Real-Time Trust Region Ground Plane Segmentation for Monocular Mobile Robots

Hong Liu
Shenzhen Graduate School
Peking University
Shenzhen, China
Email: hongliu@pku.edu.cn

Yongqing Jin
Shenzhen Graduate School
Peking University
Shenzhen, China
Email: yongqingjin@pku.edu.cn

Chenyang Zhao
Shenzhen Graduate School
Peking University
Shenzhen, China
Email: chenyangzhao@pku.edu.cn

***Abstract*—Ground plane segmentation is quite a challenging fundamental problem for monocular mobile robot navigation due to the dynamic unknown environments and the initialization of coordinate system which induces outliers to the bottom region of interest. Current geometric-based methods are mostly limited to deal with multiple plane segmentation in stationary known scene from depth sensor. In this paper, we propose a robust real-time trust region ground plane segmentation method to handle the unknown environments with a single camera. The proposed method utilizes Radius Outlier Removal filter to exclude the outliers of candidate points generated by the state-of-the-art method, Direct Sparse Odometry (DSO), then candidate points in the trust region are provided to fit the ground plane. The coefficients of fitted plane will be used to remove the outliers and to compensate omissive points. Therefore the ground plane segmentation is refined iteratively. Comprehensive experiments on the TUM monoVO dataset demonstrate that our method outperforms the random sample consensus (RANSAC) methods on time consumption and robustness in the unknown scenes, even when the initial coordinate system is pitched and rolled.**

I. Introduction

The critical process of visual-based navigation systems is developed to provide real-time robust obstacle avoidance function and plane segmentation which is considered as the preprocess of obstacle recognition. Plane segmentation technology has been widely applied in computer vision applications, for instance, augmented reality applications (AR) [1], cleaning robot [2], assisting device for visually impaired people [3] and driverless car [4]. Planar structures are computed by depth information [5], [6]. However, the high cost of 3D scanners and the need of proper calibration of stereo cameras render both sensor modalities unfavorable for consumer grade applications. Therefore, this motivates researches for better alternatives. There are two kinds of plane segmentation methods: classifier-based and geometric-based methods. Classifier-based methods need large 3D ground truth datasets to train the classification model for objects segmentation which is computationally expensive. For geometric-based methods, existing ground plane segmentation in navigation system also faces several difficulties. First, during robot initialization, the location of coordinate system may cause some of original plane points misidentified as obstacle points, whose coordinate value is relatively small. Second, the outliers of point clouds increase the difficulty of plane segmentation. From the perspective of the density, point clouds are classified into sparse and dense point clouds. Dense point clouds generated by depth sensors, which require a large amount of computation. And sparse point clouds reconstructed environments using a single camera, which is an alternative for its low cost and computation in spite of bringing some extra problems. Comparing with dense point clouds, the sparse point clouds lost the connectivity and object construction information to exchange for high efficiency. No matter which kind of point clouds is used for ground plane segmentation, they all need to face the challenge of not knowing the whole shape of objects in advance when environments are reconstructed through Simultaneous Localization and Mapping (SLAM) system or Visual Odometry (VO).

There are different plane segmentation methods in the literature. Work shown in [7], using a classifier that has access to full 3D models results in better performance in segmentation than using a single-view point clouds, while this method is only limited to specified training datasets. Besides, the research of full scene fused from multiple views and dynamic growth scene structures built from SLAM are more valuable than a single image [8], [9], [10], [11]. Semantic maps are built from dense 3D maps to help robot understand environments [12], [13]. Supervised 3D semantic segmentation using joint segmentation and recognition [7], [14], [15], [16] like Conditional Random Fields (CRF) [17] is employed to capture scene features and complex relationships between different labels of 3D elements. The demerit of those modern semantic methods is that users have to train large 3D ground truth datasets to get the classification model. Unfortunately, manually annotating 3D data requires more human efforts than that of 2D images. Moreover, the process of training is usually time-consuming and resource-consuming, which is unsuitable for real-time robot exploration and navigation system.

