Mitigating Shadows in LIDAR Scan Matching Using Spherical Voxels

Matthew McDermott and Jason Rife, *Member, IEEE*

***Abstract*—In this letter we propose an approach to mitigate shadowing errors in LIDAR scan matching, by introducing a pre-processing step based on spherical gridding. Because the grid aligns with the LIDAR beam, it is relatively easy to eliminate shadow edges which cause systematic errors in LIDAR scan matching. As we show through testing on real and synthetic data from a mechanically spinning multi-channel LIDAR unit, our proposed algorithm provides better results than ground plane removal, the most common existing strategy for shadow mitigation. Unlike ground plane removal, our method applies to arbitrary terrains (e.g. shadows on urban walls, shadows in hilly terrain) while retaining key LIDAR points on the ground that are critical for estimating changes in height, pitch, and roll. In our experiments, we demonstrate how our technique drastically reduces error in NDT scan registration (compared to a standard Cartesian voxel grid) on real LIDAR point cloud data, and then conduct Monte-Carlo trials in a simulated environment to demonstrate how our proposed technique eliminates the systemic bias introduced by range-shadowing.**

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

I. Introduction

LIDAR Odometry is a dead reckoning technique in which sequences of LIDAR range scans are registered on top of each other to estimate vehicle movement. In contrast with sensing modalities like GNSS and INS, the accuracy of a LIDAR scan match (and thus the resulting odometry estimate) is a strong function of both sensor quality and the geometric properties of the LIDAR's immediate surroundings. In this letter we propose a new approach to mitigate shadows, which occur when regions of the surroundings are occluded (shadowed) by a closer object. The edges of the shadowed regions move when the LIDAR moves. Moving shadows violate the assumption that the surroundings are static and, thereby, cause systematic biases in the scan-matching process.

Broadly speaking, existing scan-matching methods fall into one of three categories: end-to-end machine-learning (ML) methods that directly relate two point clouds, feature-based methods that classify groups of points as geometric structures recognizable across point clouds, and voxel-based methods that cluster points via a grid to enable alignment between clouds. In concept, our shadow removal method can be applied as a pre-processing step for any of these methodologies. As such, it is instructive to provide a brief overview of each category.

End-to-end ML methods estimate transformations directly from a pair of raw point clouds, either through direct estimation [1] or by projecting the 3D point cloud to a 2D *spin image* and extracting contextual information from the scene [2]. ML methods are advantageous in that they have the potential to achieve high accuracy because they can capture subtle effects in real-world data not easily captured by simple geometric models. However, ML-based methods are potentially limited because of their unpredictable patterns of error [3] and because of persistent challenges in registering point clouds that are only partially overlapping [4]. Applying shadow-removal as a preprocessing step may help reduce the complexity of the ML-based model required to compute the transformation between two scans.

Feature-based scan-matching methods identify structures in the point cloud and match those structures between scans. The simplest such approach, dubbed *Iterative Closest Point* (ICP), simply matches each individual point between two scans [5]. Early ICP algorithms are quite computationally intensive because of the large number of possible correspondences, so ICP variants have been developed that accelerate processing by limiting correspondence to a local neighborhood [6], by down-sampling the number of points used in each scan [7], [8], or by improving initialization [9], [10]. Other feature-based methods identify and correspond edges [11], [12], planes [13], or regions of similar LIDAR reflectivity [14] between scans. Mixed methods also exist that extract then match features using an ML-based approach, such as a deep neural-network [15]. Shadow removal has the potential to aid feature-based methods by ensuring that shadow edges are not labeled as valid features.

Voxel-based methods subdivide the scene into a grid of adjacent volume elements (aka, *voxels*). The distribution of points within each voxel is characterized, and scans are aligned to maximize the similarity of the distributions within each voxel. The best known voxel-based method is the *Normal Distributions Transform* (NDT), which describes the distribution of points in each voxel using a Gaussian density function [16]. Unlike feature-based methods, NDT does not ascribe any specific shape to characterize surfaces, a property which makes it more robust to processing scenes with arbitrary terrain [17]. NDT is also known for being computationally efficient [17]. Another

Manuscript received 29 June 2022; accepted 14 October 2022. Date of publication 25 October 2022; date of current version 2 November 2022. This letter was recommended for publication by Associate Editor A. Nuechter and Editor S. Behnke 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 the Office of the Assistant Secretary for Research and Technology (OST-R), and in part by NSF under Grant CNS-1836942. *(Corresponding author: Matthew McDermott.)*

The authors are with the Department of Mechanical Engineering and Automated Systems and Robotics Laboratory, Tufts University, Medford, MA 02155 USA (e-mail: matthew.mcdermott@tufts.edu; jason.rife@tufts.edu).

