

Robust Place Recognition using an Imaging Lidar

Tixiao Shan, Brendan Englot, Fábio Duarte, Carlo Ratti, and Daniela Rus

Abstract—We propose a methodology for robust, real-time place recognition using an imaging lidar, which yields image-quality high-resolution 3D point clouds. Utilizing the intensity readings of an imaging lidar, we project the point cloud and obtain an intensity image. ORB feature descriptors are extracted from the image and encoded into a bag-of-words vector. The vector, used to identify the point cloud, is inserted into a database that is maintained by DBoW for fast place recognition queries. The returned candidate is further validated by matching visual feature descriptors. To reject matching outliers, we apply PnP, which minimizes the reprojection error of visual features’ positions in Euclidean space with their correspondences in 2D image space, using RANSAC. Combining the advantages from both camera and lidar-based place recognition approaches, our method is truly rotation-invariant, and can tackle reverse revisiting and upside down revisiting. The proposed method is evaluated on datasets gathered from a variety of platforms over different scales and environments. Our implementation is available at <https://git.io/imaging-lidar-place-recognition>.

I. INTRODUCTION

Place recognition plays an important role in many mobile robotics applications, such as solving the kidnapped robot problem, localizing a robot in a known map, and maintaining the accuracy of simultaneous localization and mapping (SLAM). During the last two decades, a variety of place recognition methods have achieved great success in tackling such problems using camera, lidar, and other perceptual sensors. Camera-based place recognition methods often extract visual features from textured scenes and find candidates using a bag-of-words approach. However, such methods are subject to illumination and viewpoint change. On the other hand, lidar-based place recognition methods, which often extract local or global descriptors from a point cloud, are invariant to such changes. The long detection range and wide aperture of lidar permit the capture of many structural details of an environment. Yet such details are often discarded during descriptor extraction, which may result in false positive detections when surrounded by repeating structures. Due to the prevalence of low lidar resolution, camera-based methods cannot typically be applied to lidar data. Conversely, lidar-based methods cannot typically be applied to camera data due to a lack of structural information.

However, with the recent availability of high-resolution lidars, such as the Ouster OS1-128 and Velodyne VLS-128, we can begin to bridge the gap between camera-based

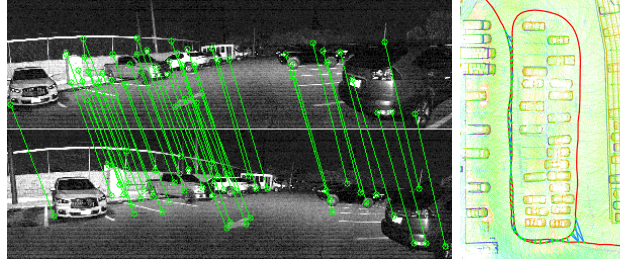


Fig. 1: A demonstration of the proposed method applied to a mapping task. Left: a loop is found when the place is revisited. Grayscale images are intensity images projected from point clouds. Green lines connect the matched features. Right: top-view point cloud map of a parking lot. Red line indicates the traversed trajectory. Blue segments along with green dots indicate detected loop closures using our method. Note that features are extracted from the traffic arrow on the ground for place recognition.

and lidar-based place recognition methods. We refer to such high-resolution lidar that gives image-quality 3D scans as *imaging lidar*. Driven by the prospects of this technology, we present a method for robust place recognition using an imaging lidar. We first project the high-resolution point cloud with intensity information onto an intensity image. We then extract Oriented FAST and rotated BRIEF (ORB) feature descriptors from the intensity image. The extracted descriptors are converted into a bag-of-words (BoW) vector, which forms a compact representation for the original point cloud. A DBoW database is built with these vectors and queried for place recognition. If a candidate is found, we match the ORB descriptors to ensure enough features can be matched between these two places. To reject matching outliers, we formulate the matching problem as an optimization problem by applying Perspective-n-Point (PnP) Random Sample Consensus (RANSAC). A representative example of our method is shown in Figure 1. The main contributions of our work, which combines techniques from both camera and lidar-based place recognition methods, are as follows:

- Real-time robust place recognition that is designed for imaging lidar, and to our knowledge, the first that uses projected lidar intensity images for place recognition.
- The proposed method, which is invariant to sensor attitude changes, can detect reverse revisiting, and even upside down revisiting.
- Our method is extensively validated with data gathered across different scales, platforms, and environments.

II. RELATED WORK

Our work draws upon concepts used in both camera-based and lidar-based place recognition methods. Due to their low hardware cost requirement and robustness in texture-rich

T. Shan, F. Duarte and C. Ratti are with the Department of Urban Studies and Planning, Massachusetts Institute of Technology, USA, {shant, fduarte, ratti}@mit.edu.

