Lidar Scan Registration Robust to Extreme Motions

Simon-Pierre Deschênes, Dominic Baril, Vladimír Kubelka, Philippe Giguère, François Pomerleau
Northern Robotics Laboratory, Université Laval, Quebec City, Quebec, Canada
{simon-pierre.deschenes, francois.pomerleau}@norlab.ulaval.ca

***Abstract*—Registration algorithms, such as Iterative Closest Point (ICP), have proven effective in mobile robot localization algorithms over the last decades. However, they are susceptible to failure when a robot sustains extreme velocities and accelerations. For example, this kind of motion can happen after a collision, causing a point cloud to be heavily skewed. While point cloud de-skewing methods have been explored in the past to increase localization and mapping accuracy, these methods still rely on highly accurate odometry systems or ideal navigation conditions. In this paper, we present a method taking into account the remaining motion uncertainties of the trajectory used to de-skew a point cloud along with the environment geometry to increase the robustness of current registration algorithms. We compare our method to three other solutions in a test bench producing 3D maps with peak accelerations of $200\,\mathrm{m/s^2}$ and $800\,\mathrm{rad/s^2}$. In these extreme scenarios, we demonstrate that our method decreases the error by $9.26\,\%$ in translation and by $21.84\,\%$ in rotation. The proposed method is generic enough to be integrated to many variants of weighted ICP without adaptation and supports localization robustness in harsher terrains.**

***Index Terms*—lidar, point weighting, registration, ICP, localization, mapping, de-skewing, collision, extreme motion**

I. Introduction

Time-critical operations, such as search and rescue, require mobile robots to be deployed in increasingly complex environments and at higher velocities. In these conditions, the risk of such a system to sustain a collision is inevitably increased. Thus, the ability to recover from collisions is fundamental to expand the autonomy of robots in large and unstructured environments. This research topic has received a lot of attention over the last few years but was mostly related to Unmanned Aerial Vehicles (UAVs) being mechanically collision-resilient [1]–[3]. Also related to flying, highly reactive controllers have been explored to recover from collisions [4], [5]. However, to our knowledge, no work has been done on increasing the robustness of localization and mapping specifically to collisions.

High velocities and accelerations cause severe distortions in data acquired by lidar sensors. This is due to the assumption that all points from the same scan are measured from the same origin, even if the lidar is subject to high motion during a scan. Data distortions are manifested through a skewing effect in the measured point cloud, causing errors in point locations. This skewing effect is proportional to the lidar speed during a scan, which is typically high when the sensor sustains a collision, as shown in Fig. 1. If significant enough, the scan skewing can lead to localization failure due to inconsistencies of the environment representation. A naive solution would be to ignore incoming scans when sustaining a collision, but this would cause discontinuities in the robot trajectory, also potentially leading to localization failure. To mitigate scan skewing, point cloud de-skewing methods have proven effective [6]. Point cloud de-skewing consists of applying a correction to the measured point cloud, taking lidar motion within a scan into account. However, state-of-the-art methods assume accurate odometry [7], which is jeopardized during extreme motions.

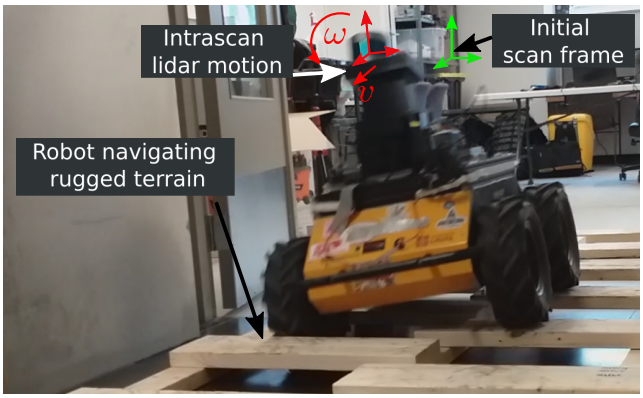

Fig. 1: A mobile robot navigating on controlled rugged terrain. Due to the high velocities sustained by the robot, to which a lidar sensor is attached, rapid motion is induced within the acquisition time of a scan. This motion introduces a point cloud skewing effect, potentially leading to large localization errors. In green is the fixed reference frame of the scan and in red is the moving frame during a scan.

In this work, we address this challenge by proposing a model to compute weights for each point in a scan based on motion uncertainties and environment geometry. These weights are then directly used in the minimization process of the registration algorithm to increase localization accuracy and robustness. The key contribution of this work is threefold:

1) a novel motion uncertainty-based point weighting model, allowing the robustness of registration algorithms to be improved when subject to high velocities and accelerations, requiring neither high-fidelity pose estimation nor prior knowledge of the environment;
2) experimental comparison of our proposed approach with state-of-the-art approaches on a dataset gathered at high velocities and accelerations; and
3) a highlight of the impact of point cloud skewing on the accuracy of localization and mapping.

II. Related Work

Over the last few decades, the Iterative Closest Point (ICP) registration algorithm [8], [9] has been widely used in robotics

© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

for localization and mapping. Since its introduction, many variants of ICP were proposed to increase its robustness. For instance, in the work of Phillips *et al.* [10], the Fractional Root Mean Squared Distance (FRMSD) measure was introduced to allow simultaneous minimization of alignment error and estimation of overlap ratio between two point clouds. This allowed the registration of point clouds with varying overlap ratios, without having to manually tune this parameter each time the algorithm is run, making registration more robust. Granger *et al.* [11] proposed an ICP variant based on the Expectation-Maximization (EM) algorithm, which registers point clouds at multiple scales to increase registration robustness. However, these solutions work at the point cloud level and do not take into account external factors that can affect scan accuracy. Therefore, when point clouds suffer from distortions due to rapid sensor motion, the accuracy of the algorithm is reduced. In this work, we propose a weighting model that can be integrated with prior efforts on the robustness of variable overlaps without modification.

