (20) and the σ_z^2 entries in (12)) while no entries of $\overline{\Phi}$ are being observed cannot be overstated; too little uncertainty may prevent the association of future observations with $\overline{\Phi}$ (false negative), while too much uncertainty may incorrectly associate an observation with an entry of $\overline{\Phi}$ (false positive). The latter case was responsible for the spike at approximately 205 s in the plots in Figure 10, where a wall whose orientation differed only slightly from an entry of $\overline{\Phi}$ was associated incorrectly.

5.4. Qualitative Comparison of Heading Estimation Methods

To summarize, a qualitative comparison of the four methods of heading estimation used in this paper is provided in Table 3. Each attribute is described in greater detail directly below.

Computation A qualitative assessment of the computational burden. Note that a proper quantitative assessment would require re-implementing both the open source scan matching algorithm and the LC in a way that ensures a direct and fair comparison.

Implementation The complexity/difficulty of implementation.

Consistency The consistency of the estimates over the course of several trials.

Robustness The robustness of the estimates to external disturbances or abrupt changes in the environment.

Error growth The rate at which heading error accumulates.

A priori Information that is required before estimation can occur.

Environmental Requirements Requirements characteristic of the environment necessary to meet the assumptions of the estimator.

6. Conclusion

This paper introduces the concept of a *LiDAR compass*, which uses very simple *a priori* geometrical information about an environment (i.e., an *axis map*) to transform range and bearing measurements from a two-dimensional scanning LiDAR into absolute and relative heading estimates. The LC was shown to correct the heading of a vehicle equipped with wheel and steering encoders, and its performance was compared to a gyroscope and scan matching in both indoor and outdoor environments. In all tested environments, the LC was found to be highly effective at bounding the growth of heading error, resulting in significantly improved localization when compared to traditional methods. As a result, in environments that meet the requirements of the LC (e.g., indoors, urban areas), relatively accurate localization can be achieved without SLAM

Table 3: A qualitative comparison of different methods of heading estimation. More detailed descriptions of the attributes being compared are provided in Section 5.4.

Attribute	Odometry	Gyroscope	Scan Matching	LC
Computation	low	low	medium	low
Implementation	low	low	medium	low
Consistency	average	good	poor	good
Robustness	good	good	poor	good
Error growth	poor	average	average	bounded
<i>A priori</i>	none	none	none	minimal*
Environmental	none	none	minimal	more**

* An axis map is required, as described in Section 2.1.

** Re-occurring, line-extractable surfaces are required.

or an initial mapping phase. Alternatively, offline mapping algorithms (e.g., pose-graph SLAM) can be provided with an excellent initial guess without the use of additional sensors. In future, the authors plan to extend the LiDAR
515 compass and axis mapping to 3D, which has interesting applications in naturally occurring environments, such as in geology and mining.

Appendix

The ground truth for the Beamish-Munro Hall data set consists of poses
520 q_A, q_B, \dots, q_J , where $q_i = (x_i, y_i, \theta_i) \in \mathbb{R}^2 \times \mathbb{S}^1$. The position of these poses were estimated by taking a series of range and bearing measurements between all poses within line-of-sight of each other, as well as between poses and known positions in the building (e.g., a corner), taken from blueprints. An overdetermined nonlinear system was formed from this set of noisy range and bearing measurements.
525 This system was solved using the popular Levenberg-Marquardt algorithm, yielding an estimate of $(x_A, y_A, x_B, y_B, \dots, x_J, y_J)$ and its associated covariance matrix. These are plotted in Figure 6. The orientations $(\theta_A, \theta_B, \dots, \theta_J)$ were estimated via hand measurements with a digital protractor relative to a nearby known orientation (e.g., a wall). The noise in these measurements is
530 accounted for in the uncertainty in Figure 7.

The ground truth for the Queen’s University campus data set was determined with a combined global positioning and inertial navigation system (GPS/INS). A Novatel SPAN-SE-2 GPS receiver and CPT IMU, alongside commercial Novatel GPS/INS software were used to estimate the true two-dimensional pose. The
535 latitude, longitude, and azimuth provided by this software was transformed to the local coordinate frame (defined by the initial pose of the vehicle) by first converting it to the Universal Transverse Mercator (UTM) coordinate system. The software also provided an estimated uncertainty in the heading, which is included in the error bounds of the plots in Figure 10. The typical standard
540 deviation of the GPS/INS solution was approximately 0.5–3.0 m in position and $< 0.5^\circ$ in heading.

Acknowledgments

This project was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under project 397985-2011 and by the
545 NSERC Canadian Field Robotics Network (NCFRN). The authors would like to thank Brian Lynch for his help obtaining the ground truth in BMH.