Geometric-based methods are classified into five varieties: (1) edge-based segmentation [12] method has the least consumption of time but poorest performance of accuracy against noise. (2) normal-based segmentation method [18] is more reliable while time-consuming. (3) RANSAC is designed to find clusters that well fit a plane model without considering the connectivity of points, which produces false detections in many situations along with computationally expensive. (4) Region-growing method [19] has been proposed to exploit the

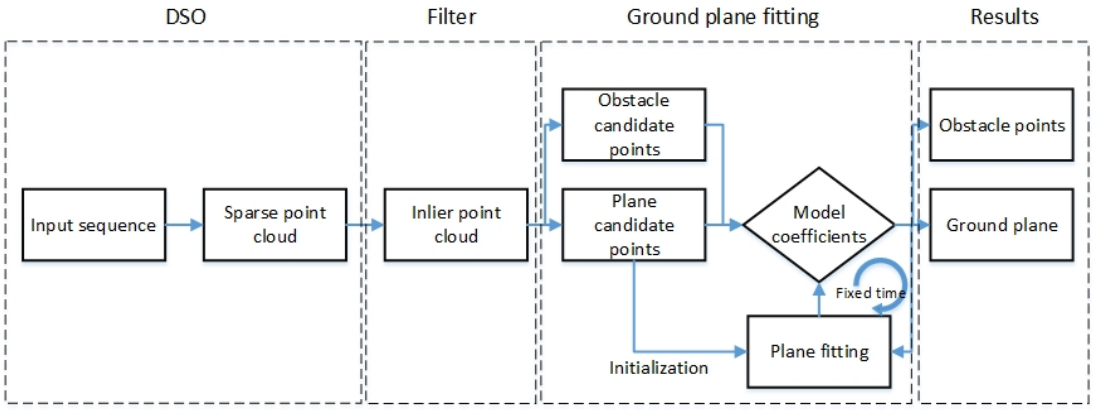

Fig. 1: Pipeline of RTTR ground plane segmentation.

neighborhood information in dense depth images, which is invalid in sparse point clouds for points without connectivity. (5) The Hough transform [20] is an alternative method to estimate model parameters from a set of measurements. The Hough transform and efficient RANSAC are combined in [21] to solve multi-resolution plane segmentation of 3D point clouds by splitting each cluster into a set of connect components, and the authors apply RANSAC to determine a fittest plane and to reject outliers robustly in each connect components in multi-resolution. However, it is extremely expensive computation to extract surface elements on multiple resolutions. RANSAC is used in [22] to fit and remove planes and other shapes such as spheres, cones from the point clouds. The drawback of this approach is that the plane is non-deterministic and sensitive to the initialization of model coefficients, meanwhile, the approach focuses on still environments. Our fitting algorithm is robust to provide the candidates generated by DSO for better planar segmentation by excluding the influence of the desk and walls point clouds. And it is compatible for dynamic growth environment. Some works [6], [23], [21] fit the plane using the depth information of indoor scenes to rebuild the construction of environment. Pham et al.[24] propose an unsupervised geometric-based approach to separate 3D point clouds into plane and other meaningful scene structures by guaranteeing geometric consistencies, for instance, walls are orthogonal to the floor. The geometric consistencies result in failing to extract ground planes individually in outdoor environments.

Above methods have their merits and demerits to be applied in plane segmentation. In order to meet the requirement of autonomous navigation and exploration task in an unknown environment, the ground plane segmentation algorithm needs to be operated in real-time without training. Therefore, a Real-Time Trust Region (RTTR) RANSAC-based plane segmentation method is proposed in this paper.

Our proposed method aims to conquer the problems brought by sparse point clouds and outperforms the random sample consensus (RANSAC) [25] method in real-time. Fig.1 is the pipeline of RTTR ground plane segmentation. The main contribution of this paper lies in proposing Real-Time Trust Region (RTTR) for ground plane fitting. Firstly, we utilize Radius Outlier Removal algorithm [26] to remove isolated noisy points in the point clouds generated by direct sparse odometry (DSO) [27]. Secondly, candidate points of trust region are utilized to fit the ground plane using RANSAC [25]. Finally, the outliers are excluded and the omitted points are appended for better segmentation in the subsequent iteration. Even though RANSAC has been used for many applications, especially for plane segmentation, to the best of our knowledge, we appear to be at the forefront to use sparse point clouds to achieve ground plane segmentation by RANSAC. Our approach is evaluated on the TUM monoVO dataset [28] which contains complex indoor and outdoor mixed scenes. The evaluated results show consistency by achieving good performance while achieving considerable speedups.

II. RTTR GROUND PLANE SEGMENTATION

We propose the Real-Time Trust Region (RTTR) ground plane segmentation for monocular mobile robot navigation system. Outdoor scene seq_25 of dataset are chosen as an example of our method in Fig.2. The sparse point clouds of RTTR depend on DSO [27] which is the state-of-the-art monocular visual odometry based on direct method. DSO realizes localization by minimizing the intensity differences between consecutive images and utilizes method proposed in [29] to refine the depth of each point. It is memory-saving to reconstruct a large scale environment with sparse spatial points but sensitive to illumination change and trouble in dealing with outliers. For instance, Fig.2(a) related to Fig.2(b) have more noise in points clouds. Therefore we filter out the outliers of the sparse point clouds in next section.

