Efficient Segmentation and Plane Modeling of point-cloud for structured environment by Normal Clustering and Tensor Voting

Ming Liu,
The Hong Kong University of Science and Technology
eelium@ust.hk

Abstract—**In this paper, we introduce an efficient point-cloud segmentation algorithm, inspired by efficient segmentation (also named as super-pixel extraction). It uses parameterised "normal words" as distance measures, which are obtained by clustering of surface normals. We estimate the surface normals by the sparse tensor voting framework, which enables adaptive structural extraction, even for the case of missing points. The output result is consist of labeled point representations regarding plane assumptions, which is validated by metrics based on information theory. We show the quality of the segmentation results by experiments on real datasets, and demonstrate its potentials in aiding 2.5D topological navigation for structured environments.**

I. Introduction

POINT-CLOUD is an important sparse representation of the work environment for mobile robots. In order to efficiently use point-clouds, segmentation is the basic operation required. The output of segmentation is usually used for various robotics applications such as topological mapping, semantical reasoning and scene reconstruction. However, it is in general considered as a difficult problem, especially for the applications using raw data points. We consider the unreliable observation primarily hinder the segmentation quality, which usually consist of the following cases, as shown in figure 1.

- *Outliers*: The outliers are commonly generated by reflection surface or sharp edges. The second case is sometimes called "Shadow-point". They are the main source of false recognition for point-cloud based application.
- *Missing Points*: The missing points are due to different view perspectives, and usually the main reason for wrong structural analysis. In this paper, we use tensor voting to comprehend these missing information.
- *Non-uniform Density*: It can be affected by different laser setups. For example, a nodding laser setup and a rotation laser may lead to different distributions, even in the same environment. In this paper, we use sampling respecting with point density to alleviate the drawbacks.

Despite of these difficulties, we choose to use the point-cloud as the representation, because of the following advantages. First, it is a raw sparse representation. The algorithms designed on point-cloud do not require preprocessing of the data, which are usually computationally expensive, such as meshing or triangulation. Second, the sparse representation implies that the conducted algorithms can be more efficient, with less memory and computational cost. Besides, the drawbacks discussed previously can be solved, at least in part, by structural reasoning algorithms such as tensor voting. Though tensor voting is an algorithm with $O(n^2)$ complexity, thanks to our recent contributions [1], it can be executed in near real-time for typical data size [2].

This work was supported by the HKUST Project IGN13EG03; partially sponsored by the Research Grant Council of Hong Kong SAR Government, China, under project No. 16206014 and NSFC Fund 6140021318.

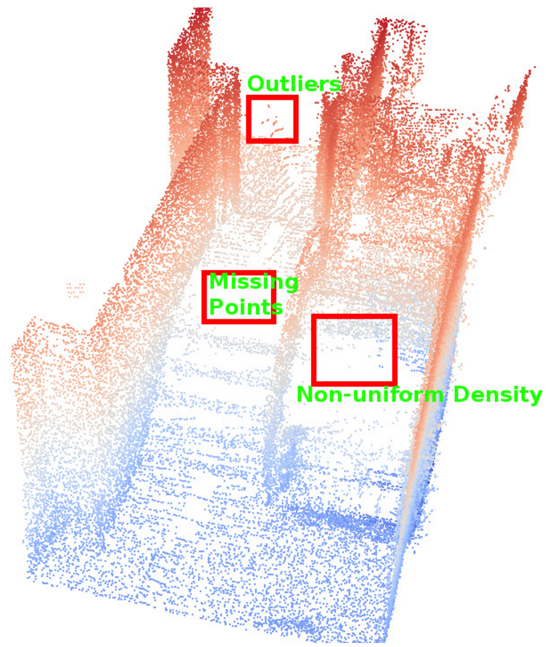


Fig. 1. Clip of a typical point-cloud and common unreliable observations.

