Multimed Tools Appl (2019) 78:713–726
https://doi.org/10.1007/s11042-017-5519-7

Real-time indoor scene reconstruction with Manhattan assumption

Zunjie Zhu[1] · Feng Xu[2] · Chenggang Yan[1] · Ning Li[1] · Bingjian Gong[1] · Yongdong Zhang[3] · Qionghai Dai[4]

Received: 8 June 2017 / Revised: 10 October 2017 / Accepted: 7 December 2017 /
Published online: 21 December 2017

Abstract This paper presents a novel end-to-end system for real-time indoor scene reconstruction, which outperforms traditional image feature point-based method and dense geometry correspondence-based method in handling indoor scenes with less texture and geometry features. In our method, we fully explore the Manhattan assumption, i.e. scenes are majorly consisted with planar surfaces with orthogonal normal directions. Given an input depth frame, we first extract dominant axes coordinates via principle component analysis which involves the orthogonal prior and reduce the influence of noise. Then we calculate the coordinates of dominant planes (such as walls, floor and ceiling) in the coordinates using mean shift. Finally, we compute the camera orientation and reconstruct the scene by proposing a fast scheme based on matching the dominant axes and planes to the previous frame. We have tested our approach on several datasets and demonstrated that it outperforms some well known existing methods in these experiments. The performance of our method is also able to meet the requirement of real-time with an unoptimized CPU implementation.

Keywords SLAM · Tracking · Depth sensor · Real-Time · AR

✉ Feng Xu
feng-xu@tsinghua.edu.cn

✉ Chenggang Yan
cgyan@hdu.edu.cn

Zunjie Zhu
zunjiezhu@gmail.com

[1] Institute of Information and Control, Hangzhou Dianzi University, Hangzhou, China

[2] School of Software, Tsinghua University, Beijing, China

[3] Institute of Computing Technology, Chinese Academy of Sciences(CAS), Beijing, China

[4] Department of Automation, Tsinghua University, Beijing, China

1 Introduction

In recent years, with the development of depth sensing technique, real-time 3D indoor scene scanning, which aims to reconstruct the 3D geometry of an indoor scene, becomes possible. Several systems are proposed [6, 11, 14] and promising results are generated. On the other hand, with Augment Reality (AR) becoming a hot topic in both academic and industry, real-time 3D scanning are eagerly demanded as the recovery of the 3D geometry of our real scene is the key to render virtual objects seamlessly aligned with the scene. In Microsoft Hololens, scanning the 3D geometry of the current room is required by many AR-based applications.

With a depth camera, as 3D information is directly recorded, the key to achieve 3D scanning is to estimate the camera motion between every two consecutive input frames. There are several ways to achieve this. First is to use Iterative Closest Point (ICP) to estimate the correspondences between the point cloud obtained by two depth frames. The best rigid motion estimated by the 3D correspondences is the desired camera motion as the scene is assumed to be static. Then the two point cloud can be merged by the estimated camera motion. An alternative solution is to estimate feature correspondences on color frames which are recoded accompanying with the depth frames for most commercial depth sensors. As the depth and color frames can be calibrated, the correspondences on 2D color frames can also be used to estimate the camera motions. Details can be found by most Structure from Motion (SfM) techniques.

However, these techniques have their own drawbacks [27, 28, 30]. In the ICP based methods, geometry features in the scene are required to robustly estimate correct camera motions. For two pure 2D planes in a 3D space, the closest point may not be the correct correspondence. In this situation, ICP may generate wrong camera motions. This extreme case may also happen in the real, for example, the sensor is recording a big white wall in a room. Besides, ICP requires a large number of sampling points and requires iterations to converge to the final correspondences, which means a relative heavy computation cost. Even though some ICP-based systems use GPU to achieve real-time performance, it is still not applicable for many real applications as GPU may be heavily occupied by other tasks, e.g. rendering, or it is a mobile application with no high performance GPU. Methods using color image features seem not suffer these drawbacks, but they are also not able to handle white walls as no sufficient confident image features can be extracted or matched. These methods require rich texture features in the scene.

We proposed a new solution to overcome the aforementioned drawbacks when scanning an indoor scene, which may contain big and texture-less walls. Our techniques explores the Manhattan assumption that the indoor scene is dominated by orthogonal plans. In practice, when a recorded frame contains sufficient orthogonal plans, large wall region may exist and usually few features can be extracted as walls are usually with consistent color. In this situation, the scheme based on the Manhattan assumption gives accurate motion estimation.