B. Englot is with the Department of Mechanical Engineering, Stevens Institute of Technology, USA, benglot@stevens.edu.

T. Shan and D. Rus are with the Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA, {shant, rus}@mit.edu.

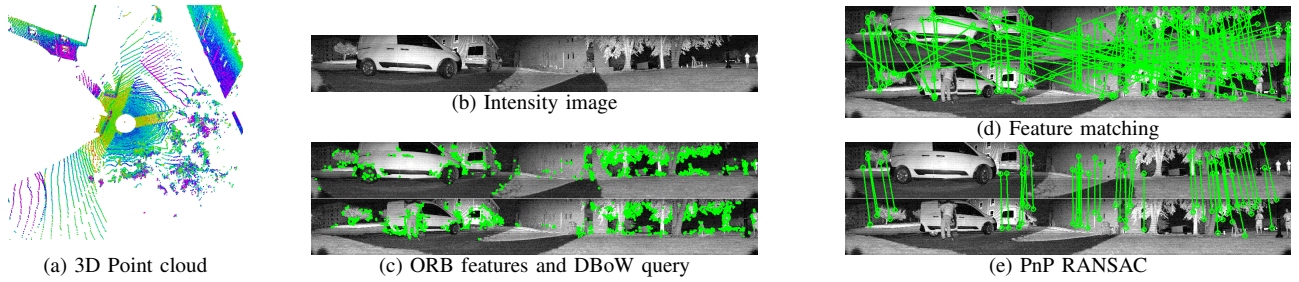


Fig. 2: Demonstration of the proposed methodology: (a) a high-resolution point cloud - color variation indicates intensity change; (b) the intensity image projected from point cloud; (c) extracted ORB features (green dots) and a pair of candidates returned from DBoW query; (d) matched ORB descriptors between two candidates; (e) matched ORB descriptors after PnP RANSAC outlier rejection.

environments, camera-based approaches have been widely used in various SLAM frameworks [1]–[4] for loop closure detection. Such approaches often extract visual feature descriptors from an image and convert them into bag-of-words vectors using DBoW [5] on a pre-trained visual vocabulary. Assuming that there exists visual overlap between images, DBoW queries the database and returns loop closure candidates based on a similarity score between vectors. Because the loop closure candidates from DBoW are prone to false detection, an extra validation step can also be introduced to reject such detection. For example, [4] introduces a two-step geometric validation method, which triangulates visual features, to verify the candidate. The detection performance of camera-based methods, however, heavily depends on the environmental appearance. They are unable to offer reliable detection if the illumination and viewpoint change drastically when a place is revisited.

Lidar-based place recognition methods can be grouped into direct methods and descriptor-based methods. Though the direct methods can operate on the raw point cloud from a lidar without any pre-processing, e.g., [6], [7], their performance is sensitive to point cloud size, initial alignment, and occlusion. Therefore, we focus our discussion on descriptor-based methods, which can offer improved matching robustness. Descriptor-based methods can be categorized into local and global descriptor methods. Local descriptor methods, such as [8], [9], and [10], extract descriptive features from specific regions of a scan, which are then encoded into a histogram for compact representation and query. Due to the application of feature extraction in 3D space, the density of the point cloud impacts the performance of these methods. On the other hand, the recently proposed scan context (SC) [11], intensity scan context (ISC) [12], and lidar iris (IRIS) [13], which are global descriptor-based methods, show superior speed and accuracy over local descriptor methods. Such methods discretize a full 3D point cloud into sectors using polar coordinates. Height or intensity information from each sector is then extracted and encoded into a 2D matrix. Thus these methods are less sensitive to the density of a point cloud. However, these methods may fail if the sensor revisits the same place with a different roll or pitch angle, which changes the signature of each sector greatly.

Similar to ISC, there is a collection of related methods that use lidar intensity information for localization and place

recognition. [14] and [15] rely on a custom-built scanning platform, which requires the robot to stop and scan the environment. Such a design scheme may limit their applications in real-world navigation scenarios. Deep learning-based methods, such as [16] and [17], have incorporated lidar intensity information as input, however, we focus our discussion on methods that can be applied to a wide range of computing platforms, especially low-power embedded systems. Finally, similar to our approach, extracting visual features from lidar intensity images was proposed in [18], which achieves visual odometry by tracking features on a frame-to-frame basis.