A. Contributions

We address the following contributions in this paper:

- Following our recent results proposed in [1], stick components from sparse tensor voting are taken as the estimation of surface normals.
- We propose an efficient segmentation approach using clustering result of surface normals.
- Information theory based assessment of point-cloud segmentation, which helps to evaluate the result in quantitative way.
- Evaluation of topological structure extraction for real point-cloud dataset. A preliminary study of its application to 3D all-terrain topological mapping is carried out.

B. Arrangement

The rest of this paper is arranged as follows. We start with introducing related works from different perspectives. The segmentation algorithm based on super-pixel extraction by bag-of-normal-words is presented in section III, followed by the information theory based evaluation and tests on real dataset in section IV. At the end, we draw conclusions and introduce our vision for future work.

II. Related Work

A. Segmentation of Point-cloud

Several works regarding point-cloud segmentation have been proposed. These works are based on different features, such as edge, projected as from 2D images [3]. Then computer vision techniques are applied to process the data in 2D. The main drawback of these works is that they rely on almost-clean dense representations of the target models for efficient retrieval [4], which is not for usual robotics applications. Moreover, the back-forth projection between two representations is time consuming.

B. Normal based Segmentation

The surface normal is a local consistent feature. Therefore it is widely used for point-cloud analysis. Regarding segmentation, one early work by Pulli et al [3] aims at segmenting range images into homogeneous regions, by decomposing x- and y-components of the normal vectors. It assumes perfect dense point-clouds without noise and the resulting algorithm only deal with segmentation in 2.5D. Normal estimation based on local constrained least square modeling [5]. Clustering by an initial segmentation in normal space, then refined in distance space [6] was proposed by Holz et al. Teutsch et al presented a clustering algorithm for subset segmentation [7], which aimed at the segmentation of point-clouds without plane-assumption. [8] introduced an incremental way to model different clusters by using both angular and distance constraints. We propose to firstly perform clustering algorithm on normal directions [9], which can be seen as prerequisites for further segmentation of point-clouds. It alleviates further computation of normals.

C. Information Theory

For most clustering algorithms, such as spectral clustering [10] or K-means, the target number of clusters of input data is a key issue, which has been extensively studied in [11]. Sometimes it is named as the cardinality of segmentation. Though several adaptations are proposed for specific applications such as self-tuning spectral clustering [12], this problem persists in general. In order to eliminate the dependency on cardinality, usually information theory can be adopted. In our previous work [13], we proposed a segmentation algorithm using Chow-Liu decomposition of the mutual information tree. When local constraints can be defined, the cardinality can be taken as parameter of global entropy maximization. Mutual information has been proved to be an optimal clustering criterion [14] for such setup. In this work, we introduce a modified Akaike Information Criterion (AIC) [15], regarding the effect of different cardinality, using efficient segmentation [16] to maximize the global entropy to obtain optimal clustering results.

D. Tensor voting

Tensor voting [17] is originated in computer vision. It has been extended to several applications related to segmentation [18], [19]. Several works have been proposed on structure extraction of point-clouds using Tensor voting as well [20], [21]. We consider it is one of the most important algorithms for structural analysis, because of its extraordinary performance in tolerance to noise and missing data, its consistency for local information and intuitive extraction of evidence saliency, etc. Nevertheless, the computational cost of Tensor voting is high. Thanks to the CUDA based tensor voting open-source library, which we recently proposed [1], the algorithm is able to launch in near real time. As result, both surface normals and geometrical saliency can be extracted efficiently.

E. Topological mapping

Autonomous navigation in 2.5D terrain for mobile robots is a very hard problem in general [22], [23], especially when a raw point-cloud is used. However, it is feasible and lower requirements on computation when a topological map is conducted [24], [25]. Segmentation, as the main technique for topological mapping, is the basis for these point-cloud based applications. Several work deal with the extraction of topological regions from metric maps. Intuitively, a topological map is defined as a graph structure, which is composed of *nodes* and linkages among them. We follow the definitions of *nodes* as interesting regions such as that in [26], [24], [13], [27], namely region-based topological maps.