Our major contribution is a real-time camera motion estimation scheme based on Manhattan assumption. We first estimate the normal directions of all 3D points of a recorded frame. Then three orthogonal primary directions are estimated via principle component analysis (PCA). As the primary directions are estimated from all the depth points, random noise in the recorded depths is largely filtered out. Then we further estimate the coordinates of the planes and corresponding the dominant axes(i.e. normals) and planes coordinates of each frame. Finally we estimate poses of sensor and reconstruct target scenes.

In summary, there are three key contributions in this work:

- We fuse the Manhattan assumption into a camera motion estimation scheme.
- Propose a solid method to extract dominant normals efficiently.
- We develop a method to directly extract mutually orthogonal planes.
- Our system is the first known dominant planar indoor scene reconstruction algorithm capable of real-time (30 fps) CPU only execution.

2 Related work

In recent years, registration of 3D data has been a popular front-end solution. The registration method can be separated into local method and global method. The most representative solution of local method is ICP algorithm [2], which is used to estimate sensor pose [26, 29]. In practice, this makes tracking vulnerable and led researchers to design back-end such as loop closure detection [25], pose graph optimization [21] to recover sensor tracking failures. Some variants of the ICP algorithm have been used for frame-to-frame sensor pose tracking. For example, a solution can be found for registration by point-to-point [1, 24], point-to-line [15], point-to-plane [17], line-to-line [31], line-to-plane [4], and plane-to-plane [7] correspondences.

A perhaps more prevalent uses point-to-point method such as keypoint matching. This method should first detect a set of interest points in color images or depth map and assigned descriptor of each point based on appearance, then match keypoint pairs between adjacent frame. Many researchers has focus on efficiency, and research of feature detection [18], description [3, 19], and matching [13]. Lepetit et al. [13] employ random forests to match discrete features in images, Jamie et al. [20] goes beyond former by using regression approach. Henry et al. [9] extracts keypoint from color images and using RANSAC to get sensor poses, which were final refined by ICP. However these registration method that merely depend on points encounter insufficient correspondences or mismatch in texture-less and feature-less regions or regions with many repeated features.

To avoid the situation, Lee et al. [12] addressed the plane-to-plane registration, they first extract all planes in images and presented an available approach to compute correspondences using constraint estimation. Trecvor et al. [23] use two sensors for both plane-to-plane and line-to-plane correspondences. However, their method all have the drawback that their correspondence step costs a considerable computational resource in a mobile device such as a head-mounted device for VR.

Driven by these issues, our approach aiming to lower the computational costs, and execute correspondence step by using mutually orthogonal planes assumed by Manhattan-world. This method has not been addressed before and we present an efficient heuristic method to extract normals of the mutually orthogonal planes(we call these planes the dominant plane). Different from Taguchi et al. [22] which extract all planes in images, we need only extract three dominant planes and corresponding two frames by using plane normals and plane positions on dominant axis.

3 Method

We estimate the sensor coordinate only by three dominant planes in each frame. This section describes our procedure for identifying these planes and calculating transformations of sensor in each frame.

Given a set of depth images, the first of our algorithm is to reconstruct 3D point cloud of each image and calculate normals of each 3D point. The normals are then used to extract dominant axes(i.e. normals of dominant planes) for the scenario of current frame, and the axes are later used to confirm plane coordinates. Finally, we estimate camera pose for a frame by a rigid body transformation matrix which is acquired by plane coordinates and axes. The pipeline of our system is shown in Fig. 1.

3.1 Extract dominant axes

The dominant planes in Indoor scenes(such as Wall, Floor and Ceiling) are perpendicular to each other, so does their normals. Thus, We set these three normals of the plane to be the dominant axes in Manhattan-World. Meanwhile, the three planar fill most of its field of view in texture-less and feature-less scenes. Under the circumstances, We design a method to robustly and efficiently extract dominate axes in a frame.