To solve the issue of distortions, point cloud de-skewing algorithms were introduced. In the work of Bosse *et al.* [12], scan matching and lidar trajectory were optimized jointly to de-skew and register point clouds. Their algorithm performed well at linear speeds up to $6\,\mathrm{m/s}$, but at low angular speeds. Moreover, it was tested under unknown accelerations. In a two-step approach, Zhang *et al.* [13] first registered consecutive scans to estimate the lidar motion between them. Then, they used this motion estimate to correct point cloud distortions, by assuming constant linear and angular velocities during a lidar scan. While showing interesting results at speeds up to $0.5\,\mathrm{m/s}$, their algorithm was not tested at high speeds and accelerations. In the work of Renzler *et al.* [7], a high-accuracy odometry system is used to compute the wheeled vehicle pose at a rate of $50\,\mathrm{Hz}$, enabling correction of motion distortion in lidar point clouds. Their algorithm was tested at linear speeds up to $25\,\mathrm{m/s}$ and under linear accelerations up to $10\,\mathrm{m/s^2}$, but at unknown angular speeds. Zhao *et al.* [14] developed a four-step odometry and mapping method to achieve real-time mapping on highways. They estimated the vehicle pose by integrating Inertial Measurement Unit (IMU) measurements, allowing for point cloud de-skewing. Their method was tested at linear speeds up to $20\,\mathrm{m/s}$ and at angular speeds up to approximately $1.05\,\mathrm{rad/s}$, but at low accelerations, inherent to highway navigation. For their part, Qin *et al.* [15] developped an Error-State Kalman Filter (ESKF) to localize in real time. Using IMU and lidar odometry, they were able to de-skew point cloud features. Their algorithm was tested at unknown speeds and accelerations. The Ceres solver was used in the work of He *et al.* [6], to refine the IMU trajectory during a scan, thereby minimizing point cloud deformations after de-skewing. Their algorithm was tested at a linear speed of up to $6.94\,\mathrm{m/s}$, but at unknown angular speeds and accelerations. Furthermore, to avoid distortion caused by high displacement during a scan, they discarded $10\,\%$ of the data. Our experimental setup pushed the validation of our model far beyond de-skewing algorithm capabilities, with linear speeds and accelerations up to $3.5\,\mathrm{m/s}$ and $200\,\mathrm{m/s^2}$, with rotational speeds and accelerations up to $11\,\mathrm{rad/s}$ and $800\,\mathrm{rad/s^2}$.

As can be seen, the aforementioned work either lacked testing at high speeds and high accelerations, did not perform well or needed the use of a high-accuracy odometry system to work properly when subject to aggressive motion. Instead, we propose a solution to mitigate the effect of such motion using a weighting approach similar to Al-Nuaimi *et al.* [16]. In their work, weights were assigned to each point of a scan, based on azimuth scanning angle and local surface curvature to scale the alignment error of matched points. While showing significant improvements in simulation, their solution was tested at unknown speeds and accelerations on real-world data and showed little improvement. However, in the method we propose, more elaborate weights will be used and compared with their model. We will demonstrate significant improvements on real lidar scans acquired at high speeds and under high accelerations. Moreover, our solution can be used in conjunction with de-skewing approaches for further improvement of localization accuracy, allowing an increase in performance of the aforementioned methods.

III. Theory

Typical 3D lidars sense their environment by sweeping it with an array of laser beams rotating around their vertical axis. In order to construct a point cloud from the data acquired during a scan, the lidar is assumed to be stationary throughout the scan. All points are consequently presumed to be located within the same coordinate frame. However, this assumption does not hold most of the time for a mobile robot, as it is highly probable that the sensor will move within a scan. Consequently, when all points of a scan are expressed in the same coordinate frame without consideration for this lidar displacement, the resulting point cloud is skewed. This skewing effect increases proportionally with the linear and angular displacements of the lidar during a scan.

Skewed point clouds will degrade the accuracy of any localization and mapping framework, starting with the registration algorithm. Given a reading point cloud $\mathcal{P}$ and a reference point cloud $\mathcal{Q}$, the ICP algorithm aims at finding the transformation $\boldsymbol{T}$ that minimizes the alignment error between $\mathcal{P}$ and $\mathcal{Q}$. In the context of localization and mapping, the reading $\mathcal{P}$ is the latest point cloud acquired with the lidar and the reference $\mathcal{Q}$ is the map of the environment available after registering the last point cloud. We can recover the pose of a robot over a trajectory by concatenating all of the rigid transformations $\boldsymbol{T}$ obtained from past registrations since the start of the trajectory. Of course, this pose estimation will drift as small errors will be integrated throughout the complete trajectory, and skewed point clouds will precipitate this phenomenon.

For example, given a globally-coherent map $\mathcal{Q}$ and a skewed reading $\mathcal{P}$ of a square room, the distortion will prevent both point clouds from aligning perfectly. ICP will thus find a transformation that distributes this alignment error throughout the whole reading $\mathcal{P}$, as shown in Fig. 2. Therefore, because of the skew in the reading $\mathcal{P}$, the rigid transformation $\boldsymbol{T}$ will

not represent the actual displacement of the lidar since the last registration, causing a loss of accuracy in localization. Furthermore, if a skewed reading $\mathcal{P}$ is merged into the map, the newly added points will blur the overall structure, further increasing the loss of accuracy of the pose estimation. Given that the mapping and the localization processes alternate at each scan, the overall robustness of the pose estimation framework can be rapidly jeopardized.

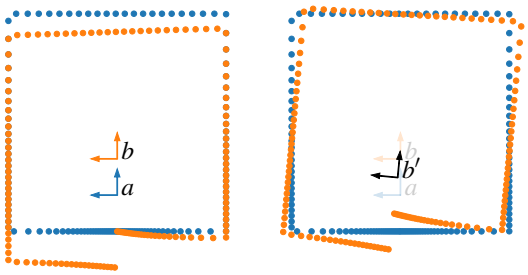

Fig. 2: A toy example demonstrating the impact of motion on the registration of a square room. Blue points represent the globally consistent map $\mathcal{Q}$ of the environment, with its origin pose a marked with blue arrows. Yellow points were acquired while the lidar was moving from pose a to pose b, resulting in a heavily skewed point cloud. *Left*: Before registration. *Right*: After registration, where the new pose b' is extracted from the resulting rigid transformation $\boldsymbol{T}$ computed by ICP. Although the map combining both point clouds seems crisper on the right, the estimated pose of the lidar is wrong.

