Novel SKIP Features for LIDAR Odometry and Mappings

Novel SKIP Features for LIDAR Odometry and Mappings

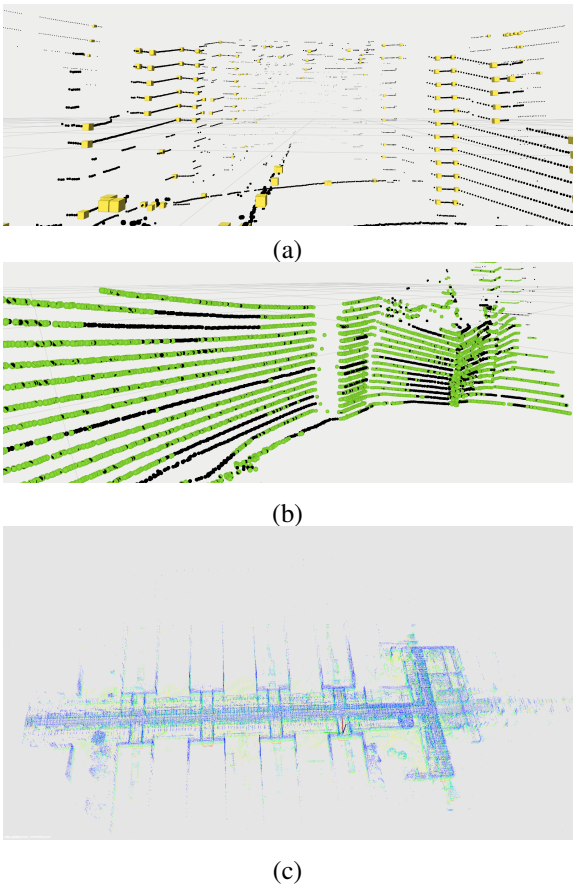
Asad Ullah Khan[1] and Dario Lodi Rizzini[1,2]
[1]RIMLab, Department of Engineering and Architecture
[2]CIDEA Centro Interdipartimentale per l'Energia e l'Ambiente
University of Parma, Italy
{asadullah.khan, dario.lodirizzini}@unipr.it

***Abstract*—This work proposes a refined feature extractor for the LiDAR Odometry and Mapping (LOAM) algorithm often rely on features extracted from point clouds. This paper proposes a novel detection algorithm SKIP-3D (SKeleton Interest Point) for extraction of features namely edges and planner patches from multi-layer LiDAR scan. SKIP-3D make use of the organization of LiDAR measurements to search for salient points in each layer through an iterative bottum-up procedure. In the process it removes the low curvature points to find edges and classifies the points from point clouds acquired from different view points are associated and used for their alignment. Experimental results showed that Fast LiDAR Odometry and Mapping (F-LOAM) based on SKIP-3D feature extractor performs at least better than original F-LOAM feature extractor.**

I. Introduction

Mapping and spacial estimation from sensor data are among the most important problems of autonomous robots. Sensors like LiDARS or depth cameras acquire an accurate 3D representation of the scene near robot in the form of point clouds. Their measurements can be aligned to estimate the robot motion or to be arranged into a consistent map. In particular, LiDARs can provide high range accurate geometric measurements with high resolution in quick time. Such qualities have led to the fast growth of LiDAR robot applications in autonomous driving as well as in industrial applications [1]. A single LiDAR scan will result in 3D sparse representation of the sensor neighborhood, in the form of a point cloud. However, in order to reduce processing time, several algorithms detect salient features from point clouds. Then, these features are used for registration after associating and matching them. This approach is followed by most of the state-of-the-art systems for LiDAR scan registration [2], [3], [4], [5], [6], [7], [8]. The outcome of registration is also used to correct the odometry and the global pose of the robot w.r.t. an initial reference frame.

