2016 IEEE International Conference on Automation Science and Engineering (CASE)
Fort Worth, TX, USA, August 21-24, 2016

A Novel RGB-D SLAM Algorithm Based on Points and Plane-Patches

Ruihao Li[1], Qiang Liu[1], Jianjun Gui[1], Dongbing Gu[1] and Huosheng Hu[1]

Abstract— **In this work, we present a novel RGB-D SLAM algorithm. The novelty of the proposed algorithm lies in the use of both feature points and plane patches for pose estimation. A plane patch is defined as a small-sized patch constructed by using a feature point with small curvature. The feature points with small curvature are called plane points. The remaining feature points are classified as either smooth points which represent smooth surfaces or structural points which represent edges or corners of some objects. Two criteria (coplanarity and overlap) are used to match the plane patches from two frames. The proposed algorithm is able to weight various types of feature points in order to improve the robustness of SLAM algorithms in favouring plane patches but not the isolated feature points, and maintain the real-time performance of SLAM algorithms in favouring small plane patches, but not the large size of planes. We evaluate the proposed algorithm on multiple benchmark datasets. A large scale experiment is also presented to show the robustness of our algorithm.**

I. INTRODUCTION

In the past few years, Simultaneous Localization and Mapping (SLAM) has been widely used in autonomous mobile robotic systems, augmented reality and virtual reality, search and rescue, etc [1], [2]. SLAM technique enables robots to localize themselves and construct the map in an unknown environment. Visual SLAM using commercial monocular [3], stereo [4] or RGB-D cameras [5] provides great opportunity for potential applications considering their low-cost, efficiency and real-time performance.

Nowadays RGB-D SLAM algorithms can be divided into feature-based [5], [6] and direct-based [7], [8], [9]. Feature-based algorithms are the main focus of this work, which extract and use invariant features as primitives to perform geometrical registration. Most current algorithms operate on low-level feature points, such as SIFT, SURF and ORB. These low-level primitives are generally noisy, redundant and require mass storage. Therefore, high-level primitives, such as planes, are introduced into the SLAM [10], [11]. They provide more stable parametric representation and compact interpretation of the scene. However the number of macroscopic-visible planes is usually very small and sometimes less than three in an image. And macroscopic-visible plane extraction is time-consuming due to the fact of plane normal calculations for all points. Furthermore, parameters of macroscopic-visible planes are sometimes different from those of constituted microscopic plane patches due to the noise from depth sensor and real geometry structure.

In this work, we introduce a novel RGB-D SLAM algorithm that uses both feature points and microscopic plane patches as primitives. We first extract feature points and use Principal Component Analysis (PCA) method to calculate the curvature and normal of each feature point. Then we classify feature points into plane points, smooth points and structural points. Next a feature matching process is carried out by combining feature point matching and plane patch matching. Feature points together with plane patches (called point-plane patch method in this paper) are used to estimate the egomotion.

In the next section, we provide a review of related works. In section III, we state the problem and give an overview of our SLAM system. Section IV presents the method of plane patch extraction and matching. Section V addresses the issue of camera pose estimation using point-plane patch method. Section VI demonstrates the experimental results and analyses the performance of our system. In section VII, we conclude our work.

II. RELATED WORK

MonoSLAM was the earliest system to show SLAM performance using a single camera [3]. PTAM implemented a real-time SLAM algorithm by using parallel tracking and mapping threads [12]. DTAM improved the robust performance in textureless regions and fast motion situations with direct-based method [13].

As to RGB-D SLAM, Endres at el. [5] demonstrated an RGB-D SLAM system which includes feature based pose estimation and pose graph optimization. Mur-Artal at el. [6] proposed an ORB-SLAM system which adopted ORB features and demonstrated real-time performance in tracking and mapping for monocular cameras and RGB-D cameras.

