Proceedings of the 2017 18th
International Conference on Advanced Robotics (ICAR)
Hong Kong, China, July 2017

3D Feature Points Detection on Sparse and Non-uniform Pointcloud for SLAM

Prarinya Siritanawan, Moratuwage Diluka Prasanjith, and Danwei Wang
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore

***Abstract*—In this paper, we propose a novel 3D feature point detection algorithm using Multiresolution Surface Variation (MSV). The proposed algorithm is used to extract 3D features from a cluttered, unstructured environment for use in real-time Simultaneous Localisation and Mapping (SLAM) algorithms running on a mobile robot. The salient feature of the proposed method is that, it can not only handle dense, uniform 3D point clouds (such as those obtained from Kinect or rotating 2D Lidar), but also (perhaps more importantly) handle sparse, non-uniform 3D point clouds (obtained from sensors such as 3D Lidar) and produce robust, repeatable key points that are specifically suitable for SLAM. The efficacy of the proposed method is evaluated using a dataset collected from a mobile robot with a 3D Velodyne Lidar (VLP-16) mounted on top.**

I. Introduction

The Simultaneous Localisation and mapping (SLAM) problem is considered to be the most important part of any mobile robot [1] due to the fact that the ability to accurately estimate its pose in possibly unknown, unstructured environments is an essential prerequisite for any robotics application such as exploration, 3D mapping and search and rescue missions to name a few. The three most common approaches found in the literature to solve the SLAM problem are feature based, grid based and topology based solutions. Out of these techniques, feature based SLAM solutions are considered to be the most suitable approach for 3D SLAM solutions since the computational complexity (in terms of data association and measurement updates) is lower compared to the other two, making them a stronger candidate approach. However, unlike 2D environments, 3D environment present some more challenges that needs to be properly addressed in order to produce successful SLAM solutions. For a feature based SLAM algorithm, one of the most important requirement is the ability to extract robust, repeatable features from the possibly cluttered, unstructured environment.

Given an unorganized 3D pointcloud, feature points representing the distinguish landmarks locations are essential. These landmarks can be any type of representations if they are mutually and steadily localized by all robots. They can be corners, edges, curvatures, surface variations, or any specific objects information.

This paper presents a real-time, robust, and stable, 3D feature points extraction. The algorithm is designed for SLAM in full three dimensional free space. The algorithm can operate across multiple unmanned ground vehicles (UGVs) and/or unmanned aerial vehicles (UAVs) in unstructured environments, without any restriction from prior assumptions of 3D scenes. In addition, the algorithm is also freely operational in many other applications such as feature matching, registration, and etc.

In this paper, we are particularly interested in surface variation estimation approach. We developed a real-time 3D feature detector for SLAM by a set of processes, which consists of ground plane removal, keypoint candidates selections by using Multiresolution Surface Variation (MSV), and 3D clustering method. The features were calculated from 3D pointcloud data given by sparse rotating Lidar sensor (Velodyne VLP-16). This type of sensors usually produces non-uniform distribution of points, i.e. the pointcloud is dense at the closer range and become sparser at the far range. Given this data inconsistency, mesh operation is not applicable as the triangulation would yield incrementally larger errors on the far distances.

Another major issue with the feature detection is how to specify the radius size for computing a keypoint score around a query region. This leads to the optimization issue which require manual adjustments to get a proper parameter for a specific environment. This exhaustive parameter tuning is insufficient and undesired. The systems with such a regulation will eventually fail to reconcile with other environments. For example, a robot with indoor setting is employed to outdoor environment.

Our algorithm aims to generalize the feature detection component to operate in a broader range of environments. In order to avoid the bias from selection of computing radias, this paper suggested to estimate a keypoint score with varying computation scales scheme. Our proposed MSV feature detection calculates the eigenvectors representing a sub-region of pointcloud repetitively with varying computational scales. Then, the resultant surface variation scores of each scale are propagated and combined. This method can work in various environments with less effort of tuning and capable to work in real-time.