In this paper, we propose a novel robust place recognition method that combines the benefits of both camera and lidar-based methods. We extract visual feature descriptors from an intensity image that is projected from the lidar point cloud. With these descriptors, we represent the point cloud using a bag-of-words vector, which is similar to lidar-based global descriptor methods. Then, we perform efficient query of these vectors using DBoW, which resembles the process of camera-based methods. At last, we verify a loop closure candidate by performing feature matching. Matched outliers are rejected by minimizing the projection error of a feature’s position in Euclidean space with their correspondences in 2D image space. The method is described in detail below.

III. METHODOLOGY

In this section, we describe the proposed place recognition method, intended for use with an imaging lidar. We perform a series of processing steps that includes: intensity image projection, feature extraction, DBoW query, feature matching, and PnP RANSAC. An illustrative example of each process step is shown in Figure 2.

A. Intensity Image

The intensity information from a lidar represents the energy level of a return, which is generally influenced by the object surface reflectance and is invariant to ambient light. When a 3D point cloud \mathbb{P} is received, we project it onto an intensity image \mathbb{I} . Each valid pixel in \mathbb{I} can be associated with a point in \mathbb{P} . The value of the pixel is determined by the intensity value of the received point. We then normalize all the pixel values to lie between 0 and 255, which essentially treats the intensity image as a grayscale image and enables

us to process it with various existing image processing approaches. Pixels with no valid points associated are assigned to be zero-valued. An illustrative example of a 3D point cloud is shown in Figure 2(a), where color variation indicates intensity change. The obtained intensity image is shown in Figure 2(b), where the bright and dark pixels correspond to high and low intensity values respectively.

B. Feature Extraction

We next perform feature extraction on the intensity image \mathbb{I} . Rather than assuming a fixed sensor mounting solution [11]–[13], we assume the lidar sensor may undergo aggressive orientation change, which greatly extends the application scenarios of our approach. Therefore, we choose to extract ORB features [19] due to their efficiency and invariance to rotation change. ORB feature descriptors are obtained by first extracting FAST corner features [20] and then describing them using BRIEF descriptors [21]. Due to sensor motion, the scale of an object observed in the intensity image is a function of the distance between the sensor and the object. Similarly, the orientation of the object is also subject to sensor orientation. To increase extraction robustness at various scales and orientations, we apply an eight-level image pyramid with a down-sample ratio of 1.2 to obtain eight intensity images at different resolutions. ORB features are detected using the FAST algorithm in each of the images. The orientation of a feature is determined by computing the intensity change in a circular region that is centered at the feature. At last, the BRIEF algorithm is used to convert a corner feature to a descriptor. We extract a total number of N_{bow} ORB feature descriptors, which are denoted as \mathbb{O} . Note that since we associate each 3D point in \mathbb{P} to each pixel in \mathbb{I} , every feature descriptor in \mathbb{O} is also associated with a 3D point in \mathbb{P} . An example of the extracted ORB features overlaid on an intensity image is shown in Figure 2(c).

C. DBoW Query

We utilize DBoW [5] to convert the ORB feature descriptors \mathbb{O} into a bag-of-words vector using the visual vocabulary proposed in [3]. Thus, the 3D point cloud is now efficiently represented using a sparse bag-of-words vector, which is used to build a database with DBoW. When a new bag-of-words vector is received, we query the database by measuring the similarity between the new vector and the previous vectors using the $L1$ distance. If the similarity between two vectors is larger than a threshold λ_{bow} , we assume a potential revisit candidate is found. The new bag-of-words vector is inserted into the database after the query. Denoting the timestamps at the current and previous vectors as i and j , we send \mathbb{O}_i and \mathbb{O}_j to the processes described in the following sections for further verification. A matched candidate returned by DBoW is shown in Figure 2(c), where the top and bottom images represent \mathbb{I}_i and \mathbb{I}_j respectively.

D. Feature Matching

Usually, the candidates from a DBoW query consist of many false detections. To validate a detection, we match

the descriptors from \mathbb{O}_i and \mathbb{O}_j . We note that matching every descriptor in \mathbb{O}_i and \mathbb{O}_j is not only computationally expensive, but also often results in numerous false matches. To increase the matching success rate, we rank all the descriptors of \mathbb{O}_i in descending order based on their corner scores [22]. The first N_s descriptors, where $N_s < N_{bow}$, with the largest corner scores are selected and denoted as $\mathbf{O}_i : \mathbf{O}_i \subset \mathbb{O}_i$. For each descriptor in \mathbf{O}_i , we find its best match in \mathbb{O}_j . The distance between two descriptors is calculated using the Hamming distance. The matched descriptors are then ranked in ascending order based on their Hamming distance. At last, we introduce a distance test to reject false matches - only matches with Hamming distance less than λ_h are retained for further validation. We set λ_h to twice the smallest Hamming distance of all current matches. In practice, we find that our distance test performs better than [23], which rejects many true positive matches. The matched descriptors are denoted as $\mathbf{O}_i : \mathbf{O}_i \subset \mathbf{O}_i \subset \mathbb{O}_i$ and $\mathbf{O}_j : \mathbf{O}_j \subset \mathbb{O}_j$ respectively, where we have $\|\mathbf{O}_i\| = \|\mathbf{O}_j\|$.