The SLAM using plane as primitives can be traced back to the work done by Weingarten at el. [14]. They used 3D laser to detect planes and performed plane-to-plane registration with an ICP algorithm. Trevor at el. [15] combined RGB-D camera and 2D laser to detect the planar surfaces for pose estimation. Taguchi at el. [11] introduced a novel SLAM algorithm which treated both points and planes as primitives. Ataer-Cansizoglu at el. [16] improved Taguchi's SLAM system by predicting planes with a pose prediction model so that the time spent on plane extraction was decreased. Salas-Moreno et al. [17] presented a novel dense planar SLAM algorithm which perceived, merged and compressed the arbitrary planar regions and used them for registration. Gao [18] proposed to use planar points in pose estimation by extracting all planes first and then testing if feature points are on the planes. Kaess [10] proposed a minimum

[1] Ruihao Li, Qiang Liu, Jianjun Gui, Dongbing Gu and Huosheng Hu are with School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK {rlig, qliui, jgui, dgu, hhu}@essex.ac.uk

978-1-5090-2409-4/16/$31.00 ©2016 IEEE

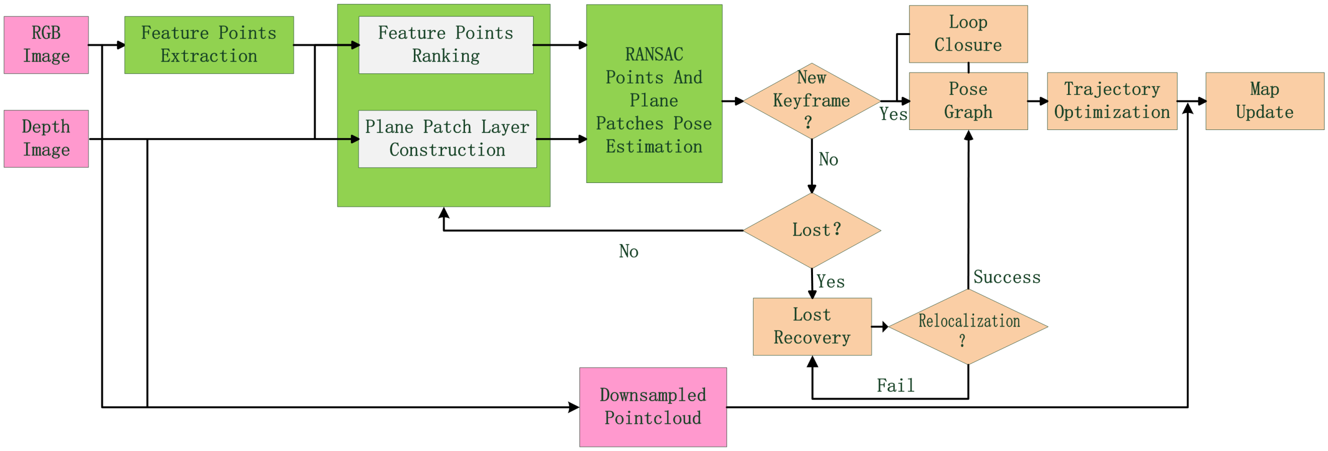

Fig. 1: System overview.

representation of plane parameters in order to produce a solution to the least-square estimation problem using Gauss-Newton methods. Cupec [19] used both planar surfaces and line segments to recognize places and relocalize the camera. Li at el. [20] presented to use points and surface normals to realize pose estimation which is similar with our algorithm and almost proposed at the same time compared with our paper. The difference is that normals can only estimate rotation and are powerless to estimate translation.

In summary, macroscopic-visible planes potentially are able to produce the compact representation, but time consuming on plane extraction.

III. PROBLEM DESCRIPTION AND SYSTEM OVERVIEW

As with most SLAM algorithms, the proposed SLAM algorithm is also separated into a front-end and a back-end. The difference is that two layers—feature point layer and plane patch layer—are used in the front-end of the proposed algorithm. The purpose of using both of them is to ensure the robust and real time performance in texture-less and large-scale environment. Our algorithm first extracts feature points in an image, and judges whether or not they are in plane patches based on feature points and their neighbourhood area. Then the plane patch's parameters are calculated. This is the work done in the plane patch layer construction (see Fig. 1). A plane patch consists of many robust plane points, leading to more compact representations and more accurate plane parameters. Meanwhile, the feature points are ranked according to their geometrical information. If a point is in a plane or in a smooth surface, it has less uncertainty in depth and contains more reliable information. If a point is on the edge of some objects and its depth is very different from neighbour points (its depth gradient is very large in one direction), it could have large noise in depth and may produce some bad effects in the process of pose estimation. Different weights are assigned to the points according to their uncertainties. The weighted features are able to improve the robustness and accuracy of the algorithm.

In the back-end, the algorithm selects keyframes and adds them into a pose graph. Then loop closures are detected by comparing current keyframe with previous nearby keyframes and keyframes having similar attitude. The general framework g2o [21] is used for graph optimization. The trajectory will be optimized and the map will be updated. If the algorithm loses the trajectory under some circumstances, it will perform a relocalization process from the lost frame. The overview of our approach is shown in Fig. 1.