In this paper, we propose a novel detection algorithm SKIP-3D (SKeleton Interest Point) for the extraction of edges (sharp regions) and plane (smooth regions) from multi-layer LiDAR scans. The proposed SKIP-3D feature detector is integrated with the state-of-the-art Fast LiDAR odometry and mapping (F-LOAM) [8] system and substitutes the original features. Figure 1 illustrates an example of the outcome of F-LOAM with SKIP-3D features. The main characteristic of SKIP-3D is the exploitation of the multi-layer structure and uniform point distributions of LiDAR scans for detecting salient parts. An important issue of multi-layer LiDARs is the unequal horizontal and vertical sampling combined with the relatively large amount of measurements to be processed in real-time. A scan usually covers a horizontal field-of-view (FoV) of 360 degree which is densely sampled (e.g. standard angular resolution is about $0.2°$) whereas its vertical FoV is more limited (few decades of degrees) and sparsely sampled (about $1-2°$). The scattered and non-uniform point clouds acquired



(a) (b) (c)

Fig. 1. Overview of F-LOAM with SKIP-3D features: (a) Yellow points represents an example of SKIP-3D edge points; (b) example of SKIP-3D surface points shown with green points; (c) the complete map of indoor DIA dataset estimated using F-LOAM and SKIP-3D.

978-1-6654-0976-6/21/$31.00 ©2021 IEEE

from LiDAR scan should be handled accordingly while performing registration between different views and specially in associating the points. Although registration algorithms only temporally use features, the huge amount of spatial information available in point clouds should be refined and organized so that it cant be constructively recognized.

The original LOAM algorithm [2] as well as the recent improved variant F-LOAM [8] exploits the roughly semantic classification of points into edge and planar patches to match them and perform registration. SKIP-3D is an alternative algorithm designed to exploit the scan organization into layers or rings. Salient points are obtained through bottom-up removal of less significant points from the ring. Point significancy is measured by a metric proportional to the local curvature. Surface points are found by checking the points between a pair of salient points. In particular, surface points are the ones with a limited distance from the segment connecting the corresponding salient points. The proposed algorithm has been implemented and integrated with F-LOAM and tested on real datasets.

The paper is organized as follows. Section II reviews the state of the art of 3D registration methods. Section III illustrates the F-LOAM system. Section IV presents the proposed SKIP-3D features. Section V reports the experimental results. Section VI gives the concluding remarks.

II. Related Work

Research about 3D LiDAR registration intersects the general point cloud registration as well as localization and mapping problems have been extensively investigated in the past decades. Standard approaches to point cloud registration include Iterative Closest Point (ICP) [9], [10] or Normal Distribution Transform (NDT) [11]. Since point-to-point association is a difficult task, semantic-like classification has been applied to improve the estimation [3], [12]. These works extract edges, planar patches, salient geometries that can be more easily matched between two point clouds. Such geometric features are represented either explicitly, e.g. with a parameterized equation, or implicitly, e.g. labeling points.

Odometry and mapping systems designed for LiDARs adopt more specialized techniques than simple registration algorithms. LOAM [2] is among the first effective algorithms using crafted feature extraction for LiDARs or customized LiDARs obtained with a rolling planar laser scanner. The original LOAM salient points are dependent on the swinging motion pattern of the laser scan, but the algorithm has been adapted to 3D LiDARs. Other registration and mapping tools integrated features extraction from point clouds with other sensors. LIO (Lidar Inertial Odometry) [6], [7] and IN2LAMA (INertial Lidar Localisation And MApping) [5] combines registration and inertial sensors. An augmented version of LOAM combines the geometric features with vision keypoint features [8]. While inertial measurements or vision features can effectively improve registration through an initial guess to the estimation, they also increase the constraints on the input data. The measurements acquired from different sensors must be calibrated w.r.t. spacial extrinsic and temporal parameters. Moreover, the system setup is more complicated. In contrast, F-LOAM (Fast LOAM) [8] only requires 3D LiDAR measurements in the form of organized point clouds. Actually, F-LOAM does not require points with matrix indices, but it recover the layer index from each point. Then each ring is processed to extract high curvature points (edge) and plane points (surface) used by the odometry module. F-LOAM is the closest work and is presented in more detail in the following section.

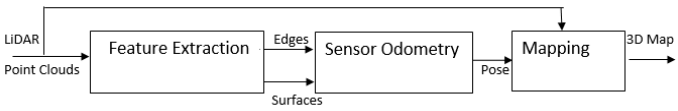


Fig. 2. LOAM algorithm block diagram.

III. LIDAR Odometry and Mapping

F-LOAM [8] is a lightweight system for estimation of sensor odometry and mapping. Its only input consists of point clouds acquired by LiDAR 3D sensors and organized in rings. The system computes the transformation between each current cloud and the previous ones in order to achieve a corrected odometry. Moreover, it computes the complete map through registration and accumulation of previous LIDAR measurements. It does not implement any loop closure procedure or map correction and the resulting map is the outcome of effective registration.

