Degradation-Aware Point Cloud Sampling in Robot Ego-Motion Estimation

Pavel Petracek*, Nikhil Khedekar•, Morten Nissov•, Kostas Alexis•, and Martin Saska*

Abstract—**The typical point cloud sparsification methods used in the state estimation of mobile robots are unaware of the perceptual degradations arising in geometrically symmetrical and structureless environments. In these environments, failing to handle geometrical degeneracies can render estimation ineffective or even impossible. We propose a novel method for a degeneration-aware point cloud sampling that is capable of lowering the effects of geometrical degeneracies by sampling informative points, which can be efficiently exploited in optimization along the degenerate directions. The proposed method is an alternative to the commonly used sparsification methods that normalize the density of points to comply with constraints on the real-time capabilities of a robot. In contrast to density normalization, our method leverages the concept of geometrical flow as an indirect measurement of a robot's ego-motion, exploits the shape of commonly used loss functions, and reuses the information that is expensively computed in previous iterations of an estimation process. Preliminary results show that the proposed sampling technique outperforms fine-tuned point-density normalization in geometrically-degenerate settings, while also being 26 % faster.**

I. Introduction

Spatial AI and classical methods of 3D perception utilize an abundant amount of data as input. To cope with the abundance, pipelines typically lower the cardinality of the input before any other action is taken and then optionally transform the reduced input into a feature domain. Random downsampling and discretization (voxelization) of the data to fixed-sized 3D cubes (voxels) remain prevalent among methods for lowering the cardinality of 3D data.

Although voxelization maintains the spatial distribution of the input and seems reasonable when computational tractability is essential, it may negatively influence the performance of algorithms, which iteratively find correspondences from the input to some target domain. This is due to the voxelization removing the level of detail below the fixed resolution, thus limiting the algorithms in the utilization of fine details in their pipeline. Therefore, voxelization is feasible with data containing high variance and salient structures. However, exploiting the finest level of detail is crucial when coping with spatial degeneracies consisting of geometrically symmetrical and structureless environments. Such degenerated environments include long and narrow corridors, tunnels, or chimneys.

A promising alternative to voxel filtering is importance sampling, a term adopted from early works on particle filtering and path planning. In the context of spatial perception, importance sampling represents a way of downsampling the input data such that the salient points, which are essential to any consequent task, remain in the data, whereas the rest are removed. Feature extraction is a similar concept; however, transforming the data from input into the feature domain can be expensive, sensitive to handcrafting, and sometimes undesired as a certain level of detail is inevitably lost.

However, the problem of selecting data in the input domain is non-causal, as the knowledge about the importance of a point in optimization is unknown before its point-map correspondence is established. The voxelization property of normalizing density in the input data while also maintaining spatial distribution makes it a safe option when applied in most state-of-the-art robotic solutions. Is that, however, the optimal way to reduce the task's complexity?

We propose three approaches to sampling the input data with respect to the following metrics — 1) a measure of a geometric flow indirectly quantifying the robot motion, 2) a heuristic maximizing the expected point contribution, and 3) output-to-input feedback maximizing the sum of information value. The proposed metrics, described in detail in Sec. IV, are deterministic and do not depend on any data-based learning procedure. In Sec. V, we further propose a greedy algorithm to fuse the three metrics in point cloud sampling. To experimentally validate the effects, we analyze the methods in a task of robot ego-motion estimation from 3D LiDAR point clouds. In this task, voxelization removes the detail below the fixed resolution, which limits the correspondences to a discretized grid. The discretization hinders the estimation of any robot motion smaller than the resolution, making the estimation more sensitive to correct initialization. Our experiments show that the proposed method focuses on exploiting the structure of the input domain and is capable of speeding up the entire pipeline, while improving its performance in degenerated environments.

