2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Dense RGB-D SLAM for Humanoid Robots in the Dynamic Humans Environment

Tianwei Zhang[1], Emiko Uchiyama[1], and Yoshihiko Nakamura[1]

***Abstract*— These two problems block the SLAM method applications for humanoids. On the one hand, humans are often considered as moving obstacles or moving targets in the humanoids working spaces, which result the dynamic environment problem. On the other hand, the disturbances caused by the executions of biped locomotion and the environment structure discontinuity caused by the falling down case make big challenge for SLAM approaches. In this paper, we propose a robust dense RGB-D environment reconstruction method for humanoids working in dynamic humans space. The proposed approach efficiently detects humans and fast reconstructs the static environments through deep learning-based human body detection, and then implement a graph-based segmentation on the RGB-D point clouds, which separates detected moving humans from the static environment. Finally, the separated static environments are aligned with using state-of-the-art frame-to-model scheme. Experimental results on both public benchmark and a newly developed HRP-4 humanoids SLAM dataset indicate that the proposed approach achieves outstanding performance in full dynamic environments.**

I. Introduction

Biped walking is the biggest property of a humanoid robot. However, the disturbance due to the dynamics of biped walking and the visual information's discontinuity that may arise from the falling down situations make it hard to implement a visual Simultaneous Localization and Mapping (SLAM) solution on a humanoid robot. On the other hand, the dynamic human objects, which are often considered as interaction targets, usually occlude the environment features, which leads to visual odometry and localization failures.

In this paper, we propose a robust dense RGB-D SLAM solution for humanoids in multiple humans dynamic environments. By which, a humanoid robot can efficiently detects humans and fast reconstructs the static environments. We develop our previous approach PoseFusion (PF) [1] into humanoid platform, named PoseFusion for Humanoids (PFH) , which is able to handle the dynamic motions come from both the dynamic environments and as well the humanoid itself. As a combination of OpenPose [2] and ElasticFusion (EF), PF is a dense RGB-D SLAM framework for dynamic humans environment applies human pose detection in the frame-to-model scheme. To handle the special dynamic motions of humanoids, PFH adjust the RGB frame tracking method to handle camera shaking during biped walking, then develop long term key-frame strategy to reduce the moving objects' effect on landmarks, so as to enhance the performance of random fern loop closing method [3] to deal with the humanoids falling down situations.

[1] Department of Mechano-Informatics, School of Information Science and Technology, the University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan.

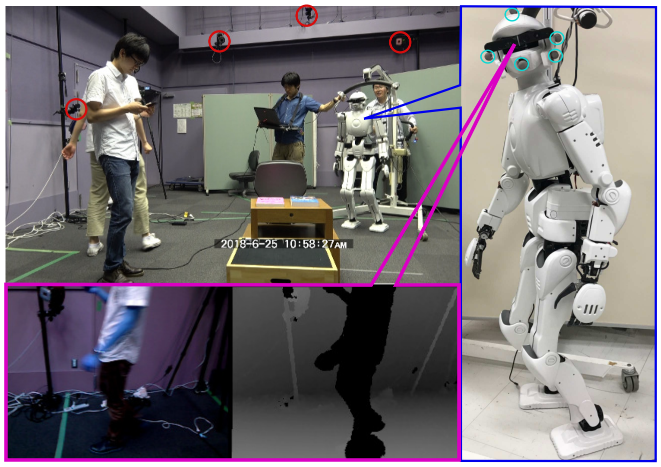


Fig. 1: HRP-4 humanoid dynamic SLAM scene. Cyan circles and red circles show the markers and cameras of motion capture system. These five markers are used to estimate the robot head mounted camera's trajectory. the purple subscreens are the RGB and Depth images in the robot's view.

Our method is tested on the well known Freiburg RGB-D SLAM dataset dynamic serials [4], and it achieved smallest camera trajectory error compared to other state-of-the-art dynamic SLAM methods. We then build up a dynamic RGB-D SLAM HrpSlam database using a humanoid robot. The proposed method also achieved promising performance.

