Correcting Motion Distortion for LIDAR Scan-to-Map Registration

Matthew McDermott, *Graduate Student Member, IEEE*, and Jason Rife, *Member, IEEE*

***Abstract*—Because scanning-LIDAR sensors require finite time to create a point cloud, sensor motion during a scan warps the resulting image, a phenomenon known as *motion distortion* or *rolling shutter*. Motion-distortion correction methods exist, but they rely on external measurements or Bayesian filtering over multiple LIDAR scans. In this letter we propose a novel algorithm that performs snapshot processing to obtain a motion-distortion correction. Snapshot processing, which registers a current LIDAR scan to a reference image without using external sensors or Bayesian filtering, is particularly relevant for localization to a high-definition (HD) map. Our approach, which we call Velocity-corrected Iterative Compact Ellipsoidal Transformation (VICET), extends the well-known Normal Distributions Transform (NDT) algorithm to solve jointly for both a 6 Degree-of-Freedom (DOF) rigid transform between a scan and a map and a set of 6DOF motion states that describe distortion within the current LIDAR scan. Using experiments, we show that VICET achieves significantly higher accuracy than NDT or Iterative Closest Point (ICP) algorithms when localizing a distorted raw LIDAR scan against an undistorted HD Map.**

***Index Terms*—Localization, SLAM, range sensing.**

I. Introduction

A SCANNING LIDAR records data by sweeping an array of laser-ranging sensors across a scene, typically completing a full sweep once every 50–200 ms. Even modest motion of the sensor unit during the sweep creates discernible distortion in the scanned image. By contrast, instantaneous sweeps are assumed in conventional scan-matching algorithms such as Iterative Closest Point (ICP) [1], the Normal Distributions Transform (NDT) [2], and their variants [3], [4], [5], [6]. Although the instantaneous-sweep, distortion-free approximation is reasonable for a static platform observing a static scene, distortion is usually present in LIDAR data captured by a moving platform. Such motion distortion effects impact all LIDAR registration algorithms, whether keypoint-based [7], [8], distributions-based [4], [6], or ML-based [9], [10].

We seek a new approach for LIDAR-based distortion correction that does not rely on external measurements or an estimator, such that snapshot processing is possible, performing registration using only a reference image and the current LIDAR scan. This is particularly important for map-based navigation, where instantaneous position is estimated by matching a single scan to a pre-existing map.

We also seek a distortion-correction methodology that can be integrated with an algorithm that produces a meaningful prediction of measurement uncertainty (e.g. a covariance matrix that quantifies the error for aligning two point clouds). This is especially important for safety-critical navigation systems such as autonomous vehicles [11]. Even sequential-estimator based distortion correction methods, like VICP and LOAM, do not provide meaningful uncertainty quantification, in part because they use scan-matching methods for which accuracy is difficult to predict. Considering residuals between point-to-point correspondences does not adequately describe the true noise in the environment due to sensor noise, surface roughness, and discrete sampling effects [4], [6].

We seek to achieve both goals by reformulating the Normal Distribution Transform (NDT) [2], a scan-matching algorithm for which uncertainty quantification methods exist, dating back to Stoyanov [6]. We recently reformulated NDT (via an algorithm called ICET) to streamline error estimation while also excluding a significant source of error related to the distinction between random noise and structural patterns within covariance ellipsoids fit to the points in each voxel [4], [12]. Our primary contribution in this letter is to modify NDT by adding motion-distortion compensation, in order to enhance registration accuracy and remove systematic biases not accounted for in NDT's uncertainty quantification.

The remainder of the letter is organized as follows. In Section II we discuss related works in the motion distortion space. In Section III refine our definition of motion distortion and demonstrate how relative distortion between two scans can produce ambiguity in the scan-registration process. In Section IV, we formulate our new algorithm, VICET, and discuss specific implementation details. In Section V we compare VICET performance against benchmark algorithms of NDT and ICP. Finally, in Section VI we summarize our key findings and discuss avenues of future work.

II. Related Work

Because motion distortion is a ubiquitous issue for LIDAR scan matching, we compare existing techniques for combating

Manuscript received 25 August 2023; accepted 13 December 2023. Date of publication 25 December 2023; date of current version 5 January 2024. This letter was recommended for publication by Associate Editor T. Stoyanov and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the U.S. Department of Transportation Joint Program Office (ITS JPO) and in part by the Office of the Assistant Secretary for Research and Technology (OST-R). *(Corresponding author: Matthew McDermott.)*

Matthew McDermott is with the Mechanical Engineering Ph.D. Program, Tufts University, Somerville, MA 02145 USA (e-mail: matthew.mcdermott@tufts.edu).

Jason Rife is with the Department of Mechanical Engineering, Tufts University, Somerville, MA 02145 USA (e-mail: jason.rife@tufts.edu).

We recommend the reader explore our open-source code and visualizations at https://github.com/mcdermatt/VICET, which supplements this manuscript.

Digital Object Identifier 10.1109/LRA.2023.3346757

motion distortion and discuss their strengths and limitations. One simple method of alleviating motion-distortion bias is to augment raw LIDAR data with external sensor measurements that describe platform velocity and/or angular velocity, for example, from Global Navigation Satellite System (GNSS), Interial Navigation System (INS), or wheel odometry data. Al-Nuaimi et al. quantify errors introduced by this skewing and propose using inertial data to reduce its effects [13]. Repositories of LIDAR benchmark data often publish pre-rectified point clouds in which distortion effects have already been mitigated using external sensors [14], [15], [16]. Although distortion can be mitigated somewhat, external sensors also introduce new distortion due to sensor noise. This noise may be substantial. For example, GNSS velocity estimates in clear-sky conditions are roughly 3–5 cm/s (one-sigma) [17]) and, in high-multipath environments like urban canyons, are much worse [18]. By comparison, velocity estimates generated from the LIDAR data, itself, can achieve higher accuracy, often better than 1 cm/s.