II. Related work

A. Point cloud downsampling

There are many ways to downsample input data, ranging from binning image pixels through 2D and 3D convolution, to random or informed sampling. As our focus is narrowed to 3D ego-motion estimation from LiDAR data, the spectrum of classical methods is reduced mainly to random sampling (sometimes used in vanilla ICP) and voxel filtering, which is the most prevalent deterministic method. A typical implementation of a voxel filter uses the Octree structure [1] or a simple numerical discretization, such as that implemented in Point Cloud Library (PCL) [2].

The downsampling methods have also been studied in the context of machine learning. SampleNet [3] learns for task-specific (classification and geometry reconstruction) point

Authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University (CTU) in Prague, Czech Republic (*) and the Autonomous Robots Lab, Norwegian University of Science and Technology (NTNU), O. S. Bragstads Plass 2D, 7034, Trondheim, Norway (•).

This work was supported by CTU grant no. SGS23/177/OHK3/3T/13, by the Czech Science Foundation (GACR) under research project No. 23-07517S and by the Research Council of Norway under project REDHUS (NO-317773). Corresponding author: `pavel.petracek@fel.cvut.cz`.

cloud sampling. The method in [4] learns features and selects the points with the greatest contribution to the global max-pooling. DGCNN [5], FoldingNET [6], KCNET [7] construct a graph out of an unordered point cloud and downsample the graph using a graph-based max-pooling that takes the maximum features over the neighborhood of each node using a pre-built k-NN graph. The disadvantage of these methods is the absence of deterministic guarantees that the sampling will be invariant to the type of environment and that it will maximize point relevancy for optimization.

B. Degeneracy

Geometrical degeneration can be divided into two cases — strong and weak degeneracy. In the least common case of strong degeneracy, the input data contain no information whatsoever to be exploited for some degrees of freedom (DoF). This makes some directions of the optimization under-constrained, and constraints for these directions have to be supplied by other means, typically with other modalities. An example of such a solution is MIMOSA [8], which fuses inertial, visual, and LiDAR modalities for coping with their respective degeneracies. On the other hand, weakly-degenerated data contain at least a minor set of informative points that can be exploited for constraining any optimization direction. An example is given in Fig. 1a, where only a small ratio of data measures the end wall of a narrow corridor, which makes estimation along the corridor's main axis challenging. When the weight of this minority is not increased relative to the data cardinality, these points can easily get filtered out, be marked as noise or outliers, or fail to constrain the respective optimization direction sufficiently.

Only a few methods have been proposed to tackle the effects of weak degeneracy. A point cloud subset selection such that the direction of the normals of the points uniformly fill the normal-vector space was proposed in [9]. However, [9] requires a normal of a point to be known, which is expensive to achieve in real-time systems. Other methods utilize the detection of degeneracy in the optimization itself. A recent learning-based approach for detecting whether a problem is degenerated was introduced in [10], but the most used degeneracy detection is the older work [11], where the strength of an optimization problem is defined by the eigenvalues of the information matrix. AdaLIO [12] adaptively selects one of two fixed voxel filter parametrizations. The first parametrization is set to high resolution and is used when the problem strength at any DoF is below a degeneracy threshold [11]. In the opposite case of well-constrained problems, the second low-resolution parametrization is used. Nevertheless, a high-resolution voxelization may not reduce the data to a sufficient degree for real-time computing. Greedy-based method [13], KFS-LIO [14], and X-ICP [15] sample the residuals created from data-to-map correspondences with respect to their contribution to the optimization problem. Although this contribution can be quantified similarly to [11], these approaches require the correspondences to be known, which might be expensive. Our proposed methodology focuses on sampling in the input domain before any correspondence matching, residual computation, linearization, and optimization.