A. De-skewing

De-skewing methods take into account the lidar displacement to move each point $\boldsymbol{p}_i$ in a common coordinate frame to correct a point cloud $\mathcal{S}$ with a cardinality $n = |\mathcal{S}|$. One can decide that the origin of this coordinate frame is the pose of the lidar at the acquisition time of the first range measurement. An assumption is made that each individual point $\boldsymbol{p}_i$ has its own timestamp t_i. Let ${}^1_i\boldsymbol{T}$ be the rigid transformation to convert between the lidar coordinate frame at time t_i and the lidar coordinate frame at the time of acquiring the first point of the scan and ${}^i\boldsymbol{p}_j$ be the j^{th} point of a scan expressed in the lidar coordinate frame at time t_i. To construct the de-skewed point cloud $\hat{\mathcal{S}}$, we need to compute

$$\hat{\mathcal{S}} = \left\{ {}^1\boldsymbol{p}_1,\ {}^1_2\boldsymbol{T}\,{}^2\boldsymbol{p}_2,\ {}^1_3\boldsymbol{T}\,{}^3\boldsymbol{p}_3,\ \cdots,\ {}^1_n\boldsymbol{T}\,{}^n\boldsymbol{p}_n \right\}. \tag{1}$$

In the remainder of this article, we will simplify the notation by stating that a point is expressed in its own coordinate frame when no subscript is present (i.e., $\boldsymbol{p}_i = {}^i\boldsymbol{p}_i$). From Eq. 1, we can see that it is necessary to know the pose of the lidar each time a range measurement occurs in order to de-skew point clouds. In other words, we need a finer-grained approximation of the trajectory over time than what a registration algorithm can give.

B. Trajectory estimation

In order to de-skew point clouds in real-time, it is typical to use IMU measurements and point cloud registration jointly [14], [15]. Indeed, to compute the de-skewed point cloud $\hat{\mathcal{S}}$, we must first compute the pose ${}^1_i\boldsymbol{T}$ of the lidar at each measurement time t_i, which occurs at a frequency of 20 000 Hz. To do so, we use linear interpolation between lidar positions and orientations estimated at a frequency of 100 Hz. For lidar orientation estimation, a Madgwick filter [17] based on gyroscope and accelerometer measurements is used. It provides an orientation estimate at the same frequency as IMU measurements (i.e., 100 Hz). As for the position of the lidar, it is estimated by integrating its linear velocity. The latter is obtained by fusing two velocity estimates. The first one corresponds to the single integration of the accelerometer measurements and yields estimates at the same frequency as IMU measurements. The other, which is more precise but less frequent, is computed by deriving the two last positions outputted by our localization and mapping framework and yields estimates at the scanning frequency (i.e., 10 Hz). With those two computed linear velocities, we obtain a reliable velocity estimate $\boldsymbol{v}$ at the same frequency as we receive IMU measurements, without long-term drift. This reliable estimate can then be integrated to compute the lidar position at a frequency of 100 Hz. Using the aforementioned techniques, we are able to compute a reasonable estimate of the pose of the lidar ${}^1_i\boldsymbol{T}$ within a scan. The de-skewed point cloud $\hat{\mathcal{S}}$ is then computed using Eq. 1 and registered in the map using our localization and mapping framework to obtain a more precise pose estimate. The frequencies at which the sensors operate and at which the different estimates are computed are shown in Fig. 3.

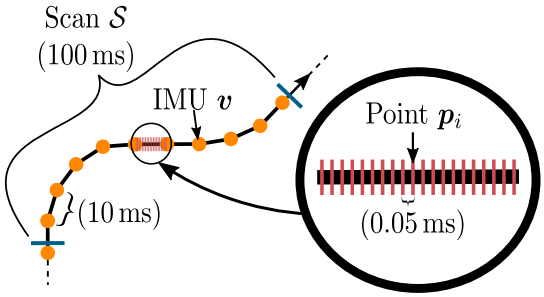

Fig. 3: Measurement delays for all sensors used in this framework. The blue lines represent the start and end of a lidar scan, acquired at a rate of 10 Hz. The orange dots represent the IMU measurements, acquired at a rate of 100 Hz, used to estimate the lidar velocity $\boldsymbol{v}$ during a scan. The red lines represent individual lidar measurements $\boldsymbol{p}_i$, acquired at a rate of 20 000 Hz.

Although the reconstruction of a trajectory using an IMU and a lidar yields a reasonable estimate, limitations remain. As described by Underwood *et al.* [18], common error sources in mapping include sensor noise, timing errors, and sensor miscalibration. Even though the trajectory is estimated over a short period of time (i.e., between two scans), accelerations need to be integrated twice, while the angular velocities need to be integrated once. These integrals will accumulate error over the duration of the scan leading to an increase of uncertainty over the trajectory. Smaller IMU typically found in mobile robots rely on Micro Electro-Mechanical Systems (MEMS) to measure linear acceleration and rotational velocity. Their

measurements suffer from scaling errors and bias, component misalignment errors, non-linearities and vary depending on the temperature [19]. Both the IMU and lidar must produce a globally coherent timestamp, which may require dedicated hardware to avoid drift and resolve any offsets. Finally, to be able to estimate the trajectory tracking the centre of the lidar, the pose of the IMU must be calibrated with respect the lidar. This calibration will be estimated up to some accuracy, and will be subject to wear of the fixation during the use of the robot. Any of these sources of uncertainty will produce a point cloud deformed with residual errors. Thus, we propose a way to estimate such uncertainties and to exploit this information in the registration process.