III. Segmentation Algorithm

The proposed algorithm pipeline is shown in figure 2. The

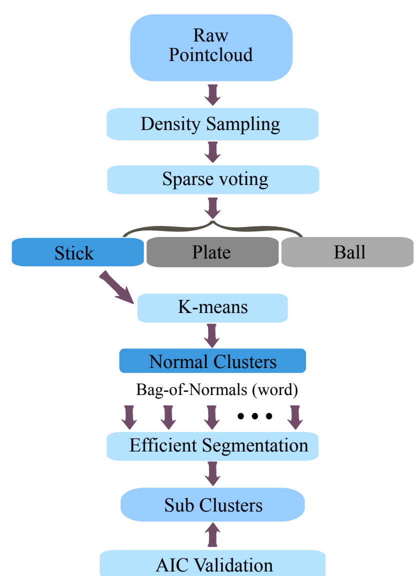


Fig. 2. The pipeline of the proposed algorithm

density based sampling helps to maintain the stableness of tensor voting, for which readers are referred to [21] for more details. Using *sparse voting*, the local tensor field for the neighborhood of each point can be decomposed into “stick” “plate” and “ball” saliency, with corresponding eigenvectors as shown in (2). We use the stick components as approximation of local normal directions. Inspired by the classical bag-of-word [28] model, we first build a so-called “normal word” by k-means, then use these words as inputs for efficient segmentation [16] algorithm. The clustering result is validated by AIC information criteria shown in (9).

A. Surface Normal estimation by tensor voting

We propose to use tensor voting as the primary algorithm to estimate [21] surface normal. It allows us to adapt to the terrain structure in a more flexible way, by varying the kernel size σ of the sparse voting field, such that:

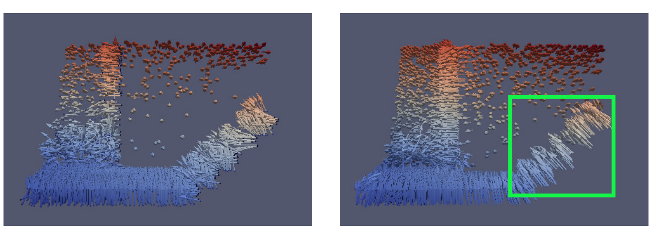
$$Decay(d, \sigma) = e^{-\frac{d^2}{\sigma^2}} \tag{1}$$

where d is the Euclidean distance between a pair of voter and votee. A typical voting result T_{3x3} for an arbitrary point-cloud can be decomposed as follows.

$$\begin{aligned} T &= \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T + \lambda_3 \hat{e}_3 \hat{e}_3^T \\ &= (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + && \text{(stick component)} \\ &(\lambda_2 - \lambda_3)(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) + && \text{(plate component)} \\ &\lambda_3 (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \hat{e}_3 \hat{e}_3^T) && \text{(ball component)} \end{aligned} \tag{2}$$

where λ_i's are eigenvalues sorted in decreasing length sequence, $\hat{e}_i$'s are the corresponding eigenvectors. We use the stick component $\hat{e}_1$ from (2) as a normal estimation for local planes. The readers are referred to [17] for more details regarding tensor voting.

Upon different applications for the robot, the voting kernel size can be chosen differently. In figure 3, we show different results by changing voting kernel size σ for the side view of the point-cloud in figure 1. The green rectangle in figure 3(b) highlights the area of a section of stairs. We can see that for a bigger σ, the structural information can be easily smoothed and refined. In practice, the kernel size is chosen as the navigation footprint of the robot. However, the computational cost of tensor voting is high ($O(N^2)$). In order to alleviate that, we use an existing GPU based tensor voting computation framework for surface normal calculation in [1], by which we can get surface normal per point in real-time.