III. Problem Definition

In our case, a six DoF robot ego-motion estimation from LiDAR data employs a scan-matching pipeline in its core. Scan matching can be defined as a minimization task finding the change in pose $\mathbf{T}^* \in SE(3)$ between a source $\mathcal{P} = \{\mathbf{p} \in \mathbb{R}^3\}$ and a target $\mathcal{Q} = \{\mathbf{q} \in \mathbb{R}^3\}$ point clouds (sets of points). In this definition, the objective function minimizes the squared sum of residual functions $\mathbf{r_T} \in \mathbb{R}^3$ over the two point clouds

$$\mathbf{T}^*(\mathcal{P},\mathcal{Q}) = \underset{\mathbf{T}\in SE(3)}{\arg\min} \sum_{(\mathbf{p},\mathbf{q})\in\mathbb{C}_{\mathcal{Q}}^{\mathcal{P}}} \rho\left(||\mathbf{r_T}(\mathbf{p},\mathbf{q})||_2^2\right), \tag{1}$$

where $\mathbb{C}_{\mathcal{Q}}^{\mathcal{P}}$ represents the set of correspondence pairs from $\mathcal{P}$ to $\mathcal{Q}$ and ρ is a robust kernel. Given an input point cloud $\mathcal{P}_L$, we propose to precede the optimization problem defined in Eq. (1) with a single-metric selection task

$$\mathcal{P} = \underset{\mathcal{P}_l\subseteq\mathcal{P}_L}{\arg\max}\, g(\mathcal{P}_l), \qquad \text{subject to } |\mathcal{P}_l| > \lambda|\mathcal{P}_L|, \tag{2}$$

where the constraint defines the minimum cardinality of the output set by a factor $\lambda \in (0,1\rangle$ and $g(\cdot)$ is a metric to be defined in Sec. V. The metric $g(\cdot)$ ranks the potential of a point in bringing information to the optimization to achieve the desired selection. By Eq. (2), the total amount of information $\mathcal{I}(\mathcal{P})$ available in $\mathcal{P}$ during the optimization is $\mathcal{I}(\mathcal{P}) \leq \mathcal{I}(\mathcal{P}_L)$. In an ideal case, the optimal selection metric yields $\mathcal{I}(\mathcal{P}) = \mathcal{I}(\mathcal{P}_L)$. However, in real-world data subjected to noise and inaccuracies, a feasible selection metric should minimize $\mathcal{I}(\mathcal{P}_L) - \mathcal{I}(\mathcal{P})$.

The selection aims to reduce the cardinality in the input domain to reduce the complexity of a scan-registration pipeline comprising iterative correspondence search, residual computation, linearization, and optimization. Although a certain amount of information is inevitably lost during the selection, the preliminary experiments (Sec. VI) show that using a lower-cardinality point subset may yield better convergence performance in speed and accuracy, if the subset maximizes the information exploitable in the optimization. The reasoning for this arises from the implication that fewer points result in less computation per iteration, which speeds up the process. A faster registration pipeline thus implies a faster convergence rate, assuming the optimum of the objective function remains unchanged during the selection process. With the convergence rate improved, the registration pipeline finds the global optimum faster. And faster convergence can result in better results in real-time pipelines constrained by termination criteria (e.g., number of iterations, error magnitude), which may stop the optimization before the optimum is found.

IV. Importance Metrics

We propose three different metrics $g(\cdot)$ as defined in Eq. (2) — geometric flow g_Δ, the maximum expected value g_Φ, and information contribution $g_\mathcal{I}$. The algorithm for combining the metrics is described in Sec. V, whereas the metrics are visualized in Fig. 1 and described in the following subsections.

A. Geometric flow

In computer vision, optic flow represents the apparent motion of objects in a rigid visual scene caused by the relative motion of a camera. Inspired by optic flow, we define geometric flow