IV. PLANE PATCH LAYER CONSTRUCTION

A. Plane patches extraction

A plane patch is defined as a small-sized patch constructed by using a feature point with small curvature. Feature points with small curvature are called plane points. Each plane point is treated as center of a potential plane patch, and the patch size is defined as 21×21 in pixel dimension, PCA technique is adopted in extraction. For each patch, we have a set of points $\{\mathbf{p}_i = (x_i, y_i, z_i)^{\mathbf{T}}\}$, i = 1, ..., N. Let $\overline{\mathbf{p}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i$, then the 3×3 covariance matrix Σ can be calculated as below:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{p}_i - \overline{\mathbf{p}})(\mathbf{p}_i - \overline{\mathbf{p}})^{\mathbf{T}} \tag{1}$$

Three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and their corresponding eigenvectors are computed if $rank(\Sigma) = 3$. Let $\lambda_1 > \lambda_2 > \lambda_3$, the eigenvector with minimum eigenvalue λ_3 represents the normal of this patch. In order to judge whether the selected patch is a plane or not, the criterion shown below is used:

$$\varepsilon = \frac{\lambda_3}{\sum_{i=1}^{3}\lambda_i} < th_p \tag{2}$$

Where ε is the curvature of the selected patch. If the eigenvalues of a covariance matrix meet the condition (2), the corresponding patch is treated as a plane and added into the plane patch layer. A plane is usually represented by a vector $\pi = (\pi_1, \pi_2, \pi_3, \pi_4)^{\mathrm{T}}$. For any point $\mathbf{p} = (x, y, z)^{\mathbf{T}}$ on the plane, the following equation is true.

$$\pi_1 x + \pi_2 y + \pi_3 z + \pi_4 = 0 \tag{3}$$

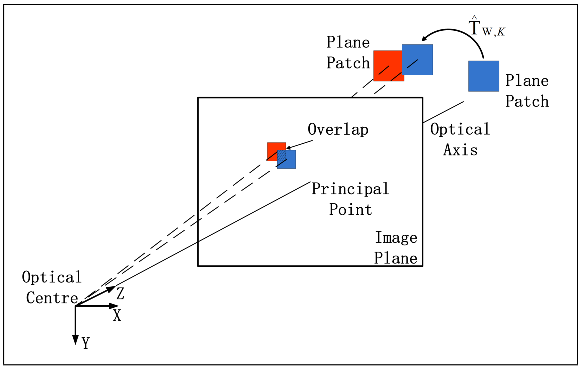

Fig. 2: Plane patch in previous keyframe is transformed from global coordinate to current frame using pose prediction $\hat{\mathbf{T}}_{W,K}$, and the coplanarity criterion is checked first. Then both of the plane patch in current frame and the one predicted from previous keyframe are projected to the image plane in current frame to check the overlap criterion.

Plane normal $\mathbf{n} = (\pi_1, \pi_2, \pi_3)^{\mathrm{T}}$ is equal to the eigenvector with minimum eigenvalue λ_3, and $\pi_4 = -\mathbf{n}^{\mathrm{T}}\overline{\mathbf{p}}$ represents the distance from origin to the plane. If $\|\mathbf{n}\| = 1$ and $\pi_4 \geqslant 0$, the plane representation is unique.

Assume the transformation from global frame to local frame is T, then we have:

$$\pi_l = \mathbf{T}^{-\mathrm{T}}\pi_g \qquad (4)$$

B. Plane patches matching

To match plane patches, two criteria—coplanarity criterion and overlap criterion—are used. The coplanarity criterion is to check if the normals and the distances from origin to plane between the pair of planes are the same. Here we propose to use the minimum representation of a plane for coplanarity criterion instead of the normal and the distance to reduce the computational complexity. The minimum representation of a plane is $\rho(\pi) = (\pi_1\pi_4, \pi_2\pi_4, \pi_3\pi_4)^{\mathrm{T}}$, which represents the intersection of a plane with the normal which passes through origin. The coplanarity can be evaluated by the Mahalanobis distance [19]:

$$d_c = (\rho(\pi_l) - \rho(\hat{\mathbf{T}}_k^{-\mathrm{T}}\pi_g))^{\mathrm{T}}\mathbf{Q}_c^{-1}(\rho(\pi_l) - \rho(\hat{\mathbf{T}}_k^{-\mathrm{T}}\pi_g)) \qquad (5)$$