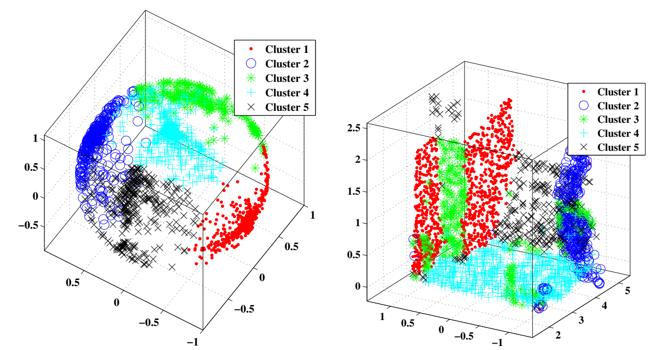
(a) Surface normal estimation by tensor voting $\sigma = 0.1m$ (b) Surface normal estimation by tensor voting $\sigma = 0.5m$

Fig. 3. Different results by varying kernel size σ

B. Clustering of surface normals

We have the assumption that the structured environment is composed of major planes. Regarding the structural smoothing introduced by tensor voting, the definition of planes is flexible. It means that by adapting the size of the voting kernel, continuous stairs can be considered as planar as well. This assumption is very helpful to extract topological models of given structured environment, with potential applications such as topological navigation for all-terrain robots which will be discussed in section V.

Taking the surface normals resulted by tensor voting, the clustering result of surface normals, using partially the dataset shown in figure 1, is illustrated in figure 4(a). Although it is a small dataset for this specific test, the major features and advantages of the proposed algorithm are to be revealed and compared in section IV. By back-projecting the clustering results, we obtain a clustering result of the raw point-cloud shown in figure 4(b). Figure 4(a) and 4(b) show the same clustering result in normal direction space and Euclidean space respectively. This intermediate result leads to clean separations of normal directions (including the smoothed stair structure as Cluster 5). The next problem is how to further divide the point-cloud into separate planes. Because the cardinality of the clustering is unknown, a rational segmentation algorithm independent of this cardinality information is required.



(a) Clustering results for the normal directions on a unit sphere. Color denotes different normal words (b) Normal based clustering results for points in Euclidean space. Color denotes different normal words

C. Efficient segmentation inspired algorithm

As discussed previously, the cardinality of clustering is a general issue. Several works investigated and reasoned for a proper cardinality from the data perspective [11] and Bayesian tests such as Bayesian Information Criterion (BIC)[29]. However a bad parameter selection will corrupt the result easily as described in [30]. The efficient segmentation proposed in [16] avoided this problem from data perspective, which considered each data point as a vertex in graph, and starting with one cluster per data point. This algorithm is highly efficient when the distance between data points and clusters is simple or homogeneous, such as color for pixels in the original work. As far as point-cloud is concerned, the output of surface normal estimation by tensor voting greatly smoothed out the noise, resulting a uniform distributed feature space. Therefore, it is possible to follow the idea of efficient segmentation for point-cloud clustering. Compared with the most related work [31], we use more robust criteria to assess similarities between points; the proposed normal based features allow the clustering in area with non-uniform distribution, which is not applicable for other similar approaches. The efficient segmentation is summarised in algorithm 1, where $MInt$ is the minimum internal difference function [16]. kNN is the number of nearest neighbours considered in the normal estimation. Please refer to [1] for more details about the effect of kNN to the estimation precision.