Digital Object Identifier 10.1109/LRA.2022.3216987

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

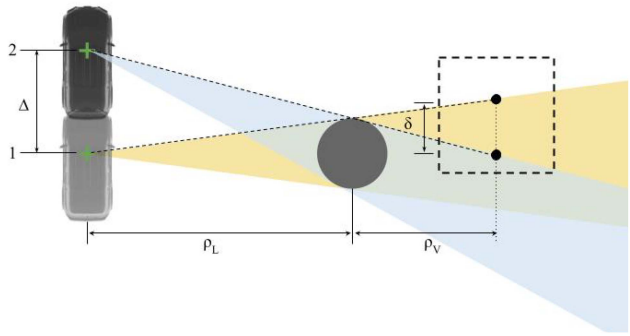


Fig. 1. Schematic of shadows created by a moving LIDAR (Vehicle graphic credit: ShapeNet [23]).

recently introduced voxel-based approach is the *Iterative Closest Ellipsoidal Transformation* (ICET), a method notable for analytically predicting solution accuracy as a function of scene geometry [18], [19]. ICET also includes a capability to recognize and exclude point groupings compromised by *Aperture Ambiguity* [20]. Shadow removal can benefit voxel-based scan matching by mitigating systematic errors that corrupt voxel point distributions. As we will show, shadow mitigation can also reformulate the voxel grid for more efficient and accurate computations.

The key contributions of this letter are to introduce a novel shadow-removal method, which acts as a preprocessing step before LIDAR scan-matching, and to characterize the benefits of the shadow-removal method via experiments on real and simulated LIDAR point clouds. In our first experiment, we apply our proposed voxel method to NDT and demonstrate how it can drastically reduce registration error on the KITTI [21] and Argoverse 2 [22] LIDAR datasets. In the subsequent experiment, we focus on ICET due to its ability to predict the covariance of the output states. This allows us to demonstrate clearly that shadow mitigation is effective if simulations show the predicted accuracy matches the true accuracy after shadow mitigation is applied (but not before). In the following sections, we will explain the shadowing error mechanism, detail our shadow-removal approach, and evaluate its performance.

II. Range Shadowing Effect

To help explain the impact of range shadowing on voxel-based scan matching, this section provides an illustrative geometric model. Consider the scenario shown in Fig. 1. The LIDAR moves with a vehicle through a distance Δ from one position to another (from the cross labeled 1 to the cross labeled 2). The LIDAR beam intersects a column (shown as a circular cross-section in the figure) at a distance ρ_L from the LIDAR. The column casts a shadow, which extends through many voxels including a voxel of interest (dashed rectangle). The voxel is a distance of approximately ρ_V from the column.

As the LIDAR moves, the edge of the shadow shifts by a distance δ (which we measure parallel to Δ). The LIDAR beam and shadow edge are tangent to the column at a point that moves little in response to LIDAR movement, assuming the radius of the column is small compared to ρ_L. For this reasonable case, we can use similar triangles to give:

$$\delta = \frac{\rho_V}{\rho_L}\Delta \tag{1}$$

As the LIDAR moves (toward the top of the page), the edge of the shadow within the voxel moves in the opposite direction, toward the bottom of the figure. This decreases the size of the shadowed region and increases the width of the point cloud in the voxel. Algorithms like NDT and ICET track the mean location of points within the voxel, so as the shadow recedes, the mean shifts down by $|\Delta\bar{x}|$. Using a one-dimensional, uniform-density approximation for the first moment, the change in the mean is:

$$|\Delta\bar{x}| = \frac{1}{2}\delta \tag{2}$$

Combining (1) and (2) gives:

$$|\Delta\bar{x}| = \frac{\rho_V}{\rho_L}\frac{\Delta}{2} \tag{3}$$

According to this equation, the point cloud has an apparent shift $|\Delta\bar{x}|$ equal to half Δ, scaled by a distance ratio. The distance-ratio is $\rho_V/\rho_L \in [0, \infty)$, which is zero just behind the column and increasingly large farther behind it.

Note that (3) suggests the error $|\Delta\bar{x}|$ grows very large when ρ_L is small. In fact, errors are capped by the size of a voxel, with larger shadow-motion effects confined to edge voxels (those containing both shadow and non-shadow regions) as interior voxels (those fully shadowed) have no LIDAR points and cannot contribute to scan matching.

From our analysis of Fig. 1, we conclude importantly that the shadow i) impacts many voxels in which it ii) amplifies apparent vehicle motion. The amplification of vehicle motion occurs because, from the point of view of the moving LIDAR, the column appears to shift downward (on the page) while the shadow appears to shift even farther downward. Since the shadow edge cuts through many voxels, a systematic bias is introduced into the perceived motion inferred from all of those voxels. This systematic bias affects all voxels along the edge of the shadow, and according to (3), the bias increases moving progressively farther away from the column.