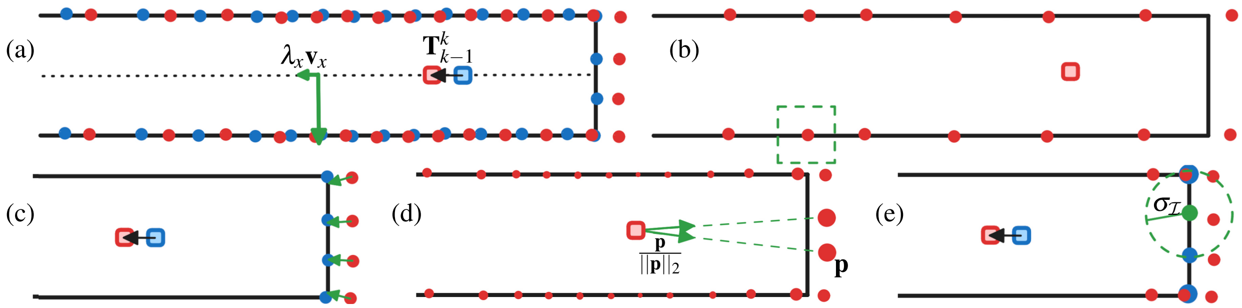

Fig. 1: Proposed metrics quantifying the importance of points on pose-change estimation, which estimates the transformation $\mathbf{T}_{k-1}^k$ from two point clouds taken at two different times. (a) The point cloud $\mathcal{P}_L^{k-1}$ is given in blue ●, $\mathcal{P}_L^k$ in red ●, and the Eigenspace of $\mathcal{P}_L^{k-1}$ is denoted by green →. The vector $\lambda_x \mathbf{v}_x$ is a scaled Eigenvector corresponding to the axis visualized by the dashed black line. Since the opt. problem can exploit only a few points for estimating motion along this axis, it is weakly degenerated. (b) Voxelization can remove the points important for mitigating the degradation when the voxel size (in green) is improperly set to the environment. (c) Robot motion directly produces *geometric flow* (→) in the environment. Multiplying the flow by the Eigenspace scales up the areas exploitable for optimizing along the degenerated direction. (d) The MEV scales up points by their parallelism to the degenerated directions. The scheme shows an example of MEV (larger ● implies greater g_Φ) corresponding to the point-to-point loss function. (e) The IC scales up points by opt. contribution of the points from $\mathcal{P}_L^{k-1}$ in their proximity (only the points contributing along the degenerated axis are shown).

g_Δ as a per-pixel change in depth (or scale) caused by the relative motion of a depth camera or a LiDAR. The geometric flow indirectly encodes information about the robot's velocity and naturally behaves as a cheap and accurate edge detector. Having a point cloud $\mathcal{P}_L^k$ at time k in an ordered matrix form, we define the geometric flow at pixel (i,j) as

$$g_\Delta^k(i,j) = \left\| \Gamma_{\Lambda_{k-1}} \left[\mathcal{P}_L^k(i,j) - \mathcal{P}_L^{k-1}(i,j) \right] \right\|_2, \tag{3}$$

with $g_\Delta^k(i,j)$ being equal to zero if either $\mathcal{P}_L^k(i,j)$ or $\mathcal{P}_L^{k-1}(i,j)$ is undefined or missing. The per-axis scaling factor $\Gamma_{\Lambda_{k-1}} \in \mathbb{R}^{3\times3}$ represents a *degeneracy-correcting factor* (DCF). Inspired by [11], we employ Eigenspace decomposition of an information matrix Λ_{k-1}, and we novelly define DCF of matrix Λ_{k-1} as

$$\Gamma_{\Lambda_{k-1}} = \lambda_{\min} \Sigma^{-1} \mathbf{V}^\intercal, \tag{4}$$

where $\mathbf{V} \in \mathbb{R}^{3\times3}$ is a matrix of column-wise Eigenvectors of Λ, $\Sigma \in \mathbb{R}^{3\times3}$ is a diagonal matrix of Eigenvalues corresponding to the Eigenvectors $\mathbf{V}$, and $\lambda_{\min}$ is the minimum Eigenvalue of Σ. Note, the $k-1$ indices have been omitted for simplicity. In our case, the decomposed information matrix is given as $\Lambda_{k-1} = \mathbf{J}_{k-1}^\intercal \mathbf{J}_{k-1}$, where $\mathbf{J}_{k-1} \in \mathbb{R}^{3\times3}$ is the sub-matrix related to the translation part in the Jacobian of the optimization problem defined at time $k-1$. The use of Λ_{k-1} represents a form of feedback that informs the input at time k about the influence of the geometric properties of the environment on the optimization.