A second method compensates for motion distortion by estimating platform motion with batch or sequential filtering, for instance, using a Kalman Filter [19]. A filter can suppress the noise from an external sensor, but the filter also introduces lag [20], resulting in suppression of high-frequency velocity-variations and making velocity corrections difficult to match to the correct LIDAR time step, particularly when the platform changes speed rapidly.

A third approach enhances the performance of estimation filters through tighter coupling with the LIDAR image-generation process. For instance, Setterfield exploits factor-graph optimization to break each LIDAR image into highly localized SURF features, each of which represents a sample interval short enough ($<$1 ms) to be essentially distortion free [21]. Similarly, Moosmann et al. deconstruct LIDAR scans into range images from which related features are extracted and used to dewarp the scans using a graph-based approach at the beginning and end of registration [22]. These approaches, however, extract a small number of time-aligned features and discard the remaining LIDAR points; moreover, the approaches rely on highly textured terrain and struggle when viewing large smooth surfaces [23]. Given these limitations, other tightly-coupled estimators like VICP [24] and LOAM [7] mitigate motion distortion for the entire point cloud by estimating a time-varying transform with linear scaling across the interval of a full scan. Both VICP and LOAM obtain velocity estimates from the LIDAR data directly, by analyzing data in batches and constructing a local submap in the process. These tightly-coupled algorithms do not function well unless processing a long series of sequential frames. A large number of recent papers on this topic [25], [26], [27] reaffirm that this is an important open issue for the community.

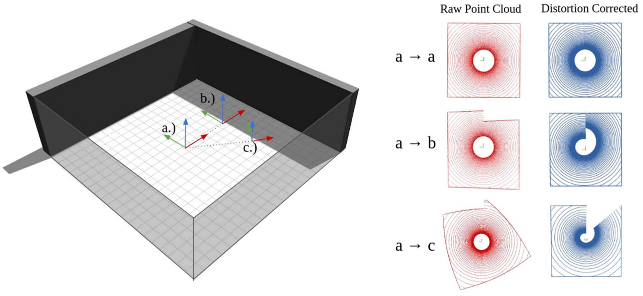

Fig. 1. Simple test scene to study motion-distortion correction. The isometric room view shows three possible LIDAR locations, labeled a, b, and c. At each location, the orientation of the LIDAR unit is described by a set of orthonormal basis vectors. For each configuration, assume the LIDAR beam begins aligned with the red arrow and rotates counterclockwise about the vertical (blue) axis. During the scan, the LIDAR unit either remains stationary (a → a), undergoes forward linear motion (a → b), or undergoes composite translation and rotation (a → c). In each case, the LIDAR beam spins 360° in the frame of the LIDAR stator, which itself moves, resulting in distorted raw point clouds (viewed from above, shown in red). Compensating for stator motion, the raw image can be transformed into room-fixed coordinates (shown in blue), where the square shape of the room is recovered.

III. Motion Distortion Example

Before describing our solution, it is helpful to develop a better understanding of motion distortion through a simple simulation. Consider a mechanically spinning LIDAR unit placed inside a rectangular room, as illustrated in Fig. 1. A LIDAR image (or *scan*) is generated each time the LIDAR spins 360° about the LIDAR's vertical (z) axis. Define the LIDAR scan to begin when the rotating beam aligns with the positive x direction in a coordiante system attached to the LIDAR stator, which we will label the *body* frame. In the figure, starting and ending locations of the LIDAR are indicated by vector-bases where the spin axis is shown in blue and the scan-initiation axis is shown in red. Now consider three cases, one in which the sensor remains stationary during the scan (starting and ending at the pose labeled a), a second in which the sensor undergoes pure translation (starting at a and ending at b), and a third in which the sensor undergoes both translation and rotation (starting at a and ending at c). Each of the three cases creates a distinct LIDAR point cloud shown, from an aerial view, in a column labeled "Raw Point Cloud" (red). In all cases, a circle appears in the middle of the point cloud, reflecting an elevation cutoff, with the simulated LIDAR unit unable to generate samples below 30° under the horizon. The point clouds were generated assuming a uniform rate of motion between the start and end points. Importantly, the raw images are distorted (except in the stationary case), such that the walls of the room do not form a perfect square. Accounting for platform motion, however, points can be shifted from body-frame coordinates to world-frame coordinates. Unwarped point clouds are shown in Fig. 1 as a column of images (blue) labeled "Distortion Corrected." Importantly, these distortion-corrected images all recover the correct room shape, bounded by a square wall. In the last case ($a \rightarrow c$), a wedge of missing data appears because the LIDAR stator rotates clockwise, opposite the counterclockwise rotation of the LIDAR rotor, such that the entire room is not visualized during a single scan.

For map-matching applications, pose is estimated relative to a reference image. For instance, the reference image might be constructed by registering a series of sequential scans from a moving LIDAR, in order to create a mosaic image or high-definition (HD) map. The performance of this registration operation is greatly enhanced if the current scan (captured in the frame of the LIDAR) can be unwarped (converted to the world frame) before registration to the map. In the case of the rectangular room of Fig. 1, the map will look very much like the scan for the static case ($a \rightarrow a$). If a new LIDAR image were generated during translation ($a \rightarrow b$) or combined translation and rotation ($a \rightarrow c$) the resulting raw image (red) will be harder to align with a map of the room than the corresponding distortion-corrected image (blue).