3.1.1 Normals computing

As is showed in Fig. 2, while calculating the 3D normal of a pixel at frame f, we should first translate these pixels into 3D coordinate system, and then employ 3D coordinates of adjacent pixels to get the normal.

$$\mathbf{v}(u, v) = (\mathbf{D}_f(u + \kappa, v) - \mathbf{D}_f(u - \kappa, v) \tag{1}$$

$$\mathbf{v}'(u, v) = \mathbf{D}_f(u, v + \kappa) - \mathbf{D}_f(u, v - \kappa) \tag{2}$$

$$\mathbf{n}_f(u, v) = \Psi\left[\mathbf{v}(u, v) \times \mathbf{v}'(u, v)\right] \tag{3}$$

where × is the cross product, k is an adjustable parameter which represent the distance between two pixels, and we can adjust k to get the most accurate normals of a frame. Ψ is a function which convert a normal to unit vector:

$$\Psi[\mathbf{n}] = \mathbf{n} \cdot ||\mathbf{n}||^{-1} \tag{4}$$

3.1.2 Vector statistic

There are a plenty of points on a dominant plane, and normal directions of these point will be very similar. Meanwhile, because of disturbance and noises that generated from irregular objects and sensor itself, the normal clusters we have computed will contain a set of vectors which have a large angle with dominant axes. Thus we design a method to compute a histogram of normal directions over a unit sphere, aiming to extract candidate axes, i.e. normals of dominant planes.

We should confirm that the normals computed before were in camera coordinate system. And now We purposefully convert these normal $\mathbf{n} = (x, y, z)$ to the angle form $n = (\alpha, \beta, \gamma)$, where α, β, γ are angles between normal and camera coordinate axes. We

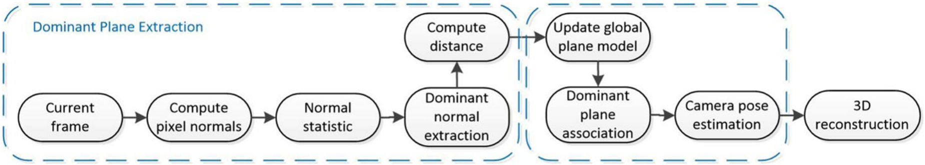


Fig. 1 Schematic pipeline of our system

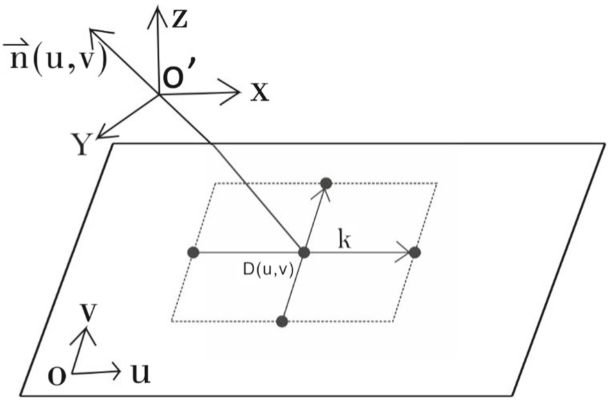


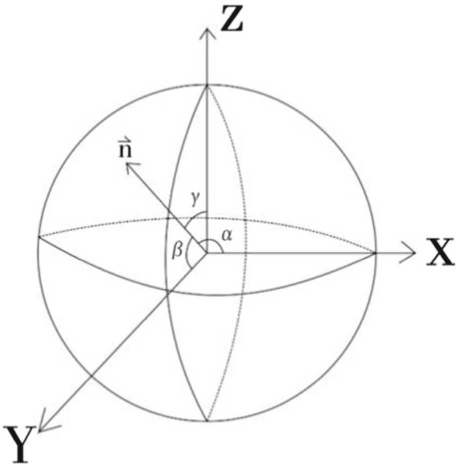
Fig. 2 This figure is the plane model which shows normal computing. Ouv is pixel coordinate system and $O'XYZ$ is camera coordinate system. $D(u, v)$ is 3D coordinate of the investigation point (u, v) which is on the pixel coordinate system. Another 4 points are k pixel length away from the investigation point

subdivided each angle into 180 bins, i.e. the bandwidth of each bin is 1 degree. To begin with, we put each normal $\mathbf{n}_f(u, v)$ into corresponding bins and set the first candidate dominant axis $l_1 = (\alpha_1, \beta_1, \gamma_1)$ to the coordinate of largest bin. Then, we search for second candidate axis $l_2 = (\alpha_2, \beta_2, \gamma_2)$ in bins that are 80 to 100 degrees away from l_1. The constraint formula set as follow:

$$\theta_1 < \Theta(\alpha_1, \alpha_2) + \Theta(\beta_1, \beta_2) + \Theta(\gamma_1, \gamma_2) < \theta_2 \tag{5}$$

where $\Theta(a, b) = \cos(a) * \cos(b)$, and $\theta_1 = \pi * 100/180$, $\theta_2 = \pi * 80/180$.
Then, We find l_3 in the region that is in the range 80 to 100 degrees away from both l_1 and l_2. We finally mark normals which is within 5 degrees away from l_1, l_2, l_3 as the candidate dominant axes cluster $\mathbf{V}_p$ (Fig. 3).