In the literature, we observe two mechanisms that have been proposed for mitigating the effects of shadowing. The first approach is to remove all voxels on or near the ground plane [12], [24]. This approach is a widely-used heuristic, but it comes with a cost. Removing the ground plane signficantly reduces the ability of LIDAR to estimate pitch, roll, and vertical translation. Moreover, the heuristic assumes that shadows lie predominantly on a flat plane (e.g. the road), an assumption that may not be valid in some urban areas (when LIDAR shadows are cast on walls) or in offroad terrain.

The second approach looks for jumps in a 1D ranging signal [25]. These jumps correspond to adjacent LIDAR measurements that reflect from different surfaces (e.g. the column or the ground behind it) and aid in detecting the edge of a shadow. Because these methods only apply to 2D line scans, our new approach, introduced in the following section, adapts the jump-detection concept to apply to 3D scans (e.g., sampled over both azimuth and elevation).

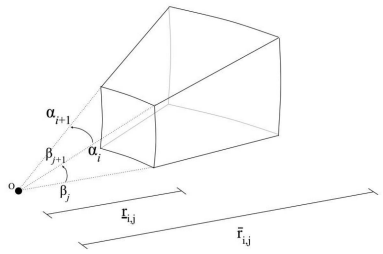


Fig. 2. Voxel boundaries.

III. Voxel-Based Jump Detection

In this section, we apply a spherical-gridding approach in a novel way, to mitigate LIDAR shadows. Our approach operates before each scan match, excluding problematic points from the primary scan and the secondary scan, which is transformed to align with the primary.

A. Primary Scan

It is straightforward to convert a point from a Cartesian vector $\mathbf{q}$ to a spherical form with radius r, azimuth α, and elevation β. They are related by:

$$\mathbf{q} = \begin{bmatrix} r\cos(\alpha)\cos(\beta) \\ r\sin(\alpha)\cos(\beta) \\ r\sin(\beta) \end{bmatrix} \tag{4}$$

If the LIDAR unit is modeled as a point, the spherical description is useful because each beam emanates radially outward along a particular azimuth α and elevation β. In a spherical grid, LIDAR beams do not cross grid boundaries, so it is easy to recognize the nearest radial object and exclude any objects behind it, which are likely to be shadowed. By extension, it is not necessary to fully populate the spherical grid with voxels in the radial direction, since we need only consider the voxel(s) associated with the nearest object. In fact, it is convenient to define only one radial voxel for each azimuth and elevation direction, where the radial limits are adapted to fit the nearest object. This concept for a voxel with adaptive radial boundaries is illustrated in Fig. 2; an algorithm for obtaining the adaptive boundaries is described below and summarized as Algorithm 1.

In constructing our voxel grid, we start with a set of wedge-shaped voxels (no radial limits) indexed by lower azimuth and elevation limits α_i and β_j, respectively. If the LIDAR points in scan K are indexed $k \in K$, then each point is characterized by the coordinates ${}^{(k)}r$, ${}^{(k)}\alpha$, and ${}^{(k)}\beta$. For voxel (i, j), define the set of radial coordinates to be $V_{i,j}$:

$$\begin{aligned} V_{i,j} = \{ {}^{(k)}r \} \quad \forall\, k \quad \text{s.t.} \\ {}^{(k)}\alpha \in [\alpha_i, \alpha_{i+1}) \\ {}^{(k)}\beta \in [\beta_j, \beta_{j+1}) \end{aligned} \tag{5}$$

Our approach to segmenting the nearest object is to add an outer radial limit $\bar{r}_{i,j}$ if a radial jump between scan points is detected. This generalizes the 2D jump-detection methods of [25] to a spherical grid in 3D. An inner radial bound $\underline{r}_{i,j}$ is also introduced, to screen out low-density stray points that may appear near the LIDAR unit. Note that the algorithm depends on two parameters, the jump-distance threshold T and the minimum number of points N that define a valid object. In Section VI we include a simple parameter study on selecting optimal values for N and T.

Algorithm 1: Adaptive Radial Boundaries.