A. Radius outlier removal

The point clouds generated by DSO still have many noisy points as shown in Fig.2.(a) and Fig.2.(b). In our proposed method, depth of point cloud is calculated by triangulation by visual odometry while the scale of ground truth cannot be

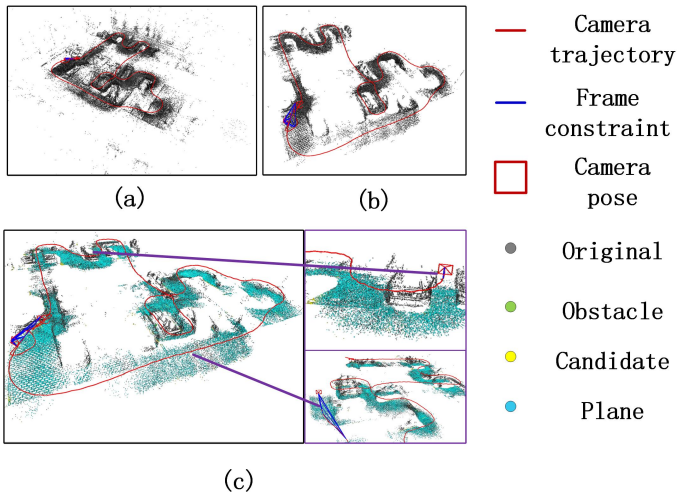

Fig. 2: Examples of RTTR ground plane segmentation method. The points colored in gray, green, yellow and blue represent the original point clouds, obstacle points, candidate plane points and fit plane points, respectively. Red line is the trajectory of camera and the red square is the pose of current camera, and the blue line is the constraint of current frame. For seq_25, (a), (b) and (c) represent the original point cloud before filtering, the point cloud after filtering and the overall effect of our method with two zoomed in parts, respectively. **(Best viewed in color)**

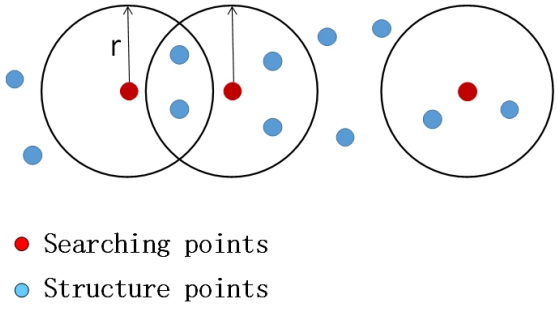

Fig. 3: Schematic diagram of Radius Outlier Removal. The red points are the searching points and the blue points represent the neighborhood points, while the black circle is the searching radius with radius of r. **(Best viewed in color)**

obtained. Most of those noisy points are caused by error depth estimation, normally, and those points are isolated from the construction point cloud. The region where the point cloud is more dense represents richer information. Our method utilizes Radius Outlier Removal to remove noisy points far away from the point clusters. Fig.3 illustrates that specified number of neighbors in the region of the circle constructed by the center with a specified radius remain in the point cloud, otherwise, the center point is excluded from the point clouds. The parameters of radius and threshold of neighbors are given in section four.

B. Trust region

Once the outliers have been removed from sparse point clouds, the ground plane is identified from the remaining point clouds. As several plane fitting methods have been mentioned before, we aim to use RANSAC to fit the ground plane for its robustness without computationally expensive training.

However, RANSAC is non-deterministic and sensitive to the structure of environment, which results in failing to extract the ground plane in the long narrow corridor for considering the wall as ground plane shown in seq_11 of Fig.7. So the quality points should be provided for plane segmentation. Trust region of ground plane points is proposed to solve this problem.

1) Candidate points: By intuition we know that points at the bottom of coordinate system are more likely to be a ground plane in the case of known robot pose. However, the establishment of the coordinate is related to the initial direction of movement of the robot. For instance, once the robots pitch or roll, the points in the lowest region may not still be the real ground plane points. Fig.4 indicates that robot pitching leads to the change of the distribution of points in the coordinate system. When robot initial translation is along to X-axis with pitching, the lowest region points are close to ground plane like Fig.4(a). We assume $\mathbf{p}$ is a vector contains all points generated in current pose. z_i is the height value of a point $\mathbf{p}_i$ and is stored in a vector $\mathbf{H}$. The vector $\mathbf{H}$ is used to obtain threshold of H_{thre} based on initial lowest region as in Eq.(1):

$$H_{thre} = \mathbf{H}_{min} + (\mathbf{H}_{max} - \mathbf{H}_{min})/t, \tag{1}$$

where t is a parameter used to divide the vector $\mathbf{H}$ into equal parts. If the z_i meets the requirement of Eq.(2), points p_i will be considered as the initial lowest region points:

$$0 < z_i < H_{thre}. \tag{2}$$

2) Sliding windows: Normally, the initialization of robot is accompanied with a slightly shaking, which leads to the situation like Fig.4(b). Mobile robots can not see the ground due to the camera is being blocked or perspective change. Both above reasons cause that some ground plane points are no longer in the scope. Even worse, some structure points or obstacles are appended. The mixed of true and false ground points gradually increase as the robot moves. Hence, we follow the approach by Leutenegger [30] with maintain points instead of keyframes by sliding window. Fig. 5 illustrates how the sliding window works in our paper. In Fig.5(a), specified size of sliding window makes pose 9 perceive the points which are insight of pose 7 but out of sight current pose. Points at the bottom of the bounded sliding window are trust region of interest which are considered as candidate plane points to fit a robust ground plane. At the same time the sliding window determines the degree of our method depending on the previous point clouds. The bigger size of sliding window is, the more we depend on the previous points. As Fig.5 shown, the size of sliding window is set larger when the robots are

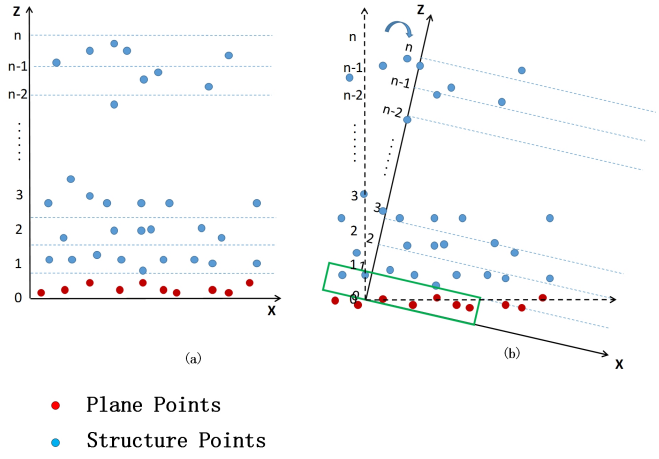

Fig. 4: Schematic diagram of the situation that initial coordinate system is pitched or rolled, red points means the true ground plane points while the blue points means the other points of sparse point cloud.(b) is the situation of coordination rotation of (a). **(Best viewed in color)**

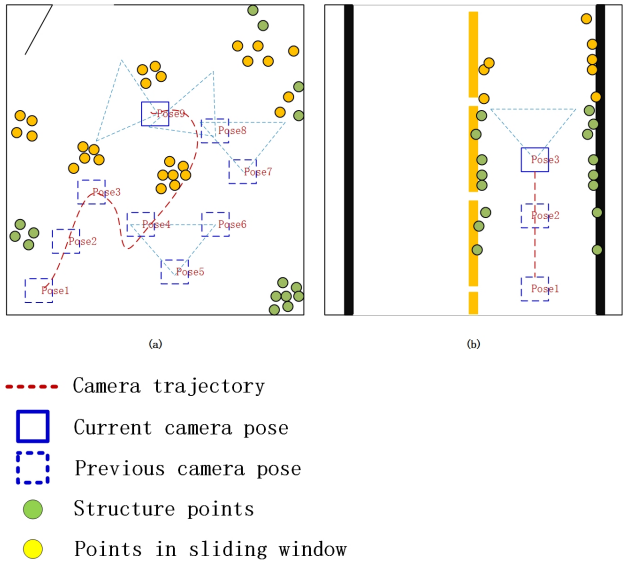

Fig. 5: Schematic diagram of the sliding window. (a) is top view of the indoor scene and the (b) is top view of the outdoor scene. The blue solid line (pose 9) rectangle is the current pose and blue dash line rectangles are previous pose of camera. The dash line triangle is the visual filed of the camera. The structure point cloud in the sliding windows present yellow color circles. **(Best viewed in color)**

in the narrow room filled with objects which will block the sight of camera. In contrast, smaller size of sliding windows will be set in a scene without so many obstacles.