The main focus of this letter is that distortion correction is possible using only the current LIDAR scan and a map. If LIDAR motion is not known, then we can infer the motion by registering the current image to a map while jointly solving for a set of unwarping states to produce the best possible alignment. We develop this idea for simultaneous estimation of pose and motion-correction states in subsequent sections.

We intentionally restrict our analysis to cases where the reference image is an undistorted map. In concept, the process of aligning and unwarping images could also be performed on sequential images, for instance to implement LIDAR odometry. If the reference image were distorted, our estimated motion-distortion correction would reflect relative velocity and angular velocity (changes between scans) but not absolute velocity and angular velocity (motion relative to the ground). In other words, our approach does not correct distortion in the reference image.

IV. Implementation

This section develops a snapshot minimum-squared error (MSE) estimator, which infers both the relative-pose (translation and rotation) and motion-distortion parameters for a current LIDAR scan by comparison to an undistorted reference map. We call the algorithm Velocity-corrected Iterative Compact Ellipsoidal Transformation (VICET).

A. Algorithm Formulation

To start, we define two relevant reference frames: one associated with the map and a second associated with the LIDAR stator. We characterize the map frame by an origin Q_M and a coordinate system defined by the basis M. We define the LIDAR stator frame, which we also called the *body* frame, with a reference point Q_B and a body-fixed coordinate system B. The body frame translates and rotates relative to the map frame over time, as shown in Fig. 2. Although the LIDAR rotor spins about a fixed axis of B, we do not define a distinct reference frame for the rotor, since the LIDAR system converts measurements to the stator frame B using encoder data.

A map matching algorithm seeks to find the rigid transform that aligns the raw LIDAR scan to the map at the beginning of the scan, at time $\tau = 0$. The motion states are inferred to unwarp the LIDAR scan given its motion between the beginning of the scan and the end of the scan, which occurs at $\tau = T$. Using a zero-order hold model, we approximate that the velocity and angular velocity of B relative to M remains constant for times $\tau \in [0, T]$. Now consider a LIDAR measurement vector $\mathbf{p}_\tau^B$. The vector $\mathbf{p}_\tau^B$ describes the position of a feature in the world (or on the map) relative to the body-frame reference point Q_B using body-fixed coordinates (specified by the trailing superscript B) at a time τ (specified by the trailing subscript). The following transformation converts this measurement to $\mathbf{p}_\tau^M$, which expresses the same measurement in map coordinates relative to the map origin Q_M:

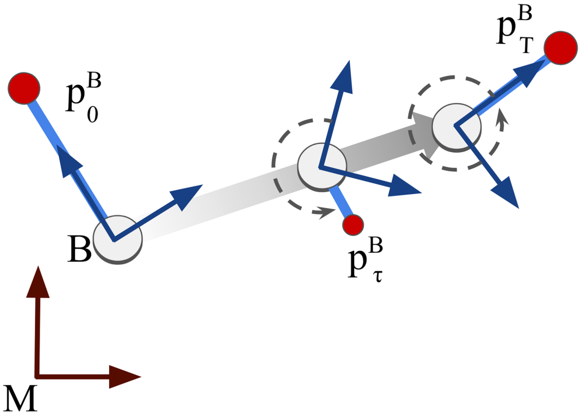

Fig. 2. Frame of the LIDAR stator (or *body*) B moves relative to the map frame M. For illustration purposes, the bases are shown in 2D, viewed from above. The progress of time is indicated by the shaded gray arrow, with the LIDAR beam recording measurements (red dots) as it sweeps a full circle between time $\tau = 0$ and time $\tau = T$.

$$\mathbf{p}_\tau^M = {}^M\mathbf{R}_\tau^B \mathbf{p}_\tau^B + \mathbf{t}_\tau^M \tag{1}$$

Here the rotation matrix ${}^M\mathbf{R}_\tau^B$ converts from the B basis to the M basis at time τ. The translation $\mathbf{t}_\tau^M$ expresses the vector from Q_M to Q_B at time τ, in map-fixed coordinates.

As a next step, we decompose the terms in (1) to relate them to initial states and change states. This decomposition is easiest to develop for the translation term $\mathbf{t}_\tau^M$, which is linear. Given the zero-order hold assumption, we can rewrite the translation using the lever rule to interpolate between an initial translation $\mathbf{t}_0^M$ and a final translation $\mathbf{t}_T^M$. The lever rule can be written

$$\mathbf{t}_\tau^M = \mathbf{x}_0 + s\Delta\mathbf{x} \tag{2}$$

This formulation introduces the following shorthand notation: $\mathbf{x}_0 = \mathbf{t}_0^M$, $\Delta\mathbf{x} = \mathbf{t}_T^M - \mathbf{t}_0^M$, and $s = \tau/T$. The scaled time s, derived from the LIDAR time tag for each measurement, is normalized such that $s \in [0, 1]$. The variables $\mathbf{x}_0$ and $\Delta\mathbf{x}$ are identified specifically because these are states that the estimator will infer.

Just as initial and change states are defined for translation, similar states must be defined for rotation. In our implementation, we use an array of three Euler angles $\mathbf{\Theta_0}$ to describe the initial orientation of the body frame. The rotation matrix is constructed as a function f of the Euler angles $\mathbf{\Theta_0}$.

$${}^M\mathbf{R}_0^B = f(\mathbf{\Theta_0}) \tag{3}$$

The orientation change is described by a set of three angles $\Delta\mathbf{\Theta}$. The final rotation matrix at time T can be related, as follows, to the initial rotation matrix and $\Delta\mathbf{\Theta}$.

$${}^M\mathbf{R}_T^B ({}^M\mathbf{R}_0^B)^{-1} = f(\mathbf{\Delta\Theta}) \tag{4}$$

Compiling the initial and change states together, we can define a combined state vector $\mathcal{X} \in \mathbb{R}^{12\times1}$, which we will estimate by comparing the LIDAR scan to a map.