C. Registration point weighting models

To limit the impact of uncertainty in the trajectory estimate, we propose to lower the weights w_i of individual points $\boldsymbol{p}_i$ of the reading $\mathcal{P}$ and reference $\mathcal{Q}$ point clouds that were more probably affected by motion distortion. The ICP algorithm relies on the distance $\boldsymbol{e}$ between all matched points m to minimize a cost function J. In this process, it is typical to compute a covariance matrix $\boldsymbol{\Sigma}_l$ representing the local structure around a point in the reference $\mathcal{Q}$. In this work, we also add the covariance matrix $\boldsymbol{\Sigma}_n$ defining the sensor noise and express it as an isotropic noise using the pessimistic model of Pomerleau *et al.* [20], by using a scalar matrix with its diagonal elements as σ_n^2. Finally, related to the uncertainty on the position of a point $\boldsymbol{p}_i$ caused by an error in the trajectory estimation, we added the covariance matrix $\boldsymbol{\Sigma}_s$. Again, we make the assumption that the uncertainty defined by $\boldsymbol{\Sigma}_s$ is isotropic, by using a scalar matrix with its diagonal elements as σ_s^2. Following the work of Babin *et al.* [21], we can approximate the point-to-Gaussian cost function $\mathrm{J}_{\text{p-g}}$ of ICP, defined as

$$\mathrm{J}_{\text{p-g}} = \sum_{m=1} \left(\boldsymbol{e}^T \left(\boldsymbol{\Sigma}_l + \boldsymbol{\Sigma}_n + \boldsymbol{\Sigma}_s \right)^{-1} \boldsymbol{e} \right)_m , \tag{2}$$

to produce the more commonly used weighted point-to-plane cost function $\mathrm{J}_{\text{p-n}}$ using

$$\begin{aligned} \mathrm{J}_{\text{p-g}} &\approx \mathrm{J}_{\text{p-n}} \\ &= \sum_{m=1} \left(w (\boldsymbol{e} \cdot \boldsymbol{n})^2 \right)_m , \end{aligned} \tag{3}$$

where

$$w = \frac{1}{\sigma_n^2 + \sigma_s^2} \tag{4}$$

and where the approximation is that the covariance $\boldsymbol{\Sigma}_l$, related to the local structure, is simplified to use only the surface normal vector $\boldsymbol{n}$, and $\boldsymbol{e}$ is the euclidean distance between two matched points. In the following sections, three different models to compute the skew-related position uncertainty σ_s^2 are proposed in increasing order of complexity.

1) Time-based Weighting (TW): The TW model only takes into account the time at which points were acquired since the beginning of the scan to compute their skew-related uncertainty. The intuition behind this model is that, as more time passes since the beginning of the scan, the lidar can cover a greater linear and angular distance, thus point position uncertainty increases. The position uncertainty σ_{si} of the i^{th} point of a scan is computed as $\sigma_{si} = c_1 t_i$. where $c_1 = 0.25$ is a scaling factor identified empirically to account for modelling errors.

The uncertainty grows linearly with time to be pessimistic and to account for possible bias in the velocities estimated by our odometry. The left-most plot of Fig. 4 shows the uncertainty that would be computed for each point of the toy example measured point cloud by using the TW model.

2) Velocity and Time-based Weighting (VTW): The VTW model takes into account the lidar linear and angular speed uncertainties σ_v and σ_ω about each axis to compute the skew-related uncertainties of the points of a scan $\mathcal{S}$. The intuition behind this model is that if the estimated lidar linear and angular velocities $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are erroneous, point clouds will be inaccurately de-skewed, thus point position uncertainty will increase. In order to compute the position uncertainty of a point, the point is rotated around the origin and translated by an amount proportional to σ_v and σ_ω. The translation parameters $\boldsymbol{\delta} = \left[\delta_x, \delta_y, \delta_z\right]^T$ to apply to the points are computed in the following way:

$$\delta_x = \int_{t=0}^{t=t_i} \sigma_v(v_x(t))dt, \tag{5}$$

where $v_x(t)$ is the linear speed of the lidar along the x axis at time t. δ_y and δ_z are computed similarly, but along their respective axis. The rotation parameters $\boldsymbol{\theta} = \left[\theta_x, \theta_y, \theta_z\right]^T$ to apply to the points are computed in the following way:

$$\theta_x = \int_{t=0}^{t=t_i} \sigma_\omega(\omega_x(t))dt, \tag{6}$$

where $\omega_x(t)$ is the estimated angular speed of the lidar around the x axis at time t. θ_y and θ_z are computed similarly, but around their respective axis. Our odometry linear $\sigma_v(\cdot)$ and angular $\sigma_\omega(\cdot)$ speed uncertainty models are described in Sec. V-A. The position uncertainty σ_{si} associated with the point $\boldsymbol{p}_i$ of the currently-processed scan is computed as follows with this model:

$$\sigma_{si} = \frac{c_2}{2} \max(\|(\mathrm{rot}(\boldsymbol{p}_i, \pm\boldsymbol{\theta}) \pm \boldsymbol{\delta}) - (\mathrm{rot}(\boldsymbol{p}_i, \pm\boldsymbol{\theta}) \pm \boldsymbol{\delta})\|), \tag{7}$$

where $c_2 = 2$ is a scaling factor identified empirically to account for modelling errors and $\mathrm{rot}(\boldsymbol{p}, \boldsymbol{\theta})$ is the function that applies the rotation parametrized by $\boldsymbol{\theta}$ to the point $\boldsymbol{p}$. Eq. 7 allows the computation of the highest de-skewing error a point can suffer from by testing 16 directions in which the velocity estimation error could be made. The second plot from the left of Fig. 4 shows the uncertainty that would be computed for each point of the toy example measured point cloud by using the VTW model.