In our experiments in Sec. VI, the employed pose estimation method KISS-ICP [16] uses the point-to-point metric, which Jacobian's translational part is identity. For that reason, we utilize the rotational part $\mathbf{J}_{k-1} = \sum_{(\mathbf{p},\mathbf{q})\in\mathbb{C}_\mathcal{Q}^\mathcal{P}} \frac{\partial \mathbf{r}_\mathbf{T}(\mathbf{p},\mathbf{q})}{\partial \mathbf{R}}$ instead.

The objective of DCF is to compensate for low geometric flow generated during a slow ego-motion along a degenerated axis. In other words, DCF scales down (relatively to the worst-constrained axis) geometric flow along the axes that are well constrained in the opt. problem at the previous time step.

B. Maximum expected value

The *maximum expected value* (MEV) g_Φ exploits the shape of the loss function used in the optimization. This metric values points by their potential to constrain degenerated optimization directions. Since the most common loss functions utilized in cloud-to-cloud registration are the point-to-point, point-to-plane, and point-to-line functions, we define MEV as

$$g_\Phi(\mathbf{x}) = \left\| \Gamma_{\Lambda_{k-1}} \frac{\mathbf{x}}{||\mathbf{x}||_2} \right\|_2, \tag{5}$$

where $\mathbf{x}$ is a point $\mathbf{x} \in \mathcal{P}_L$ for the point-to-point, the normal of a planar surface for the point-to-plane, and the unit direction of a line for the point-to-line loss function, all represented in the LiDAR frame. Note that, in our experiments in Sec. VI, we target estimation without the expensive estimation of the underlying surface and employ the point-to-point metric, which requires no additional information. Projection of $\mathbf{x}$ onto the Eigenspace by the DCF maximizes the potential for the high-g_Φ points to generate residuals important for axes valued by the Eigen decomposition as degenerated. In other words, MEV serves as a preemptive measure that greatly values the points at which future correspondences yield the maximum potential in generating residuals constraining degenerated axes.

C. Information contribution

During a typical iterative pipeline, the point-to-domain correspondences, their corresponding residuals, and their linearizations are deleted once the process converges and a change in robot pose is found. The geometric flow and MEV metrics have utilized this information only in its condensed form of the problem information matrix. The *information contribution* (IC) $g_\mathcal{I}$ is a method that further reuses expensively-computed per-correspondence information when solving the same problem in the previous iteration. Let $\mathcal{F}_{k-1} = \left\{ \left(\mathbf{p}_i \in \mathbb{R}^3, \mathbf{J}_i \in \mathbb{R}^{3\times3} \right) \right\}$ be a set of correspondences remaining after the convergence at time $k-1$, with $\mathbf{J}_i$ being the Jacobian of the i-th residual and $\mathbf{p}_i \in \mathcal{P}_L^{k-1}$ being the linearization point. Then, let $\mathcal{X}_i = \left\{ \mathbf{x} \mid ||\mathbf{x} - \mathbf{p}_i||_2 < \sigma_\mathcal{I}, \mathbf{x} \in \mathcal{P}_L^k \right\}$ be a set of points in a neighborhood of point $\mathbf{p}_i$ within a distance of $\sigma_\mathcal{I}$ (m). The IC is then given $\forall \mathcal{X}_i$, $\forall \mathbf{x}_i \in \mathcal{X}_i$ as

$$g_\mathcal{I}(\mathbf{x}) = \max\left(g_\mathcal{I}(\mathbf{x}), \left\| \Sigma_i \mathbf{V}_i^\intercal \Gamma_{\Lambda_{k-1}} \right\|_2 \right), \tag{6}$$

where $\mathbf{V}_i \in \mathbb{R}^{3\times3}$ is a matrix of column-wise Eigenvectors of $\Lambda_i = \mathbf{J}_i^\intercal \mathbf{J}_i$ and $\Sigma_i \in \mathbb{R}^{3\times3}$ is a diagonal matrix of Eigenvalues corresponding to the Eigenvectors $\mathbf{V}_i$. The purpose of `max` is to maintain the maximum $g_{\mathcal{I}}$-value for points located in multiple $\mathcal{X}_i$. Points not located in any $\mathcal{X}_i$ have $g_{\mathcal{I}}$ equal to 0.