Algorithm 1 Generic Efficient Segmentation

Data: Pointcloud $P = \{p_1, \ldots, p_i, \ldots, p_n\}$
Surface normals $N = \{n_1, \ldots, n_i, \ldots, n_n\}$
Result: Cluster identity for each point
function GENERATEGRAPH(P)
  Graph $G = (V, E)$,
  node set $V = \phi$, edge set $E = \phi$
  for all $p_i \in P$ **do**
    $V \leftarrow$ node p_i
    $kNN_i \leftarrow$ kNearestNeighbours(p_i, P)
    for all $b \in kNN_i$ **do**
      $E \leftarrow$ edge $e = Distance(b, p_i)$
  return G
$G(V, E)$ = GENERATEGRAPH(P)
sortedEdges = sort(E)
build 1-element component c_i for $\forall p_i$
repeat
  for all $e_j \in$ sortedEdges **do**
    $(u_j, v_j) \leftarrow e_i$
    $s \leftarrow c_{u_j}$
    $t \leftarrow c_{v_j}$
    if $s! = t\ \ \&\ \ e_i \leq MInt(s, t)$ **then**
      FUSECLUSTER(s, t)
      UPDATECLUSTERLABELS(e_j)
until Converge
Filter Results by fusing small clusters into bigger one
return Clusters $c_1, \ldots, c_K$

The definition of the distance function is the primary design issue, since efficient segmentation is scale sensitive. It means even with the same distance measuring function, scaling on measurements may lead to different results. Comparing with different distance functions, such as radius difference of normal directions, degree difference, exponential of the differences, we find "degree difference" is an optimal criterion, by which the distance function is defined as:

$$\Delta(\overrightarrow{u}, \overrightarrow{v}) = mod\left(rad2deg(\arccos\frac{\overrightarrow{u} \cdot \overrightarrow{v}}{|\overrightarrow{u}||\overrightarrow{v}|}), 180\right) \tag{3}$$

where $\overrightarrow{u}$, $\overrightarrow{v}$ are normal directions for two arbitrary points in the point-cloud. By manually tuning parameters, the optimal result after filtering is shown as figure 4(a).

We could see that several extra batches within same region are extracted. The major difficulty is that the distance function is hard to select, adapting with regional efficiency $MInt(s, t)$ described in algorithm 1. We consider that the ideal case is that the distance function is symbolically defined, by which the neighbours can be equally treated. In order to achieve that, we use the normal clustering results by k-means in figure 4(b). The modified distance function can be defined as:

$$Distance(\overrightarrow{u}, \overrightarrow{v}) = \begin{cases} \Delta(\overrightarrow{u}, \overrightarrow{v}) \in (0, +180), \overrightarrow{u}, \overrightarrow{v} \text{ with same normal word} \\ 0, \text{otherwise} \end{cases} \tag{4}$$

The result is shown in figure 4(b). We could see that the number of final clusters K is greatly reduced. Quantitatively, in the next section, we use information theory to assess the quality of the result, and determine the optimal K, based on non-parametrical density estimation.

IV. Information based Validation

Clustering is a NP-complete problem, which is able to be evaluated by a quality score. We consider the segmentation problem as distributional clustering. It means that each point is described as a distribution. The relative entropy then becomes a natural measure of the distance between distributions. Given such a choice, mutual information has been proved to be an optimal clustering criterion [14]. The clustering of point-cloud $P = \{p_1, p_2, \ldots, p_N\}$ into K clusters is a function:

$$C : P \rightarrow \{1, \ldots, K\}$$

K is sometimes named as the cardinality of the clustering results as well.