References

- [1] K. V. Mardia, P. E. Jupp, *Directional Statistics*, John Wiley & Sons, Ltd., West Sussex, England, 2000.
- 550 [2] G. Hu, K. Khosoussi, S. Huang, Towards a reliable SLAM back-end, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 37–43.
- [3] C. Stachniss, *Exploration and Mapping with Mobile Robots*, Phd thesis, Institute of Computer Science, University of Freiburg (Apr. 2006).
- 555 [4] M. Bosse, P. Newman, J. Leonard, S. Teller, MIT Killian Court [data set], retrieved from http://www.ijrr.org/contents/23_12/abstract/1113.html (Dec. 2004).
- [5] M. Drumheller, Mobile robot localization using sonar, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9 (1987) 325–332.
- 560 [6] I. J. Cox, Blanche- An experiment in guidance and navigation of an autonomous robot vehicle, *IEEE Transactions on Robotics and Automation* 7 (2) (1991) 193–204.
- [7] J. Gonzalez, A. Ollero, A. Reina, Map building for a mobile robot equipped with a 2D laser rangefinder, in: *Proc. IEEE International Conference on Robotics and Automation*, Vol. 3, San Diego, CA, 1994, pp. 1904–1909.
- 565 [8] A. Garulli, A. Giannitrapani, A. Rossi, A. Vicino, Mobile robot SLAM for line-based environment representation, in: *Proc. IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 2041–2046.
- [9] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, N. Tomatis, Orthogonal SLAM: a step toward lightweight indoor autonomous navigation, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 5007–5012.
- 570 [10] B.-W. Kuo, H.-H. Chang, Y.-C. Chen, S.-Y. Huang, A light-and-fast SLAM algorithm for robots in indoor environments using line segment map, *Journal of Robotics* 2011.
- 575 [11] P. de la Puente, D. Rodriguez-Losada, Feature based graph-SLAM in structured environments, *Autonomous Robots* 37 (2014) 243–260.

- [12] F. Labrosse, The visual compass: Performance and limitations of an appearance-based method, *Journal of Field Robotics* 23 (10) (2006) 913–941.
- [13] N. Bellotto, K. Burn, E. Fletcher, S. Wermter, Appearance-based localization for mobile robots using digital zoom and visual compass, *Robotics and Autonomous Systems* 56 (2) (2008) 143–156.
- [14] J. Košecká, W. Zhang, Video Compass, in: *Proc. 7th European Conference on Computer Vision*, 2002, pp. 476–491.
- [15] J. Montiel, A. Davison, A visual compass based on SLAM, in: *Proc. IEEE International Conference on Robotics and Automation*, 2006, pp. 1917–1922.
- [16] G. L. Mariottini, S. Scheggi, F. Morbidi, D. Prattichizzo, An accurate and robust visual-compass algorithm for robot-mounted omnidirectional cameras, *Robotics and Autonomous Systems* 60 (9) (2012) 1179–1190.
- [17] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, R. Vincent, Efficient sparse pose adjustment for 2D mapping, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 22–29.
- [18] K. Konolige, K. Chou, Markov localization using correlation, in: *Proc. of the 16th International Joint Conference on Artificial Intelligence*, Vol. 2, Stockholm, Sweden, 1999, pp. 1154–1159.
- [19] E. Olson, Real-time correlative scan matching, in: *Proc. IEEE International Conference on Robotics and Automation*, IEEE, Kobe, Japan, 2009, pp. 4387–4393.
- [20] M. J. Gallant, J. A. Marshall, B. K. Lynch, Estimating the heading of a Husky mobile robot with a LiDAR compass based on direction maps, in: *Proc. 10th International Conference on Intelligent Unmanned Systems*, Montreal, Canada, 2014.
- [21] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart, A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2005, pp. 1929–1934.
- [22] S. T. Pfister, S. I. Roumeliotis, J. W. Burdick, Weighted line fitting algorithms for mobile robot map building and efficient data representation, in: *Proc. IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 14–19.
- [23] S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2006.

- [24] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Cambridge, MA, 2005.
- 620 [25] S. J. Julier, J. K. Uhlmann, A new extension of the Kalman filter to non-linear systems, in: *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, 1997, pp. 182–193.
- 625 [26] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A. Kleiner, J. Tardos, A comparison of SLAM algorithms based on a graph of relations, in: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009, pp. 2089–2095.
- [27] I. Dryanovski, W. Morris, Laser scan matcher, http://wiki.ros.org/laser_scan_matcher, accessed: 2014-10-30.
- 630 [28] A. Censi, An ICP variant using a point-to-line metric, in: *Proc. IEEE International Conference on Robotics and Automation*, Pasadena, CA, 2008, pp. 19–25.