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}_0 & \Delta\mathbf{x} & \mathbf{\Theta_0} & \Delta\mathbf{\Theta} \end{bmatrix}^T \tag{5}$$

To relate (1) to the states in (5), we use a Taylor series expansion, in which we expand (1) about a reasonable guess. In implementation, we seed the initial position and Euler angles with a rough guess, and we seed the change parameters with zero values. The affine terms of the Taylor series are:

$$\mathbf{p}_\tau^M = \left({}^M\hat{\mathbf{R}}_\tau^B \cdot \delta {}^M\mathbf{R}_\tau^B \right) \mathbf{p}_\tau^B + \hat{\mathbf{t}}_\tau^M + \delta\mathbf{t}_\tau^M. \tag{6}$$

Here we employ a hat notation to identify terms computed using the initial guess. Also, we introduce a δ notation to identify linear perturbations.

Because the translation term is linear, the translation perturbation can be obtained from the derivative of (2):

$$\delta\mathbf{t}_\tau^M = \delta\mathbf{x}_0 + s\delta\Delta\mathbf{x}. \tag{7}$$

The rotation-perturbation $\delta\mathbf{R}$ is obtained from the rotation-matrix derivative, which in the small angle limit is

$$\delta {}^M\mathbf{R}_\tau^B = [\delta\mathbf{\Theta}\times] \ {}^M\hat{\mathbf{R}}_\tau^B. \tag{8}$$

Here $\delta\mathbf{\Theta}$ is an angular correction converted to a cross-product by the $[\_\_\times]$ operator. This matrix, for a generic vector $\mathbf{v}$ with elements $\{v1, v2, v3\}$ is:

$$[\mathbf{v}\times] = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{9}$$

Though Euler angles do not sum in general, summation is a good approximation after linearization, and so $\delta\mathbf{\Theta}$ can be written in terms of initial and change states:

$$\delta\mathbf{\Theta} = \delta\mathbf{\Theta}_0 + s\delta\Delta\mathbf{\Theta} \tag{10}$$

Substituting (7)-(10) into (6), we obtain a linear system of equations relating the correction states to the LIDAR data:

$$\mathbf{p}_\tau^M = {}^M\hat{\mathbf{R}}_\tau^B \mathbf{p}_\tau^B + \hat{\mathbf{t}}_\tau^M + \mathbf{J}_\mathcal{X}\delta\mathcal{X}. \tag{11}$$

Here the the state-perturbation vector $\delta\mathcal{X} \in \mathbb{R}^{12\times 1}$ is

$$\delta\mathcal{X} = \begin{bmatrix} \delta\mathbf{x}_0 & \delta\Delta\mathbf{x} & \delta\mathbf{\Theta_0} & \delta\Delta\mathbf{\Theta} \end{bmatrix}^T \tag{12}$$

and the Jacobian $\mathbf{J}_\mathcal{X} \in \mathbb{R}^{3\times 12}$ is

$$\mathbf{J}_\mathcal{X} = \begin{bmatrix} \mathbf{I} & s\mathbf{I} & -\mathbf{P} & -s\mathbf{P} \end{bmatrix}, \tag{13}$$

where the cross-product is embedded in $\mathbf{P} = [{}^M\hat{\mathbf{R}}_\tau^B \, \mathbf{p}_\tau^B \times]$ Here, $\mathbf{I}$ represents a 3×3 identity matrix.

In order to invert (11) and obtain the state corrections (12), two important details remain. One issue is data association, which is to say that the body-frame point described by $\mathbf{p}_\tau^B$ must be matched to the corresponding point $\mathbf{p}_\tau^M$ in the map data. A second issue is that (11) contains only 3 equations, so several such systems must be solved simultaneously to enable an independent solution for the twelve states in (12). Moreover, to avoid condition-number deficiencies, points must be compared from a wide range of s values (meaning a wide range of angles through the LIDAR rotational sweep); otherwise if the range of s is narrow, the structure of (7) undermines the independence of $\delta\mathbf{x}_0$ and $\delta\Delta\mathbf{x}$, and similarly the structure of (10) undermines the independence of $\delta\mathbf{\Theta}_0$ and $\delta\Delta\mathbf{\Theta}$. In the rest of this section, we assume a broad swathe of features are imaged across the full range of s, such that a well-condition set of equations can be obtained. Thus, our main focus is data association.

To address the data association problem, we use a voxel-based strategy, as employed by the popular NDT algorithm [28] and our variant ICET [4]. In these algorithms, point clouds are situated within a three-dimensional grid, consisting of volume elements called *voxels*. If the voxel grid is defined in frame M, it is trivial to assign each LIDAR point from the HD-map to the voxel containing it. Points from the current scan can also be associated with voxels, after the scan is transformed to frame M using the initial guess. An improved registration can then be inferred by aligning the distribution of map points to the distribution of current-scan points within each voxel. Ultimately, the grid enforces spatial associations, thereby avoiding association problems that arise in ICP, LOAM and other methods in the form of ambiguities caused by incorrect matches of points (or extracted features) between the map and the current scan.