C. Plane segmentation

Initial candidate points of the trust region are extracted from the sliding windows, even though a small amount of outliers in the candidate points. It is enough for RANSAC to calculate the first coefficient of plane model. Meanwhile, it is more efficient for limited size of candidate points in sliding windows related to all points. Assuming (x_{i1}, x_{i2}, z_i) is the 3D-dimensional coordinate of a point $\mathbf{p}_i$, vector $\mathbf{x}_i$ contains $[x_{i1}, x_{i2}, 1]^T$, and the plane model coefficient $\mathbf{w}$ is $[w_1, w_2, w_3]^T$. Thus the ground plane fitting problem is computed by Eq.(3):

$$\min_{\mathbf{w}} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_i - z_i)^2, \tag{3}$$

where N is the number of all candidate points. Taking all the initial candidate points in sliding window into account, $\mathbf{X}$ is a vector that contains all X-axis and Y-axis values of all initial candidate points, and $\mathbf{Z}$ is the vector made up of Z-axis values of all initial candidate points, so the first model coefficient can be obtained as follows:

$$w = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Z}. \tag{4}$$

Once the first plane model coefficient is estimated from the Eq.(4), the model will be used to exclude the outliers of the candidate points and obstacle points. In this paper, the value of ε is the allowable deviation of fitted ground plane. The initial value of ε is equal to H_{thre} in Eq.(1). Excluding outliers can be described as:

$$z_i < \mathbf{w}^T \mathbf{x}_i + \varepsilon. \tag{5}$$

Then the subsequent candidate points within the sliding window are calculated by first plane model coefficient. At the same time, the processing of plane fitting carries on the quality points in the previous fitted plane. Then allowable deviation ε will update as follow equation:

$$\varepsilon = \alpha\varepsilon + (1-\alpha) H_{Goodthre} \quad if \quad H_{Goodthre} < H_{thre}, \tag{6}$$

where α is the stability coefficient which determines the ratio of original ε. $H_{Goodthre}$ is range of quality points in the previous fitted plane. This equation is available when $H_{Goodthre}$ is smaller than H_{thre} in Eq.(1). $H_{Goodthre}$ becomes smaller after successful plane fitting which results in smaller value of ε and a more stringent plane fit condition. In this paper, we set the minimum limit of allowable deviation ε as a half of the H_{thre} for convenience.

D. Obstacle detection

Related work in obstacle detection system for monocular vision real-time navigation is few for sparse density of point clouds and huge computation. This paper continues to use the sliding windows to achieve a local obstacle detection system, meanwhile, octree [31] is used to improve the searching speed of obstacle detecting. The octree is a hierarchical data structure for spatial subdivision in 3D that used to reduce the memory usage of point cloud. Taking the camera position as the searching center, our method measure the distance from the searching center to points in the size of searching

TABLE I: The time usage of our method with different size of sliding windows(density $= 19$) .

scene	seq	the time usage with different size of sliding windows (ms)									
		1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Indoor	seq_01	0.025	0.099	0.135	0.256	0.324	0.394	0.435	0.509	0.579	0.645
	seq_11	0.029	0.104	0.148	0.193	0.225	0.273	0.301	0.362	0.418	0.432
	seq_28	0.042	0.077	0.105	0.165	0.208	0.242	0.273	0.302	0.339	0.366
	seq_38	0.026	0.076	0.138	0.191	0.214	0.272	0.323	0.362	0.396	0.440
	seq_41	0.035	0.056	0.114	0.127	0.181	0.225	0.250	0.254	0.276	0.311
Outdoor	seq_20	0.101	0.187	0.278	0.317	0.405	0.447	0.520	0.665	0.702	0.869
	seq_21	0.004	0.067	0.154	0.198	0.233	0.304	0.339	0.392	0.429	0.538
	seq_31	0.021	0.083	0.135	0.195	0.256	0.303	0.327	0.381	0.423	0.463
	seq_48	0.006	0.032	0.061	0.076	0.095	0.106	0.140	0.150	0.196	0.207
	seq_50	0.009	0.041	0.087	0.111	0.142	0.174	0.209	0.236	0.266	0.303
Both	seq_44	0.014	0.046	0.109	0.135	0.176	0.213	0.243	0.281	0.342	0.344

sliding windows. those points whose distance are shorter than safe distance will be considered as obstacle points. In the searching procedure, consuming a single octree data structure including n nodes with a tree depth of d, the complexity can be performed as $\mathcal{O}(d) = \mathcal{O}(log_8 n) = \mathcal{O}(log n)$. This optimization of searching method is effective for real-time system.