Where $\rho(\pi_l)$ is the minimum representation of plane π_l in local kth frame, $\rho(\pi_g)$ is the minimum representation of plane π_g in global frame. $\hat{\mathbf{T}}_k = \mathbf{T}_{k-1}\mathbf{T}_{k-2}^{-1}\mathbf{T}_{k-1}$ is the pose prediction transformed from the global coordinate to the kth frame, where $\mathbf{T}_{k-1}$ and $\mathbf{T}_{k-2}$ are the pose estimation of $(k-1)$th frame and $(k-2)$th frame, respectively. $\mathbf{Q}_c$ is related to the covariance matrix Σ_ρ representing the uncertainty of measured plane and the covariance matrix Σ_t representing the uncertainty of pose prediction. So the coplanarity criterion can be expressed by:

$$d_c \leqslant th_{co} \qquad (6)$$

Assume d_c obeys χ^2 distribution, then the threshold th_{co} can be determined by a desired probability.

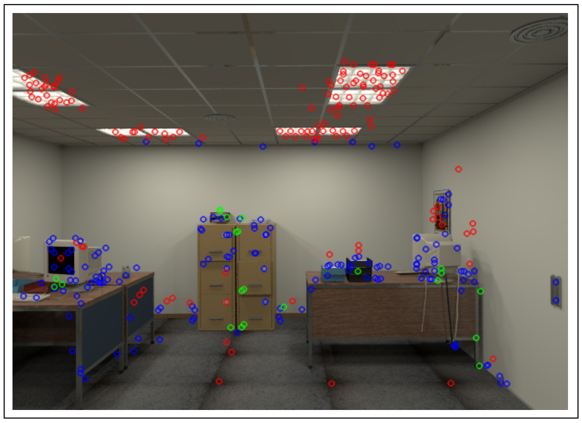

Fig. 3: Different types of feature points. Red points, blue points and green points represent plane points, smooth surface points, and structural points respectively.

The overlap criterion is to check if there exists overlaps between two plane patches. As illustrated in Fig. 2, a plane patch in previous keyframe is first re-projected back into current frame. Then the center and size of plane patch are used to check the overlap based on the equation below:

$$d_o = \left(\frac{\mathbf{c}_l - \hat{\mathbf{c}}_{gl}}{a_l + \hat{a}_{gl}}\right)^{\mathrm{T}}\mathbf{Q}_o^{-1}\left(\frac{\mathbf{c}_l - \hat{\mathbf{c}}_{gl}}{a_l + \hat{a}_{gl}}\right) \qquad (7)$$

Where $\mathbf{c}_l$ and a_l are the center and size of the plane patch re-projected into the 2D image from local frame, and $\hat{\mathbf{c}}_{gl}$ and $\hat{a}_{gl}$ are the center and size of the plane transformed by the pose prediction $\hat{T}_k$ from global coordinate to local frame. $\mathbf{Q}_o$ is the covariance matrix related to Σ_c representing the uncertainty of center points and the covariance matrix Σ_t representing the uncertainty of pose prediction. The overlap criterion is expressed as:

$$d_o \leqslant th_{ov} \qquad (8)$$

Assuming d_o obeys χ^2 distribution, then the threshold th_{ov} can be determined by a desired probability.

V. CAMERA POSE ESTIMATION WITH POINTS AND PLANE-PATCHES

A. Feature points ranking

The feature points are separated into three subsets. The components of the first subset are the called plane points which have small curvature. The remaining feature points are separated by their patch curvatures into smooth surface points indicating they are on smooth surfaces, but not on plane patches, and structural points representing edges or corners of some objects. They are shown in Fig. 3 with different colors. Following from (2), the following two criteria are used for smooth surface points (9) and structural points (10), respectively,

$$th_p < \varepsilon < th_s \qquad (9)$$

$$\varepsilon > th_s \qquad (10)$$

Each feature point is assigned with a weight. The weight parameter for a point should be inversely proportional to the uncertainty of the point or the curvature ε. Let the maximum curvature be ε_{max} and minimum curvature ε_{min}, the weight of feature points is defined below:

$$\begin{cases} \omega_i = \dfrac{\varepsilon_{min}(\varepsilon_{max} - \varepsilon)}{\varepsilon(\varepsilon_{max} - \varepsilon_{min})}, & plane\ point\ or\ surface\ point \\ \omega_i = \quad \dfrac{1}{2}, & structual\ point \end{cases} \tag{11}$$

Each plane patch has a weight parameter ω_j which is equal to the weight parameter of its corresponding plane point.

