PoseFusion: Dense RGB-D SLAM in Dynamic Human Environments

Tianwei Zhang, Yoshihiko Nakamura

PoseFusion: Dense RGB-D SLAM in Dynamic Human Environments

Tianwei Zhang and Yoshihiko Nakamura

Abstract RGB-D Simultaneous Localization and Mapping (SLAM) in indoor environments is a hot topic in computer vision and robotics communities, and the dynamic environment is a remaining problem. Dynamic environments, which are often caused by dynamic humans in indoor environments, usually lead to the camera pose tracking method failure, feature association error or loop closure failure. In this paper, we propose a robust dense RGB-D SLAM method which efficiently detects humans and fast reconstructs the static backgrounds in the dynamic human environments. By using the deep learning-based human body detection method, we first quickly recognize the human body joints in the current RGB frame, even when the body is occluded. We then apply graph-based segmentation on the 3D point clouds, which separates the detected moving humans from the static environments. Finally, the left static environment is aligned with a state-of-the-art frame-to-model scheme. Experimental results on common RGB-D SLAM benchmark show that the proposed method achieves outstanding performance in dynamic environments. Moreover, it is even comparable to the performance of the related state-of-the-art methods in static environments.

1 Introduction

The task of Simultaneous Localization and Mapping (SLAM) is to estimate the visual sensor's pose and reconstructed the three-dimensional (3D) static backgrounds at the same time. Except for some special applications, most of the SLAM approaches assume that the robot works in static environments. The dynamic environment is a challenging problem for visual SLAM since that the foreground dynamic objects occlude static background features and then result in failing or wrong features corresponding. Humans are often considered as moving obstacles in indoor environments. Particularly, they may occlude visual features during the human-robot interactions.

Generally, a SLAM framework can be divided into a front-end and a back-end part. The task of the front-end is to extract environment features, which are used for image or point clouds alignment. The back-end part takes care of maintaining the graph structure, saving keyframe information, loop finding and dealing with camera pose drift. In [1], Saputra et al. survey the dynamic SLAM methods by the year

Tianwei Zhang and Yoshihiko Nakamura are from Department of Mechano-Informatics, School of Information Science and Technology, the University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan. e-mail: `[zhang,nakamura]@ynl.t.u-tokyo.ac.jp`

2016. They classify the existed dynamic SLAM methods to three types based on different application and outputs: Robust Visual SLAM, Dynamic Object Segmentation, and Joint Motion Segmentation and Reconstruction. Follow this taxonomy, we propose a novel dense RGB-D SLAM method for dynamic humans environments based on human motion segmentation and static environment reconstruction.

Our method works in multiple humans dynamic environment, it efficiently detects humans and fast reconstructs the static environments. We detect moving objects by integrating the OpenPose [2] into our front-end algorithm, which is an advanced deep learning based human detection method for a 2D color image. Dynamic point clouds are then removed by using the Min-Cut Segmentation [3]. Finally, we input the static environment point clouds to a state-of-the-art dense RGB-D SLAM framework, ElasticFusion [4] for static environment reconstruction. As a combination of OpenPose and ElasticFusion, our method is named as PoseFusion, which indicate that this SLAM framework applies human pose detection and frame-to-module schemes. Our method is tested on the well known Freiburg RGB-D SLAM dataset dynamic serials [5], and it achieved the smallest camera trajectory error compared to other state-of-the-art dynamic SLAM methods.

2 Related Works

2.1 Dynamic SLAM Methods

In dynamic SLAM survey [1], Saputra et al. divide the exited dynamic SLAM and visual odometry by their fundamental techniques, such as background/foreground initialization, deep learning, optical flow, ego-motion constraints, geometry constraints, and feature based. In this section, we only introduce recent works for RGB-D dense reconstruction, which are similar to our setting.

1. EF [4] and Kintinous [6] are extended from KinectFusion [7], which is an early dense RGB-D reconstruction framework. [7], [4], [6] perform great in static environments. They adopt a module maintaining scheme. Different from the others, EF is a state-of-the-art method designed for static environments within slightly dynamic scenes. It can handle small-scale environment changing since it benefits from the deformation graph based non-rigid module fusion. When a slightly changed scene occurs, EF can fuse it into the saved key-frame modules and ignore small scene changing.
2. Co-Fusion [8] (CF) is a state-of-the-art approach for tracking and reconstructing multiple moving objects using EF framework. However, CF has to first reconstruct the map in a static environment, and then they enable the dynamic object detection and tracking abilities within that reconstructed map.
3. Jaimez et al. [9] proposed an odometry method named The joint visual odometry and scene flow (SF). SF localizes the moving camera in dynamic scenes by segmenting intensity point clouds into a number of clusters (called super-voxels),