Fig. 3 This is an unit sphere in 3D cartesian coordinate system. α is the angle between normal n and axis X, β is the angle between normal n and axis Y, and γ is the angle between normal n and axis Y



We use 3D matrix to store numbers of each bins, the length of each dimension equal to the range number of object angle. Mentioned by the angle form of vectors, we could know these matrix is sparse, so the calculate quantity during statistic can be efficiently reduced.

3.1.3 PCA

As we have mentioned, dominant axes are mutually orthogonal. However, to extract axes only by computing the average of the normals within several bins like Yasutaka et al. [8] can not get the most accurate dominant axes, and the axes they extracted is not perpendicular.

In our method, we choose Principal Components Analysis(PCA) to extract dominant axes. We set all candidate dominant axes V_p and their inverse to the input of PCA, then PCA will extract main directions of these axes. Finally we get three mutually orthogonal feature vectors and set these vectors to be the dominant axes l_1, l_2, l_3.

3.2 Extracting dominant planes

For each 3D point $\mathbf{D}(u, v)$, we can easily get its projection position P_k in an axis l_k by setting camera optical center to be the origin of coordinate. And for a plane with normal equal to axis l_k, the distance d_k between plane and origin of coordinate must equal to projection positions of 3D points, which are located in the plane: $\mathbf{D}_f(u, v) \cdot l_k = d$. At frame f we compute a set of position P_k^f and use 1D mean shift clustering [5] to extract peaks, and we set ω to be the bandwidth of mean shift algorithm. The distance is the coordinate of highest peak (See Fig. 4). Some smaller peaks may also be observed, they represent clusters which containing small number of samples, and they also represent small planes. However we only need dominant plane to implement registration, thus we exclude clusters with fewer than ν samples. The size of ν is corresponding with number of pixel and the bandwidth of the mean shift algorithm.

3.3 Reconstruction

We represent camera pose estimated for a frame a by a 4-to-4 transformation matrix:

$$T_{a,b} = \begin{bmatrix} R & \mathbf{t} \\ 0 & 1 \end{bmatrix} \tag{6}$$

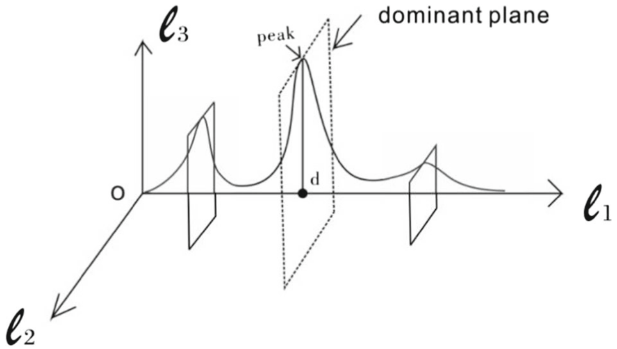


Fig. 4 l_1, l_2, l_3 are dominant axes we extracted. The curve represent point density on l_1. d is the distance between the dominant plane and origin

This maps the camera coordinate at frame a into frame b. We set camera coordinate of first frame to the global coordinate, such that a 3D point $\mathbf{p}_f$ in camera frame f is transferred into the global coordinate $\mathbf{p}_{global}$ via formulate:

$$\mathbf{p}_{global} = T_1 \cdot T_2 \cdot \dots \cdot T_f \cdot \mathbf{p}_f \tag{7}$$

where T_1 is identity matrix. We have acquired the information of three perpendicular dominant planes through the above steps. And in this step we use plane normals, i.e. dominant axes l_k to calculate rotation matrix R from frame g to f:

$$R_f = (l_1^f, l_2^f, l_3^f) \cdot (l_1^g, l_2^g, l_3^g)^{-1} \tag{8}$$

and obtain translate vector $\mathbf{t}$ via computing the offset of plane at adjacent frame and transform it into camera coordinate:

$$\mathbf{t}_f = \sum_k (d_k^f - d_k^g) l_k^g, k = 1, 2, 3 \tag{9}$$

After calculating transformation matrix of each frame, we finally use these matrices to reconstruct 3D point cloud of scenes rendered by OpenGL.