3) Geometry, Velocity and Time-based Weighting (GVTW): The GVTW model takes into account the lidar linear and angular speed uncertainties σ_v and σ_ω about each axis and the geometry of the environment to compute the skew-related uncertainties of the points of a scan $\mathcal{S}$. The intuition behind the GVTW model is that erroneous velocity estimates $\boldsymbol{v}$ and

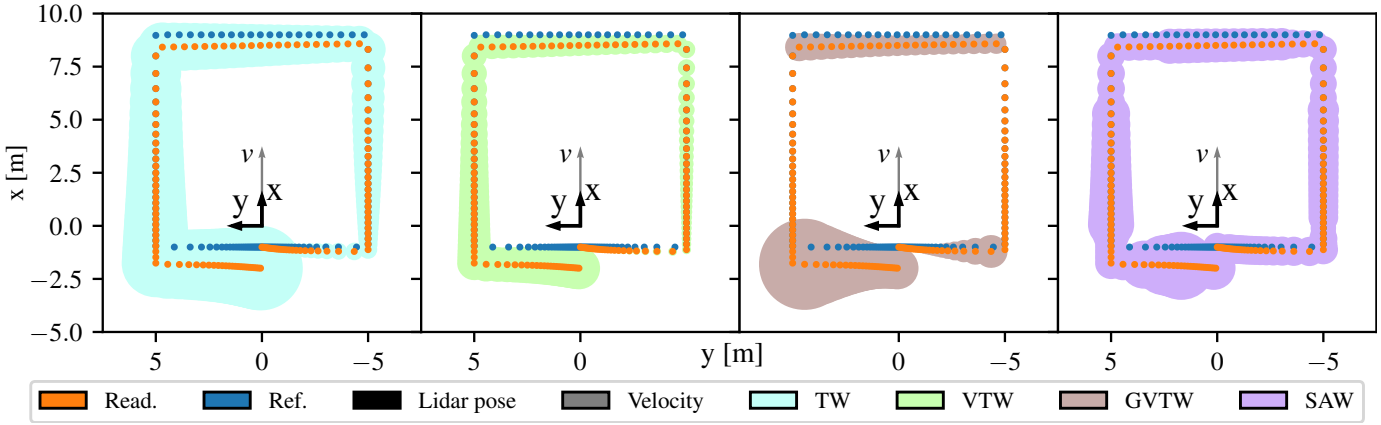

Fig. 4: Uncertainty propagation on a toy example. The lidar scans anti-clockwise, starting from the negative x direction. The measurement $\mathcal{P}$ is orange, and the reference $\mathcal{Q}$ is blue. The initial lidar position is shown in black and its velocity direction in gray. The shaded areas show the skewing uncertainty for each point of the measurement $\mathcal{P}$, for each model.

$\boldsymbol{\omega}$ will lead to an inaccurate de-skewing of a scan $\mathcal{S}$, but not for points on all surfaces. Indeed, if a surface is parallel to the direction in which a velocity estimate error is made, points measured on that surface will still be valid after de-skewing, since such an error will cause points to *slide* along the surface. To compute the position uncertainty of a point $\boldsymbol{p}_i$, the point is rotated around the origin and translated by an amount proportional to to σ_v and σ_ω. Then, the distance that would have been measured if the laser had been fired in this direction is computed. The translation $\boldsymbol{\delta}$ and rotation $\boldsymbol{\theta}$ parameters to apply to the point are computed using Eq. 5 and Eq. 6. The position uncertainty σ_{si} associated with the point $\boldsymbol{p}_i$ of the currently-processed scan is computed as follows with this model:

$$\sigma_{si} = \frac{c_3}{2} \max(|\mathrm{d}(\boldsymbol{p}_i, \pm\boldsymbol{\theta}, \pm\boldsymbol{\delta}, \boldsymbol{n}_i) - \mathrm{d}(\boldsymbol{p}_i, \pm\boldsymbol{\theta}, \pm\boldsymbol{\delta}, \boldsymbol{n}_i)|), \tag{8}$$

with

$$\mathrm{d}(\boldsymbol{p}, \boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{n}) = \frac{(\boldsymbol{p} - \mathrm{rot}(\boldsymbol{\delta}, \boldsymbol{\theta})) \cdot \boldsymbol{n}}{\mathrm{rot}\left(\frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}, \boldsymbol{\theta}\right) \cdot \boldsymbol{n}}, \tag{9}$$

where $c_3 = 4$ is a scaling factor identified empirically to account for modelling errors, and $\boldsymbol{n}_i$ is the surface normal vector of $\boldsymbol{p}_i$. Eq. 8 allows the computation of the highest de-skewing error a point can suffer from by testing 16 directions in which the velocity estimation error could be made. Eq. 9 allows the computation of the distance that would have been measured for a point $\boldsymbol{p}$ if the laser had been fired from a different location and in a different direction. We observed better results when points with an uncertainty above 1 m were removed from the map. This helps registration by matching scan points only with significant points of the map. The third plot from the left of Fig. 4 shows the uncertainty that would be computed for each point of the toy example measured point cloud by using the GVTW model.

4) Scanning Angle-based Weighting (SAW): In order to evaluate the weighting models proposed previously, we use the SAW model proposed by Al-Nuaimi *et al.* [16] as a baseline because, to our knowledge, it is the only prior work addressing point cloud skewing using a point weighting model. The SAW model takes into account the horizontal scanning angle and surface curvature to compute the weight of each point in a scan. Let $w_{si} = \cos(\gamma_i/4)$ be the part of the weight of point $\boldsymbol{p}_i$ which is associated with its horizontal scanning angle γ_i. Let $w_{ci} = \max(0.25, \min(c_i/c_r, 1))$ be the part of the weight of point $\boldsymbol{p}_i$ which is associated with its surface curvature c_i and where c_r is an environment-dependent constant equal to the curvature of a highly curved surface. The final weight w_i of point $\boldsymbol{p}_i$ is the maximum between w_{si} and w_{ci}.

In the work of Al-Nuaimi *et al.* [16], this weighting function was used for the evaluation with non-simulated data. The right-most plot of Fig. 4 shows the uncertainty that would be computed for each point of the toy example measured point cloud by using the SAW model.