An example of matched descriptors is shown in Figure 2(d). Note that there are still many false positive matches across the images. Though we can choose a smaller λ_h to filter such matches, many true positive matches may be rejected as well. If the number of successful matches after the distance test is larger than N_m , we then proceed to the PnP RANSAC technique described next in Section III-E.

E. PnP RANSAC

If the candidate returned by a DBoW query is incorrect, we may still obtain enough matches from the process described in Section III-D. To further validate the candidate, we formulate the validation problem as a PnP problem [24]. Knowing the 3D Euclidean position of features in \mathbf{O}_i and the 2D image position of features in \mathbf{O}_j , PnP minimizes the reprojection error of the 3D points and their 2D correspondences, and estimates the relative sensor pose between i and j . However, PnP is prone to errors due to false matches in \mathbf{O}_i and \mathbf{O}_j , which is shown in Figure 2(d). To increase the robustness of PnP, we utilize RANSAC [25] here to reject outliers among the matches. Figure 2(e) shows the correct feature matches after outlier rejection. Note that the matched features surrounding the observed person, who is not observed near the van in the latest frame, are rejected after performing PnP RANSAC. If the number of inliers is beyond N_p , we treat this candidate as a correct detection. The estimation of relative sensor pose between i and j , which is a byproduct of PnP, can also be used in a full SLAM framework to facilitate frame-to-frame registration.

IV. EXPERIMENTS

We now describe a series of experiments to quantitatively analyze the proposed method. The sensor used in this paper is the Ouster OS1-128 imaging lidar. The horizontal and vertical field-of-view of the sensor are 360° and 45° respectively. The resolution of the sensor in both directions is 0.35° when it operates at 10Hz, thus resulting in an intensity image with a resolution of 1024 by 128. We compare the proposed method