III. Experiments

It is undeniable that classifier-based method get a good effect by training a lot of scene in the prior. However, the excellent performance of those methods are limited in the dataset similar environment. So we focus on the unsupervised methods and compare our method directly against geometric-based methods. Following the work in the paper [24], we compare our method with standard RANSAC method. In this paper, Pham et al. have mentioned that currently there is no proper 3D benchmarking available, thus quantitative evaluation of our method is difficult. One has to fall back on conventional 2D method for the issue of 3D maps segmentation. Meanwhile, work of [32] also mentioned that currently there were no evaluation methods for 3D data available because some problems arise when benchmarking 3D point cloud segmentation. Pham et al. chose the NYUv2 dataset which is proposed in the work of [33] published by Nathan Silberman. As the author demonstrates in the paper, this dataset only has indoors scenes and leads to better object segmentation. The road part of KITTI benchmark [34] contains discrete images which are unable to reconstruct environments. In order to prove that our approach can segment ground plane in the both indoor scenes and outdoor scenes, we provide qualitative comparison on the TUM monoVO dataset [28]. This dataset contains 50 photometrically calibrated sequences, comprising 105 minutes of video recorded in dozens of different indoor and outdoor environments. It is recorded by handing evaluate so that it contains small slight shake and coordination rotation which can demonstrate adaptability and robustness of our method. Eleven sequences, which contain five indoor scenes, five outdoor scenes and one mixed scene, are chosen to verify our method in the following four aspects: the high-efficiency to different size of sliding windows, the high-efficiency to density of sparse point cloud, robustness to density and robustness

TABLE II: The time usage of our method with different density (size of sliding windows $= 1000$).

scene	seq	the time usage with Density (ms)				
		1	5	10	15	19
Indoor	seq_01	4.680	0.904	0.223	0.052	0.025
	seq_11	5.247	0.976	0.337	0.095	0.029
	seq_28	4.701	0.846	0.135	0.107	0.042
	seq_38	4.452	0.694	0.105	0.059	0.026
	seq_41	4.199	0.868	0.183	0.038	0.035
Outdoor	seq_20	4.904	1.023	0.355	0.137	0.101
	seq_21	4.420	0.758	0.079	0.017	0.004
	seq_31	4.614	0.865	0.155	0.036	0.021
	seq_48	3.686	0.515	0.056	0.013	0.006
	seq_50	3.757	0.566	0.050	0.017	0.009
Both	seq_44	3.999	0.591	0.094	0.025	0.014

to different scenes. In our experiment, we set the radius as 0.2cm, and the number of neighbors equals to 5 for taking into account the density of point cloud and the effect of filtering. All experiments run 5 times and the average are accounted.

The high-efficiency to different size of sliding windows. As shown in Table I, computing time (obtained on an Intel i7 quad core @ 3.6GHz with 16Gb memory) of our method is less than 1ms each frame with the size of sliding windows increasing from 1000 to 10000. In this experiment the density is set to 19 which means point cloud is the most sparse. It is acceptable time usage for DSO which is less than 33ms per frame.

The high-efficiency to density of sparse point cloud. Table II sets the size of sliding window to 1000. The density becomes smaller as the number is from 1 to 19. Our method takes less than 1ms when the value of density larger than 5, which meets our expectations in real-time.

Superior performance in the case of different densities and different sliding windows makes our method achieve high-efficiency against the change of density and size of sliding window in both indoor scenes and outdoor scenes.

Robustness to density Robust performance to density in ground plane segmentation is shown in Fig.6. The points colored in gray, green, yellow and blue represent the original point clouds, obstacle points, candidate plane points and fit plane points, respectively. Almost all fitted ground plane points (blue) distribute at the bottom of scene viewed by side view