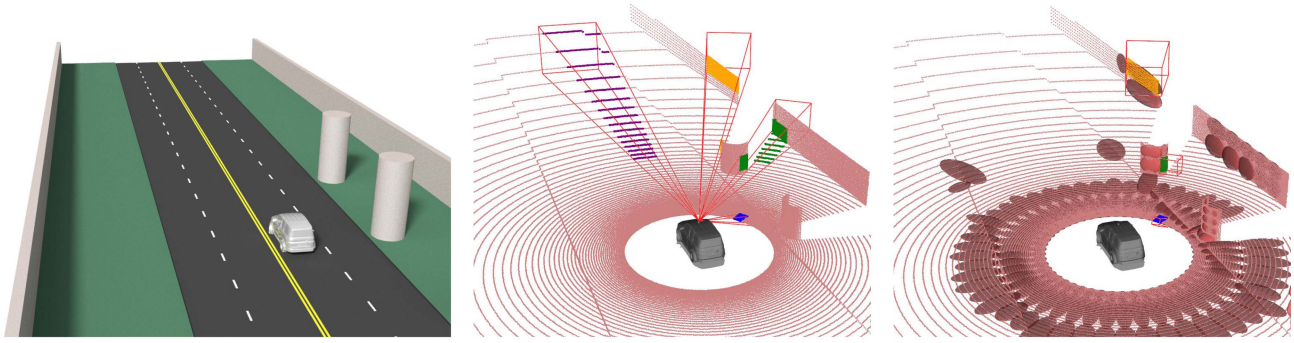
```
Initialize point cloud K
Transform all points k to spherical coordinates with (4)
for each wedge-shaped voxel (i, j) do
  Extract radii to obtain V_{i,j} with (5)
  Sort V_{i,j} (ascending) and assign indices l ∈ [0, L]
  Set default index of inner bound to l_min = 0
  Set default index of outer bound to l_max = L
  for each ordered point l (after point 0) do
    compute difference e_l = r_{i,j}(l) − r_{i,j}(l − 1)
    if jump detected (e_l > T) then
      if sufficient points (l − l_min > N) then
        reset outer bound to l_max = l − 1
      else
        reset inner bound to l_min = l
      end if
    end if
  end for
  Set r_{i,j} = r_{i,j}(l_min) and r̄_{i,j} = r_{i,j}(l_max)
  Exclude all points with l < l_min or l > l_max
end for
```

B. Secondary Scan

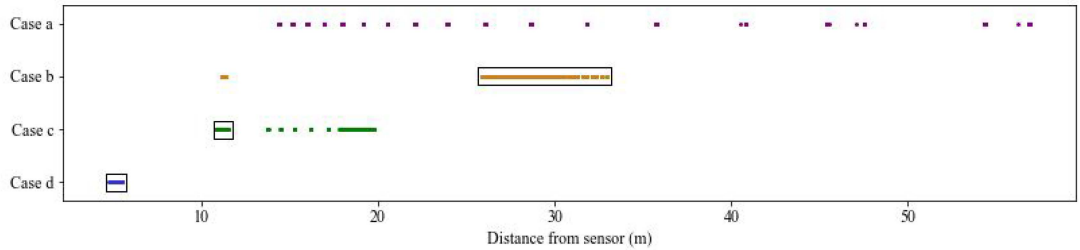
Scan matching methods compute a rigid transformation that aligns (or *registers*) a secondary scan to the primary. In general, the shadow-mitigation process of Algorithm 1 is intended as a preprocessing step for both scans. In voxel-based methods, however, the benefits of shadow-matching can be attained by applying the algorithm only to the primary scan. Since a common voxel grid is used for both scans, since points for either scan are only analyzed if they fall into a voxels, and since the voxels exclude shadowed regions of the scene, then the shadow-mitigation benefits established for the first scan transfer to the second through their shared voxelization. This simplification gives a slight efficiency benefit with no apparent impact on accuracy, so we adopt it for our evaluations described below.

C. Iterative Scan Matching

Once shadow mitigation has been applied, the rigid transformation relating the scans can be estimated using existing scan matching techniques. For compatibility with existing algorithms, we frame the rigid transformation in Cartesian coordinates, in terms of a rotation matrix $\mathbf{R}$ and a Cartesian translation through the scalars $\{x, y, z\}$. For each point k in the secondary scan, the rotation and translation can be applied to map the second-scan location ${}^{(k)}\mathbf{p}$ to its equivalent location



(a) Rendered scene (b) Wedge-shaped bins (c) Truncated voxels



(d) Radial distribution of points wedges in Fig. 3b

Fig. 3. Simulated four-lane highway.

${}^{(k)}\mathbf{q}$ in primary-scan coordinates:

$$ {}^{(k)}\mathbf{q} = \mathbf{R}\,{}^{(k)}\mathbf{p} - \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{6} $$

For voxel-based scan-matching methods like NDT and ICET, Algorithm 1 both eliminates shadowed regions of the scene and defines a useful voxel grid. Any points from the primary or secondary scan whose locations ${}^{(k)}\mathbf{q}$ fall outside the defined voxels are excluded from analysis. Point distributions are then compared for each voxel that contains a sufficient number of points from both scans, in order to compute the rotation matrix $\mathbf{R}$ and the translation $\{x, y, z\}$.

IV. Illustrative Example

In order to better understand the proposed shadow-mitigation algorithm, it is helpful to explore an example in detail. Consider the simulated scene shown in Fig. 3, where a vehicle using LIDAR for navigation (the *ego-vehicle*) passes a series of round columns that occlude portions of the ground plane and wall. The figure shows four views of the same scene. A visual rendering of the vehicle and scene are shown in Fig. 3(a). Simulated LIDAR scan points observed as the vehicle passes the last two columns are shown in Fig. 3(b); the same image also shows four wedge-shaped bins (in selected directions). The voxels generated by Algorithm 1 are shown in Fig. 3(c); the same image also shows the covariance ellipsoids describing the distribution of points in those voxels. In moving from Fig. 3(b) to Fig. 3(c), it is clear that Algorithm 1 prunes voxels in some directions. In order to understand the pruning process (and the process of determining the inner and outer radial bounds for each voxel), Fig. 3(d) visualizes the point distributions in the four wedges identified in Fig. 3(b).