F-LOAM consists of three modules illustrated in Figure 2 corresponding to three different nodes in the ROS implementation. These modules are *feature extraction*, *sensor odometry* and *mapping*.

Feature extraction operates on the sparse input point cloud $\mathcal{P}$ in order to detect two kind of features, edges and surfaces. The point cloud $\mathcal{Q}$ is partitioned into subsets $\mathcal{P}_r$ ($r = 1, \ldots, k$) corresponding to the rings or layers of the LIDAR 3D. The number of the layers k depends on the sensor: for example, Velodyne VLP-16 has $k = 16$, Velodyne HDL-64 $k = 64$. The points belonging to a given layer are acquired by the same emitter/receiver oriented with a different altitude angle and samples a 360 deg horizontal FoV. If the point cloud is organized, then the index r is directly available. Otherwise it is obtained from the altitude angle of each point. The partition into rings is exploited by the original feature extractor as well as by SKIP algorithm described in section IV.

The output of *feature extraction* block consists of two point clouds representing respectively the edge and surface regions. Roughly speaking edge points are the sharp points, i.e. high curvature points on the ring, whereas surface points are the low curvature ones. The criterion used by original feature extraction algorithm is to classify sharp and planar points based on the smoothness of score. Let $\mathcal{R} = \mathcal{P}_r$ be a ring of the point cloud, $\mathbf{p}_{r,i}$ be the point i-th of ring r, $\mathcal{N}_{r,i}$ be the set of adjacent points of $\mathbf{p}_{r,i}$ on the ring r. Then, the

smoothness score is defined as

$$\sigma_{r,i} = \frac{1}{|\mathcal{N}_{r,i}|} \sum_{j \in \mathcal{N}_{r,i}} \|\mathbf{p}_{r,j} - \mathbf{p}_{r,i}\| \tag{1}$$

The value of $\sigma_{r,i}$ is small for flat surface points and large for the edges points. This ring-oriented processing is motivated by efficiency and by the unequal resolution of LIDAR scans along horizontal and vertical directions. Along the rings points are more dense and reliable whereas vertical sampling does not allow the same accuracy.

Sensor odometry block exploits the extracted features to compute the motion of LiDAR sensor and accordingly, of the robot. The motion is evaluated by comparing the features corresponding to two successive 3D scans and by finding the transform that better align them. The registration procedure is briefly presented in next section III-A. Registration based on features is more efficient and robust than raw point-based registration, since it operates on a subset of the input cloud and associates the points according to a label.

Map block builds a 3D map by merging the feature clouds. The alignment among the keyframes of the cloud is provided by the sensor odometry. The odometry block supplies transformation between two consecutive scans and the transformation w.r.t. the origin frame obtained through integration of partial data. The frequency of the transform is crucial for the quality of the reconstructed path and of the generated map. The global map consists of two different global maps, one for edges and one for planar patches. Internally the update is made every time a new keyframe is initialized.

A. Motion and Pose Estimation

According to LOAM work distortion is improved within a sweep, which recursively estimate the pose transform between two successive point clouds (received at two different time stamps). This process of finding the transformation matrix (rotational matrix and translation vector) needs to be perform repetitively which in turn is not effective for large data. On the other hand in F-LOAM a two stage distortion reparation is applied in order to speed up the computation process. In first stage assuming constant velocity (angular, linear) for short interval hence, anticipating the motion and correcting the distortion. Secondly, after pose estimation the distortion is re-computed and the map is updated. The undistorted edge and planner features are ties up with global map with the help of pose estimation. The global map consist of edge and surface features which are updated and kept separately. Finally, once reliable features are obtained then linked global features can be found for each feature points from undistorted edge/plane features. In doing so with this correlation the optimal pose between current frame and global map is estimated by minimizing the distance between feature points and global features. The distance between current edge and global feature is defined as

$$\mathcal{F}_e(\mathbf{p}_e) = \mathbf{p}_n^\top \cdot ((\mathbf{T}_k \ \mathbf{p}_e - \mathbf{p}_e^g) \times \mathbf{n}_e^g) \tag{2}$$