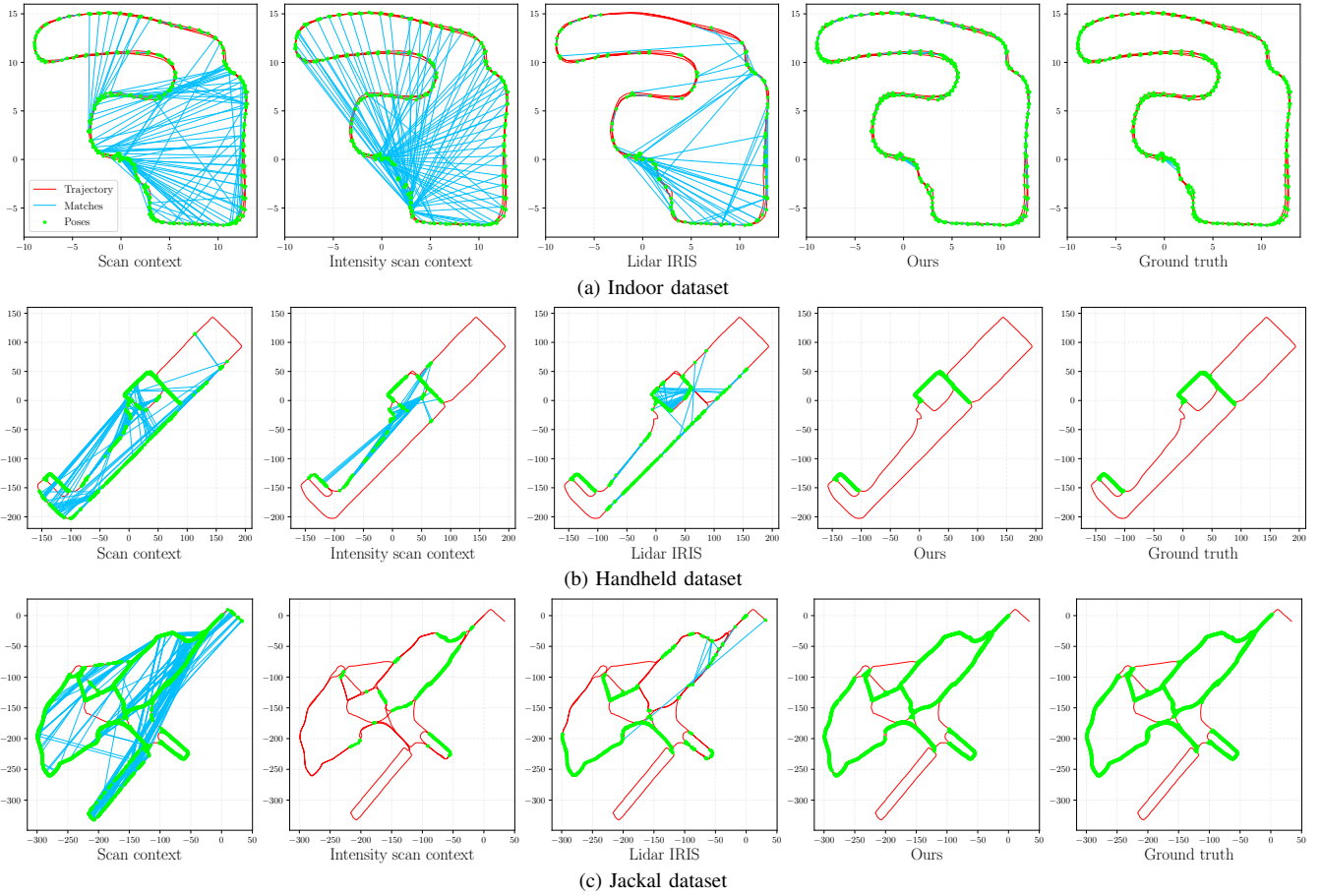


Fig. 3: Detected loop closures of each method. The trajectory of the dataset is colored red. The green dots and blue segments indicate the reported loop closure matches, where the green dots represent the positions of loop closures. Axis units are in meters.

with SC [11], ISC [12], and IRIS [13]. All the methods are implemented in C++ and executed on a laptop equipped with an Intel i7-10710U 1.1GHz CPU.

For validation, we gathered three different datasets across various scales, platforms and environments. These datasets are referred to as *Indoor*, *Handheld*, and *Jackal*, respectively. The lidar scans from the OS1-128 are registered using LIO-SAM [26], which is a tightly-coupled lidar-inertial odometry framework built atop a factor graph. Similar to the previous implementation of SC using [27], we associate each place detection to a node in the factor graph of LIO-SAM, which allows each method’s performance to be validated. In addition to evaluating each place recognition method atop the LIO-SAM solution, we also compare against a ground truth solution in which all possible loop closures more than 30 seconds apart are identified. In our comparison, we use the default parameters from the available implementations of SC, ISC, and IRIS. The parameters of our method are chosen as: $\lambda_{bow} = 0.015$, $N_s = 500$, $N_{bow} = 2500$, $N_m = 15$, $N_p = 15$ for all experiments. Supplementary details of the experiments performed can be found at the link below¹.

A. Indoor Dataset

The *Indoor* dataset is gathered by an operator carrying the sensor walking in an indoor environment, which passes

through doors, corridors, and areas populated with furniture. During data-gathering, the operator follows the same trajectory three times, which start and finish at the same location. When traversing the environment for the third time, the operator turns the sensor completely upside down. Ideally, a robust loop closure method should start reporting detections when the environment is passed the second and third time.

The detected loop closures of each method are shown in Figure 3(a). The LIO-SAM trajectory is colored red. The green dots and blue segments indicate the loop closure matches, where the green dots represent the position of the node in the factor graph of LIO-SAM. If the position between the matched nodes is less than 2m, we consider this detection a true positive, otherwise a false positive. The *Indoor* dataset has 245 ground truth loop closures. Due to the detection mechanisms of SC, ISC, and IRIS, fine details of the environment are discarded in the process of obtaining their point cloud descriptors. Thus, many false positives are reported by these methods, especially when the trajectory is traversed for the third time with the sensor upside down.

Four representative loop closures detected by our method are shown in Figure 4(a). For each detection, the top and bottom row images indicate intensity images captured at the current and previous times. Matched ORB features are connected using green lines. The second example shows