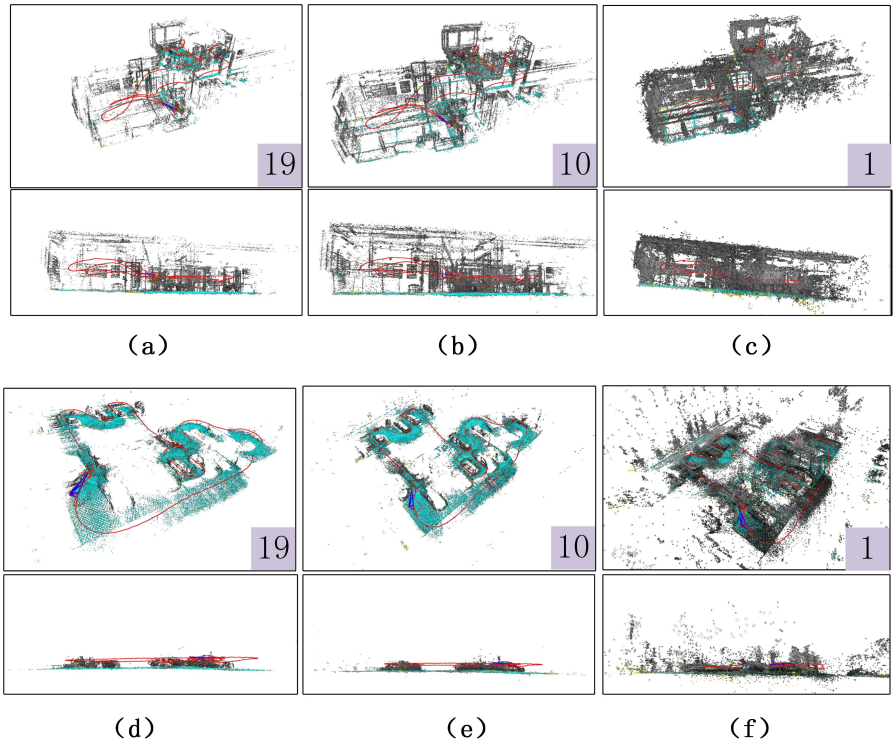

Fig. 6: Example of the performances of our method with different densities. (a), (b) and (c) represent both stereogram and side view of the same scene of seq_01 in the case of the density value is equal to 19, 10 and 1 respectively. (d), (e) and (f) represent both stereogram and side view of the same scene of seq_25 in the case of the density value is equal to 19, 10 and 1 respectively.**(best viewed in zoomed mode)**

in both indoor scenes and outdoor scenes. The candidate plane points (yellow) is sparse and only a small part of them distribute on the ground plane for the initialization rotation of coordination.

Robustness to different scenes. Comparison of experimental performance of traditional RANSAC method with our method in five sequences are shown in the Fig.7. For seq_20 contains upstairs which causes the huge change on the height of camera pose, the RANSAC method can only segment the plane which contains the maximum number of points, while our method split a number of ground plane benefited from the sliding window. The RANSAC method has a wrong segmentation by splitting the wall in seq_11. Correct division in those sequences using our method for the robustness of trust region.

IV. Conclusion

This paper presents a novel approach, namely the Real-time Trust Region ground plane segmentation, for dealing with the difficulties of segmentation in dynamic unknown environment and the rotation of initialization coordination. Experiments show more robustness and high-efficiency than the RANSAC method due to RTTR providing quality inliers for ground plane segmentation. At the same time, RTTR shows considerable speed-ups in computational times since limited candidate points by sliding windows are used to fit the ground plane. The excellent performance of our method are shown on the both indoor and outdoor scenes of the TUM monoVo dataset. A continuation to this study could be using our method on real mobile robots to achieve autonomous navigation with a single camera. It would also be interesting to segment more objects and recognize the split objects in real-time.

Acknowledgment

This work is supported by National High Level Talent Special Support Program, National Natural Science Foundation of China (No. 61340046, 61673030, U1613209), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130001110011), Natural Science Foundation of Guangdong Province (No. 2015A030311034). Specialized Research Fund for the strategic and prospective industrial development of Shenzhen city (No. ZLZBCXLJZI20160729020003).

References

[1] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8(2-3):73–272, 2015.

[2] L. Buonocore, dos Santos, A. A. Neto, and C. L Nascimento. Fastslam filter implementation for indoor autonomous robot. pages 484–489, 2016.

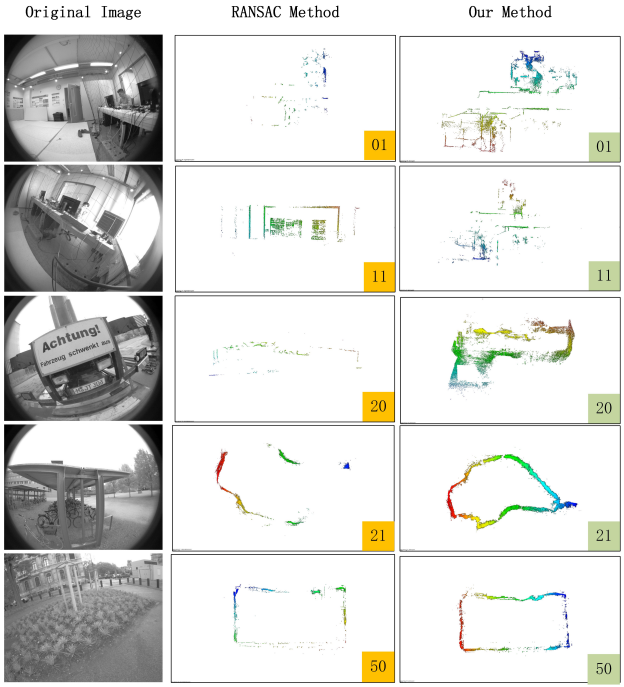

Fig. 7: The qualitative comparison of RANSAC and our RTTR method. The first row is an image from relative sequence, and segmentation plane of each Sparse Point cloud is represented with a top view, and all those colorful points are ground points. The lower right corner of the picture shows the sequence number, the yellow represents the RANSAC method and the green means our method. **(best viewed in zoomed mode)**

[3] A Aladren, G Lopez-Nicolas, L Puig, and J. J Guerrero. Navigation assistance for the visually impaired using rgb-d sensor with range expansion. *IEEE Systems Journal*, (99):1–11, 2014.