In other words, IC represents the maximum information brought into the optimization at the previous iteration by a spatial volume defined by $\sigma_{\mathcal{I}}$. This way, IC exploits the assumption that important points remain spatially close among two consecutive point clouds.

V. Importance Sampling

By having the three different metrics, we propose to redefine Eq. (2) and simultaneously propose a solution by defining

$$\mathcal{P} = \underset{\mathcal{P}_\Delta \subseteq \mathcal{P}_L}{\arg\max}\, g_\Delta(\mathcal{P}_\Delta) \cup \underset{\mathcal{P}_\Phi \subseteq \mathcal{P}_L}{\arg\max}\, g_\Phi(\mathcal{P}_\Phi) \cup \underset{\mathcal{P}_\mathcal{I} \subseteq \mathcal{P}_L}{\arg\max}\, g_\mathcal{I}(\mathcal{P}_\mathcal{I}), \tag{7}$$

$$\text{subject to } |\mathcal{P}_\Delta| = \lambda_\Delta|\mathcal{P}_L|,\ |\mathcal{P}_\Phi| = \lambda_\Phi|\mathcal{P}_L|,\ |\mathcal{P}_\mathcal{I}| = \lambda_\mathcal{I}|\mathcal{P}_L|, \tag{8}$$

where $\mathcal{P}$ is the union of three subsets of cardinality given by factors $\lambda_\Delta, \lambda_\Phi, \lambda_\mathcal{I} \in (0,1\rangle$, which maximize one of the respective metrics. Greedy algorithm solving Eq. (7)-(8) is summarized in Alg. 1.

Algorithm 1 Pseudocode of the importance sampling routine

```
Input:
    $\mathcal{P}_L^{k-1}, \mathcal{P}_L^k$    ▷ point clouds at time $k-1$ and $k$
    $\Lambda_{k-1} \in \mathbb{R}^{3\times3}$    ▷ information matrix of the opt. problem at time $k-1$ (translation part)
    $\mathcal{F}_{k-1}$    ▷ set of correspondences at time $k-1$
    $\sigma_\mathcal{I} \in \mathbb{R}^+$    ▷ $g_\mathcal{I}$ influence radius (m)
    $\lambda_\Delta, \lambda_\Phi, \lambda_\mathcal{I} \in (0,1\rangle$    ▷ minimum ratio of sampled points (%)
Begin:
    $\Gamma_{\Lambda_{k-1}} = \text{EigenDecomposition}(\Lambda_{k-1})$    ▷ Eq. (4)
    ▷ compute the metrics for all points in $\mathcal{P}_L^k$
    $\mathcal{H}_\Delta = \text{geometricFlow}\left(\mathcal{P}_L^k, \mathcal{P}_L^{k-1}, \Gamma_{\Lambda_{k-1}}\right)$    ▷ Eq. (3)
    $\mathcal{H}_\Phi = \text{maximumExpectedValue}(\mathcal{P}_L^k, \Gamma_{\Lambda_{k-1}})$    ▷ Eq. (5)
    $\mathcal{H}_\mathcal{I} = \text{informationContribution}(\mathcal{P}_L^k, \Gamma_{\Lambda_{k-1}}, \mathcal{F}_{k-1}, \sigma_\mathcal{I})$    ▷ Eq. (6)
    ▷ sample three fixed-cardinality $g$-maximizing subsets
    return  $\text{maxN}(\lambda_\Delta|\mathcal{P}_L^k|, \mathcal{H}_\Delta) \cup \text{maxN}(\lambda_\Phi|\mathcal{P}_L^k|, \mathcal{H}_\Phi) \cup$
            $\text{maxN}(\lambda_\mathcal{I}|\mathcal{P}_L^k|, \mathcal{H}_\mathcal{I})$
```