¹<https://youtu.be/yH1hLBFaNoI>

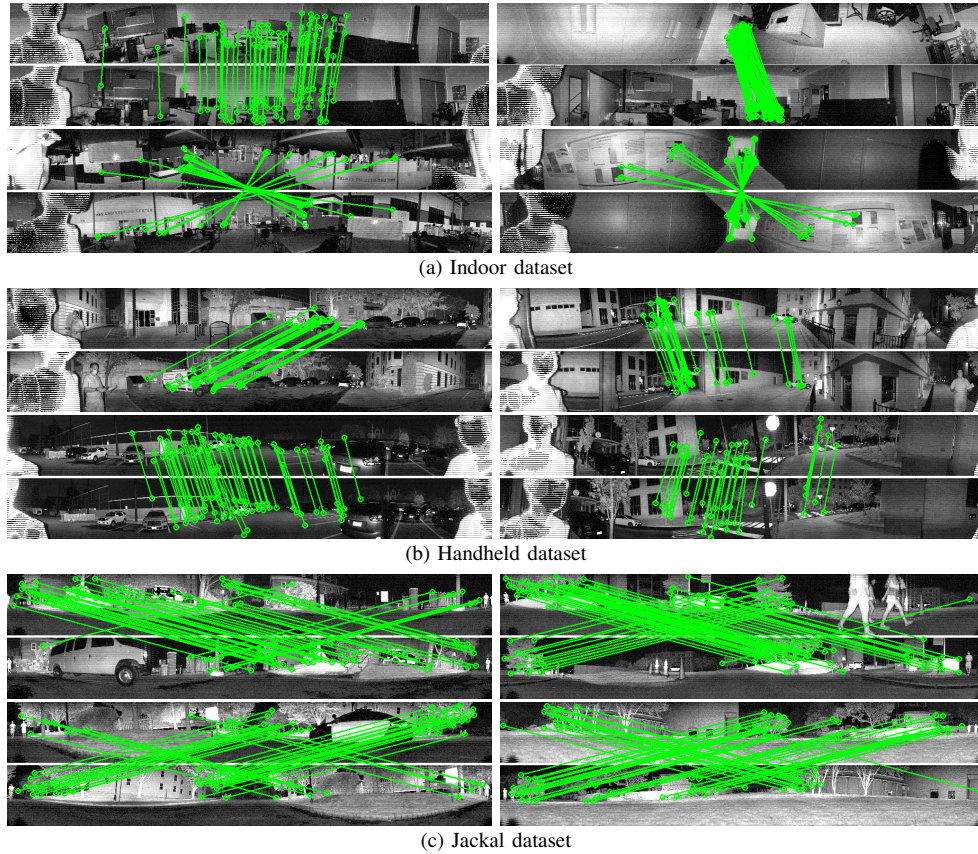


Fig. 4: Twelve representative loop closure detection examples using our method. For each example, the top and bottom row images indicate intensity images captured at the current and previous times, respectively. Matched ORB features are connected using green lines.

detection when we rotate the sensor close to 90 degrees around its forward axis. The third and fourth examples show our method detecting loop closures with the sensor upside down, which is a 180 degree rotation about its forward axis. Note that in the fourth example, our method extracts features from posters hanging in the corridor to aid detection.

The number of true and false positives reported by each method is shown in Table I. Among all the detections recorded, the positive detection rates of SC, ISC, IRIS, and our method are 58%, 51%, 47%, and 100% respectively. We also use receiver operating characteristic (ROC) curves to benchmark the detection accuracy of each method. The ROC curves, which are shown in Figure 5(a), plot the true positive rate against the false positive rate. The area under the curve (AUC) is provided for comparison of prediction accuracy.

B. Handheld Dataset

The *Handheld* dataset is gathered in an outdoor environment with the operator walking with the sensor. This dataset features urban structures and vegetation, moving cars and pedestrians. Since it is not mounted on a fixed platform, the sensor undergoes aggressive attitude change. The distance between matches for considering true positive detection is increased to 4m due to environment scale change.

The reported loop closure detections of each method are shown in Fig. 3(b). Note that SC, ISC, and IRIS report many false positives when we pass through two long streets, which consist of many repetitive scenes (shown in the lower left of

the figure). Our method, on the other hand, can reject false detections by utilizing the fine details of the environment. Four examples of detected loop closures by our method are shown in Fig. 4(b), where cars, windows, and street markings are utilized for detection. As shown in Table I, our method achieves a 98% true positive detection rate, as opposed to 54% for SC, 60% for ISC, and 31% for IRIS. We also achieve the highest AUC of all methods in Fig. 5(b).

C. Jackal Dataset

In the *Jackal* dataset, we mount the sensor on a Clearpath Jackal unmanned ground vehicle (UGV), driving the UGV on asphalt roads, concrete and brick sidewalks, and ground covered by grass and soil. We mainly test the reverse loop closure detection capability of all methods. Though SC is able to detect 1429 out of 1447 ground truth loop closures, it reports 409 false positives, which accounts for 22% of all detections. ISC and IRIS report significantly fewer detections compared with SC and our method. Our method detects 1245 loop closures with 98% of them being true positives. Again, our method achieves the highest AUC in Figure 5(c).