II. Related Works

Feature points represent the salient areas of the scene by a tiny collection of 3D points $\{(x, y, z)\}$, which is a subset of the original pointcloud. The signatures of feature points must be sufficiently unique enough to be distinquished from their surrounding irrelevent points. Moreover, feature points must be repeatable whenever the objects-of-interest are present in the scene perception regardless to robot positions, viewpoints, or

978-1-5386-3157-7/17/$31.00 ©2017 IEEE

noise. However, it is commonly known that there is no absolute rule of which attributes of a 3D surface must be defined as a feature point.

Generally, 3D keypoint extraction ideas are usually active around the idea of formulating corners or curve surfaces representation. Much research extended famous 2D keypoints extraction methods to 3D such as Harris corner detector [2] or SIFT [3]. A 3D variant of Harris detector was done by Siripan and Bustos [4], and 3D SIFT can be found in Flint et al. [5] and Godil et al. [6].

Other famous 3D keypoint extraction methods can be found in Unnikrishnan and Hebert [7]. They represented the surface by Laplace Beltrami Scale Space (LBSS). Similar approach were proposed by Sun et al. with the Heat Kernel Signature (HKS) [8]. However, the these methods either require mesh representation, or they are computational inefficient for a large environment.

In this paper, we are interested in surface variation approach by using eigen method. The algorithm is very simple, easy to implement, but yet fast and powerful. Most importantly, it work well with sparse and non-uniform distributed pointcloud. The algorithm basically have two steps: first is to compute the eigenvectors and eigenvalues of the local pointcloud, second is to impose filtering rules, which are based on the characteristic or relationship of each eigenvalues. The filtering criteria can be different from work-to-work. One of the early work by Pauly applied the criteria $\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} < \tau$ as a rule to filter the keypoints [9]. Similarly, Ho and Gibbins used $\frac{\lambda_1}{\lambda_1+\lambda_2+\lambda_3} < \tau$ [10]. In contrast, Matei et al. directly employed smallest λ_3 to determine the potential keypoints [11]. Zhong et al. exploited the intersection of two criterias $\frac{\lambda_2}{\lambda_1} < \tau_1$ and $\frac{\lambda_3}{\lambda_2} < \tau_2$ in their feature descriptor paper [12]. On the top of above filtering criterias, Mian et al. proposed an additional criteria to rules out the unwanted keypoints by Keypoint Quality metric (KPQ) [13]. According to the evaluation by Salti et al. [14], KPQ yields the best overall repeatability against many famous keypoint extraction methods. However, the KPQ was argued its drawback on high computation complexity [14].

Furthermore, the methods mentioned above were mainly tested on static 3D object dataset with simulation of varying viewpoints and noises. The 3D objects are recorded in form of full 3D mesh or dense pointcloud from 3D scanner. Due to the sparse and non-uniform characters of pointcloud from Velodyne VLP-16 sensor, it is interesting to investigate the effectiveness of feature detection methods on such data. Our empirical test with Velodyne VLP-16 sensor is shown in the experimental results section.

III. Proposed Method

The feature point observations F_k in the current framework have been defined by the object cluster locations. In order to reduce the burden to SLAM component and maximize the repeatability of features. The procedure is in three steps; ground plane removal, candidate points selection, and 3D clustering.

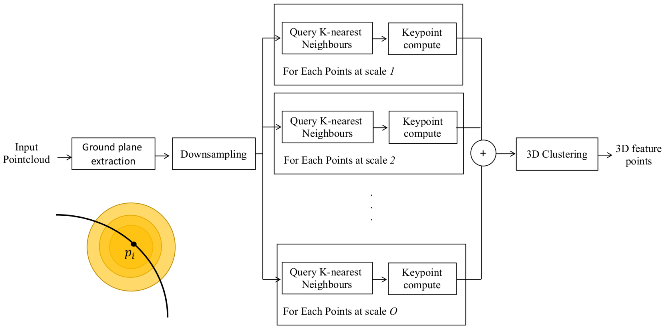

Input Pointcloud
Ground plane extraction
Downsampling
Query K-nearest Neighbours
Keypoint compute
For Each Points at scale 1
Query K-nearest Neighbours
Keypoint compute
For Each Points at scale 2
Query K-nearest Neighbours
Keypoint compute
For Each Points at scale O
+
3D Clustering
3D feature points
p_i