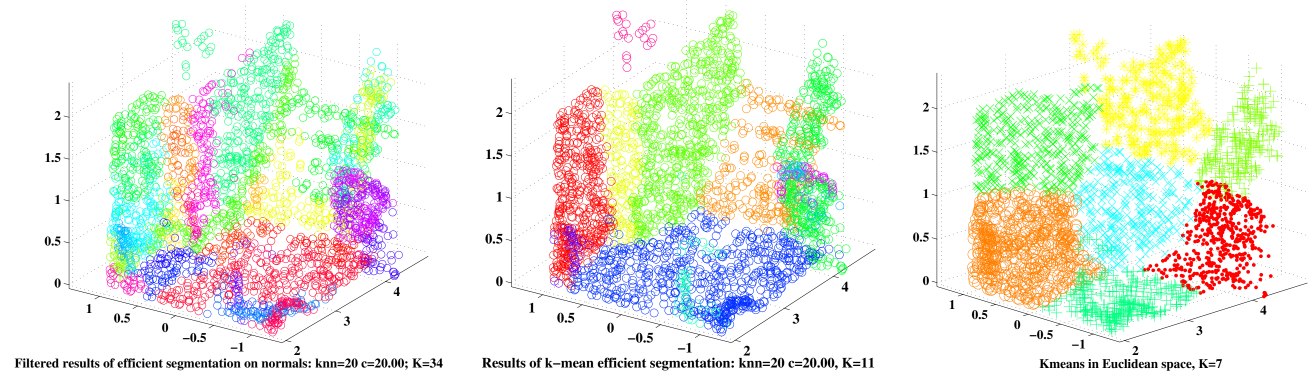
By introducing clustering function C, the data points are better organized. We want to maximize this optimization. In other words, we want to maximize the mutual information between raw point-cloud and the clustering, namely $I(P, C)$. Based on information theory, the mutual information can be further representation by $I(P, C) = H(P) - H(P \mid C)$. In general, the goal of clustering is to maximize the divergence between clusters and minimize the "dynamics" within the same cluster. In this case, $H(P)$ is determined with the raw data, therefore independent on any specific clustering. The target problem is thus converted as an optimization problem, respecting with the minimization of a score as:

$$S_{MI}(C) = H(P \mid C) = \sum_{k=1}^{K} \frac{n_k}{N} H(P \mid C = k) \tag{5}$$

where n_k is the number of points assigned to cluster k, N is the total number of points. Score S_{MI} describes the average entropy of the data points from clustering function C. For an optimal clustering C, the score will be minimized, resulting in maximized mutual information $I(P, C)$. The intra-cluster entropy of points p^k's assigned with cluster k:

$$H(P \mid C = k) = -\sum_{p^k}^{n_k} p(p^k \mid C = k) \log p(p^k \mid C = k) \tag{6}$$

Result by local metric (3)	Proposed normal clustering (4)	Result of Euclidean k-means



(a) $S_{AIC} = 0.769$ (b) $S_{AIC} = 0.068$ (c) $S_{AIC} = 0.695$

Fig. 4. Clustering results using efficient segmentation (a)(b) and k-means (c) on the same structured dataset (after density based sample: 2588 points). A lower S_{AIC} indicates a better result.

We must notice that the distribution of the clustered points $p(p^k \mid C = k)$ is non-Gaussian, nor uniform etc. It depends on the sparse representation of point-cloud in the 3D space. Therefore, parametrical modeling based on Expectation Maximization (EM) etc. are not applicable in this case. We need to use a non-parametrical approximation of this entropy.

A. Non-parametrical approximation of intra-cluster Entropy

Faivishevsky et al proposes a non-parametrical information clustering (NIC) estimator using MeanNN differential entropy [32], which leads to an inter-cluster entropy estimation that can be summarized by:

$$H(X \mid C = j) \approx \frac{d}{n_j(n_j - 1)} \sum_{i \neq l \mid c_i = c_l = j} \log ||x_i - x_l|| \quad (7)$$

where $|| \cdot ||$ is the Euclidean distance between two datapoints x_i and x_l (in $\mathbf{R}^d$). Please notice that it does not require the intra-cluster distribution as spherical Gaussian. We can rewrite equation 5 in the NIC form as [1]:

$$S_{NIC}(C) = \frac{1}{N} \sum_b \frac{3}{n_k - 1} \sum_{i \neq l \mid c_i = c_l = j} \log ||x_i - x_l|| \quad (8)$$

One major limitation of (8) is that it can only determine the quality of a segmentation with given cardinality. Inspired by the discussion of Akaike Information Criterion (AIC) in chapter 16 of [15], the clustering cardinality K can be embedded in the NIC cost function as part of evaluation, resulting in the AIC score as follows:

$$S_{AIC}(C) = S_{NIC}(C) + \psi K \quad (9)$$

where ψ is a weighting factor of the influence of variation of number of clusters K. In this work, it is empirically chosen as 0.1. A smaller AIC value indicates a better segmentation in statistical meaning. We compare the score with three different algorithms on the same dataset as shown in figure 4. We could see that the proposed algorithm shown the best quality in figure 4(b). Intuitively, the walls and stairs are correctly clustered. On the contrary, the distance metric by (3) leads to too many small clusters, due to difference of local normals. The results of k-means shown in (c) are not interesting since the clusters are gathered blobs that only lie in Euclidean space.