B. RANSAC registration using points and plane patches

All feature points and plane patches are used to estimate transformation, with different weight parameters to enhance the robustness and accuracy of registration. Assume two matched sets of feature points from two frames are $\{\mathbf{p}_{gi}\}$ and $\{\mathbf{p}_{li}\}$, i = 1, ..., N. Let $\overline{\mathbf{p}}_g = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_{gi}$ and $\overline{\mathbf{p}}_l = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_{li}$. The matched plane patches are $\{\pi_{gj}\}$ and $\{\pi_{lj}\}$, j = 1, ..., M. To estimate the rotation R, we minimize the cost below:

$$\min(\sum_{i=1}^{N}\omega_i\|(\mathbf{p}_{li} - \overline{\mathbf{p}}_l) - \mathrm{R}(\mathbf{p}_{gi} - \overline{\mathbf{p}}_g)\|^2 + \sum_{j=1}^{M}\omega_j\|\mathbf{n}_{lj} - \mathrm{R}\mathbf{n}_{gj}\|^2)$$

Where $\mathbf{n}_{gj}$ and $\mathbf{n}_{lj}$ are unit normal vector of π_{gj} and π_{lj}. To estimate the translation $\hat{\mathbf{t}}$, we minimize the cost below:

$$\min(\sum_{i=1}^{N}\omega_i\|\mathbf{t} - (\overline{\mathbf{p}}_l - \mathrm{R}\overline{\mathbf{p}}_g)\|^2 + \sum_{j=1}^{M}\omega_j\|\mathbf{n}_{lj}^{\mathrm{T}}\mathbf{t} - (d_{lj} - d_{gj})\|^2)$$

Where d_{lj} and d_{gj} are π_4 component of plane patches π_{lj} and π_{gj}.

Random sample consensus (RANSAC) is very effective in distinguishing inliers and outliers. For point only SLAM, three non-collinear points have to be used as initial points. For our proposed approach, two plane points are randomly selected as initial features, then totally four features - two plane points and two plane patches are used to estimate transformation. The requirement for them is that the line passing through two feature points can not be parallel to both normals of two plane patches. If there is no enough features, one plane point and one feature point without plane patch can be used for pose estimation.

The penalty function to be minimised by RANSAC is

$$C = \sum_{i=1}^{N} D(\mathbf{p}_l, \mathbf{p}_g, \hat{\mathbf{T}}) + \sum_{j=1}^{M} D(\rho_l, \rho_g, \hat{\mathbf{T}}) \tag{12}$$

and

$$D(\mathbf{p}_l, \mathbf{p}_g, \hat{\mathbf{T}}) = \begin{cases} \|\mathbf{p}_l - \hat{\mathbf{T}}\mathbf{p}_g\|^2, & \|\mathbf{p}_l - \hat{\mathbf{T}}\mathbf{p}_g\|^2 < th_{ra1} \\ \mathrm{th}_{ra1}, & \|\mathbf{p}_l - \hat{\mathbf{T}}\mathbf{p}_g\|^2 \geq th_{ra1} \end{cases} \tag{13}$$

$$D(\rho_l, \rho_g, \hat{\mathbf{T}}) = \begin{cases} \|\rho_l - \hat{\mathbf{T}}\rho_g\|^2, & \|\rho_l - \hat{\mathbf{T}}\rho_g\|^2 < th_{ra2} \\ \mathrm{th}_{ra2}, & \|\rho_l - \hat{\mathbf{T}}\rho_g\|^2 \geq th_{ra2} \end{cases} \tag{14}$$

where ρ_l and ρ_g are the minimum representations of plane, th_{ra} is a threshold for the distance of matched features. If the distance is small than the threshold, they will be treated as inliers, otherwise, outliers.

TABLE I: Comparison of ATE RMSE between our method, point-only method and RGB-D SLAM on different kinds of benchmark datasets. The unit of RMSE is in meters.