Algorithm 1 Removal of gaps in a point cloud ring

function REMOVEGAP($\mathcal{R} = \{\mathbf{p}_i\}_{i=1\ldots n}$, q_{gap}, m_{gap})
  $\mathcal{Q} \leftarrow \emptyset$
  $i_{first} \leftarrow -1$, $i_{last} \leftarrow -1$
  for $\mathbf{p}_i \in 1 \ldots n$ **do**
    if $\mathbf{p}_i = nan$ **then**
      continue
    end if
    $prev(i) \leftarrow i_{last}$
    if $0 \leqslant i_{last} \leqslant n$ **then**
      $next(i_{last}) \leftarrow i$
      $r_{curr} \leftarrow \|\mathbf{p}_i\|$, $r_{last} \leftarrow \|\mathbf{p}_{i_{last}}\|$
      $r_{mid} \leftarrow (r_{curr} + r_{last})/2$
      if $|r_{curr} - r_{last}| < q_{gap} + m_{gap}\ r_{mid}$ **then**
        $score(\mathbf{p}_{i_{last}}) \leftarrow dist(\mathbf{p}_{prev(i_{last})}, \mathbf{p}_{i_{last}}, \mathbf{p}_i)$
        $\mathcal{Q} \leftarrow push(\mathcal{Q}, \mathbf{p}_{i_{last}})$
      else
        $\mathcal{G} \leftarrow \mathcal{G} \cup \{\mathbf{p}_{i_{last}}\}$
      end if
    else
      $i_{first} \leftarrow i$
    end if
    $i_{last} \leftarrow i$
  end for
  if $0 \leqslant i_{last}, i_{first} \leqslant n$ **then**
    $prev(i_{first}) \leftarrow i_{last}$
    $next(i_{last}) \leftarrow i_{first}$
    $r_{curr} \leftarrow \|\mathbf{p}_{i_{last}}\|$, $r_{last} \leftarrow \|\mathbf{p}_{i_{first}}\|$
    if $|r_{curr} - r_{last}| < q_{gap} + m_{gap}\ r_{mid}$ **then**
      $score(\mathbf{p}_{i_{last}}) \leftarrow dist(\mathbf{p}_{prev(i_{last})}, \mathbf{p}_{i_{last}}, \mathbf{p}_{i_{first}})$
      $\mathcal{Q} \leftarrow push(\mathcal{Q}, \mathbf{p}_{i_{last}})$
    else
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{\mathbf{p}_{i_{last}}\}$
    end if
  end if
  return priority queue $\mathcal{Q}$, gap points $\mathcal{G}$
end function

where $\mathbf{p}_e$ represents edge feature point, $\mathbf{T}_k$ is a transformation matrix represents robot pose at $k-th$ scan (transform between two consecutive frames) and $\mathbf{p}_n$ is the unit vector and $\mathbf{n}_e^g$ is the eigenvector associated with the largest eigenvalue. The distance between planner features and global plane is found by

$$\mathcal{F}_s(\mathbf{p}_s) = (\mathbf{T}_k \mathbf{s} - \mathbf{p}_s^g)^\top \cdot \mathbf{n}_s^g \tag{3}$$

where $\mathbf{n}_s^g$ is the eigenvector associated with the smallest eigenvalue.

For further refining the matching process weight functions are introduced for both features. the optimal pose estimation is derived by solving non-linear equation through Gauss-Newton method and finally the Jacobian matrices of residuals are calculated, which contributes in finding new correspondences and new odometry.

IV. SKIP FEATURES

The main contribution of this paper lies in the SKIP-3D features extracted from the point clouds acquired through 3D LiDARs. These point clouds are organized according to the physical structure of LiDARs, which consist a battery of impulse emitters that fire each firing impulses in different