The four wedges were selected to highlight different aspects of how Algorithm 1 functions. The wedges are labeled *case a* through *case d*. In *case a*, marked with purple, the points lie on parallel scan lines crossing the ground plane. Because they are separated by equal intervals in elevation angle, scan lines are separated by an increasing distance along the ground moving toward the horizon. The scan lines are never close enough to accumulate N points without a radial distance jump larger than the threshold T. As a consequence, all of the points in the wedge are excluded, so no associated voxel appears in Fig. 3(c). In *case b*, marked with yellow, a very small cluster of points occurs at a short distance (about 12 m) and a much larger cluster of points begins at a farther distance (about 25 m). The nearer points are the edge of a column; the farther points are returns from an oblique section of the wall. There are too few points in the near cluster (less than N), so the inner and outer radial bounds are determined based on the farther cluster, with bounds indicated by the black box shown for *case b* in Fig. 3(d). The same yellow-coded voxel appears in 3D in Fig. 3(c). Arguably, *case b* represents a mis-classification. In practice, we have observed our value for N to be suitable, such that surfaces with fewer than N points did not introduce a strong enough shadow to observably bias performance, as evidenced by the results of the next section. At the same time, N cannot be arbitrarily high, as the edge of a nearby object might be excluded, thus failing to remove a shadow. By contrast, *case c*, marked with green, captures the other edge of the same column, but with more than N points in

the nearer cluster. As a consequence, the radial bounds of the voxel bracket the column and exclude the wall behind, as shown by the black box for *case c* in Fig. 3(d) and by the green-coded voxel in Fig. 3(c). Finally, *case d*, marked by blue, captures a section of the ground plane near to the ego vehicle. All of the blue points in the wedge shown in Fig. 3(b) and in Fig. 3(c) are close together, and so no points are excluded from the voxel. The inner radial bound $\underline{r}$ and outer bound $\bar{r}$ capture all points in the wedge, as shown by the black box for *case d* in Fig. 3(b).

In this illustration, the bins are uniformly 7.2° wide in azimuth and elevation, the distance threshold is $T = 0.5$ m, and the minimum point count is $N = 50$. The four cases described above illustrate the tradeoffs in setting these parameters. The selected threshold (0.5 m) is low enough to clearly distinguish most occluded objects (e.g. the wall and the column behind) and high enough to cluster points on most oblique surfaces (e.g. the ground plane near the ego vehicle and the vertical wall at the edge of the road); however, the threshold is not high enough to link points on a highly oblique and distant horizontal surface (e.g. the ground plane far from the ego vehicle). Increasing T would allow us to capture more of the ground plane, but at the expense of losing the ability to exclude some shadows.

An additional important consideration is that it is helpful to pad the radial limits slightly, to increase the size of each voxel. The reason is that surfaces seen in the secondary image may not align exactly with those in the primary image. For example, marginally farther points on a column or nearer points on the back wall might become visible in the secondary scan. To account for this we modify Algorithm 1 slightly to pad the cell volume. The padding reduces the inner radial limit and increases the outer radial limit by as much as 0.5 m (or half the distance to the next excluded point, whichever is less).

V. Performance Evaluation

In this section we evaluate the degree to which our shadow-mitigation methods enhance algorithm performance. To this end, in our first experiment we performed computations using four algorithm variations: conventional NDT using a Cartesian grid, NDT using the spherical grid proposed via Algorithm 1, GP-ICP, and LOAM. In these tests, we compare the root-mean-square error for the output of each algorithm against the "ground truth" data provided in each benchmark. In our analysis we did not perform any loop closure or low frequency correction (as is prescribed in [11]), and instead limit registration estimates to the information present in the two sequential frames being considered for each trial.

For NDT analyses on Cartesian grids, cubic voxels were used, with 0.5 m edge length for the simulated and KITTI datasets, and 1.0 m for the Argoverse dataset (this dimension was tuned for favorable registration accuracy [17], so as not to artificially penalize the Cartesian analysis). Shadows on the ground were mitigated through ground plane removal. We also applied the simple outlier rejection routine commonly used in NDT as described by [26]. In our testing with spherical grids, we achieved optimal performance with an elevation and azimuth resolution of 4°.

For these experimental datasets, accuracy is computed as the root-mean square error (RMSE) over multiple pairs of sequential scans, obtaining error by subtracting the estimated LIDAR motion from the ground-truth provided by GPS-inertial fusion. Because LIDAR errors between frames (sub cm-level) are generally smaller than the GPS ground-truth errors (cm-level), the GPS errors dominate, and so the experiments provide a means to compare only relative accuracy (but not true accuracy) for various LIDAR algorithms.