Dataset	Point-plane patch	Point-only	RGB-D SLAM
fr1/desk	0.025	0.030	**0.023**
fr1/desk2	0.053	0.059	**0.043**
fr1/xyz	**0.014**	0.015	0.014
fr1/rpy	**0.022**	0.024	0.026
fr1/plant	**0.060**	0.074	0.091
fr2/desk	**0.047**	0.061	0.057
fr2/xyz	0.015	0.015	**0.008**
fr3/long	**0.031**	0.044	0.032
ICL-NUIM/living0	**0.021**	0.033	0.045
ICL-NUIM/living1	**0.011**	0.019	0.032
ICL-NUIM/office0	**0.022**	0.029	0.027
ICL-NUIM/office2	**0.018**	0.030	0.026
Our Arena lab/local	**0.038**	0.043	0.048
Our Arena lab/circle	**0.050**	0.058	0.056
Our Arena lab/wall	**0.037**	0.049	0.053
Our Arena lab/long	**0.062**	0.068	0.074

VI. EXPERIMENTAL EVALUATION

The evaluation of proposed SLAM algorithm is conducted on benchmark datasets first. Then tests in our lab are presented, where the ground truth trajectory is provided from a motion capture system. Finally a large-scale indoor environment test is demonstrated. All the experiments are run on a laptop equipped with Intel Core i7-3610QM 2.3Hz CPU and 8G RAM, no GPU is used for computing.

A. Tests on benchmark datasets and in our lab

In this part, we first use the widely used open datasets from TUM [22] and ICL-NUIM [23] which provide color images and depth images along with ground truth all associated by timestamp. The ground truth is obtained by a motion capture system. Then we run our point-plane patch SLAM to further verify its effectiveness in our Robotic Arena equipped with a motion tracking system. Eventually we test our method on 16 benchmark datasets including different scenes such as office room, living room, large lab, desk and so on.

Absolute trajectory error (ATE) introduced in [22] measuring the distance between estimated trajectories and ground truth trajectories is used to evaluate our system. We compared our point-plane patch method with point-only method and another feature-based state-of-the-art RGB-D SLAM [5]. The ATE RMSE results using three methods are shown in Table I. It can be seen from Table I that the RMSE using our point-plane patch method is much less than that using point-only method in most situations.

The plots of some trajectory differences are shown in Fig. 4 and the 3D point cloud maps constructed by our method are shown in Fig. 5. It can be seen that our SLAM algorithm can track camera positions in most scenes. The radius of our lab is about 5.5m and the height is about 8 m.

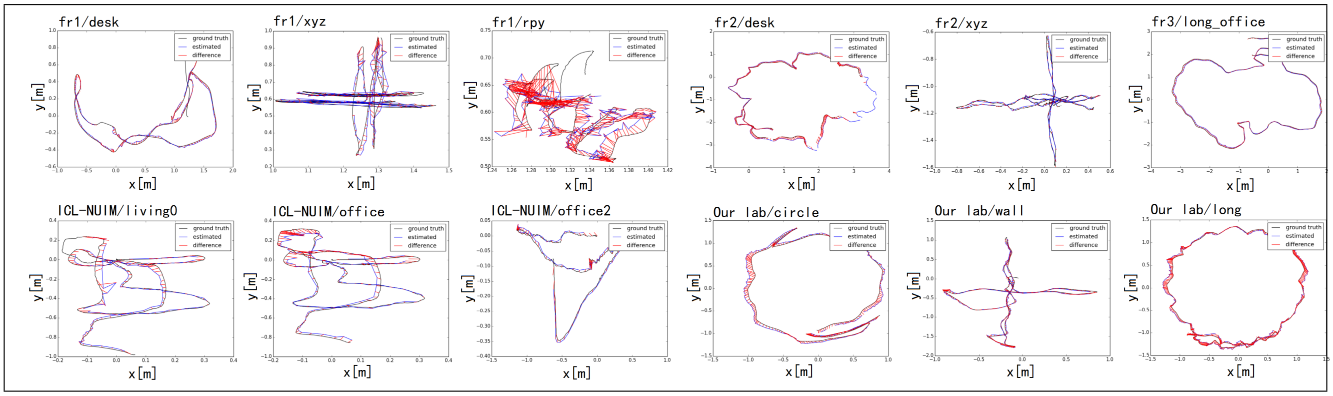

Fig. 4: Comparison between estimated trajectory using our method and ground truth trajectory