Fig. 1: Proposed procedure

A. Ground plane removal

To speed-up and improve feature extraction, ground plane is removed in prior to segmentation step. This paper has used the concept of voxel based ground plane removal which is similar to the idea in Douillard et al. work [15]. Due to performance and accuracy issue, we improved the algorithm with $B-R-H$ coordinate system. B is a bearing angle (in degree) rotating around the VLP-16 sensor where its zero degree is located at the frontal direction of robot. R is a range distance (in meter) from sensor toward a point. H is a height distance (in meter) from robot platform and perpendicular to B and R axes.

Given a pointcloud $\mathcal{P}_k = \{p_1, p_2, ..., p_N\}$. A point in Cartesian coordinate is denoted by $p_n^{(xyz)} = \{x_n, y_n, z_n\}$. N is total number of points in a pointcloud. Firstly, we convert each point to *B*-*R*-*H* coordinate system $p_n^{(xyz)} \mapsto p_n^{(brh)}$. Then, we assign these points to the voxel grid in *B*-*R*-*H* coordinates as shown in Fig. 2.

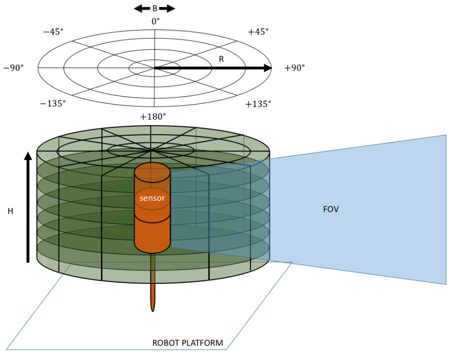


Fig. 2: $B-R-H$ grid system for ground plane removal component.

For each voxel grid, we calculate the means and variances of pointcloud along height direction $\{m_H(\acute{b}, \acute{r}, \acute{h}),\ \sigma_H(\acute{b}, \acute{r}, \acute{h})\}$, where $\acute{b}, \acute{r}, \acute{h}$ are grid indices. To extract the ground plane, we determine potentially ground plane grids by checking the height variances of each grid that satisfy $\sigma_H(\acute{b}, \acute{r}, \acute{h}) < \tau_\sigma$. Then, the grids are clustered and merged. The largest cluster with corresponding average height is considered as ground plane grids. Finally, we remove the ground plane points in the pointcloud $\mathcal{P}_k$. The non-ground plane pointcloud is denoted by

$\hat{\mathcal{P}}_k = \{p_1, p_2, ..., p_I\}$, where I is total number of non-ground points, and $I < N$ (Fig. 3).

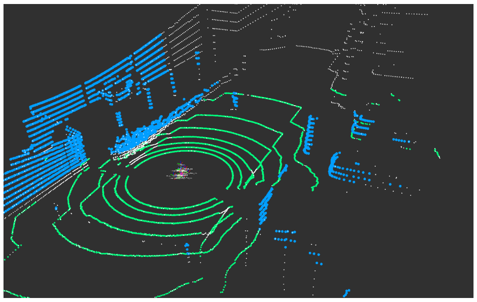


Fig. 3: Example result from ground plane extraction. Green colour represents ground points, and Blue colour represent non-ground points

B. Candidate Points Selection

1) Surface Variation: For a given non-ground pointcloud $\hat{\mathcal{P}}_k$, the local regions of 3D surface is sampled. Surface Variation (SV) at p_i can be computed according to procedure shown in Fig. 4.

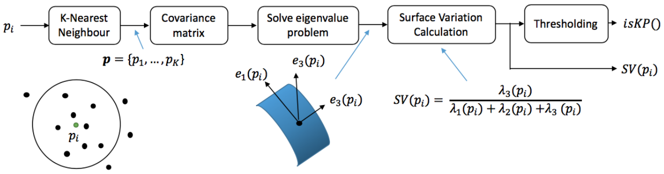

p_i
K-Nearest Neighbour
Covariance matrix
Solve eigenvalue problem
Surface Variation Calculation
Thresholding
$isKP()$
$SV(p_i)$
$\boldsymbol{p} = \{p_1, ..., p_K\}$
$e_1(p_i)$
$e_3(p_i)$
$e_3(p_i)$
$SV(p_i) = \frac{\lambda_3(p_i)}{\lambda_1(p_i) + \lambda_2(p_i) + \lambda_3(p_i)}$
p_i

Fig. 4: Procedure of Surface Variation calculation.