4 Experimental result

We use color camera on mobile phone and structure sensor which is a hand held depth sensor that provides depth maps at a resolution of 640 × 480 pixels. Figure 6a, b show some

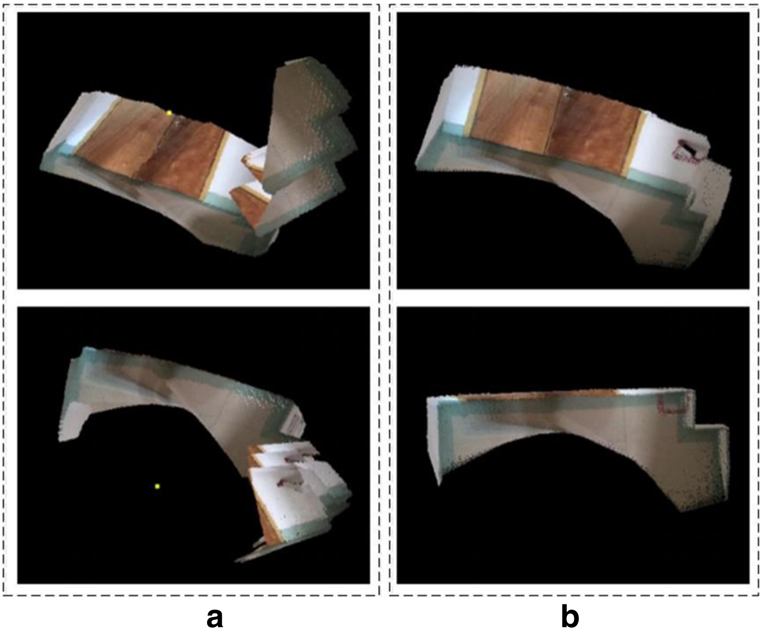


Fig. 5 This figure shows the comparison between traditional point-based method and our dominant plane method in different views. Two images on the left (**a**) are the snapshots reconstructed by point-based method, the upper one is front view and the bottom one is vertical view. Images on the right (**b**) are the result of SLAM visualization reconstructed by our method

color images and depth maps extracted from the sequence. Figure 6c shows snapshots of the SLAM result with our method. .

4.1 Comparison

Figure 5 compares the 3D models reconstructed by our approach and the conventional approach that uses feature-point correspondences. The conventional approach using only points produced serious drift when reconstructing the white wall on the right. While our method maintained accurate registration. The reasons produce this distinctive contrast are as follow:(1) feature-point method can hardly extract keypoints in texture-less regions and regions with repeated patterns. (2) plane correspondences are more stable and robust than point correspondences which is mainly because the high rate of mismatch of keypoint. Note that point method could make up the drawback by using ICP algorithm and RANSAC, but it also brings additional computational costs. Moreover, in some common cases, our method found a good solution while the traditional point-based method failed without extracting any available keypoint.

In Fig. 7 we compared our method with the state-of-the-art, InfiniTAM v2 [11, 16], which supports dense volumes (using an implementation based on the KinectFusion paper published by Newcombe et al. [14]) and sparse volumes (using an implementation based on [11]). The sequence we used in Fig. 7 is different from Fig. 6, and The part surrounded by red bounding boxes are results reconstructed by each method to the same area(corner of wall). Figure 7k is the reconstruction result by traditional point-based method, and we can easily observe that this method is totally failed to reconstruct the corner. Figure 7l is the