IV. Experimental Setup

We conducted experiments with a Clearpath Husky on rugged terrain, as shown in Fig. 1, but the motions the robot underwent were not aggressive enough to induce visible scan deformations. Therefore, to compare the models described in Sec. III-C, we built the mobile sensing platform shown in Fig. 5. A Robosense RS-16 lidar is located on top of the platform, producing scans at a rate of 10 Hz. Additionally, an Xsens MTI-30 IMU returns its own body linear acceleration and angular velocity at a rate of 100 Hz. This platform was mounted on a 30 m linear rail specialized for sensor calibration. It was moved for a wide variety of linear velocities and accelerations, in both directions, while recording sensor measurements. This same sensing platform was then mounted on a turntable and rotated in both directions while recording sensor measurements. Both of the aforementioned experiments were conducted in order to submit the sensors to a maximum excitement range for linear and angular speeds. Afterwards, we added a protective cage to the platform, which can be seen in Fig. 5. A rope was attached to the cage and an operator pulled on the rope in order to initiate the rolling of the cage on the ground at high angular velocities.

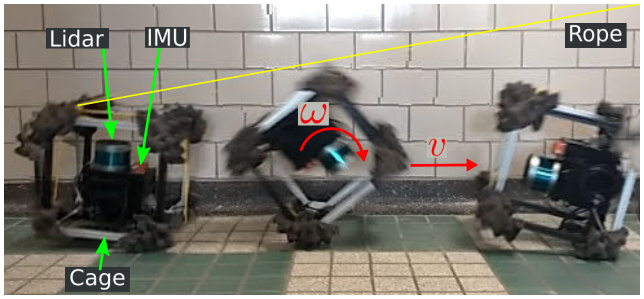

Fig. 5: The experimental platform used for this work. A lidar and an IMU are mounted on a sensing platform, which is protected by a cage. A rope is attached to the platform and pulled on by an operator to induce high velocities and accelerations during lidar scans.

V. Results

Some of the weighting models described in Sec. III-C rely on having an estimate of the speed estimation uncertainties σ_v and σ_ω for each axis. To establish the latter, we first modelled the linear and angular speed uncertainties for one axis of our odometry system from the collected data presented in Sec. V-A. We then compared the different weighting models on real lidar scans acquired at high linear and angular velocities and under high accelerations. Finally, the effects on localization error and mapping are reported in Sec. V-B.

A. Linear and angular speed uncertainties σ_v and σ_ω

To compute the weights of the different models introduced in Sec. III-C, we must find a continuous function that allows the estimation of the linear speed uncertainty σ_v along an axis as a function of the estimated linear speed v along that same axis. Similarly, we should be able to compute the angular speed uncertainty σ_ω as a function of the estimated angular speed ω around an axis.

To do so, we develop equations linking the uncertainties σ_v and σ_ω to registration residuals that we can measure in our data together with v and ω. De-skewing a point cloud with a linear velocity error ϵ is equivalent to not applying de-skewing to a point cloud acquired with a lidar travelling at a velocity of $-\epsilon$. Therefore, the position error of a point $\boldsymbol{p}_i$ along an axis can be retrieved by multiplying its timestamp t_i with the linear speed estimation error σ_v along that axis. Assuming that the velocity throughout a scan is constant, the mean residual registration error r_v caused by de-skewing a point cloud using a linear speed estimate with an error σ_v along an axis can be computed. Indeed, it is approximately the displacement of the lidar in the middle of its scan. Thus, we have

$$r_v \approx \frac{\sigma_v \tau}{2}, \tag{10}$$

where τ is the time needed to complete a whole scan. This equation allows us to express our linear speed uncertainty model σ_v along an axis as a function of the linear speed v estimated by our odometry along that axis. Similarly, it can be shown that the mean residual registration error norm r_ω can be computed as a function of the angular speed estimation error σ_ω around an axis using the following equation:

$$r_\omega \approx \frac{\sigma_\omega \tau d}{2}, \tag{11}$$

where d is the mean distance measurement of the points in the point cloud.

In order to obtain the uncertainty models σ_v and σ_ω from real data, we mounted the platform on the rail and the turntable described in Sec. IV, to acquire scans at different linear and angular speeds. Then, we computed the mean norm r_v or r_ω of registration residuals of the de-skewed scans against a previously-built dense 3D map of the environment.

For the linear speed uncertainty σ_v, these residuals were computed using the euclidean distances $\boldsymbol{e}_i$ projected on surface normals $\boldsymbol{n}_i$ only for points $\boldsymbol{p}_i$ lying on surfaces perpendicular to lidar motion. Those points were selected because, unlike points lying on surfaces parallel to the motion, their error increases when inaccurate de-skewing caused by linear motion occurs. Since the map is assumed to be accurate, the scan registration residuals were therefore due to *i)* the noise in lidar measurements and *ii)* the skew caused by inaccurate de-skewing. Fig. 6-a shows the median and quartiles of the mean registration residuals as a function of the linear speed at which point clouds were acquired. If the lidar is not moving,

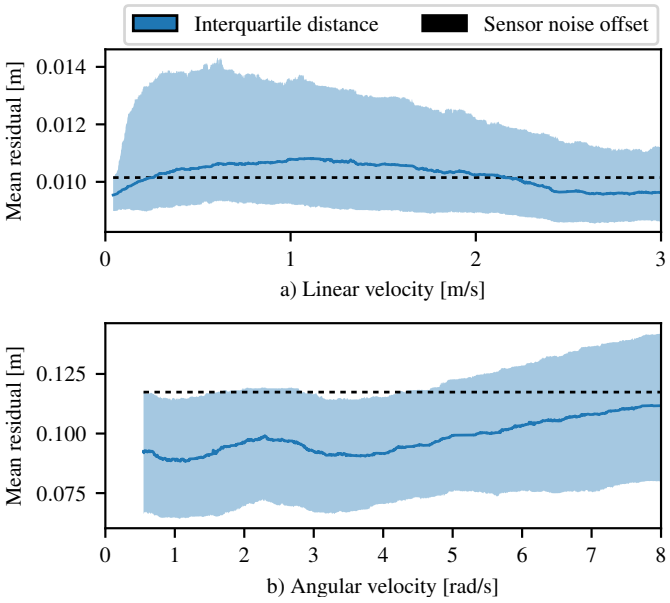

Fig. 6: Registration residual as a function of a) linear speed and b) angular speed. The median is the solid blue line, and the shaded zones represent the first and third quartiles. The black dotted line represents the residual offset caused by sensor measurement noise. For a), only residuals from surfaces *perpendicular* to the lidar motion are used.

the velocity estimation error made by our odometry system should be low enough to consider that all registration residuals come from lidar measurement noise. Furthermore, the registration residuals caused by measurement noise are assumed to be independent of lidar velocity. Therefore, by removing

the registration residual computed at a speed of $0\,\mathrm{m/s}$ from computed residuals r_v, we obtain the residual caused by inaccurate de-skewing. This error is depicted in Fig. 6-a with a black dotted line.