To compare distributions of points in a given voxel, we take the expected value of (11) across all points in the voxel, which gives the following.

$$E\left(\mathbf{p}_\tau^M\right) = E\left({}^M\hat{\mathbf{R}}_\tau^B \mathbf{p}_\tau^B\right) + E\left(\hat{\mathbf{t}}_\tau^M\right) + E\left(\mathbf{J}_\mathcal{X}\delta\mathcal{X}\right) \tag{14}$$

On the right side of the equation, the rotation matrix ${}^M\hat{\mathbf{R}}_\tau^B$, the translation vector $\hat{\mathbf{x}}_\tau^M$, and the state-correction vector $\mathcal{X}$ are the same for all points in the cloud (and therefore also for all points in voxel). Accordingly, these three terms can be moved outside the expected-value operator.

$$E\left(\mathbf{p}_\tau^M\right) = {}^M\hat{\mathbf{R}}_\tau^B E\left(\mathbf{p}_\tau^B\right) + \hat{\mathbf{t}}_\tau^M + E\left(\mathbf{J}_\mathcal{X}\right)\delta\mathcal{X} \tag{15}$$

In this equation, the matrix $\mathbf{J}_\mathcal{X}$ depends on s and P, which have slightly different values for each point in the voxel. Those variations are small in practice, and a very good approximation is obtained by defining ${}^{(j)}\tilde{\mathbf{J}}_\mathcal{X}$, which is equivalent to $\mathbf{J}_\mathcal{X}$ except that, in (13), the s value is determined for the midpoint of voxel j and $\mathbf{P}$ is set to $\mathbf{P} = \left[{}^M\hat{\mathbf{R}}_{mid}^B \ {}^{(j)}\boldsymbol{\mu}^B \times\right]$, with the rotation matrix also evaluated at the voxel midpoint. Using this simplification, we can closely approximate (15) by setting ${}^{(j)}\tilde{\mathbf{J}}_\mathcal{X} \approx E(\mathbf{J}_\mathcal{X})$. The remaining two expected-value operations are averages across LIDAR points in a given voxel j. Accordingly, we invoke the j index and introduce the notation ${}^{(j)}\boldsymbol{\mu}^M = E(\mathbf{p}_\tau^M)$ and ${}^{(j)}\boldsymbol{\mu}^B = E(\mathbf{p}_\tau^M)$. The vector ${}^{(j)}\boldsymbol{\mu}^M$ is the mean position for all points from the map that fall in voxel j. The vector ${}^{(j)}\boldsymbol{\mu}^B$ is the mean position, in body-frame coordinates, for all current-scan points falling in voxel j. Substituting ${}^{(j)}\tilde{\mathbf{J}}_\mathcal{X}$, ${}^{(j)}\boldsymbol{\mu}^M$, and ${}^{(j)}\boldsymbol{\mu}^B$ into (15) gives the following equation.

$${}^{(j)}\boldsymbol{\mu}^M = {}^M\hat{\mathbf{R}}_\tau^B \ {}^{(j)}\boldsymbol{\mu}^B + \hat{\mathbf{t}}_\tau^M + {}^{(j)}\tilde{\mathbf{J}}_\mathcal{X}\delta\mathcal{X} \tag{16}$$

The state-correction vector $\delta\mathcal{X}$ is obtained by solving this equation simultaneously over all voxels. With a sufficient number of geometrically diverse voxels, the system is overdetermined, and a weighted least-squares solution can be used to mitigate measurement noise.

$$\delta\mathcal{X} = \left(\mathbf{J}^T\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{W}\,\mathbf{y} \tag{17}$$

Here $\mathbf{W}$ is a weighting matrix, $\mathbf{J}$ is the concatenation of the matrices ${}^{(j)}\tilde{\mathbf{J}}_{\mathcal{X}}$ over all voxels j, and the $\mathbf{y}$ is the concatenation of the observables ${}^{(j)}\mathbf{y}$ over the same voxels, where

$${}^{(j)}\mathbf{y} = {}^{(j)}\boldsymbol{\mu}^M - {}^M\hat{\mathbf{R}}^B_\tau\ {}^{(j)}\boldsymbol{\mu}^B - \hat{\mathbf{t}}^M_\tau. \tag{18}$$

The weighting matrix $\mathbf{W}$ is constructed from the measurement-error covariance for $\mathbf{y}$, as discussed in [12]. The same reference also provides numerical tools that allow (17) to be solved using a block decomposition of $\mathbf{J}$, $\mathbf{W}$, and $\mathbf{y}$. The decomposition enhances computational efficiency and reduces memory requirements.

Once the correction states are computed, the initial guess $\hat{\mathcal{X}}$ can be updated using the following equation.

$$\hat{\mathcal{X}} \leftarrow \hat{\mathcal{X}} + \delta\mathcal{X} \tag{19}$$

Newton's method can be used to converge on a solution, iterating between transforming/de-warping the distorted pointcloud with (17) and refitting voxel-based correspondences using the updated point locations. In practice, we obtained better results using Levenberg-Marquardt optimiztion [29].

B. Additional Details

In this section we identify practical details related to instantiating the above VICET algorithm in code.

Timestamps: Timestamps of LIDAR points can be approximated by using beam angle ψ. It is important not to group points recorded at different times (e.g., at the beginning and end of the scan). We address the aliasing issue by defining a voxel boundary at 0° (start of scan) and then removing occasional aberational points with $\psi < 0°$ or $\psi \geq 360°$.

Initialization: As mentioned in connection with (6), VICET assumes a reasonable estimate of initial pose. We obtained the initial pose by first running a standard scan-match, specifically NDT; VICET then improves this estimate by compensating for motion distortion.

Extended surfaces: NDT fails to recognize that walls (and other flat surfaces stretching across voxel boundaries) provide useful information only in the surface-normal direction; to enhance convergence reliability and accuracy, we incorporate ICET [4] extended-surface suppression into VICET.

V. Experiment: HD-Map Registration

In this section we evaluate the performance of snapshot scan-to-map localization using VICET, and we compare with two other widely used snapshot algorithms: NDT and ICP.