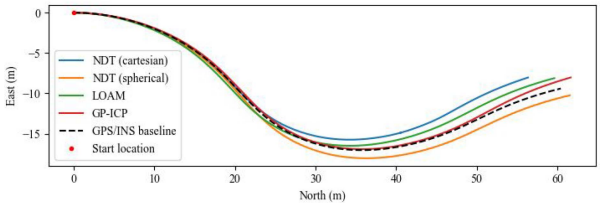
To complement these experiments, we studied absolute errors using a simulation of the roadway scene shown in Fig. 3, featuring sound-absorption walls beside the road and a series of 10 cylindrical pillars, two of which are shown in Fig. 3(a). Here, multiple Monte-Carlo trials were computed with randomized LIDAR errors. Since true motion is known in the simulated environment, scan-matching estimates were compared to the truth to compute an error, which was characterized statistically over the set of Monte-Carlo trials. In all, our analysis considered 120 trials with 3 samples for scan pairs in each of 40 locations along the roadway. In an attempt to maintain consistency with other benchmarks, we calibrated our virtual LIDAR sensor to match the specifications of the *Velodyne HDL-64E* used in the KITTI dataset [21]. For the roadway scene, the LIDAR translates 20 m along the center of its lane at a constant velocity of $5\frac{m}{s}$. Here, the columns along the roadway cause shadowing both on the ground plane and vertical walls behind.

For performance-evaluation purposes, our shadow-matching algorithm was integrated with the ICET scan matching algorithm, a least-squares variant of NDT that improves solution accuracy by suppressing voxel measurements aligned with extended surfaces and that also provides an *a priori* prediction of accuracy [18], [19]. Because true pose is known in the simulation, true errors can be computed and compared, to test not only whether shadow matching improves the pose-estimation errors for these voxel-based algorithms, but also to test whether shadow-matching improves the estimated accuracy (1σ) predicted by ICET.

VI. Results

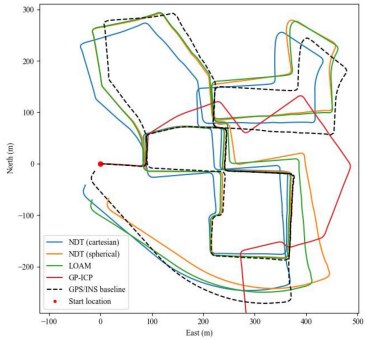
For the three experimental datasets, fofrward translation and yaw rotation estimates were compared to ground-truth, resulting in the RSME error values in Tables I-III (and resulting in accumulated North-East errors as depicted in the accompanying plots). Convergence failures occurred either when the algorithm diverged entirely or when it failed to escape the local minima of the initialization. These situations were easily detectable, and they were omitted from our analysis. The results demonstrate a consistant improvement in performance when switching from a Cartesian voxel grid to our proposed spherical voxel method. Even with ground-plane removal to mitigate shadowing, NDT with Cartesian voxels has significantly higher RMSE in translation (second column of tables) as compared to GP-ICP and LOAM; however, with our proposed spherical voxels NDT is able to achieve translation errors consistent with LOAM and GP-ICP. The magnitude of the RMSE in rotation (third column of tables) is relatively similar for all four algorithms, but there

TABLE I
KITTI RAW 05



	RMSE translation (cm)	RMSE rotation (deg)	Frames Rejected
NDT (cartesian)	3.179	0.0483	1/150
NDT (spherical)	1.626	0.0317	0/150
LOAM	1.674	0.0277	0/150
GP-ICP	1.515	0.0547	0/150

TABLE II
KITTI ODOMETRY 00

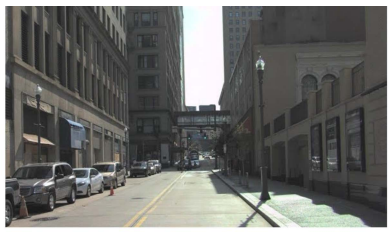


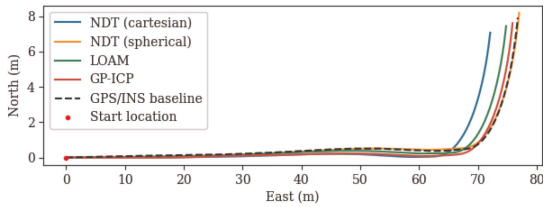
	RMSE translation (cm)	RMSE rotation (deg)	Frames Rejected
NDT (cartesian)	2.360	0.149	149/4500
NDT (spherical)	1.929	0.097	0/4500
LOAM	2.074	0.100	0/4500
GP-ICP	4.337	0.443	0/4500

is still clear improvement for NDT with spherical coordinates as compared to with Cartesian. RMSE results in all the translation directions (and all the rotation directions) followed similar trends, so results for those directions have been omitted for compactness.