Firstly, all points around a query point p_i within r radius are sampled. Then, the 3D subspace representing the local region is calculated by principle component analysis (PCA). The method basically solves eigenvalues problem such that it can minimize the variance S of 3D points distribution.

Given total M points, the point p_m is a column vector with D dimensions, where D=3 dimensions. m is the point index. The average of all local points $\bar{p}$ within searching radius, and correspondence covariance matrix S are given by:

$$\bar{x} = \frac{1}{M}\sum_{m=1}^{M} p_m \tag{1}$$

$$S = \frac{1}{M}\sum_{m=1}^{M}(p_m - \bar{x})(p_m - \bar{p})^T \tag{2}$$

Then, the eigenvectors $\phi = [e_1, e_2, e_3]^T$ and corresponding eigenvalues $\lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ are calculated by:

$$S\phi = \lambda\phi \tag{3}$$

Since the input is three dimensional vector, three eigenvectors and eigenvalues, which represent a distribution character of local regions, are computed in this process. The eigenvectors with two largest eigenvalues represent the most fitting plane of the local surface. The third eigenvector is a tangential normal vector perpendicular to the first and second eigenvectors representing the variation of surface.

However, the performance of SV approach is heavily biased to the size of interested region depend on the operated environments as shown in Fig. 5. If the interested region is too small, it cannot capture the details of large objects. The method will also fail at far range where the density of pointcloud become sparser. In contrast, too large region of interest tend to permeate all the irrelevant pointcloud, including plain area, and recognize them as the high SV area. There are cases that the method may dissolve the desired objects to the plain area. Moreover, the effective range and computation time of the algorithm is largely depended on the chosen scale. Thus, fine-tuning of this parameters is required differently from one environment to another.

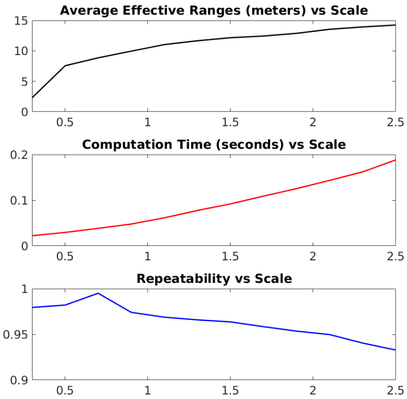


Fig. 5: Radius size (in meter) of SV Computating Scale (fix-scale) versus (Top) Average Effective Ranges, (Middle) Computation time, and (Bottom) Repeatability, from the tested indoor sequence

2) Multiresolution Surface Variation: The feature points for SLAM is expected to be free from scale dependency and environments. Multiresolution Surface Variation (MSV) is proposed to handle sparse and non-uniform distribution. The method reduce the bias of fix-scale computation by computing the average of surface variation from different computation scale. The MSV score is calculated by:

$$MSV(p_i) = \frac{1}{O}\sum_{o \in O}\frac{\lambda_3(i,o)}{\lambda_1(i,o) + \lambda_2(i,o) + \lambda_3(i,o)} \tag{4}$$

where O is total number of calculation scales. Example of MSV score is shown in Fig. 6, where the plot is colorized with *jet* colourmap scheme. Red colour represent lowest surface variation level (0) and blue color represents highest surface variation level (1). Finally, the non-keypoints are filtered out by thresholding procedure leaving only the salient keypoints.