Algorithm 2 Detection of SKIP-3D features

1: **function** DETECTSKIP($\mathcal{R} = \{\mathbf{p}_i\}_{i=1\dots n}$, d_{th}, q_{gap}, m_{gap})
2: $(\mathcal{Q}, \mathcal{G}) \leftarrow RemoveGap(\mathcal{R}, (q_{gap}, m_{gap}))$
3: **while** not $empty(\mathcal{Q})$ and $score(top(\mathcal{Q}) < d_{th}$ **do**
4: $\mathbf{p}_i \leftarrow pop(\mathcal{Q})$
5: *// Removal of i: scores of its next and prev are changed*
6: **if** $changed(\mathbf{p}_i)$ **then**
7: $score(\mathbf{p}_i) \leftarrow dist(\mathbf{p}_{prev(i)}, \mathbf{p}_i, \mathbf{p}_{next(i)})$
8: $changed(\mathbf{p}_i) \leftarrow false$, $\mathcal{Q} \leftarrow push(\mathcal{Q}, \mathbf{p}_i)$
9: **else**
10: $next(prev(i)) \leftarrow next(i)$
11: $prev(next(i)) \leftarrow prev(i)$
12: $changed(prev(i)) \leftarrow true$
13: $changed(next(i)) \leftarrow true$
14: **end if**
15: **end while**
16: *// Points still in queue are edges* $\mathcal{E}$
17: $\mathcal{E} \leftarrow copy(\mathcal{Q})$
18: *// Surface points* $\mathcal{S}$ *as flat points between edge or gap points*
19: $\mathcal{U} \leftarrow \mathcal{E} \cup \mathcal{G}$
20: sort $\mathcal{U}$ by point index
21: **for** $\mathbf{p}_{i_j} \in \mathcal{U}$ **do**
22: define segment $\overline{\mathbf{p}_{i_j}\mathbf{p}_{i_{j+1}}}$
23: **for** $k = i_j + 1 \dots i_{j+1} - 1$ **do**
24: **if** $dist(\mathbf{p}_k, \overline{\mathbf{p}_{i_j}\mathbf{p}_{i_{j+1}}}) < s_{th}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{p}_k\}$
25: **end if**
26: **end for**
27: **end for**
28: **return** edges $\mathcal{E}$, surfaces $\mathcal{S}$
29: **end function**

vertical directions. The azimuth is the horizontal rotation angle of the LiDAR w.r.t. its head reference whereas the altitude is the vertical angle. Each emitter fires a beam with a specific altitude and acquires the measurements of a layer, also called ring or channel. Thus, the point cloud collected by a 3D LiDAR is partitioned into layers that can be processed independently.

SKIP-3D algorithm exploits the multi-layer structure to extract interest point. The salient points are obtained through bottom-up simplification of the polyline according to the procedure inspired by [13]. The input data consists of a single laser scan with field-of-view of 360 deg represented as a curve connecting adjacent points. The curve is split in correspondence to gaps caused by occlusion, i.e. where the distance between two consecutive points is above a certain threshold, and these points are marked as *gap points*. The procedure to split the ring into intervals at gap points is illustrated by Algorithm 1. The exceptions are the gap points, i.e points in strong range discontinuities due to occlusion and limitation of the FoV. A layer is not represented by a closed curve and the points laying on the gaps are removed from the list of potential points. Lines 3-15 of Algorithm 2 illustrates the procedure for computation of SKIP points from the ring intervals previously computed. The procedure extracts intervals from a priority queue according to cornerness score. The main data structure is the priority queue $\mathcal{Q}$ containing the points $\mathbf{p}_i$ ordered by increasing score.

The score of SKIP-3D Algorithm measures the cornerness

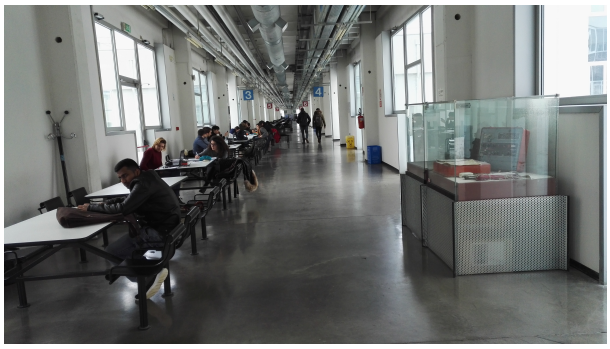


Fig. 3. Environment where the indoor dataset DIA used in the experiments have been obtained.