The Cartesian NDT algorithm occasionally failed to converge, with the algorithm solution never leaving the basin of attraction of the initial guess. These cases are reported as "frames rejected," in the last column of Tables I-III. Spherical NDT, LOAM, and GP-ICP all converged reliably.

TABLE III
ARGOVERSE 2 LIDAR TRAIN 1.31





	RMSE translation (cm)	RMSE rotation (deg)	Frames Rejected
NDT (cartesian)	2.481	0.0217	3/156
NDT (spherical)	1.664	0.0120	0/156
LOAM	2.074	0.0374	0/156
GP-ICP	1.032	0.0633	0/156

TABLE IV
SIMULATED ROADWAY: SCATTER OF ERROR

	RMSE x (cm)	RMSE y (cm)	RMSE z (cm)	RMSE ϕ (deg)	RMSE θ (deg)	RMSE ψ (deg)
ICET(c)	0.239	0.023	0.195	0.0078	0.0169	0.00118
est. σ	0.109	0.021	0.182	0.0083	0.0129	0.00107
ICET(s)	0.087	0.009	0.072	0.0010	0.0010	0.00060
est. σ	0.086	0.010	0.072	0.0010	0.0009	0.00058

We subsequently ran simulations to qualify true errors. For the simulated roadway scene, we used the ICET algorithm to evaluate both true RMSE error and the algorithm's predicted covariance. These results are reported in Table (IV), with the true RMSE error reported on the first line and the estimated σ (root of diagonal elements of predicted covariance matrix) on the second, for each of two algorithms: ICET with Cartesian coordinates, labeled "ICET (c)," and with Spherical coordinates, labeled "ICET (s)." The effect of the spherical grid are pronounced in the along-track (or x direction). Translational errors are larger in the x-direction than in y-direction (by a factor of about 10 times), because there is a relative dearth of features to mark progress along the road. For the Cartesian grids, shadowing increases the actual x-direction errors substantially above the prediction (actual σ more than twice predicted value). By comparison, our new shadow-mitigation algorithm achieves an actual error that is very close to the predicted error in the x-direction (actual σ within 0.4% of the predicted value).

RMSE grows when random noise is larger, and also when a systematic bias is introduced. In Table IV, the fact that the RMSE in x for Cartesian ICET is much larger than the predicted σ (which accounts for random noise) indicates the presence of a systematic bias, likely the systemic bias caused by shadowing. Shadow mitigation with our spherical-grid approach is much more effective than with ground plane removal in this case, because ground plane removal addresses only shadows that

TABLE V
PARAMETER STUDY: THRESHOLD T

	RMSE translation (cm)	RMSE rotation (deg)	Frames Rejected
T = 0.05	1.677	0.0334	0/150
T = 0.20	1.643	0.0328	0/150
T = 0.50	1.626	0.0317	0/150
T = 1.00	1.789	0.0342	0/150
T = 2.00	1.919	0.0383	0/150

TABLE VI
PARAMETER STUDY: MIN POINTS PER CELL N

	RMSE translation (cm)	RMSE rotation (deg)	Frames Rejected
N = 25	5.157	0.1386	37/150
N = 50	1.626	0.0317	0/150
N = 100	1.697	0.0303	0/150
N = 200	1.800	0.0314	0/150

project on the ground, but our algorithm also addresses those that project on vertical walls.

As a final step, we returned to the experimental dataset (Kitti raw 05) to run a parameter sensitivity study on the spherical grid algorithm applied to NDT. The sensitivity study varied the key parameters T and N. Resulting RMSE values for forward translation and yaw rotation are compiled in Tables V (for T) and VI (for N). The parameter T represents the minimum distance required for two surfaces to be considered distinct. RMSE is flat over a wide range of T unless T becomes very small (less than 5 cm) or very large (greater than 2 m), as shown in Table VI. The parameter N represents the fraction of LIDAR points in a voxel as compared to the total number of points in that direction (where the later is a constant determined by the angular width of the spherical grid). If N is a small fraction of the nominal value (e.g. less than 10%, which corresponds to $N = 25$ in Table VI), then an obvious anomaly has occurred causing a very low point count; for higher N, parameter variations have little impact on performance.

VII. DISCUSSION

As shown by Tables I-III, our proposed spherical-grid methods reduce RMSE in translation by 18%–49% and rotation by 34%–45% when compared to Cartesian NDT with shadow-mitigation via ground-plane removal. Moreover, the spherical grid also enhances accuracy prediction, as shown in simulation (Table IV). Our sensitivity study (Tables V and VI) showed the spherical gridding method is relatively insensitive to variations in the key parameters N and T.