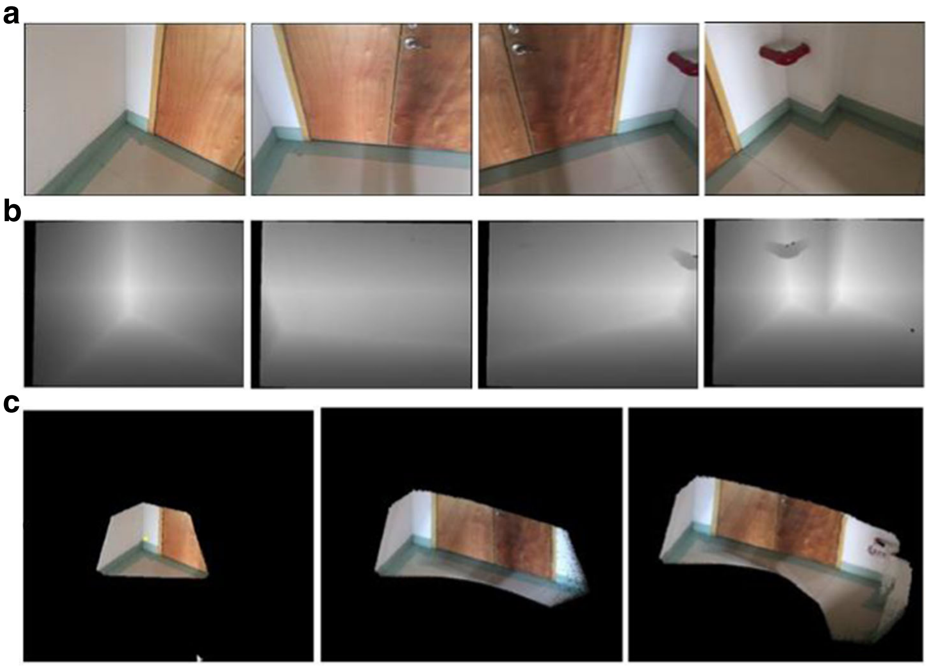
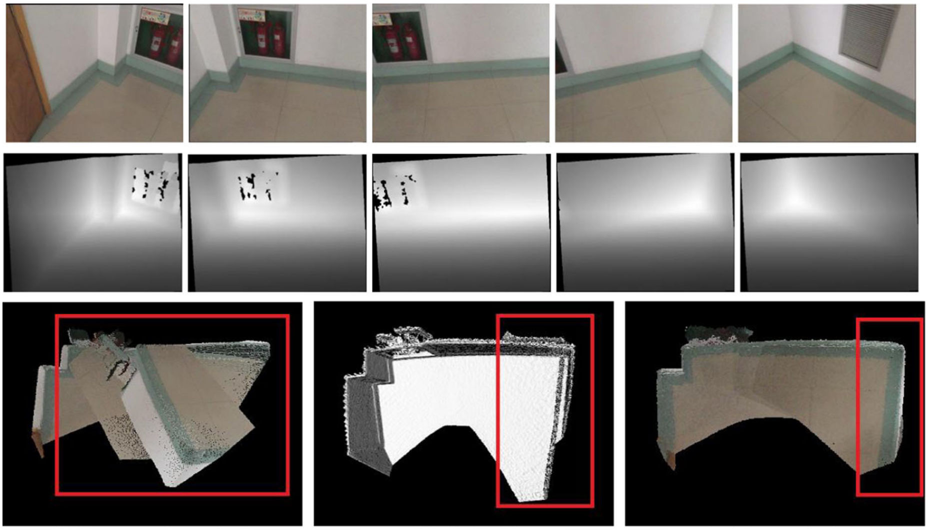


Fig. 6 An example of 3D reconstruction using a hand-held device. **a** Color images and **b** depth maps from the captured sequence. **c** Snapshots of our interactive visualization system



(k) Feature-point method (l) InfiniTAM (m) Our method

Fig. 7 **k**, **l**, **m** are top views of visualization system, reconstructed by infiniTAM, feature-point method and our method. The parts surrounded by red bounding boxes are results reconstructed by each method to the same area(corner of wall)

result of InfiniTAM v2, and (m) is the result of our method. When reconstruct the corner of wall, InfiniTAM produced a drift while our method remain accurate result.

4.2 Results

We reconstructed 3 different indoor scenes. The first one is showed in Fig. 6, which have 5 walls(containing 2 small walls) and floor. Figure 6 shows the second result, it also successfully reconstructed floor and 5 walls including a big white wall. In the third reconstruction result showed in Fig. 8, we reconstructed a more complex scene which contain many objects such as chair, table, carton and besom. The result shows that our method is available for non-planar object as well, and perform good in complex scene.

4.3 Performance

In the Table 1, we summarizes the average processing time for each step of the system over the sequence shown in Table 1. Currently our system runs more than 25 frames per second

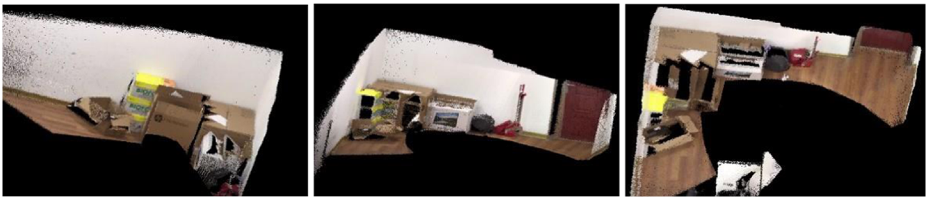


Fig. 8 An example of reconstructing 180 degree of an office by our dominant plane-based method

Table 1 Processing time for each steps (In msec)

Extract dominant axes	13
Extract dominant planes	11
Other (Map Update, Data Copy)	5
Total	29

in the beginning of reconstruction and is faster than InfiniTAM on the same standard PC. Notice that we implemented our method by C++ on a PC with an 3.50 GHz four core CPU and 16GB memory, and we have not used GPU or optimized CPU yet. As the reconstruction goes on, our system will slow down gradually due to ever-increasing number of 3D points that we should rendering. By using faster visualization scheme [10], we could solve the problem and further improve the speed of our system.