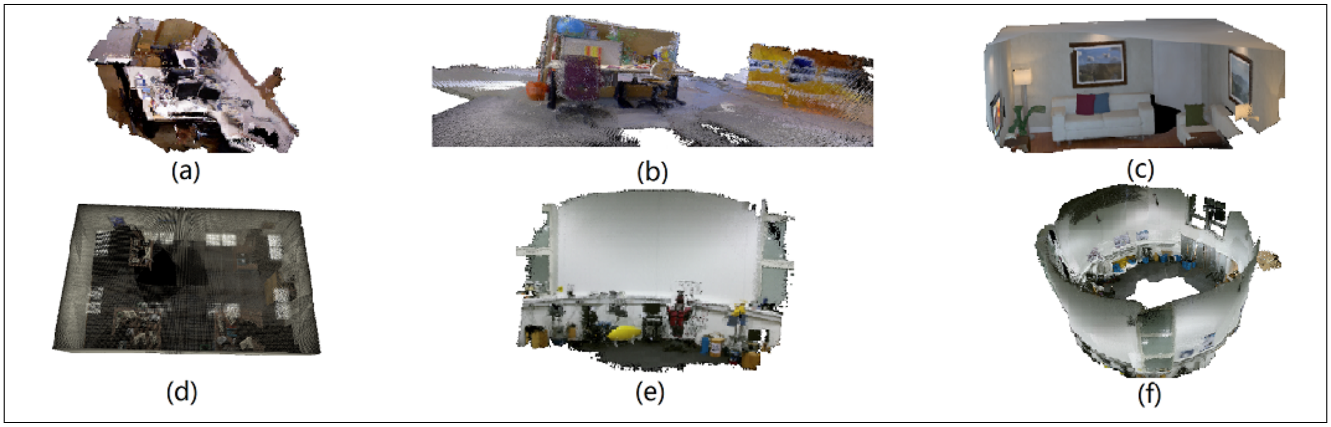

Fig. 5: 3D map reconstructed from benchmark datasets using our method. (a) fr1/desk dataset. (b) fr3/long office dataset. (c) ICL-NUIM/living room 0 dataset. (d) ICL-NUIM/office room 2 dataset. (e) Our lab/wall. (f) Our lab/long.

TABLE II: Average processing time for different procedures from 100 pairs of frames using ICL-NUIM office dataset.

Procedure	Time (ms)
Feature Points Extraction	35.03
Plane Patches Extraction	3.05
RANSAC Point-Plane Registration	7.45

TABLE III: Feature and RANSAC feature inlier composition from 100 pairs of frames using ICL-NUIM office dataset.

Feature Type	Number	Inlier Number
Plane Point	157	50
Plane Patch	157	50
Curvature Point	176	42
Structural Point	33	5
Total	523	147

B. Computational cost analysis

Most SLAM applications have a high requirement for real-time performance. The time spent on macroscopic-visible plane extraction took about 50ms to 100 ms per frame. However, the time spent on plane patch extraction saved much time. In order to analyse the computational cost of our method, we randomly chose 100 pairs of frames from ICL-NUIM office dataset to record the processing time for different procedures. The average processing time computed from these pairs of frames is shown in Table II. Here we used SURF as the point feature. The plane point curvature threshold th_p is 0.001, the curvature threshold th_s is 0.06, the RANSAC point inlier threshold th_{ra1} is 0.02^2 and the RANSAC plane inlier threshold th_{ra2} is 0.03^3. It can be seen from Table II that the average plane patch extraction time is 3.05 ms with about 366 feature points. Our SLAM system runs at about 10-15 Hz which depends on the number of feature points and of loop closures.

From Table III, it can be seen that the plane points with plane patches are more likely to be inliers while structural points are less likely to be inliers. It can mainly be explained by the depth image noise from RGB-D camera. The structural points have much more noise and larger covariance.

C. Test on a large-scale environment

In order to test the robustness of our method, we also ran our system in one large scale floor in our building that includes three large labs, four rooms, a long corridor, stairs and an elevator. The frames were over 16000, the area of the floor was about 400 m^2 and the trajectory distance our