After converting the residuals shown in Fig. 6-a to linear speed uncertainties using Eq. 10, the third quartile of the uncertainty distribution is fitted using a log-normal function. This provides an empirical model of the linear speed uncertainty σ_v along an axis of our odometry system as a function of the linear speed v it estimates along that axis, such that

$$\sigma_v(v) = \frac{\lambda}{\beta v\sqrt{2\pi}} \exp\left(-\frac{\log^2(\frac{v}{\kappa})}{2\beta^2}\right), \tag{12}$$

where $\beta = 1.1$ is the shape factor, $\kappa = 1.9$ is the scale factor and $\lambda = 0.222$ is the function gain. These values were empirically identified based on the data presented in Fig. 6-a.

Similarly, for the angular uncertainty model σ_ω, we acquired scans at different angular speeds and accelerations and computed the mean residual registration error norms r_ω using a dense 3D map of the environment. However, to highlight scan distortions, the euclidean distance $\boldsymbol{e}_i$ of all points $\boldsymbol{p}_i$ in the map was used. Euclidean distance was used because we observed that it increases more than its projection on surface normals when subject to inaccurate de-skewing caused by rotational motion. Fig. 6-b shows the median and quartiles of registration residuals as a function of the angular speeds at which point clouds were acquired. After converting the residuals shown in Fig. 6-b to angular speed uncertainties using Eq. 11, the third quartile of the uncertainty distribution is fitted using a polynomial function. This provides an empirical model $\sigma_\omega(\omega) = (\omega/\phi)^3$ of the angular speed uncertainty σ_ω around an axis of our odometry system with a scaling parameter ϕ. We have empirically identified $\phi = 16$. Although our odometry system is quite generic, the aforementioned rail and turntable experiments can be repeated with different systems to identify better-suited values for β, κ, λ and ϕ.

B. Impact of proposed method on localization and mapping

Using the platform and protective cage described in Sec. IV, we acquired scans at linear speeds and accelerations up to $3.5\,\mathrm{m/s}$ and $200\,\mathrm{m/s^2}$ and angular speeds and accelerations up to $11\,\mathrm{rad/s}$ and $800\,\mathrm{rad/s^2}$ on a total of 46 runs. For each run, we recorded scans and used the ICP algorithm with the methods described in Sec. III-C to compute the transformation between the initial and final poses of the lidar. No prior map of the environment was used to compute these transformations. They were computed using only the measurements acquired by the platform during a run. For all runs, we used point-to-plane minimization combined with FRMSD [10] to handle varying overlap ratios between point clouds. We also used a voxel grid with a cell size of $5\,\mathrm{cm}$ to down-sample input point clouds.

To compute the ground truth transformations between the initial and final poses of the lidar, we registered the first and last scans of each run in a dense 3D map of the environment built while moving slowly. This allowed us to compare the impact of de-skewing and point weighting on localization and mapping accuracy, as shown in Fig. 7. The translation and rotation localization errors of the different models are relative to the distance travelled by the platform in a run. In the graph, the blue boxes are the resulting localization errors when no point weighting is used. The left plots show the results of using point weighting models without de-skewing, and the right plots show the result of their use after point cloud de-skewing. As can be seen, without any point weighting model and without de-skewing, the median of the relative localization error is $27.23\,\mathrm{cm/m}$ and $7.04\,\mathrm{deg/m}$. However, when point cloud de-skewing alone is used, the median translation error decreases to $3.11\,\mathrm{cm/m}$ and the median rotation error decreases to $0.79\,\mathrm{deg/m}$. These results show that de-skewing methods can significantly enhance localization and mapping accuracy. Indeed, by correcting the distortion in point clouds, de-skewing methods allow greater overlaps between the scans and the map, leading to a crisper map, and thus to better localization. However, if no de-skewing method can be applied, the VTW point weighting model allows a decrease in the median localization error by a factor 2.2 in translation and by a factor 3.0 in rotation, which is the best improvement out of all of the studied models. However, the best combination consists of adding the GVTW model after point cloud de-skewing, where the median localization error is reduced by $9.26\,\%$ in translation and by $21.84\,\%$ in rotation when compared to not using point weighting on a de-skewed point cloud. Both the VTW and GVTW models also allow the interquartile range to be significantly reduced, meaning that they produce a more precise

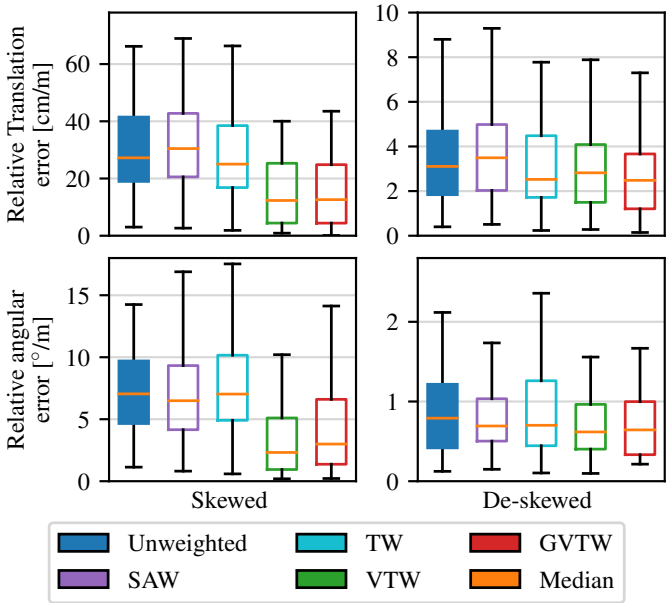

Fig. 7: Relative translation and angular errors for all models. Results were gathered offline, based on 46 runs which consisted of rolling a platform with a lidar and an IMU attached to it. The error is computed as the difference between the final estimated pose by each model and the ground truth pose. The ground truth is computed by registering the final recorded point cloud in a static map.

localization. It is also worth noting that the SAW model [16] does not improve the mean localization accuracy in our tests.