5 Conclusion and limitations

We have presented a plane-based method succeeding in reconstructing indoor scenes with less geometry and texture features, which are important and challenging for either image feature based methods and ICP-based methods. The key idea is to involve the Manhattan assumption and propose an end-to-end system which efficiently estimates dominate axes and dominate plans, and reconstruct the camera motions by aligning the axes and plans of consecutive input frames. Since the axes and planes with few freedoms are estimated from the depth input recorded by the sensor, the estimation is very robust to input noise. And as the alignment is performed on global axes and planes, we do not require local correspondences as the image feature point-based methods and ICP-based methods, we handles the scene with less geometry and texture features.

Our method requires the input containing at least two orthogonal dominant axes. Otherwise, we can not solve the ambiguity in 3D camera motion estimation. To make our system more practical and more robust for arbitrary scene styles, we need to combine our method with the existing image feature based methods or ICP-based methods. The combination requires delicate investigation to fully explore the advantages of these methods. Meanwhile, we have not implement bundle adjustment in our method like [6]. Thus we still can not handle the loop closure problem.

Acknowledgments This work is supported by National Nature Science Foundation of China (61671196, 61327902, 61671268, 61727808), Zhejiang Province Nature Science Foundation of China LR17F030006.

References

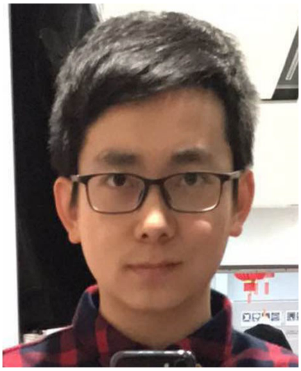
1. Arun KS, Huang TS, Blostein SD (1987) Least-squares fitting of two 3-d point sets. IEEE Trans Pattern Anal Mach Intell 9(5):698–700
2. Besl PJ, Mckay ND (1992) A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256
3. Calonder M, Lepetit V, Strecha C, Fua P (2010) Brief: Binary robust independent elementary features. In: European conference on computer vision, pp 778–792