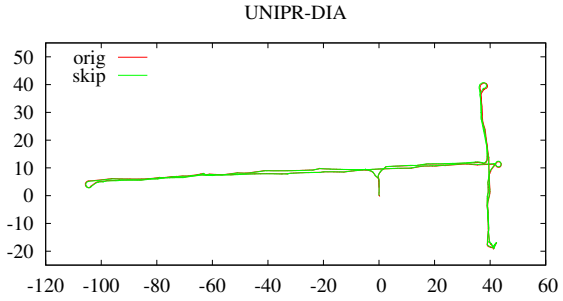


Fig. 4. Estimated trajectories obtained with F-LOAM original and FLOAM-SKIP are plotted in red and green color respectively

of each point based on its previous and next neighbors and is defined as

$$dist(\mathbf{p}_p, \mathbf{p}_c, \mathbf{p}_n) = \|\mathbf{p}_n - \mathbf{p}_c\| + \|\mathbf{p}_c - \mathbf{p}_p\| - \|\mathbf{p}_n - \mathbf{p}_p\| \quad (4)$$

in which, Let $\mathbf{p}_i$ be a generic point of the cloud in a layer with an index i. Ideally, each $\mathbf{p}_i$ has a previous point $\mathbf{p}_{pv_i}$ and a next $\mathbf{p}_{nt_i}$ in a circularly linked list representing the layer. Such indices are initialized as $pv_i = i - 1\ (mod\ n)$ and $nt_i = i + 1\ (mod\ n)$. A score function related to the curvature in a layer is used to evaluate the significance of points The saliency score of a point is given by triangular residual: given $d_{P,i} = \|\mathbf{p}_{pv_i} - \mathbf{p}_i\|$, $d_{N,i} = \|\mathbf{p}_{nt_i} - \mathbf{p}_i\|$ and $d_{C,i} = \|\mathbf{p}_{nx_i} - \mathbf{p}_{pv_i}\|$.
Since the value of the score depends on the previous and next neighbors, the score must be re-computed when a point is removed and the neighbor relations change. Flag variable $change(\mathbf{p}_i)$ in used to keep track of the point with a non-updated score.

V. EXPERIMENTS

The experiments presented in this section has been designed to assess the performance of the registration algorithm with the proposed features. Tests have been performed on both