Representative reverse loop closure detection examples (involving traversal in opposite directions) are shown in Fig. 4(c). Our method extracts features primarily from trees and buildings to support place recognition. Many moving vehicles and pedestrians are observed during the data-gathering process, and our proposed method rejects the matched features from dynamic objects using the technique discussed in

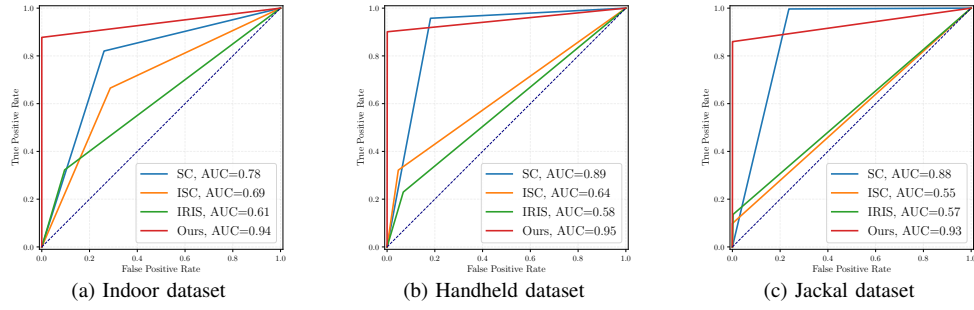


Fig. 5: ROC curves and AUC for all competing methods. The results are obtained by comparing the reported loop closure detection with the ground truth loop closures. Among all the methods, the proposed method achieves the highest AUC over various datasets.

Sec. III-E. In the fourth example of Fig. 4(c), the ground surrounding the sensor is completely covered by grass.

D. Runtime Benchmarking

The computation time per query averaged over each dataset, for each method, is shown in the last column of Table I. SC is the most time-efficient method due to its introduction of a ring key search algorithm for fast database query. Though the point cloud descriptor comparison times for ISC and IRIS are similar to SC, their computation times increase dramatically when they are applied to a full SLAM framework. ISC naively compares the current descriptor with all the descriptors in the database during a query, thus its computation time grows unbounded. Though IRIS implements a similar search algorithm as SC for query, its efficiency is not ideal due to the design of the search key, which has a dimension of 80 as opposed to 20 in SC. Though our method runs slower than SC, it is significantly faster than ISC and IRIS, while achieving the highest true positive detection accuracy. It's worth noting that average DBoW query time for the three datasets is 22.2 ms, 33.1 ms, and 58.8 ms respectively, which increases as the size of the database increases. The computation time for the remaining components of our method is similar across all datasets.

TABLE I: Quantitative results of competing methods

Dataset	Method	Detected loops	True positives	False positives	Time (ms)
Indoor (245 loops)	SC	231	134 (58%)	97 (42%)	6.17
	ISC	196	100 (51%)	96 (49%)	164.8
	IRIS	90	42 (47%)	48 (53%)	253.4
	Ours	215	215 (100%)	0 (0%)	49.5
Handheld (283 loops)	SC	499	271 (54%)	228 (46%)	6.49
	ISC	150	90 (60%)	60 (40%)	417.0
	IRIS	150	47 (31%)	103 (69%)	382.4
	Ours	256	252 (98%)	4 (2%)	65.7
Jackal (1447 loops)	SC	1838	1429 (78%)	409 (22%)	9.19
	ISC	142	134 (94%)	8 (6%)	630.4
	IRIS	202	168 (83%)	34 (17%)	423.5
	Ours	1245	1226 (98%)	19 (2%)	93.2

E. Lidar Resolution Benchmarking

Finally, we provide detection results for our method using down-sampled intensity images. The image resolutions tested are 1024 by 64, 1024 by 32, and 1024 by 16, which are equivalent to using a lidar with 64, 32, and 16 channels respectively. As shown in Table II, the detection rate when using fewer lidar channels decreases significantly. This is

because the number of extracted ORB features from a low-resolution intensity image is limited, and the performance of DBoW query and feature matching deteriorates accordingly. Our method is clearly sensitive to the available lidar resolution, and most suitable for use with hi-res imaging lidar.