Moreover, considering the state-of-the-art, RANSAC is often used for plane extraction [33], [34]. Usually RANSAC based method is optimal for single module detection. However, it may cause problem under condition of multiple planes that largely crossing-over by each other. For comparison, we adopt the implementation by Zaman [35]. By carefully tunning the parameters [2], we obtain the best results that may make sense as shown in figure 5. We have the following observations: First,

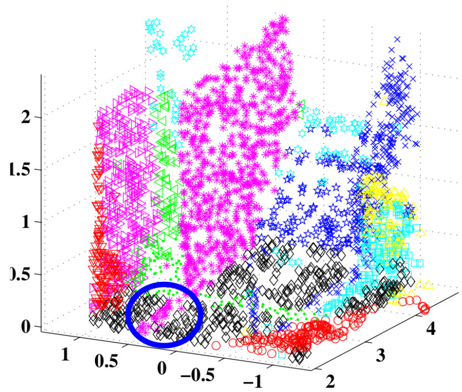


Fig. 5. Result by RANSAC based plane extraction $S_{AIC} = 0.372$. Please notice that the points highlighted by blue eclipse indicate the inherent drawbacks of RANSAC based method.

though major planes are detected after parameter tunning, the

[1] In the original paper [32], the $\frac{1}{N}$ nominator was missing.

[2] Optimized parameters: the smallest number of supporting points is 60; the threshold for point fits is 0.1 meter.

small clusters exist on the right side of figure 5. Second, as an inherent characteristic of RANSAC, whenever points that fit assumptions of a pre-detected plane, such as the magenta points highlighted by the blue eclipse, will be assigned as the same plane. This is the major drawback of RANSAC based methods comparing to the proposed algorithm. Third, even with the same set of parameters, randomness of the results can be observed, because RANSAC is based on sampling and the selection of initial points is deterministic. Last but not least, the proposed algorithm largely outperforms for the case of stairs, where the points are considered as the same "normal word" from the tensor voting results. These observations further confirm the advantages.

V. Conclusion

We described a point-cloud segmentation algorithm combining surface normal clustering and efficient segmentation algorithm. We also proposed a AIC-based criterion to assess segmentations quantitatively, by adopting information theory. We could see that the proposed algorithm provides clean and robust segmentation results, even in the case of unreliable point-cloud sensor readings. As a sample use-case, preliminary results for the topological mapping in a small scale were also introduced, indicating the potential for next research steps of navigation on point-cloud. Full topological navigation for all-terrain mobile robots will be discussed in the future.

References

[1] M. Liu, F. Pomerleau, F. Colas, and R. Siegwart, "Normal Estimation for Pointcloud using GPU based Sparse Tensor Voting," in *IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, 2012.

[2] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *The International Journal of Robotics Research*, vol. 31, no. 14, pp. 1705–1711, 2012.

[3] K. Pulli and M. Pietikäinen, "Range image segmentation based on decomposition of surface normals," in *Proceedings of the Scandinavian conference on image analysis*, vol. 2, 1993, pp. 893–893.

[4] L. Wang, M. Liu, M.-H. Meng, and R. Siegwart, "Towards real-time multi-sensor information retrieval in cloud robotic system," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*. IEEE, 2012, pp. 21–26.

[5] E. Castillo and H. Zhao, "Point cloud segmentation via constrained nonlinear least squares surface normal estimates," Technical Report CAM09-104, Computational and Applied Mathematics Department, University of California Los Angeles, Tech. Rep., 2009.