VI. Experimental Analysis

To verify the proposed importance sampling methodology, we utilize KISS-ICP [16] as the state-of-the-art implementation of the Iterative Closest Point (ICP) algorithm. ICP is a dense (without feature extraction) method estimating transformation between two point clouds. ICP is sensitive to initialization, noise, and particularly to geometrical degeneracies due to point-to-point associations, which enables verification of the required properties of the proposed sampling technique.

To validate against all kinds of different degeneracies, we use an environment inspired by the experimental verification of X-ICP [15], which is a state-of-the-art method for pose estimation in geometrically degenerated environments. The environment, shown together with the results of experiments in Fig. 2, contains long and narrow corridors, sharp turns, and a circular and square room that are challenging to rotation estimation. In the experiments, a UAV visits all parts of the structure and closes the loop by landing at the takeoff location.

The experiments evaluated qualitatively in Fig. 2 and quantitatively[1] in Tab. I show that the typical pipeline of KISS-ICP using data reduced by voxelization is capable of lowering the effects of geometrical degeneracies. Nevertheless, the voxel-size parameter is environment-dependent as the estimation fails for over-sparsified sampling (e.g., points voxelized with a resolution of 60 cm in Fig. 2). The cost of higher resolution is; however, the growing comp. time required to iteratively find correspondences, compute residuals, linearize, and optimize. On the other hand, the proposed importance sampling (parametrized with $\lambda_\Delta = 0.20, \lambda_\Phi = 0.05, \lambda_\mathcal{I} = 0.03$) shows promising potential by producing results comparable to voxelization tuned to the environment. The method lowers the degradation effects and yields the lowest drift (particularly in the z-axis) with the fewest points used in the optimization. The overhead for computing the metrics further pays off in the iterative matching process, where fewer points lead to faster convergence. Note that, the voxelization is an integral part of the KISS-ICP pipeline; a voxelized point cloud is used in both methods for building the map of the environment on the run.

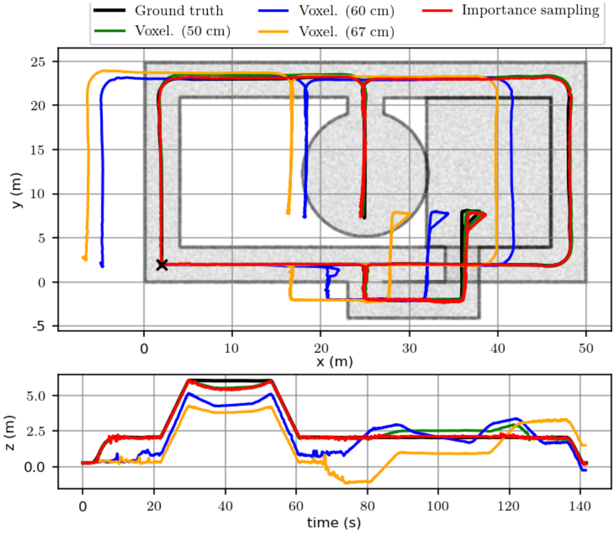

Fig. 2: Comparison of the KISS-ICP performance in a challenging environment containing various degenerated areas. The loop trajectory starts and ends at the black cross. The trajectories show the performance of KISS-ICP for input data either voxelized to a certain resolution or sampled according to Alg. 1.

Method	RMSE (m)	Median err. (m)	Avg. comp. time (ms)	Rel. point reduct. (%)
Voxelization (50 cm)	0.38	0.16	45.6	1
Voxelization (60 cm)	5.60	6.41	41.4	0.74
Voxelization (67 cm)	8.06	8.44	34.1	0.62
Importance sampling	**0.36**	**0.15**	**33.7**	**0.26**

TABLE I: Quantitative results for the trajectories shown in Fig. 2. The proposed method performs comparably to the fine-tuned KISS-ICP algorithm while sampling 4× fewer points and reducing the computational time of the entire pipeline by 26 %. Parametrization with a similar runtime performance (voxelization to 67 cm) fails to overcome geometrical degeneracy in a narrow corridor. The reported runtime is for the entire estimation pipeline, including *importance sampling*.