[4] Shweta N. Dethe, Varsha S. Shevatkar, and R. P. Bijwe. Google driverless car. *International Journal of Scientific Research in Science, Engineering and Technology*, 2, 2011.

[5] Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-time plane segmentation using rgb-d cameras. In *Robot Soccer World Cup XV*, pages 306–317, 2012.

[6] Rostislav Hulik, Vitezslav Beran, Michal Spanel, Premysl Krsek, and Pavel Smrz. Fast and accurate plane segmentation in depth maps for indoor scenes. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1665–1670, 2012.

[7] Abhishek Anand, Hema Swetha Koppula, Thorsten Joachims, and Ashutosh Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *International Journal of Robotics Research*, 32(1):19–34, 2013.

[8] Anoop Cherian, Vassilios Morellas, and Nikolaos Papanikolopoulos. Accurate 3d ground plane estimation from a single image. In *IEEE International Conference on Robotics and Automation*, pages 2243–2249, 2009.

[9] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76(1):53–69, 2008.

[10] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Learning 3-d scene structure from a single still image. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[11] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. *Advances in Neural Information Processing Systems*, 18:1161–1168, 2005.

[12] S Osswald, J. S Gutmann, A Hornung, and M Bennewitz. From 3d point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids. In *IEEE-RAS International Conference on Humanoid Robots*, pages 93–98, 2011.

[13] Aron Monszpart, Nicolas Mellado, Gabriel J. Brostow, and Niloy J. Mitra. Rapter: Rebuilding man-made scenes with regular arrangements of planes. In *ACM SIGGRAPH*, 2015.

[14] Olaf Köhler and Ian Reid. Efficient 3d scene labeling using fields of trees. pages 3064–3071, 2013.

[15] Trung T Pham, Ian Reid, Yasir Latif, and Stephen Gould. Hierarchical higher-order regression forest fields an application to 3d indoor scene labelling. In *IEEE International Conference on Computer Vision*, pages 2246–2254, 2015.

[16] Julien P. C Valentin, Sunando Sengupta, Jonathan Warrell, Ali Shahrokni, and Philip H. S Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *Computer Vision and Pattern Recognition*, pages 2067–2074, 2013.

[17] D. Wolf, J. Prankl, and M. Vincze. Fast semantic segmentation of 3d point clouds using a dense crf with learned parameters. In *IEEE International Conference on Robotics and Automation*, pages 4867–4873, 2015.

[18] K. Klasing, D. Althoff, D. Wollherr, and M. Buss. Comparison of surface normal estimation methods for range sensing applications. In *IEEE International Conference on Robotics and Automation*, pages 3206–3211, 2009.

[19] Xiaoyi Jiang and Horst Bunke. Fast segmentation of range images into planar regions by scan line grouping. *Machine Vision and Applications*, 7(2):115–122, 1994.

[20] By P Hough. Methods and means for recognizing complex pattern. 1962.

[21] Bastian Oehler, Joerg Stueckler, Jochen Welle, Dirk Schulz, and Sven Behnke. Efficient multi-resolution plane segmentation of 3d point clouds. In *International Conference on Intelligent Robotics and Applications*, pages 145–156, 2011.

[22] R Schnabel, R Wahl, and R Klein. Efficient ransac for point-cloud shape detection. In *Computer Graphics Forum*, pages 214–226, 2007.

[23] Byung Tae Oh, Ho Cheon Wey, and Du Sik Park. Plane segmentation based intra prediction for depth map coding. In *Picture Coding Symposium*, pages 41–44, 2012.

[24] Trung T. Pham, Markus Eich, Ian Reid, and Gordon Wyeth. Geometrically consistent plane extraction for dense indoor 3d maps segmentation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4199–4204, 2016.

[25] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the Acm*, 24(6):381–395, 1981.

[26] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *IEEE International Conference on Robotics and Automation (ICRA)*, May 9-13 2011.

[27] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. 2016.

[28] Jakob Engel, Vladyslav Usenko, and Daniel Cremers. A photometrically calibrated benchmark for monocular visual odometry. 2016.

[29] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *IEEE International Conference on Computer Vision*, pages 1449–1456, 2013.

[30] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *International Journal of Robotics Research*, 34(3):314–334, 2014.

[31] Donald Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147, 1982.

[32] Simon Christoph Stein, Florentin Wörgötter, Markus Schoeler, Jeremie Papon, and Tomas Kulvicius. Convexity based object partitioning for robot applications. In *IEEE International Conference on Robotics and Automation*, pages 3213–3220, 2014.

[33] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760, 2012.

[34] R. Urtasun, P. Lenz, and A. Geiger. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.