A. Pose Error Analysis

For our first evaluation, we make use of the *Newer College Dataset* [30] which contains a rosbag of raw LIDAR data packets and associated ground truth. The dataset also includes a survey-grade HD map of the structured environment recorded on a separate Leica total station. Specifically, we utilize the "Quad With Dynamics" trajectory which contains 3954 consecutive 10 Hz LIDAR scans and associated ground truth poses at the beginning of each scan, shown in Fig. 3 superimposed on the HD Map. In this sequence, a researcher runs while swinging the LIDAR sensor at head height, to approximate the dynamics of a multi-rotor drone [30].

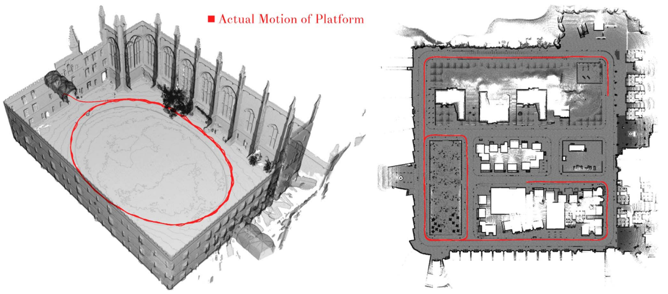

Fig. 3. Ground truth for each of the 3954 scans, superimposed on the HD Map for the Newer College dataset (left). Ground truth for each of the 5000 scans in the KITTI-CARLA Town 02 trajectory superimposed on the CARLA HD Map for Town 02 (right).

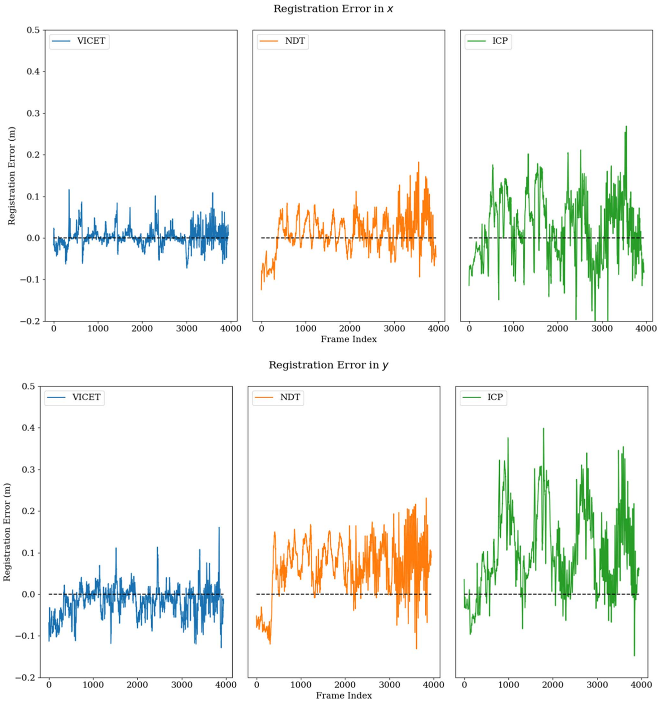

Fig. 4. Registration error in x and y for VICET, NDT, and ICP on the Newer Collge Datstet *Quad With Dynamics* trajectory.

Snapshot processing was conducted for VICET, NDT, and ICP by matching each individual scan to the map, without reference to any other scan from the sequence. NDT and ICP do not have any motion compensation applied. Note that VICET was initialized with NDT, and so the same spherical voxel grid [31] with an angular resolution of 3.6° was used for both VICET and NDT in order to ensure a controlled comparison.

Fig. 4 shows the translation errors aligned with the x and y axis of the LIDAR sensor for each frame in the Newer College

TABLE I
NEWER COLLEGE MEAN ERROR

	x (cm)	y (cm)	z (cm)	ϕ (deg)	θ (deg)	ψ (deg)
ICP	3.70	12.21	3.43	0.33	0.28	0.55
NDT	2.31	8.28	**1.06**	0.40	0.37	0.51
VICET	**0.35**	**-1.6**	1.17	**-0.29**	**-0.15**	**0.023**

TABLE II
NEWER COLLEGE RMS ERROR

	x (cm)	y (cm)	z (cm)	ϕ (deg)	θ (deg)	ψ (deg)
ICP	7.46	11.96	5.08	0.57	0.42	0.46
NDT	4.00	7.61	**2.12**	0.57	0.41	0.43
VICET	**3.32**	**5.68**	2.84	**0.39**	**0.30**	**0.30**

TABLE III
KITTI-CARLA MEAN ERROR

	x (cm)	y (cm)	z (cm)	ϕ (deg)	θ (deg)	ψ (deg)
ICP	28.36	-2.17	-1.29	0.135	-0.078	-0.453
NDT	15.23	0.93	0.23	0.0071	**0.0001**	**-0.024**
VICET	**8.93**	**-0.0035**	**-0.21**	**0.0025**	0.0062	-0.0475

TABLE IV
KITTI-CARLA RMS ERROR

	x (cm)	y (cm)	z (cm)	ϕ (deg)	θ (deg)	ψ (deg)
ICP	35.89	16.11	1.94	0.433	0.258	1.250
NDT	28.02	9.64	**0.42**	**0.0246**	**0.0107**	0.706
VICET	**14.73**	**4.43**	0.76	0.0451	0.0385	**0.593**

TABLE V
NORMALIZED CHAMFER DISTANCE (CM^2)

	Quad With Dynamics		*Dynamic Spinning*	
	Mean	STD	Mean	STD
ICP	0.869	0.092	3.317	0.758
NDT	0.814	**0.066**	3.045	0.923
VICET	**0.732**	0.097	**2.118**	**0.715**

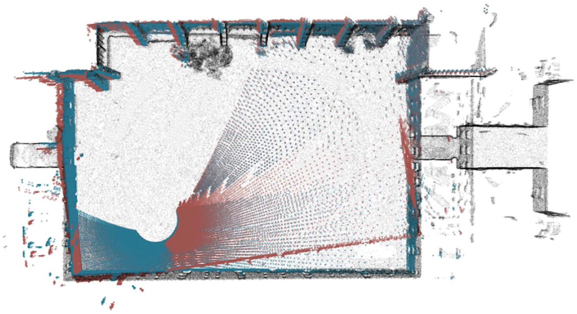

Fig. 5. Registered point clouds with and without motion-distortion compensation. As compared to the NDT-registered data (red), the VICET corrected data (blue) more closely resemble the HD-Map (black).