II. Previous Works

A. SLAM for Humanoids

The first successful implementation of real-time 3D SLAM system for a humanoid is Oliver *et al.* . [5]. They demonstrate a loop closing in the indoor environment using HRP-2 robot by tightly coupling the pattern generator, robot odometry, and inertial sensing into a standard extended Kalman filter framework. A recent work of Scona [6] fusing the Valkyrie humanoid motion information into EF [7] framework, which achieves local loop closing in slightly dynamic environments. However, both of these two humanoid SLAM works deal with the humanoids' dynamic motion and build spare environment maps. They cannot deal full dynamic unknown environment which contains both moving objects and irregular robot motions. Moreover, they do not build dense RGB-D maps, which are meaningful for the humanoids working in indoor spaces.

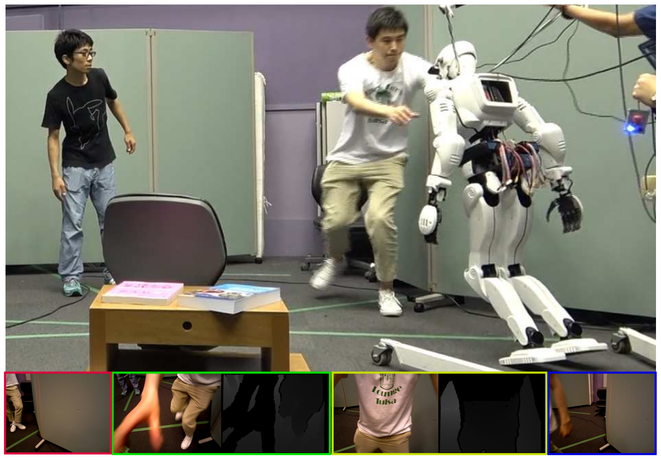


Fig. 2: Falling down situations in our humanoid dynamic SLAM experiments. The robot falls down several times within each loop. The second row shows four robot views in different color bounding box during this 4 sec long falling down case. This situation is hard for SLAM, since that, firstly, sudden falling motion leads to camera tracking, secondly, dynamic object brings a full occlusion, see the RGB and Depth inputs in the green and yellow rectangles. Thirdly, long-term discontinuity, see the difference between the red and blue rectangles.

B. Dynamic SLAM Methods

1) EF [7] and Kintinous [8] are extended from KinectFusion [9], which is an early dense RGB-D reconstruction framework. [9], [7], [8] 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 the 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 [10] (CF) is a state-of-the-art approach for tracking and reconstructing multiple moving objects using EF framework. However, CF and EF can only deal with moving obstacles after obtaining the static background module. The front-end camera tracking method in this work cannot cope with moving objects. The reason is that the dynamic environment reconstruction abilities are implemented in the back-end strategies. To be specific, at first 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.* . proposed an odometry method deal with moving objects using visual odometry and scene flow (JF) [11]. More recent, Scona and Jaimez combined JF and EF and proposed StaticFusion (SF) [12]. JF and SF localize the moving camera in dynamic scenes by segmenting intensity point clouds into a number of clusters (named super-voxels), and then, the clusters are divide into moving foregrounds and static backgrounds. Finally, the static background point clouds are feed into camera pose tracking and EF SLAM framework. They can handle both obstacles and humans if moving pixels are less than 50% per frame.

These above methods are advanced in dynamic environments. Our previous dynamic SLAM work [13] is designed for laser sensor SLAM. The idea is similar to JF, [13] tries to acquire the difference between continuous point clouds and then describe the moving objects. As the frame rate and point clouds density are quite different from laser scanners to RGB-D cameras, PF [1] is proposed to find the camera transformation from the dynamic humans environment using a deep learning based human joints detection approach.

III. PoseFusion for Humanoid Dynamic SLAM Method

Humanoids locomotion leads to unstable camera motion and the possible image frame discontinuity. To handle the special dynamic motions of humanoids, we adjust the RGB frame tracking method to handle camera shaking during biped walking. We then developed long term key-frame strategy to reduce the moving objects' effect on landmarks, so as to enhance the performance of random fern loop closing method [3] to deal with the humanoids falling down situations.