[6] D. Holz, S. Holzer, R. Rusu, and S. Behnke, "Real-time plane segmentation using rgb-d cameras⋆," in *Proc. of the 15th RoboCup International Symposium*, 2011.

[7] C. Teutsch, E. Trostmann, and D. Berndt, "A parallel point cloud clustering algorithm for subset segmentation and outlier detection," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 8085, 2011, p. 8.

[8] K. Klasing, D. Wollherr, and M. Buss, "Realtime segmentation of range data using continuous nearest neighbors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 2431–2436.

[9] M. Liu, F. Pomerleau, F. Colas, and R. Siegwart, "Information Theory based Validation for Point-cloud Segmentation by Normal Clustering and Tensor Voting," in *[submitted to] IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[10] T. Hastie, R. Tibshirani, and J. J. H. Friedman, *The elements of statistical learning*. Springer, 2009.

[11] H. Xiong, J. Wu, and J. Chen, "K-means clustering versus validation measures: a data-distribution perspective," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 2, pp. 318–331, 2009.

[12] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Advances in neural information processing systems*, vol. 17, no. 1601-1608, p. 16, 2004.

[13] M. Liu, F. Colas, and R. Siegwart, "Regional topological segmentation based on mutual information graphs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[14] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *The Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[15] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, Juli 2008. [Online]. Available: http://nlp.stanford.edu/IR-book/

[16] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[17] G. Medioni, M. Lee, and C. Tang, *A computational framework for segmentation and grouping*. Elsevier Science, 2000, vol. 1.

[18] Y. Dumortier, I. Herlin, and A. Ducrot, "4-D Tensor Voting motion segmentation for obstacle detection in autonomous guided vehicle," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2008, pp. 379–384.

[19] J. Jia and C. Tang, "Inference of segmented color and texture description by tensor voting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 771–786, 2004.

[20] H. Schuster, "Segmentation of lidar data using the tensor voting framework," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, pp. 1073–1078, 2004.

[21] B. King, "Range data analysis by free-space modeling and tensor voting," Ph.D. dissertation, Rensselaer Polytechnic Institute, 2009.

[22] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," *Robotics Research*, pp. 201–212, 2011.

[23] M. Liu and R. Siegwart, "Navigation on point-cloud - a riemannian metric approach," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 4088–4093.

[24] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[25] M. Liu, L. Oth, F. Colas, and R. Siegwart, "Incremental topological segmentation for semi-structured environments," *Autonomous Robots*, vol. 37, no. 3, 2014.

[26] J. Choi, M. Choi, and W. K. Chung, "Incremental topological modeling using sonar gridmap in home environment," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Ieee, Oct. 2009, pp. 3582–3587. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354247

[27] M. Liu, F. Colas, F. Pomerleau, and R. Siegwart, "A Markov semi-supervised clustering approach and its application in topological map extraction," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012.(IROS 2012)*, 2012.

[28] H. Wallach, "Topic modeling: beyond bag-of-words," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 977–984.

[29] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. San Francisco, 2000, pp. 727–734.

[30] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Advances in neural information processing systems 16: proceedings of the 2003 conference*, vol. 16. The MIT Press, 2004, p. 281.

[31] K. Klasing, D. Wollherr, and M. Buss, "A clustering method for efficient segmentation of 3d laser data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. Ieee, 2008, pp. 4043–4048.

[32] L. Faivishevsky and J. Goldberger, "A nonparametric information theoretic clustering algorithm," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*. Citeseer, 2010, pp. 351–358.

[33] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer Graphics Forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.

[34] R. Wahl, M. Guthe, and R. Klein, "Identifying planes in point-clouds for efficient hybrid rendering," in *The 13th Pacific Conference on Computer Graphics and Applications*, 2005, pp. 1–8.

[35] Tim Zaman, "3D Ransac for get planes from point clouds." [Online]. Available: {http://http://projects.asl.ethz.ch/datasets/doku.php}