TABLE II: Detection with differing lidar resolution

Dataset	Lidar channels	Detected loops	True positives	False positives
Indoor (245 loops)	64	181	177 (98%)	4 (2%)
	32	75	75 (100%)	0 (0%)
	16	3	3 (100%)	0 (0%)
Handheld (283 loops)	64	236	234 (99%)	2 (1%)
	32	104	103 (99%)	1 (1%)
	16	6	5 (83%)	1 (17%)
Jackal (1447 loops)	64	1040	1022 (98%)	18 (2%)
	32	435	428 (98%)	7 (2%)
	16	35	34 (97%)	1 (3%)

V. CONCLUSIONS AND DISCUSSION

We propose a novel methodology for place recognition using an imaging lidar, which demonstrates robustness in a variety of settings. Our method combines advantages from both camera and lidar-based place recognition approaches. Similar to camera-based methods, we extract ORB feature descriptors from the intensity image projected from a 3D point cloud. We utilize DBoW to represent the point clouds using bag-of-words vectors and to perform place recognition queries, which is similar to lidar-based global descriptor methods. Upon receiving a candidate from a query, we conduct ORB descriptor matching to verify its legitimacy. The outliers among the matched descriptors are rejected using PnP RANSAC. The proposed method is evaluated on datasets gathered in both indoor and outdoor environments at different scales. The results show that our method achieves higher accuracy and robustness than other lidar-based place recognition methods.

We are aware that the KITTI dataset [28] is used in [11], [12], and [13] for benchmarking. However, the lidar, Velodyne HDL-64e, used in the KITTI dataset features nonlinearly distributed channels along its spinning axis. Depending on the sensor attitude, the same object may have different appearances at different vertical locations of the intensity image. We are unable to extract consistent ORB feature descriptors for DBoW query or feature matching. Therefore, we did not include results using the KITTI dataset.

REFERENCES

- [1] C. Kerl, J. Sturm, and D. Cremers, “Dense Visual SLAM for RGB-D cameras,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2100–2106, 2013.
- [2] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” *European Conference on Computer Vision*, pp. 834–849, 2014.
- [3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [4] T. Qin, P. Li, and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [5] D. Gálvez-López and J. D. Tardos, “Bags of Binary Words for Fast Place Recognition in Image Sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.
- [6] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” *Sensor fusion IV: Control Paradigms and Data Structures*, vol. 1611, pp. 586–606, 1992.
- [7] S. Rusinkiewicz and M. Levoy, “Efficient Variants of the ICP Algorithm,” *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.
- [8] S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.
- [9] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Aligning point cloud views using persistent feature histograms,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3384–3391, 2008.
- [10] S. Salti, F. Tombari, and L. Di Stefano, “SHOT: Unique Signatures of Histograms for Surface and Texture Description,” *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.
- [11] G. Kim and A. Kim, “Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4802–4809, 2018.
- [12] H. Wang, C. Wang, and L. Xie, “Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection,” *arXiv preprint arXiv:2003.05656*, 2020.
- [13] Y. Wang, Z. Sun, J. Yang, and H. Kong, “LiDAR Iris for Loop-Closure Detection,” *arXiv preprint arXiv:1912.03825*, 2019.
- [14] J. Guo, P. V. Borges, C. Park, and A. Gavel, “Local descriptor for robust place recognition using LiDAR intensity,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1470–1477, 2019.
- [15] K. P. Cop, P. V. Borges, and R. Dubé, “Delight: An Efficient Descriptor for Global Localisation using Lidar Intensities,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3653–3660, 2018.
- [16] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer, “OverlapNet: Loop Closing for LiDAR-based SLAM,” *Proceedings of Robotics: Science and Systems (RSS)*, 2020.
- [17] I. A. Barsan, S. Wang, A. Pokrovsky, and R. Urtasun, “Learning to Localize Using a LiDAR Intensity Map,” *Conference on Robot Learning*, pp. 605–616, 2018.
- [18] T. D. Barfoot, C. McManus, S. Anderson, H. Dong, E. Beerepoot, C. H. Tong, P. Furgale, J. D. Gammell, and J. Enright, “Into Darkness: Visual Navigation Based on a Lidar-intensity-image Pipeline,” *Robotics research*, pp. 487–504, 2016.
- [19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An Efficient Alternative to SIFT or SURF,” *International Conference on Computer Vision*, pp. 2564–2571, 2011.
- [20] E. Rosten and T. Drummond, “Machine Learning for High-speed Corner Detection,” *European Conference on Computer Vision*, pp. 430–443, 2006.
- [21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary Robust Independent Elementary Features,” *European Conference on Computer Vision*, pp. 778–792, 2010.
- [22] C. G. Harris, M. Stephens *et al.*, “A Combined Corner and Edge Detector,” *Alvey Vision Conference*, vol. 15, no. 50, pp. 10–5244, 1988.
- [23] D. G. Lowe, “Distinctive Image Features from Scale-invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [24] V. Lepetit, F. Moreno-Noguer, and P. Fua, “EPnP: An Accurate $\mathcal{O}(n)$ Solution to the PnP Problem,” *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, 2009.
- [25] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [26] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, “LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [27] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain,” *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758–4765, 2018.
- [28] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision Meets Robotics: The KITTI Dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.