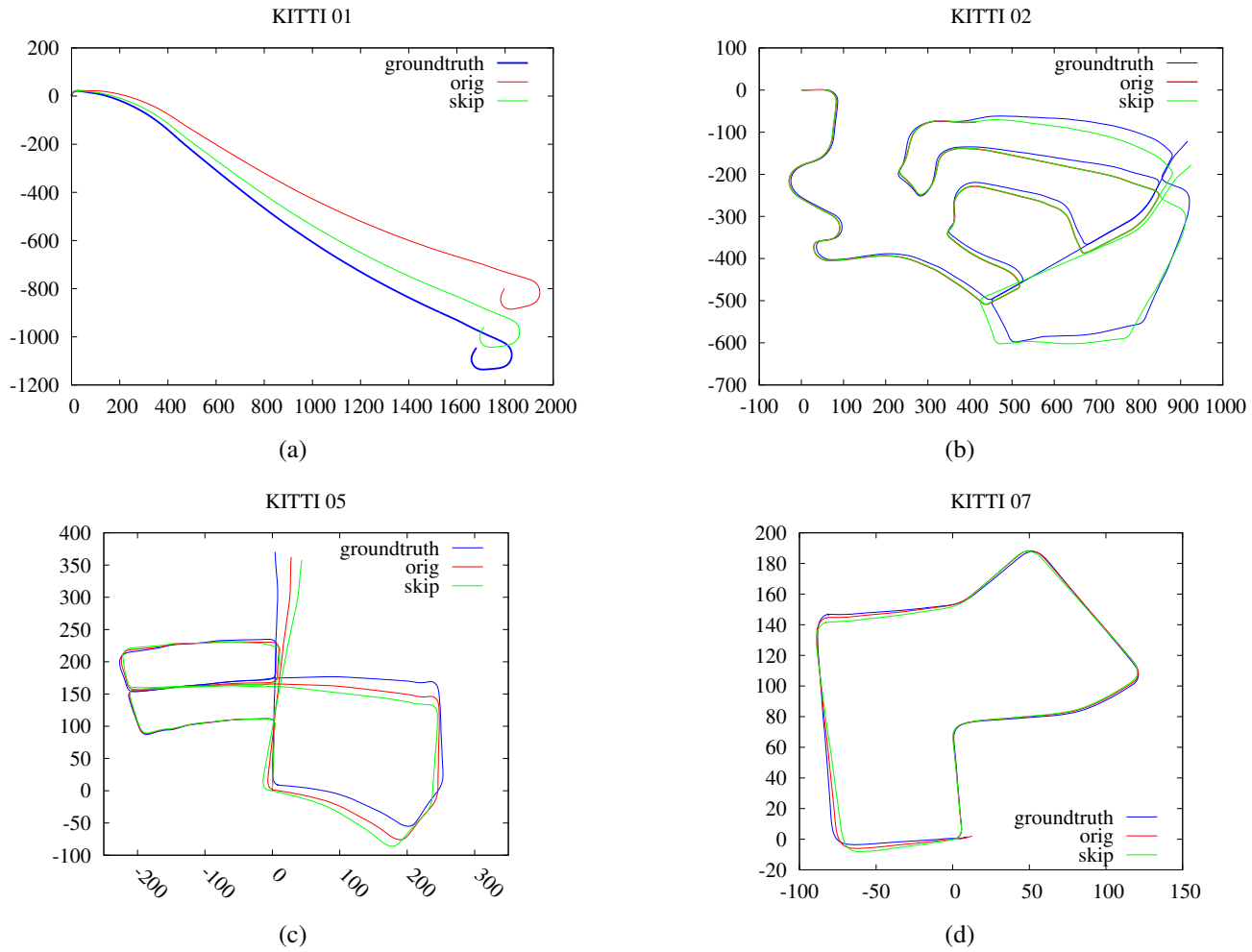


Fig. 5. Robot trajectories estimated using F-LOAM with Original (red), SKIP features (green) and Groundtruth (blue) for the following KITTI sequences: (a) KITTI-01, (b) KITTI-02, (c) KITTI-05 and (d) KITTI-07. The distances are measured in meters.

Dataset	F-LOAM *orig*		F-LOAM *skip*	
	ATE [%]	ARE [10^{-2}°/m]	ATE [%]	ARE [10^{-2}°/m]
KITTI-01	2.58385	0.669808	2.34034	0.587977
KITTI-02	8.56015	4.11018	5.11825	1.96801
KITTI-05	9.88755	4.11246	63.7162	27.2049
KITTI-07	2.67823	1.73642	9.97879	5.32276

TABLE I
AVERAGE TRANSLATIONAL ERROR (ATE) AND AVERAGE ROTATIONAL ERROR (ARE) OBTAINED BY F-LOAM WITH FEATURES *orig* AND *skip* ON THE GIVEN SEQUENCES OF KITTI DATASET.

datasets either indoor or outdoor. The indoor dataset UNIPR-DIA has been acquired by the authors in the main hallway of the Dipartimento di Ingegneria e Architettura of the University of Parma, which consists of a long corridor with branches and tables (see Figure 3). The Pioneer 3DX robot equipped with the Velodyne VLP-16 sensor has been teleoperated and collected a dataset of consecutive LIDAR point clouds. During the teleoperation the operator and other moving people have been captured in the measurements. The acquired dataset consists of 384 scan clouds, each containing 29184 points (including the invalid measurements) organized into 16×1824 matrix.

KITTI dataset [14] has been used for test in outdoor environments. We only used the data collected by Velodyne HDL-64, a multi-layer LiDAR with 64 layers. The dataset also provides accurate groundtruth. Henceafter, the F-LOAM algorithm using its original feature extractor will be referred to as *orig* algorithm and the F-LOAM algorithm using the SKIP feature extractor as *skip*. The experiments have used the implementation of F-LOAM provided by the authors[1] and the implementation of SKIP-3D feature detector.

A. Indoor dataset

These experiments qualitatively compare the trajectory obtained with *orig* and *skip* in indoor environments, where we expect to achieve effective registration due to regularity of building structures. Figures 1(a) and 1(b) show example of respectively SKIP-3D edge and surface points obtained in

[1] https://github.com/wh200720041/floam.

UNIPR-DIA. Edges are often detected on pillar borders or other sharp structures whereas surfaces lies on concrete and glass walls. The complete map of UNIPR-DIA obtained with algorithm *skip* is displayed in Figure Figure 1(c). There is no groundtruth to compare the path estimated by *orig* and *skip*, but the two outcomes can be qualitatively compared as shown in Figure 4. The paths of *orig* (red line) and *skip* (green line) largely overlap and are almost indistinguishable.