C. 3D Clustering

The amount of candidate points in sec. III-B2 can be several hundreds to thousands and it may overload the SLAM

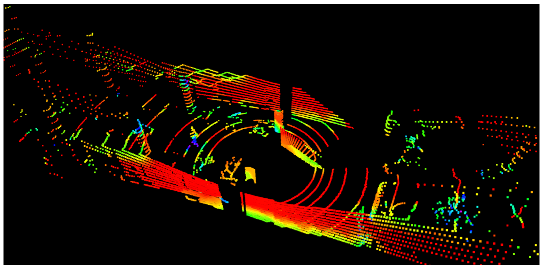


Fig. 6: Surface variation levels by using MSV technique. Pointcloud is visualized with jet colourmap from Red (lowest surface variation level) to Blue (highest surface variation level)

computation. Thus, the segmentation is used to summarize the candidate points that belong same objects. In this step, we use Euclidean Distance based clustering method provided by Point Cloud Library (PCL). By this approach, we can filter irrelevant information conveniently. Finally, the output feature points F_k are defined by the centre of each clusters as shown in Fig. 7.

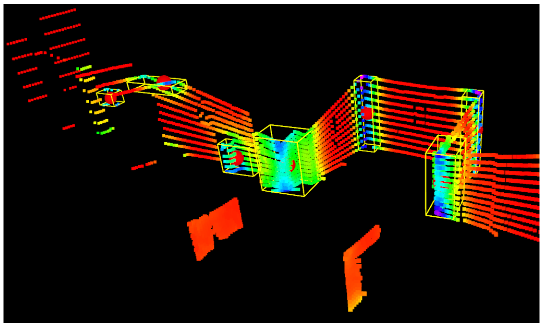
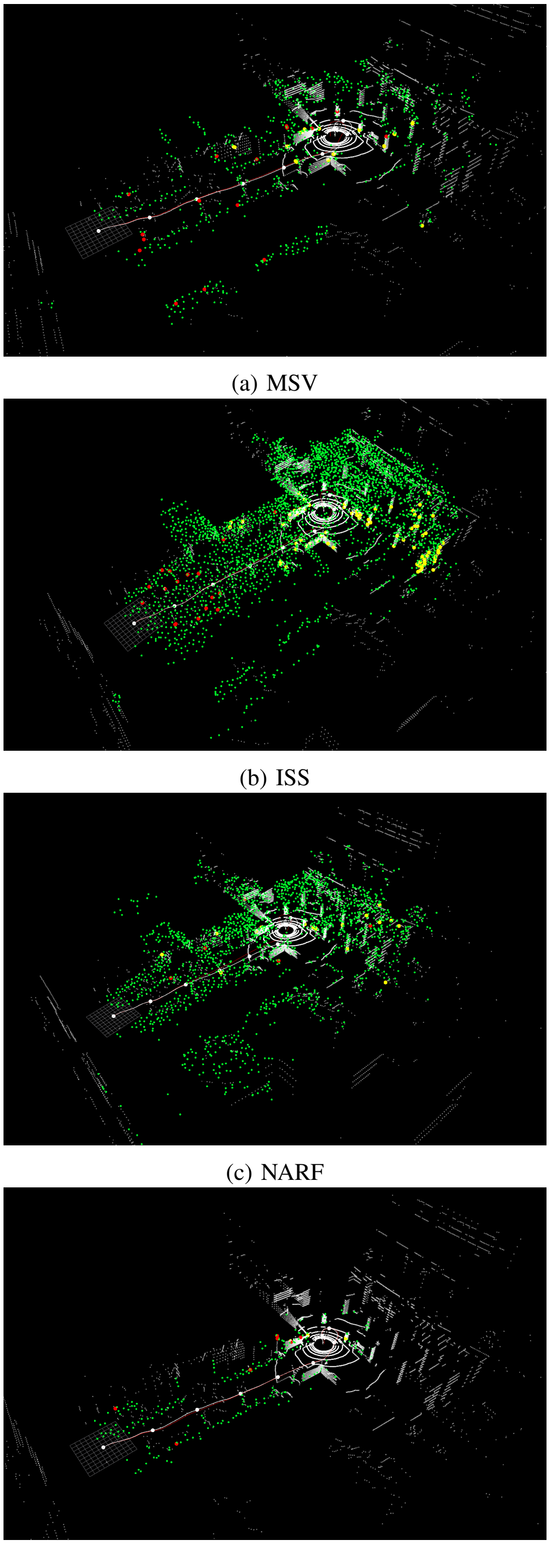