Dataset. For this sequence, the LIDAR unit was held such that the forward running direction of the experimenter was roughly aligned with the +y axis of the sensor. Similar to the $a \to b$ case in Fig. 1, the motion distortion in this experiment is most prevalent in the y component of the raw data, with some additional rotational distortion and motion in x and z due to the bobbing of the platform. The stretching of the LIDAR scans aligned with the direction of motion is problematic for the traditional rigid registration methods, as the algorithms must attempt to align the map with both undistorted points at the beginning of the LIDAR sweep, and points at the end of the sweep that are embedded with the full change in pose of the sensor that has occurred during the scan period. This effect produces an obvious bias in the y-aligned snapshot estimates for ICP and NDT, as the solutions oscillate between correct values near zero, and values roughly associated with the forward translation of the platform during the scan (0.12–0.2 m per frame). VICET is able to successfully compensate for platform motion and maintain zero-centered error in the same solution component, while the two rigid processing algorithms produce heavily biased estimates in the y direction. Mean errors for all states for the three algorithms are reported in Table I and RMS error values are reported in Table II. VICET is able to outperform ICP and NDT in x, y, ϕ (roll), θ (pitch), and ψ (yaw) for both metrics, though loses to NDT in z by a narrow margin.

A second experiment was conducted on the Town02 sequence from the KITTI-CARLA dataset [32], which simulates 5000 motion-distorted 10 Hz LIDAR scans while driving through a suburban environment. The simulated CARLA data presents some additional challenges over the Newer College data, as the KITTI-CARLA data contains moving pedestrians and other vehicles, as well as a more diverse scene geometry. Moving objects were rejected in VICET and NDT using a methodology outlined in [4]. For ICP, a similar point correspondence distance threshold was applied to reject outlier points. Results on the KITTI-CARLA dataset, shown in Tables III and IV exhibit similar bias in the direction of forward motion, which for this sequence is aligned with the x axis of the LIDAR sensor. Here, VICET achieves substantially lower mean and RMS translation error than ICP and NDT, but performs slightly worse than NDT on mean pitch and yaw, which were quite low for all three algorithms. Though it is heavily reduced compared to the rigid algorithms, VICET is not able to completely eliminate bias in forward translation on all frames of this dataset. We suspect that this is due to sparser scene geometry in some parts of the map degrading the independence of $\delta\mathbf{x}_0$ and $\delta\Delta\mathbf{x}$, as discussed in Section IV.

B. Chamfer Distance

We return to the Newer College dataset to perform an additional experiment where we quantify registration error in terms of chamfer distance, which we report in Table V. This metric effectively compares the different shapes of registered scans visualized in Fig. 5, as referenced to the map. A larger shape difference corresponds to a larger chamfer distance. Notably, the chamfer distance metric does not rely on ground truth. Here we compare results between the three algorithms on a challenging 200 frame sequence of the *Dynamic Spinning* trajectory (frames 950–1150), as well as a 200 frame sequence from the *Quad With Dynamics* trajectory (frames 2700 to 2900) which are captured from the same corner of the courtyard.

Some minimal data cleaning was required to compare chamfer distance in a meaningful way. Simply calculating the CD of the raw scans registered with the HD map is not a useful metric, as scores are dominated by points in the raw data that have no corresponding points in the map. Therefore, we suppress any points in the raw data that are not represented in the HD Map by suppressing scan points outside a convex hull for the map was inflated by 5% in our analysis. The results are compiled in Table V.

The table (column labeled *Quad with Dynamics*) shows that the average normalized chamfer distance for VICET was slightly better than for either NDT or ICP (0.73 cm^2 as compared to 0.81 cm^2 or more). This result indicates that the output from VICET is indeed a better approximations of the map than the outputs of either NDT or ICP.

The second trial on the *Dynamic Spinning* trajectory contained more motion distortion than the first. As expected, the chamfer distances were higher for this case than for the prior case (*Quad with Dynamics*). Moreover, the chamfer distance metric for VICET was again better than for either NDT or ICP (2.1 cm^2 as compared to 3.0 cm^2 or more).

VI. Conclusion

The experimental results from the prior section strongly support the importance of motion-distortion correction for high-accuracy scan-to-map matching. In this work, our primary emphasis is on achieving high-accuracy scan-to-map matching with a reasonable initial guess. VICET, by itself, does not solve the *Kidnapped Robot* problem [33], however, VICET could be layered with other algorithms that solve the global search problem, such as *DeepGMR* [34], enhancing their accuracy. Our results also demonstrate how the zero-order hold on platform velocity is valid in automotive applications with a 10 Hz sensor, where changes in speed generally happen with a bandwidth of about 1 Hz. For aircraft, the bandwidth depends on the weight and power of the platform; VICET should be quite useful for aerial survey and damage detection. For aggressive aerial motion in an indoor environment with obstacle avoidance, a method like that proposed by Setterfiled. et al. [21] may be advantageous.

In this letter we introduced VICET, a novel algorithm to solve the scan-to-map matching problem for a single LIDAR point cloud. VICET soves for twelve states, six describing a rigid transform aligning the scan to the map, and six more to account for distortion due to platform motion during the creation of the cloud. In contrast with other motion-distortion methods, our approach requires only a single LIDAR scan and no external sensor data.