B. Outdoor dataset

The performance of F-LOAM with *orig* and *skip* has also been assessed on the outdoor dataset KITTI, a benchmark largely used by robotic and computer vision community. We selected the sequences 01, 02, 05 and 07 and used only the data related to point clouds acquired by the LiDAR sensor. The point clouds have been published with rate $10\ Hz$ comparable to the acquisition rate to simulate the online execution of simultaneous odometry and mapping tasks. As for indoor datasets, we compared F-LOAM *orig* and *skip* in two consecutive trials for each sequence.

The paths obtained on KITTI sequences are illustrated in Figure 5. In particular, each subfigure displays the path estimated with *orig* (red line) and *skip* (green line) as well as the *groundtruth* (blue line), which is provided with KITTI dataset. In these experiments, the robot travels longer paths (magnituge order of unit kilometers) than indoor datasets and F-LOAM only estimates using registration without loop closure. A slight rotation error at some point of the trajectory results in irretrievable propagation of the error to all the next poses. Hence, the drift among *orig*, *skip* and *groundtruth* can be readily observed in the latter segments of each path, but they are still rather consistent for most part of the robot path. In sequence 07 (Figure 5(d)) the path estimated with *orig* prematurely interrupted.

Table I reports the ATE (average transnational error) and ARE (average rotational error) [8] obtained from *orig* and *skip*. The groundtruth and estimated paths are aligned according to the travelled distance from initial frame instead of the unavailable sampling time. ATE and ARE are computed on path slices of lengths $100, 200, \ldots, 800\ m$ sampled with steps of $10\ m$. We observe that ATE and ARE are singificantly smaller with *skip* than with *orig*. The only exception refers to sequence 05. We have investigated the reasons for such large ATE and ARE, which seem inconsistent with the paths in Figure 5(c). It seems that the possible divergence is due to failed alignment of subpaths based on distances, but further analysis is required.

VI. Conclusion

This paper have presented the novel feature detector SKIP-3D integrated in sensor odometry system F-LOAM for LiDARs. The proposed features effectively estimates points belonging to sharp items and planar patches in the scene, which can substitute the original F-LOAM feature extractor. SKIP-3D features are processed online and have been integrated with F-LOAM. The proposed and the original features have been compared in robot odometry and mapping tasks performed in indoor and outdoor environments. F-LOAM with SKIP performs similarly or better than the version with the original features and achieves generally lower position and rotation errors. In future works, we expect to apply the proposed SKIP-3D features with other robot registration and mapping algorithms or to improve the extraction algorithm.

References

[1] R. Horaud, M. Hansard, G. Evangelidis, and al., "An overview of depth cameras and range scanners based on time-of-flight technologies," *Machine Vision and Applications*, vol. 27, pp. 1005–1020, 2016.

[2] J. Zhang and S. S. Singh, "LOAM : Lidar Odometry and Mapping in Real-time," in *Proc. of Robotics: Science and Systems (RSS)*, 2014.

[3] J. Serafin, E. Olson, and G. Grisetti, "Fast and robust 3D feature extraction from sparse point clouds," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 4105–4112.

[4] J. Zhang and S. Singh, "Low-drift and Real-time Lidar Odometry and Mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2017.

[5] C. L. Gentil, T. Vidal-Calleja, and S. Huang, "In2lama: Inertial lidar localisation and mapping," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, May 2019, pp. 6388–.6394.

[6] H. Ye, Y. Chen, and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2019, pp. 3144–3150.

[7] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

[8] H. Wang, C. Wang, C. Chen, and L. Xie, "F-loam : Fast lidar odometry and mapping," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, no. 3, 2021, arXiv:2107.00822.

[9] D. Holz, A. Ichim, F. Tombari, R. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-d," *IEEE Robotics Automation Magazine*, vol. 22, no. 4, pp. 110–124, Dec 2015.

[10] J. Serafin and G. Grisetti, "Using extended measurements and scene merging for efficient and robust point cloud registration," *RAS*, vol. 92, pp. 91–106, 2017.

[11] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[12] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak, "Integrating deep semantic segmentation into 3-d point cloud registration," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 2942–2949, oct 2018.

[13] L. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1185–1190, Oct 2000.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.