Fig. 7: Example result from clustering segmentation to produce feature points. The yellow bounding box represents the cluster area, and the red sphere represents a centre location of the cluster

IV. Experimental Results

The performance of the proposed feature extraction method is evaluated using a 3D SLAM algorithm developed using Random Finite Set (RFS) representation of the feature map and the feature observations (see [16] for more details on the application of RFS methods in SLAM, and [17] for a more advanced application of RFS methods on map and features). More specifically, the developed SLAM algorithm is capable of tracking 3D features (obtained from the proposed feature extraction algorithm) from the environment in order to simultaneously map and localize itself within the possibly unknown, unstructured environment.

One of the most challenging part in this research is the evaluation of detection algorithms, what kind of quantities should be measured, and how to apply the measurement with dynamic sparse 3D scenes while the robot is moving. The quality of salient features is measured by *repeatability*. One of the best evaluation is reported by Salti et al. [14]. They evaluated the quality of 3D keypoints which mostly done on



(a) MSV

(b) ISS

(c) NARF

(d) SIFT3D

Fig. 8: Feature points are continuously collected in the global feature map while robot is moving. The transformation of feature points in sensor coodinate to map coodinate is estimated by 3D SLAM output in real-time

mesh representation of static objects. In this paper, we adopted similar metric to evaluate the quality of our proposed method on dynamic sparse and non-uniform pointcloud.

A feature point from sensor at time k is said repeatable if its transformation is close to an existing feature points in the global feature map under a small distance criteria ϵ (Eq. (5)). Then, the Repeatability at time k can be calculated by an average of repeatable points over total number of feature points.

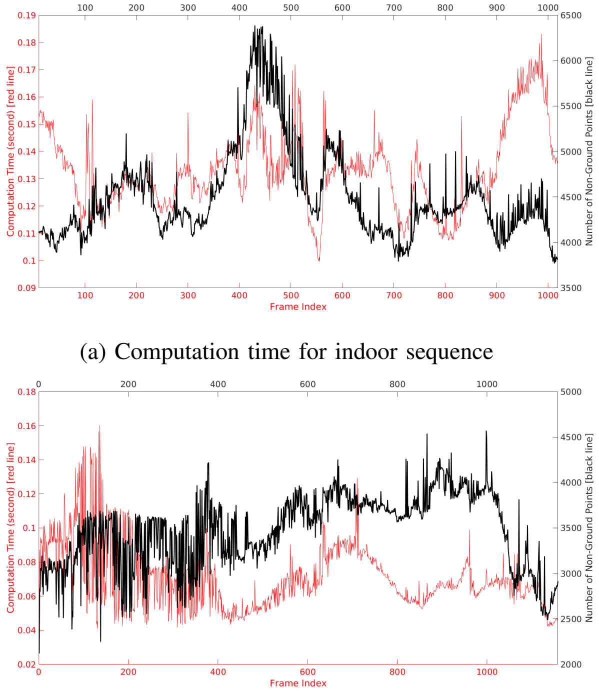
$$r_{ij}^{k} = \begin{cases} 1, & \text{if } |Rf_{sensor}^{i,k} + t - f_{map}^{j}| < \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where R and t is a rigid transformation of a new feature in sensor coodinate frame i^{th}−feature (f_{sensor}^{i}) to global feature map's j^{th}−feature (f_{map}^{j}). Suppose, the transformation (R, t) from a new sensor data to the global map is known from 3D SLAM while the robot is moving in real-time.

The experiment is tested in two scenarios: outdoor and indoor. The operation size of indoor sequence is 60 × 30 meters, and the operation size of outdoor sequence is 100 × 40 meters. Many performance metrics are measured as shown in numerical results Table. I. In order to compare with other conventional methods, we also exploited the groud plane removal and 3D clustering in the same fashion to the procedure explained in this paper. The candidate points selection is performed by using the proposed MSV and conventional feature detection methods provided in Point Cloud Libaray (PCL). We choose ISS [12], NARF [18], and SIFT3D feature extraction modules due to their viable good performances in our emphirical tests.