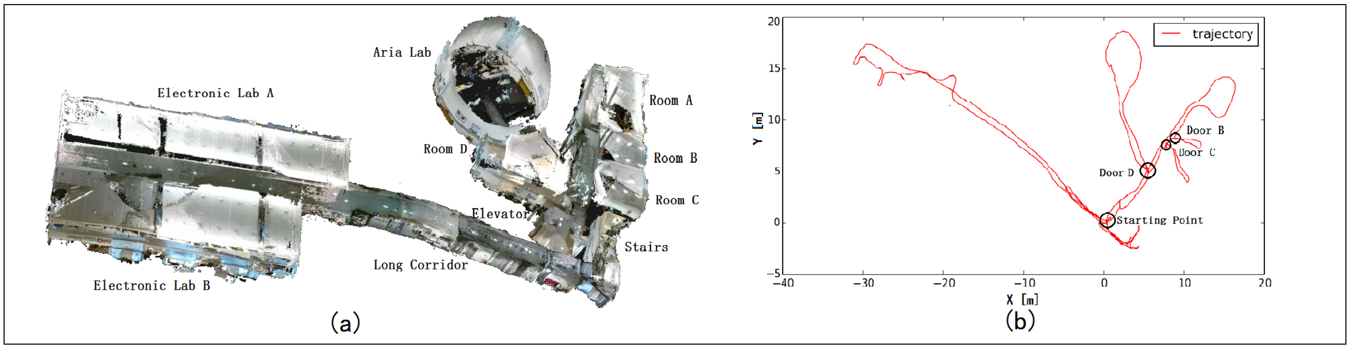

Fig. 6: (a) The 3D reconstructed dense map of one whole floor in our building. (b) The 2D trajectory while tracking.

platform moved was over 150 m. There were many challenge scenes, such as long corridor, pedestrians, featureless wall, high angular velocity movement, entering and exiting doors and so on. The reconstructed map is shown in Fig. 6(a). The trajectory accuracy can be evaluated when it passed the doors marked in Fig. 6(b). As the width of these doors is less than one meter, less than one third meter accuracy can be observed when passing the centers of the doors.

VII. CONCLUSIONS

In this paper, we present a novel point-plane patch SLAM algorithm which shows better performance in localization accuracy than point-only SLAM algorithms. This could be attributed to the usage of plane patches and different types of feature points. These "new features" are able to enhance the matching results and accordingly improve the accuracy of pose estimation. The time spent on extracting plane patches is not significant when comparing with macroscopic visible planes. The real time performance is still maintained at the same level with point only SLAM algorithms. Next step, we would like to combine SLAM with Deep Learning to realize localization improve the robustness and accuracy performance in challenge situations.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping: Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[3] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Computer Vision, Proceedings. IEEE International Conference on*, pp. 1403–1410, 2003.

[4] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 946–957, 2008.

[5] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *Robotics, IEEE Transactions on*, vol. 30, no. 1, pp. 177–187, 2012.

[6] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *Robotics, IEEE Transactions on*, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 3748–3754, 2013.

[8] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), IEEE international symposium on*, pp. 127–136, 2011.

[9] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-D SLAM with volumetric fusion," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 598–626, 2015.

[10] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 4605–4611, 2015.

[11] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 5182–5189, 2013.

[12] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Mixed and Augmented Reality (ISMAR), IEEE and ACM International Symposium on*, pp. 225–234, 2007.

[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), IEEE International Conference on*, pp. 2320–2327, 2011.

[14] J. Weingarten and R. Siegwart, "3D SLAM using planar segments," in *Intelligent Robots and Systems, IEEE/RSJ International Conference on*, pp. 3062–3067, 2006.

[15] A. J. Trevor, J. G. Rogers III, H. Christensen, *et al.*, "Planar surface SLAM with 3D and 2D sensors," in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 3041–3048, 2012.

[16] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, and T. Garaas, "Tracking an RGB-D camera using points and planes," in *Computer Vision Workshops (ICCVW), IEEE International Conference on*, pp. 51–58, 2013.

[17] R. F. Salas-Moreno, B. Glocken, P. H. Kelly, and A. J. Davison, "Dense planar SLAM," in *Mixed and Augmented Reality (ISMAR), IEEE International Symposium on*, pp. 157–164, 2014.

[18] X. Gao and T. Zhang, "Robust RGB-D simultaneous localization and mapping using planar point features," *Robotics and Autonomous Systems*, vol. 72, pp. 1–14, 2015.

[19] R. Cupec, E. K. Nyarko, D. Filko, A. Kitanov, and I. Petrović, "Place recognition based on matching of planar surfaces and line segments," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 674–704, 2015.

[20] S. Li and A. Calway, "Absolute pose estimation using multiple forms of correspondences from RGB-D frames," in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 4756–4761, 2016.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), IEEE International Conference on*, pp. 3607–3613, 2011.

[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*, pp. 573–580, 2012.

[23] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Robotics and automation (ICRA), IEEE international conference on*, pp. 1524–1531, 2014.