4. Chen HH (1991) Pose determination from line-to-plane correspondences: existence condition and closed-form solutions. IEEE Trans Pattern Anal Mach Intell 13(6):530–541
5. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
6. Dai A, Nießner M, Zollöfer M, Izadi S, Theobalt C (2017) BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface re-integration. ACM Trans Graph 2017 (TOG) 36(3). https://doi.org/10.1145/3054739
7. Eric W, Grimson L, Lozano-Perez T (1987) Model-based recognition and localization from sparse range or tactile data. Morgan Kaufmann Publishers Inc., Burlington
8. Furukawa Y, Curless B, Seitz SM, Szeliski R (2009) Manhattan-world stereo. In: IEEE conference on computer vision and pattern recognition 2009. CVPR 2009, pp 1422–1429
9. Henry Peter, Krainin Michael, Herbst Evan, Ren Xiaofeng, Fox D (2014) RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. Springer, Berlin
10. Jun HE, Hao D, Xie YQ, Liu BS (2006) Fast improved delaunay triangulation algorithm. Journal of System Simulation 18(11):3055–3057
11. Kahler O, Prisacariu VA, Ren CY, Sun X, Torr PHS, Murray DW (2015) Very high frame rate volumetric integration of depth images on mobile device. IEEE Trans Vis Comput Graph (Proceedings International Symposium on Mixed and Augmented Reality) 21(11):1241–1250
12. Lee TK, Lim S, Lee S, An S (2012) Indoor mapping using planes extracted from noisy rgb-d sensors. In: Ieee/rsj international conference on intelligent robots and systems, pp 1727–1733
13. Lepetit V, Fua P (2006) Keypoint recognition using randomized trees. IEEE Trans Pattern Anal Mach Intell 28(9):1465–79
14. Newcombe RA, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison AJ, Kohli P, Shotton J, Hodges S, Fitzgibbon A (2011) Kinectfusion: real-time dense surface mapping and tracking. In: IEEE ISMAR. IEEE, Piscataway
15. Nistér D, Stewénius H (2007) A minimal solution to the generalised 3-point pose problem. J Math Imaging Vision 27(1):67–79
16. Prisacariu VA, Kahler O, Cheng MM, Ren CY, Valentin J, Torr PHS, Reid ID, Murray DW (2014) A framework for the volumetric integration of depth images. arXiv:1410.0925
17. Ramalingam S, Taguchi Y (2013) A theory of minimal 3d point to 3d plane registration and its generalization. Int J Comput Vis 102(1):73–90
18. Rosten E, Porter R, Drummond T (2010) Faster and better: a machine learning approach to corner detection. IEEE Trans Pattern Anal Mach Intell 32(1):105–119
19. Rublee E, Rabaud V, Konolige K, Bradski G (2011) Orb: An efficient alternative to sift or surf. In: IEEE international conference on computer vision, pp 2564–2571
20. Shotton J, Glocker B, Zach C, Izadi S, Criminisi A, Fitzgibbon A (2013) Scene coordinate regression forests for camera relocalization in rgb-d images. In: IEEE conference on computer vision and pattern recognition, pp 2930–2937
21. Steinbrücker F, Kerl C, Cremers D (2013) Large-scale multi-resolution surface reconstruction from rgb-d sequences. In: IEEE international conference on computer vision, pp 3264–3271
22. Taguchi Y, Jian YD, Ramalingam S, Feng C (2013) Point-plane slam for hand-held 3d sensors. In: IEEE international conference on robotics and automation, pp 5182–5189
23. Trevor AJ, Rogers J, Christensen H (2012) Planar surface slam with 3d and 2d sensors. In: IEEE international conference on robotics and automation, pp 3041–3048
24. Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. IEEE Trans Pattern Anal Mach Intell 13(4):376–380
25. Whelan T, Johannsson H, Kaess M, Leonard JJ (2013) Robust real-time visual odometry for dense rgb-d mapping. In: IEEE international conference on robotics and automation, pp 5724–5731
26. Yan C, Zhang Y, Dai F, Xi W (2014) Parallel deblocking filter for hevc on many-core processor. Electron Lett 50(5):367–368
27. Yan C, Zhang Y, Jizheng X, Dai F (2014) Efficient parallel framework for hevc motion estimation on many-core processors. IEEE Trans Circuits Syst Video Technol 24(12):2077–2089
28. Yan C, Zhang Y, Xu J, Dai F, Li L, Dai Q, Wu F (2014) A highly parallel framework for hevc coding unit partitioning tree decision on many-core processors. IEEE Signal Process Lett 21(5):573–576
29. Yan C, Xie H, Liu X, Yin J, Zhang Y, Dai Q (2017) Effective uyghur language text detection in complex? background images for traffic prompt identification. https://doi.org/10.1109/TITS.2017.2749977
30. Yan C, Xie H, Yang D, Yin J, Zhang Y, Dai Q (2017) Supervised hash coding with deep neural network for environment perception of intelligent vehicles. IEEE Trans Intell Transp Syst
31. Zhang Z, Faugeras OD (1991) Determining motion from 3d line segment matches: a comparative study. Image Vis Comput 9(1):10–19

Zunjie Zhu is a graduate student in Hangzhou Dianzi University. His research interests include Machine Learning and Computer Vision.

Feng Xu received Ph.D. from Department of Automation in Tsinghua University and B.S. from Department of Physics in Tsinghua University. He is currently an assistant professor in School of Software, Tsinghua University. His research interests include Face Modeling and Animation, Performance Capture, 3D Vision and Graphics.

Chenggang Yan received the PhD degree in Computer Science from institute of computing technology, chinese academy of science. He is currently a professor working in Hangzhou Dianzi University. His research interests include pattern recognition, intelligent system and so on.



Ning Li is a undergraduate student in Hangzhou Dianzi University. His research interests include Computer Vision.

Bingjian Gong is a undergraduate student in Hangzhou Dianzi University. His research interests include Computer Vision.

Yongdong Zhang received the PhD degree in School of Electronic Information Engineering, Tianjin University, chinese academy of science. He is currently a professor working in Institute of Computing Technology, CAS. His research interests include Multimedia Analysis and Retrieval, Social Multimedia Analysis, Computer Vision, Multimedia Content Security, Video Coding and Streaming Media.

Qionghai Dai received the PhD degree in Northeastern University. He is currently a professor working in Department of Automation at Tsinghua University. His research interests include Computational Photography, Computational Imaging, Computer Vision, 3D Video.