Acknowledgment

The authors gratefully acknowledge the Tufts graduate funding that made this work possible. The authors also acknowledge and thank the U.S. Department of Transportation Joint Program Office (ITS JPO) and the Office of the Assistant Secretary for Research and Technology (OST-R) for partial sponsorship. Opinions discussed here are those of the authors and do not necessarily represent those of the DOT or affiliated agencies.

References

[1] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” in *Proc. SPIE*, vol. 1611, pp. 586–606, 1992.

[2] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, vol. 3, pp. 2743–2748.

[3] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, “Globally consistent 3D LiDAR mapping with GPU-accelerated GICP matching cost factors,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8591–8598, Oct. 2021.

[4] M. McDermott and J. Rife, “ICET online accuracy characterization for geometry-based laser scan matching,” 2023, *arXiv:2306.08690*.

[5] X. Lin, F. Wang, B. Yang, and W. Zhang, “Autonomous vehicle localization with prior visual point cloud map constraints in gnss-challenged environments,” *Remote Sens.*, vol. 13, no. 3, 2021, Art. no. 506.

[6] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, “Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations,” *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1377–1393, 2012.

[7] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time,” in *Robotics: Science and Systems*, vol. 2. Berkeley, CA, USA: MIT Press, 2014, pp. 1–9.

[8] A. De Maio and S. Lacroix, “Deep bayesian ICP covariance estimation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 6519–6525.

[9] Q. Li et al., “Lo-Net: Deep real-time LiDAR odometry,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8473–8482.

[10] Y. Cho, G. Kim, and A. Kim, “Unsupervised geometry-aware deep LiDAR odometry,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2145–2152.

[11] O. N. Kigotho, J. H. Rife, and H. Wassaf, “Lane-keeping safety and time correlation of navigation errors,” in *Proc. Int. Tech. Meeting Inst. Navigation*, 2022, pp. 1329–1342.

[12] M. McDermott and J. Rife, “Enhanced laser-scan matching with online error estimation for highway and tunnel driving,” in *Proc. Int. Tech. Meeting Inst. Navigation*, 2022, pp. 643–654.

[13] 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 *Proc. IEEE Int. Conf. Indoor Positioning Indoor Navigation*, 2016, pp. 1–8.

[14] G. Pandey, J. R. McBride, and R. M. Eustice, “Ford campus vision and LiDAR data set,” *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1543–1552, 2011.

[15] J.-L. Déziel et al., “Pixset: An opportunity for 3D computer vision to go beyond point clouds with a full-waveform lidar dataset,” in *Proc. IEEE Int. Intell. Transp. Syst. Conf.*, 2021, pp. 2987–2993.

[16] H. Caesar et al., “nuscenes: A multimodal dataset for autonomous driving,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.

[17] P. Misra, *Global Positioning System: Signal, Measurement, and Performance*, 2nd ed. Amherst, NY, USA: Ganga-Jamuna Press, 2010.

[18] P. Xie and M. G. Petovello, “Measuring GNSS multipath distributions in urban canyon environments,” *IEEE Trans. Instrum. Meas.*, vol. 64, no. 2, pp. 366–377, Feb. 2015.

[19] K. Inui, M. Morikawa, M. Hashimoto, and K. Takahashi, “Distortion correction of laser scan data from in-vehicle laser scanner based on Kalman filter and NDT scan matching,” in *Proc. Int. Conf. Inf. Control, Automat. Robot.*, 2017, pp. 329–334.

[20] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. London, U.K.: Pearson, 2015.

[21] T. P. Setterfield, R. A. Hewitt, A. T. Espinoza, and P.-T. Chen, “Feature-based scanning LiDAR-inertial odometry using factor graph optimization,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3374–3381, Jun. 2023.

[22] F. Moosmann and C. Stiller, “Velodyne SLAM,” in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 393–398.

[23] S. Shimojo, G. H. Silverman, and K. Nakayama, “Occlusion and the solution to the aperture problem for motion,” *Vis. Res.*, vol. 29, no. 5, pp. 619–626, 1989.

[24] S. Hong, H. Ko, and J. Kim, “VICP: Velocity updating iterative closest point algorithm,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 1893–1898.

[25] O. A. A. K. Salem, E. Giacomini, L. Brizi, L. Di Giammarino, and G. Grisetti, “Enhancing LiDAR performance: Robust de-skewing exclusively relying on range measurements,” in *Proc. Int. Conf. Ital. Assoc. Artif. Intell.*, 2023, pp. 310–320.

[26] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, “CT-ICP: Real-time elastic liDAR odometry with loop closure,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 5580–5586.

[27] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, “KISS-ICP: In defense of point-to-point ICP–simple, accurate, and robust registration if done the right way,” *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 1029–1036, Feb. 2023.

[28] M. Magnusson, “The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection,” Ph.D. dissertation, Örebro Univ., Örebro, Sweden, 2009.

[29] H. P. Gavin, “The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems,” in *Department of Civil and Environmental Engineering*, vol. 19. Duke Univ., 2019.

[30] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, “The newer college dataset: Handheld lidar, inertial and vision with ground truth,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 4353–4360.

[31] M. McDermott and J. Rife, “Mitigating shadows in LiDAR scan matching using spherical voxels,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12363–12370, Oct. 2022.

[32] J.-E. Deschaud, “Kitti-Carla: A Kitti-like dataset generated by Carla simulator,” 2021, *arXiv:2109.00892*.

[33] S. Thrun, “Probabilistic robotics,” *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[34] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, “Deepgmr: Learning latent Gaussian mixture models for registration,” in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 733–750.