VII. Future Work

Rigorous analyses of the proposed sampling have yet to be done. As part of future work, we plan to perform ablation studies to determine each proposed metric's individual contribution and failure modes. Experimental comparison to state-of-the-art point sampling techniques using datasets from different real-world domains will also be performed to highlight the advantages and limitations of the proposed metrics.

[1] Results were obtained using EVO [17].

References

[1] R. Schnabel and R. Klein, “Octree-Based Point-Cloud Compression,” in *Proceedings of the 3rd Eurographics / IEEE VGTC Conference on Point-Based Graphics*, ser. SPBG’06, 2006, pp. 111–121.

[2] R. B. Rusu and S. Cousins, “3D Is Here: Point Cloud Library (PCL),” in *IEEE ICRA*, 2011, pp. 1–4.

[3] I. Lang, A. Manor, and S. Avidan, “SampleNet: Differentiable Point Cloud Sampling,” in *IEEE/CVF CVPR*, 2020, pp. 7575–7585.

[4] E. Nezhadarya, E. Taghavi, R. Razani, B. Liu, and J. Luo, “Adaptive Hierarchical Down-Sampling for Point Cloud Classification,” in *IEEE/CVF CVPR*, 2020, pp. 12 953–12 961.

[5] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic Graph CNN for Learning on Point Clouds,” *ACM Transactions on Graphics*, vol. 38, no. 5, pp. 146:1–146:12, 2019.

[6] Y. Yang, C. Feng, Y. Shen, and D. Tian, *FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation*, 2018.

[7] Y. Shen, C. Feng, Y. Yang, and D. Tian, “Mining Point Cloud Local Structures by Kernel Correlation and Graph Pooling,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4548–4557.

[8] N. Khedekar, M. Kulkarni, and K. Alexis, “MIMOSA: A Multi-Modal SLAM Framework for Resilient Autonomy against Sensor Degradation,” in *IEEE/RSJ IROS*, 2022, pp. 7153–7159.

[9] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, “Geometrically Stable Sampling for the ICP Algorithm,” in *Fourth International Conference on 3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings.*, 2003, pp. 260–267.

[10] J. Nubert, E. Walther, S. Khattak, and M. Hutter, “Learning-Based Localizability Estimation for Robust LiDAR Localization,” 2022.

[11] J. Zhang, M. Kaess, and S. Singh, “On Degeneracy of Optimization-Based State Estimation Problems,” in *IEEE ICRA*, 2016, pp. 809–816.

[12] H. Lim, D. Kim, B. Kim, and H. Myung, “AdaLIO: Robust Adaptive LiDAR-Inertial Odometry in Degenerate Indoor Environments,” 2023.

[13] J. Jiao, Y. Zhu, H. Ye, H. Huang, P. Yun, L. Jiang, L. Wang, and M. Liu, “Greedy-Based Feature Selection for Efficient LiDAR SLAM,” in *IEEE ICRA*, 2021, pp. 5222–5228.

[14] W. Li, Y. Hu, Y. Han, and X. Li, “KFS-LIO: Key-Feature Selection for Lightweight Lidar Inertial Odometry,” in *IEEE ICRA*, 2021, pp. 5042–5048.

[15] T. Tuna, J. Nubert, Y. Nava, S. Khattak, and M. Hutter, “X-ICP: Localizability-Aware LiDAR Registration for Robust Localization in Extreme Environments,” 2022.

[16] 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 Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[17] M. Grupp, “evo: Python package for the evaluation of odometry and SLAM.” https://github.com/MichaelGrupp/evo, 2017.