In term of repeatability, the superiority of the proposed method (MSV) is hardly observable by the numerical results. However the qualitative result in Fig. 8 shows a remarkable performance of MSV over other methods by capturing most of salient objects neatly. We also found that the repeatability of SIFT3D is also descent, but the effective range is very limited. For NARF feature, it is able to capture the salient objects but its repeatability is quesionable. Eventhough the repetability and effective range of ISS yields higher numbers in the numerical result, we found that the method is susceptible to noise and distortion from sparse Lidar sensor. The plain areas, such as walls, are often mistakenly recognized as feature points. Furthermore, ISS frequently captures feature points on the leftover-ground-areas, which are remained from incomplete ground plane removal.

The computation time of the MSV method is shown in Fig. 9a and 9b. As a result, our proposed method ran at 5-15 Hz depending on the number of input non-ground points. The average number of non-ground points in the indoor sequence is higher than the outdoor sequence since the VLP-16 sensor perceived denser points in the narrow limited space. Note that, the Velodyne VLP-16 sensor is run at 10Hz. The ground plane extraction also run at 10 Hz. The machine specification used in the experiment is Intel Core i7-4710MQ 2.50GHz with 16Gb RAM. The methods in this paper are all implemented in C++ with PCL and ROS library under Linux environment.



(a) Computation time for indoor sequence

(b) Computation time for outdoor sequence

Fig. 9: Computation time (red line) for outdoor and indoor sequences with corresponding number of non-ground points (black line)

In addition to the above tests, the result 3D SLAM by using our proposed feature extraction based on MSV can be shown in Fig. 10. The octomap is ploted for each sensor captures continuously. The good map results could implicitly demonstrate the quality of our feature detection method, which is primarily contributing to 3D SLAM performance.

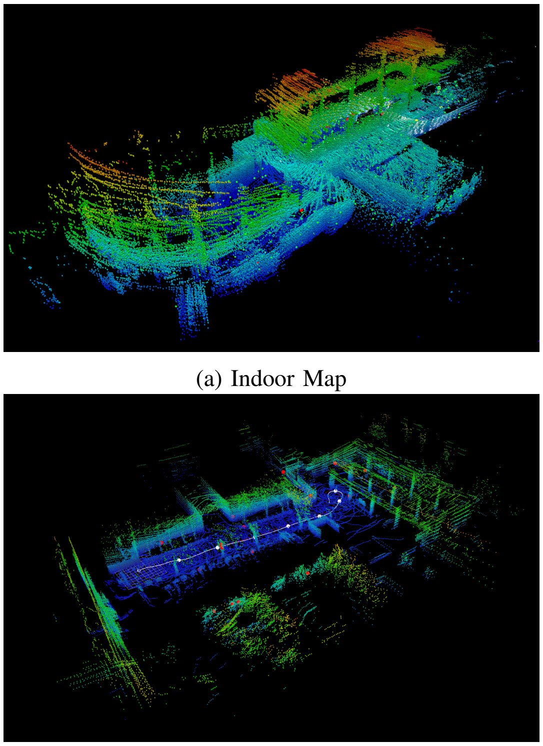
V. Conclusion

The primary challenge of this research is how to deal with the sparse and non-uniform pointcloud which is unique characteristics of rotating Lidar (Velodyne VLP-16). Due to this unusual pointclouds, many conventional 3D feature detection methods failed to estimate proper feature points. According to the experiments, our proposed method can operate with broader range of sensors, and applicable to both outdoor and indoor environments. These advantages allow us to improve generalization of the system.

However, the current system need further investigations of its capabilities and limitations. The primary concerns are the computation time requirement and the memory usage of the final integrated system, which have to meet the real-time needs in low performance and low powered CPU units. The research is expected to improve the performance of the current feature extraction component in broader diversity of environments efficiently. The component is also expected to work across the heterogeneous platforms. The large scale area and long-time operation endurance of system should be further investigated.