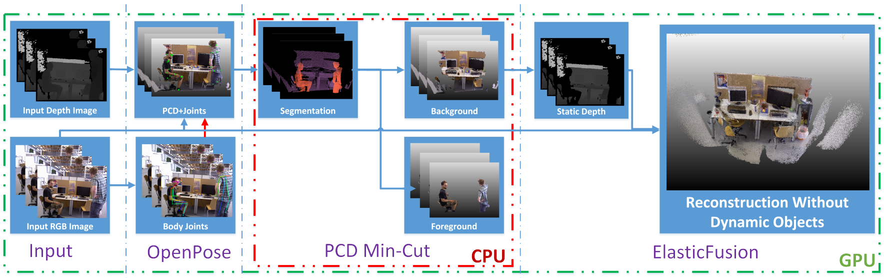


Fig. 1 PoseFusion Flowchart: Firstly input RGB images to OpenPose [2] to detect human joints. Then we project joint points to the PCD and followed by foreground removal. Finally, static backgrounds are feed into reconstruction.

and then, the clusters are divide into moving foregrounds and static backgrounds. Finally, the static background point clouds are feed into camera pose tracking.

4. Scona et al. proposed the staticFusion in [10], which extends the result of SF to environment reconstruction by combining the background segmenting and EF's dense environment reconstruction. JF achieved robust and fast dynamic segmentation and static reconstruction.

These above three SLAM methods are advanced in light dynamic environments. Same to CF and JF, the proposed method is also based on EF. Our idea is similar to SF, that is to decouple the motion of camera from the motion of moving humans. The difference is that SF tries to separate the foreground from the background by comparing the moving speeds of all segments, while we separate moving human segments by using OpenPose.

2.2 *The Human Detection Methods*

OpenPose [2] estimates the body joints as feature points of human beings such as wrists and shoulders in real time from a single RGB image. It is to estimate each joint position by deriving Part Confidence Maps using trained Convolutional Neural Networks. Moreover, [2] can estimate the vector fields between connected two joints, which make it robust to the occlusions no mater form self body nor multiple humans' bodies. OpenPose provide us the probable positions of human body joints in the 2D RGB image, thus we are able to quickly label the position of humans in 3D Point Clouds Data (PCD).

3 Multiple Humans Detection in Dynamic Environment

The Flowchart of proposed PoseFusion is illustrated in Fig. 1. PoseFusion take the RGB and Depth image pair as input. The RGB image is first used for body joints estimation using OpenPose, which tells the likelihood of the human joint positions on the input image. When the input image f is given, the feature map is extracted via CNN network and then output data:

$$[\boldsymbol{h} \times \boldsymbol{w}]_f \tag{1}$$

in which $\boldsymbol{h}$ is the detected human bodies, at a maximum 15, which means OpenPose can detect at maximum 15 humans within one frame. The $\boldsymbol{w}$ is the list of estimated joints, it presents a probability map on the RGB image plane which indicates the existence likelihood of at maximum 18 human joint. In matrix 1, each element has three components: u, v, p. They are the image pixel coordinates (u, v) and the existence likelihood ($p \in (0,1]$) of the human body joint.

The estimated joint points are converted from 2D to 3D using the pinhole camera model and then, they are used for labeling humans' positions in the PCD. This processing is the red arrow in Fig.1, and note that the green points in the point clouds stand for the projected joint positions. Then, the PCD with these green joints are inputted to Min-Cut point foreground segmentation.

Min-Cut [3] is a Graph-Cut [11] based method for segmenting objects in point clouds. Graph-cut treats every single point as a vertex and vertices are connected with their neighbors by edges. Given some vertices as foreground priors, it cuts the foreground object out of the background points by computing the weights of the edges. T apply Min-Cut, we use the human joints from Equation. 1 as foreground prior, we assign two edge weights in min-cut: the edge smooth cost C and background penalty P.

$$C = e^{-(\frac{len}{\sigma})^2} \tag{2}$$

in which len is the length of the edge, obviously, the father away the vertices are, the more is the probability the edge will be cut. The σ is a user defined parameter.