Demonstrating qualitatively the impact of our weights on mapping, Fig. 8 shows a 3D reconstruction of an environment built while applying peak accelerations up to $194.98\,\mathrm{m/s^2}$ on our sensing platform. All three maps were built using the same data, with the GVTW weighting model only used for the rightmost map. The error was computed against a ground truth map built while moving slowly the sensors. De-skewing significantly impacts the mean error by reducing it from $21.95\,\mathrm{cm}$ to $7.27\,\mathrm{cm}$. Further improvement is gained by using GVTW, leading to a mean error of $6.86\,\mathrm{cm}$. For example, one can observe an improvement in accuracy on the left wall.

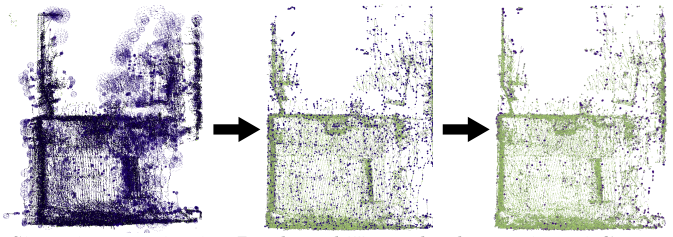

Fig. 8: Top view of an indoor garage mapped under high velocities and accelerations. Points in dark purple have high error compared to a ground truth map.

VI. Conclusion

In this paper, we address the challenge of localization accuracy when a mobile robot is experiencing extreme motions. To this effect, we introduced three novel motion uncertainty-based point weighting models to increase the robustness of point cloud registration algorithms under such conditions. We quantified that using a de-skewing solution reduces the localization error by an order of magnitude. However, uncertainty remains in the trajectory estimation generating a localization error that can be further improved by using our solution. Future improvements include removing the assumption that a scan is generated only after a full revolution, removing empirically-identified scaling factors from our method and testing with a hand-thrown device to expose our solution to even more extreme scenarios.

Acknowledgment

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through the grant CRDPJ 527642-18 SNOW (Self-driving Navigation Optimized for Winter).

References

[1] A. Briod, P. Kornatowski, J. C. Zufferey, and D. Floreano, "A Collision-Resilient Flying Robot," *JFR*, vol. 31, no. 4, 2014.

[2] P. M. Kornatowski, S. Mintchev, and D. Floreano, "An Origami-Inspired Cargo Drone," in *IROS*, 2017.

[3] L. Dilaveroglu and O. Ozcan, "MiniCoRe: A Miniature, Foldable, Collision Resilient Quadcopter," in *RoboSoft*, 2020.

[4] G. Dicker, F. Chui, and I. Sharf, "Quadrotor Collision Characterization and Recovery Control," in *ICRA*, IEEE, 2017.

[5] S. Wang, N. Anselmo, M. Garrett, R. Remias, M. Trivett, A. Christoffersen, and N. Bezzo, "Fly-Crash-Recover: A Sensor-based Reactive Framework for Online Collision Recovery of UAVs," in *SIEDS*, 2020.

[6] L. He, Z. Jin, and Z. Gao, "De-skewing lidar scan for refinement of local mapping," *Sensors*, vol. 20, no. 7, 2020.

[7] T. Renzler, M. Stolz, M. Schratter, and D. Watzenig, "Increased Accuracy for Fast Moving LiDARS: Correction of Distorted Point Clouds," in *I2MTC*, 2020.

[8] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *ICRA*, 1991.

[9] P. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," in *TPAMI*, vol. 14, 1992.

[10] J. M. Phillips, R. Liu, and C. Tomasi, "Outlier Robust ICP for Minimizing Fractional RMSD," in *3DIM*, 2007.

[11] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *ECCV*, 2002.

[12] M. Bosse and R. Zlot, "Continuous 3D scan-matching with a spinning 2D laser," in *ICRA*, 2009.

[13] J. Zhang and S. Singh, "LOAM: lidar odometry and mapping in real-time," in *RSS*, 2014.

[14] S. Zhao, Z. Fang, H. L. Li, and S. Scherer, "A Robust Laser-Inertial Odometry and Mapping Method for Large-Scale Highway Environments," in *IROS*, IEEE, 2019.

[15] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation," in *ICRA*, 2020.

[16] A. Al-Nuaimi, W. Lopes, P. Zeller, A. Garcea, C. Lopes, and E. Steinbach, "Analyzing LiDAR Scan Skewing and its Impact on Scan Matching," in *IPIN*, IEEE, 2016.

[17] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG Orientation Using a Gradient Descent Algorithm," in *ICORR*, 2011.

[18] J. P. Underwood, A. Hill, T. Peynot, and S. J. Scheding, "Error Modeling and Calibration of Exteroceptive Sensors for Accurate Mapping Applications," *JFR*, vol. 27, no. 1, 2010.

[19] U. Qureshi and F. Golnaraghi, "An Algorithm for the In-Field Calibration of a MEMS IMU," *IEEE Sensors Journal*, vol. 17, no. 22, 2017.

[20] F. Pomerleau, A. Breitenmoser, M. Liu, F. Colas, and R. Siegwart, "Noise Characterization of Depth Sensors for Surface inspections," in *CARPI*, 2012.

[21] P. Babin, P. Dandurand, V. Kubelka, P. Giguère, and F. Pomerleau, "Large-scale 3D mapping of subarctic forests," in *FSR*, 2019.