TABLE I: Experimental Results

Sequences	Candidate points selection methods	Number of non-ground points per frame	Effective range (m)			Number of candidates per frame	Number of final features per frame	Number of repeatable features per frame	10-frames relative repeatability per point (%)	10-frames weighted repeatability per point (%)	Comp. time (ms)
			Min	Max	Average						
Outdoor (100 × 40)	MSV (proposed)	3472.47	1.9	35.7	14.9	399.9	15.2	14.8 ± 5.1	71.6 ± 29.6	50.8 ± 26.9	70.5
	ISS		1.6	37.5	17.1	159.7	67	65.4±15.1	71.7 ± 30.1	45.9 ± 25.8	16.5
	SIFT3D		1.9	12.9	7.1	15.9	4.9	4.9 ± 2.3	58.1 ± 26.1	30.7 ± 17.2	41.9
	NARF		2.1	49.9	18.2	13.4	10.2	9.1 ± 3.6	39.4 ± 25.7	22.1 ± 17.6	19.7
Indoor (60 × 30)	MSV (proposed)	4504.38	1.1	34.5	9.6	928.9	21.2	20.6 ± 5.4	73.1 ± 29.8	51.1 ± 26.5	132.7
	ISS		1.1	36.7	10.3	141.9	75.4	76.3±20.1	78.6 ± 24.7	42.9 ± 20.4	30.3
	SIFT3D		1.1	13.5	6.2	54.3	16.4	16.5 ± 3.5	65.4 ± 27.2	34.6 ± 18.2	58.6
	NARF		1.2	46.2	8.7	19.2	12.4	12.3 ± 5.3	50.2 ± 30.4	27.1 ± 19.2	26.6



(a) Indoor Map

(b) Outdoor Map

Fig. 10: Octomap plot of 3D SLAM output by using our proposed feature in (a) indoor and (b) outdoor environments

Acknowledgment

The research was partially supported by the ST Engineering NTU Corporate Lab through the NRF corporate lab@university scheme

References

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[2] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, pp. 147–151, 1988.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] I. Sipiran and B. Bustos, "Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes," *The Visual Computer*, vol. 27, no. 11, p. 963, 2011.

[5] A. Flint, A. Dick, and A. v. d. Hengel, "Thrift: Local 3d structure recognition," in *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, pp. 182–188, Dec 2007.

[6] A. Godil and A. I. Wagan, "Salient local 3d features for 3d shape retrieval," 2011.

[7] R. Unnikrishnan and M. Hebert, "Multi-scale interest regions from unorganized point clouds," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8, June 2008.

[8] J. Sun, M. Ovsjanikov, and L. Guibas, "A concise and provably informative multi-scale signature based on heat diffusion," in *Proceedings of the Symposium on Geometry Processing*, SGP '09, (Aire-la-Ville, Switzerland, Switzerland), pp. 1383–1392, Eurographics Association, 2009.

[9] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Computer Graphics Forum*, vol. 22, no. 3, pp. 281–289, 2003.

[10] H. T. Ho and D. Gibbins, "Curvature-based approach for multi-scale feature extraction from 3d meshes and unstructured point clouds," *IET Computer Vision*, vol. 3, pp. 201–212, December 2009.

[11] B. Matei, Y. Shan, H. S. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert, "Rapid object indexing using locality sensitive hashing and joint 3d-signature space estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 1111–1126, July 2006.

[12] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 689–696, Sept 2009.

[13] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, vol. 89, no. 2, pp. 348–361, 2010.

[14] S. Salti, F. Tombari, and L. D. Stefano, "A performance evaluation of 3d keypoint detectors," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pp. 236–243, IEEE, 2011.

[15] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D LIDAR point clouds," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, pp. 2798–2805, IEEE, 9 2011.

[16] J. Mullane, B. N. Vo, M. D. Adams, and B. T. Vo, "A Random-Finite-Set Approach to Bayesian SLAM," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 268–282, 2011.

[17] D. Moratuwage, D. Wang, A. Rao, N. Senarathne, and H. Wang, "RFS Collaborative Multivehicle SLAM: SLAM in Dynamic High-Clutter Environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 2, pp. 53–59, 2014.

[18] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard, "Narf: 3d range image features for object recognition," in *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, (Taipei, Taiwan), October 8, 2010 2010.