The background penalty is to weight the points connected with the foreground points. In which, for a joint point $J(J_x, J_y, J_z)$, we set an input parameter rad as the maximum horizontal (X-Y plane) radius of foreground objects. Then, for a neighbor point (x, y, z) of J, its background penalty is:

$$P = \frac{\sqrt{(x-J_x)^2 + (y-J_y)^2}}{r} \tag{3}$$

As the pose points are labeled on the human body, we set the foreground rad as 20 cm, σ as 0.25 which draws a good segmentation performance. After Min-Cut, the background segments are converted to static depth images, which are the inputs for the following static environment reconstruction, together with the original RGB image. Finally, a clean static environment reconstruction is achieved through frame-to-module map fusions.

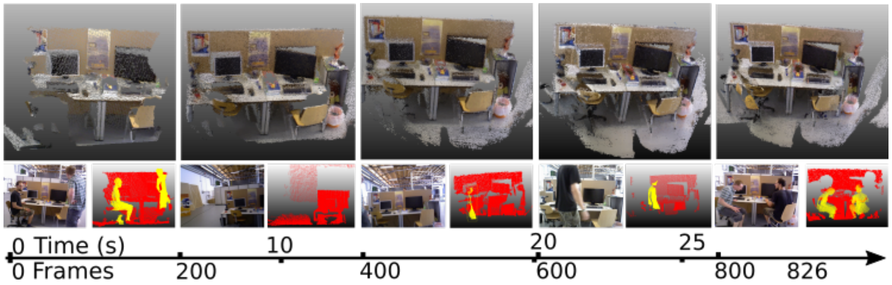


Fig. 2 Experiment Setting: RGB-D PCD frames are played as a video. The first row shows the reconstructed maps, the second row shows the respectively RGB input and foreground-background segmentations. As the frame grows, the first row reconstruction is gradually completed, the RGB viewpoint moves as the camera moves, the point clouds segmentation is also changing. There is no foreground cluster in the 4th image, the second row, since the guy walks out of the scene.

4 Experiments Setup and Results

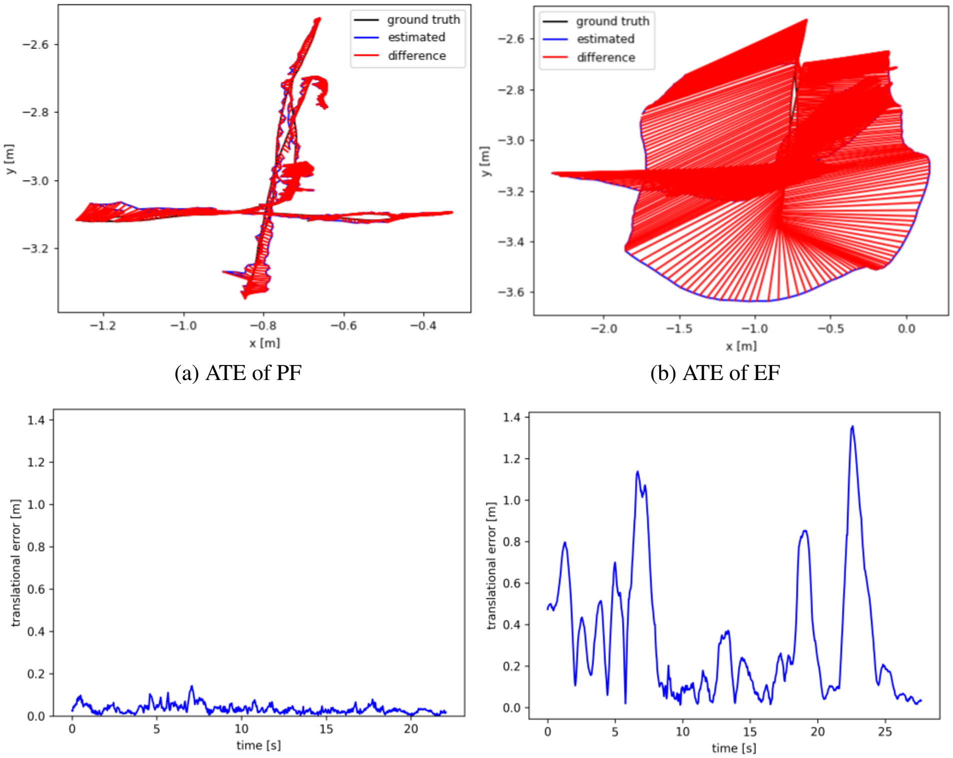
Our method is evaluated with the Freiburg SLAM benchmark, which contains a dynamic SLAM series (fr3 series). This benchmark is wildly used for dynamic scenes comparison. The experiment is shown in Fig.2, which take fr3/walk_xyz (in which the camera moved by a man walks in x, y, and z translations) as an example. Each fr3 dataset contains several hundred color (RGB) and Depth (D) image pairs, which can be transformed into RGB-D PCD frames. For instance, there are 827 RGB-D frames in fr3/walk_xyz, these RGB-D PCD frames can be played as a video.