Future work will more fully characterize Algorithm 1. We suspect one disadvantage of our spherical-grid approach is that it will not work well with search-acceleration strategies [27] which may cause convergence issues if the initial frame displacement is very large (e.g. in registering scans taken from different vehicles). Also, it is likely that scenes with large initial translation will require that Algorithm 1 be applied to both frames (and not just to one frame as described in this letter). As a final observation, future work should characterize execution speed for scan-matching using Algorithm 1. Our algorithm creates a 2D voxel grid (one voxel deep in range), which could greatly enhance run speed as compared to conventional NDT, which uses a much larger number of voxels in a 3D Cartesian grid.

VIII. CONCLUSION

This letter introduced a new method for mitigating shadowing errors in voxel-based LIDAR scan matching (e.g., using algorithms like NDT or ICET). This shadow-mitigation algorithm runs as a preprocessing step before scan matching, by creating a 2D voxel grid in spherical coordinates. This grid aligns with LIDAR rays and with shadow edges; thus, shadow edges can be eliminated by preserving only the nearest point cluster in each wedge-shaped azimuth/elevation bin. As compared to the technique of ground plane removal, commonly used for NDT processing, our proposed method offers enhanced accuracy, comparable to that obtained for GP-ICP and LOAM on real-world datasets.

ACKNOWLEDGMENT

Opinions discussed here are those of the authors and do not necessarily represent those of the DOT, NSF, or other affiliated agencies.

REFERENCES

[1] Q. Li et al., "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8465–8474.
[2] Y. Cho, G. Kim, and A. Kim, "Unsupervised geometry-aware deep LiDAR odometry," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2145–2152.
[3] O. Willers, S. Sudholt, S. Raafatnia, and S. Abrecht, "Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks," in *Proc. Comput. Safety, Rel., Secur. Workshops*, A. Casimiro, F. Ortmeier, E. Schoitsch, F. Bitsch, and P. Ferreira, Eds. Cham: Springer, 2020, pp. 336–350.
[4] Z. Zhang, Y. Dai, and J. Sun, "Deep learning based point cloud registration: An overview," *Virtual Reality Intell. Hardware*, vol. 2, no. 3, pp. 222–246, 2020.
[5] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
[6] J. Elseberg, D. Borrmann, and A. Nüchter, "One billion points in the cloud–an octree for efficient processing of 3D laser scans," *ISPRS J. Photogrammetry Remote Sens.*, vol. 76, pp. 76–88, 2013.
[7] A. Palomer, P. Ridao, D. Ribas, A. Mallios, N. Gracias, and G. Vallicrosa, "Bathymetry-based SLAM with difference of normals point-cloud subsampling and probabilistic ICP registration," in *Proc. MTS/IEEE OCEANS - Bergen*, 2013, pp. 1–8.
[8] O. Dovrat, I. Lang, and S. Avidan, "Learning to sample," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2755–2764.
[9] H. Sun, S. Tang, S. Sun, and M. Tong, "Vision odometer based on RGB-D camera," in *Proc. IEEE Int. Conf. Robots Intell. Syst.*, 2018, pp. 168–171.
[10] J. Khalife, S. Ragothaman, and Z. M. Kassas, "Pose estimation with lidar odometry and cellular pseudoranges," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1722–1727.
[11] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst.*, 2014, vol. 2, pp. 1–9.
[12] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
[13] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst.*, Seattle, WA, 2009, vol. 2, no. 4, Art. no. 435.
[14] Y. S. Park, H. Jang, and A. Kim, "I-LOAM: Intensity enhanced LiDAR odometry and mapping," in *Proc. IEEE 17th Int. Conf. Ubiquitous Robots*, 2020, pp. 455–458.
[15] Z. Li and N. Wang, "DMLO: Deep matching LiDAR odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 6010–6017.

[16] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2003, vol. 3, pp. 2743–2748.
[17] S. Pang, D. Kent, X. Cai, H. Al-Qassab, D. Morris, and H. Radha, "3D scan registration based localization for autonomous vehicles - a comparison of NDT and ICP under realistic conditions," in *Proc. IEEE 88th Veh. Technol. Conf.*, 2018, pp. 1–5.
[18] 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.
[19] M. McDermott and J. Rife, "ICET online accuracy characterization for geometric based laser scan matching of 3D point clouds," *Navigation*, pp. 1–6, 2022.
[20] 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.
[21] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, 2013.
[22] B. Wilson et al., "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in *Proc. Neural Inf. Process. Syst. Track Datasets Benchmarks*, 2021.
[23] A. X. Chang et al., "Shapenet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.
[24] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D lidar point cloud," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1887–1893.
[25] D. Huber and M. Hebert, "A new approach to 3-D terrain mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1999, vol. 2, pp. 1121–1127.
[26] M. Magnusson, "The three-dimensional normal-distributions transform: An efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Dept. Comput. Sci., Örebro Universitet, örebro, Sweden, 2009.
[27] E. Einhorn and H.-M. Gross, "Generic NDT mapping in dynamic environments and its application for lifelong SLAM," *Robot. Auton. Syst.*, vol. 69, pp. 28–39, 2015.