In Fig.2, the first row shows the reconstructed maps, and the second row shows the respective input RGB images and their foreground-background segmentation results. As the frame number grows, the first row reconstruction is gradually completed, the RGB viewpoint moves as the camera moves, the point clouds segmentation is also changing. In the fourth image of the second row, there is no foreground cluster since the people walk out of the camera view. These experiment setting intuitively indicate the dynamic ability of SLAM methods. One can compare the moving object detection and removal abilities throw checking when and how many moving object ghost shadows are integrated into the reconstructed scenes in the first row. One comparison with EF and PF is given in Fig.4.

We compare our PoseFusion (PF) method with three state-of-the-art dynamic SLAM methods: Scene Flow (SF) method from [9], ElasticFusion (EF) [4] and Co-

Table 1 Translate ATE RMSE (cm)

Dynamic DataSet [5]	SF	EF	CF	PF
fr3/sit_static	8.67	3.18	**2.93**	3.18
fr3/sit_xyz	7.65	**0.90**	1.49	**0.90**
fr3/sit_halfsphere	50.21	30.18	27.58	**2.31**
fr3/walk_static	81.07	25.13	18.77	**6.87**
fr3/walk_xyz	65.31	67.78	37.11	**3.21**
fr3/walk_halfsphere	79.97	47.90	21.23	**4.65**



(a) ATE of PF

(b) ATE of EF

(c) RPE of PF

(d) RPE of EF

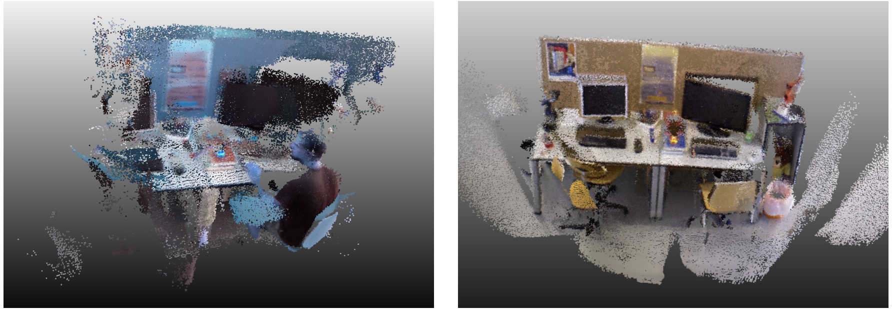
Fig. 3 Evaluation of the Proposed PoseFusion compare to ElasticFusion, both are the result of fr3/walk_xyz. (a) and (c) are absolute trajectory error (ATE) and the relative pose error (RPE) of PF, while (b), (d) are ATE and RPE of EF. PF achieves very small trajectory error, average 3.21 cm (short red line segments in (a)), while EF gets big ATE, long red line segments in (b). (c) and (d) indicate PF achieves about ten times smaller RPE than EF.

Table 2 Translate RPE RMSE (cm/s)

Dynamic DataSet [5]	SF	EF	CF	PF
fr3/sit_static	2.71	1.12	**0.81**	1.19
fr3/sit_xyz	9.61	**3.90**	4.93	3.97
fr3/sit_halfsphere	41.10	29.19	23.80	**5.19**
fr3/walk_static	27.77	20.33	18.79	**3.77**
fr3/walk_xyz	35.32	21.78	37.11	**2.11**
fr3/walk_halfsphere	69.74	77.91	31.32	**4.50**

Fusion (CF) [8]. All of them are implemented from their open source repositories. StaticFusion (JF) [10] is the most recent method and its code is not opened yet.

To evaluate these four SLAM methods, we compare their absolute trajectory error (ATE) and relative pose error (RPE). ATE is well-suited for measuring the per-



(a) ElasticFusion Reconstruction

(b) PoseFusion Reconstruction

Fig. 4 EF and PF Scene Reconstruction of fr3/walk_xyz. (a) is the result of EF, 67.78 cm absolute trajectory error makes an obvious wrong reconstruction. The wall, the table, and the desktops are not aligned and multiple people shadow corrupt the final map. While (b) is reconstructed by PF, which is well aligned with only 3.21 cm error. The dynamic performance of proposed is as good as the EF's static performance. Note that, the left chair's ghost shadow in PF reconstruction is not a wrong alignment, it is because the people moved that chair to that place for several seconds, which can be observed in our supplement video.

formance of visual SLAM systems. In contrast, the RPE is well-suited for measuring the drift of a visual odometry system, for example, the drift per second. The ATE directly measures the difference between points of the ground truth and the estimated trajectory. The RPE computes the error in the relative motion between pairs of timestamps.

Table 1 and Table 2 show the root-mean-square error (RMSE) of translate ATE (cm) and RPE (cm/s). All of these six datasets are dynamic scenes, but the first three: fr3/sit_static; sit_xyz and sit_halfsphere are low dynamic scenes, while the other three are high dynamic scenes. From these two Tables, one can obviously find that our PF method achieved the smallest estimation errors in highly dynamic situations. In the light dynamic scenes, EF achieved the best performances, since it is designed for static environments.

Fig. 3 and Fig. 4 show the comparison between PF and EF run on the fr3/walk_xyz dataset. Inside the former figure set, (a) and (c) are ATE and RPE of PF, while (b), (d) are ATE and RPE of EF. PF achieves very small trajectory error, average 3.21 cm (difference of the same time-stamp are shown as the short red line segments in (a)), while EF gets big ATE, which can be seen as the long red line segments in (b). (c) and (d) indicate that PF achieves around ten times smaller RPE than EF.

Fig. 4 intuitively indicate the performances of EF and PF. Big trajectory error of EF results in obvious wrong map reconstruction. The wall, the table, and the desktops are not aligned, as well as people shadows corrupt the final map of (a). While (b) is reconstructed by PF, which is well aligned. The dynamic performance of PF is as good as the EF's static performance. Note that, the left chair's ghost shadows in PF reconstruction is not wrong alignments, they are reconstructed since that the people moved that chair to that places for several seconds, which can be observed in our supplement video.

5 Conclusions

In this paper, we propose a novel dense RGB reconstruction approach for the dynamic human environments. We combine the advanced deep learning based human body detection method and static dense SLAM approach to deal with the dynamic environment problem using dense RGB-D data. We compare the proposed method with EF[4], CF[8], and SF[9]. Our approach significantly decreases the alignment error and archive 3.21 cm average ATE in TUM benchmark, while EF results in a 67.78 cm ATE. Moreover, We not only propose one approach for a particular dynamic environment setting but also provide a framework for solving SLAM "Type II Error", which means the front-end part doesn't "reject" the wrong features of the moving objects so that the moving shadows are integrated to key-frames on the back-end, thus breakdown loop closing. According to the proposed, one can integrate advanced learning-based object recognition methods to handle dynamic environments. For instance, involving vehicle recognition methods for high way scenes, and including semantic labeling methods for service robots in house and office scenes.

6 Acknowledgement

This work was supported by JSPS Grants-in-Aid for Scientific Research (A) 17H06291.

References

1. Saputra M R U, Markham A, Trigoni N. Visual SLAM and Structure from Motion in Dynamic Environments: A Survey[J]. ACM Computing Surveys (CSUR), 2018, 51(2): 37.
2. Cao Z, Simon T, Wei S E, et al. "Realtime multi-person 2d pose estimation using part affinity fields"[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, 1(2): 7.
3. Golovinskiy A, Funkhouser T. "Min-cut based segmentation of point clouds"[C]. Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. 2009.
4. T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison, "ElasticFusion: Dense SLAM without a pose graph", Robotics: Science and Systems (RSS), 2015.
5. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems", in IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2012.
6. Whelan T, Kaess M, Fallon M, et al. "Kintinuous: Spatially extended kinectfusion"[J]. CSAIL Tech. Rep., 2012.
7. Newcombe R A, Izadi S, Hilliges O, et al. "KinectFusion: Real-time dense surface mapping and tracking"[C], Mixed and augmented reality (ISMAR), 10th IEEE international symposium on. 2011.
8. M. Rnz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects", in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2017.
9. M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from RGB-D cameras based on geometric clustering", in IEEE Intl. Conf. on Robotics and Automation (ICRA), 2017.
10. Scona R, Jaimez M, Petillot Y R, et al. StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments[C]. Institute of Electrical and Electronics Engineers, 2018.
11. Y. Boykov and G. Funka-Lea. Graph cuts and efficient n-d image segmentation. IJCV, 70(2):109-131, 2006.