A. Remove the Image Blur From Robot Motion Shaking

Estimate and remove image blur is a classical research field in computer vision. Pertuz *et al.* compares the performance of 36 kinds of focus measure operators, in [14], discusses their performances of Shape-from-focus. These techniques are adopted by the sensor makers to acquire clear and sharp images. However, in our cases, the on-board RGB-D camera (different from the visual SLAM of human holding camera) suffers from both of the biped motion and the fast dynamic object motion. See Fig. 4, such kinds of image blur are inevitable.

To avoid the effect of image blurs, in PFH, we estimate the blur score for each RGB input image and remove the image pairs with serious motion distortion. We adopt the variation of Laplacian method form Pech-Pacheco *et al.* [15]. This method is to describe the fast changing boundaries areas of an image using the Laplace operator. As the variations of image blur areas are smaller than clear boundaries, the image blur level can be reflected by the image Laplace variations. These operations are done in three steps:

1 Grayscale the input RGB image
2 Filter the grayscale channel with 3×3 Laplace operator
3 Compute the variation score of step 2
4 If the score is smaller than the Threshold, this RGB-D image pair is removed

The Threshold in step 4 is experimentally adjusted. It is highly depends on the humanoid locomotion situations.

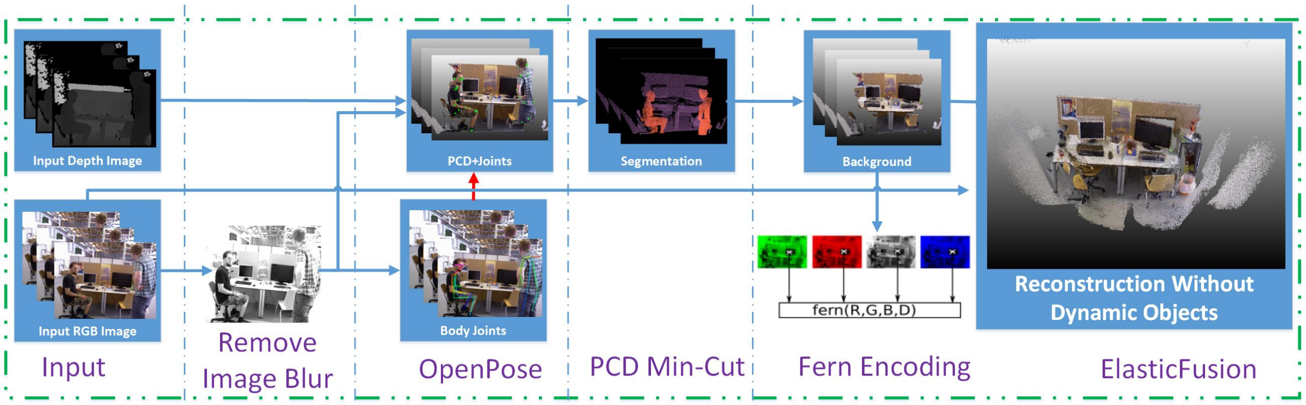
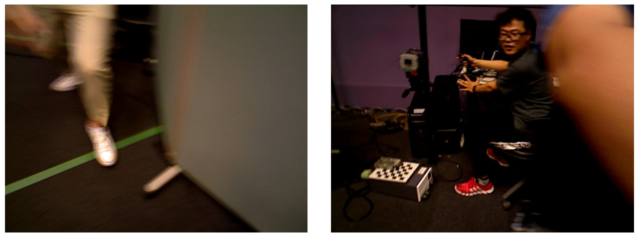


Fig. 3: Dynamic Humans Detection and Removal: PFH firstly input RGB images to OpenPose to detect human joints. Then we project joint points to the PCD and followed by foreground removal. Finally, static backgrounds are reconstructed.



(a) Image Blur Resulted From Humanoid Falling Down (b) Image Blur Resulted From Fast Dynamic Object Motion

Fig. 4: Image Blurs from the robot biped locomotion and the fast dynamic object motion.

B. Multiple Humans Detection in Dynamic Environment

The Flowchart of proposed PFH is illustrated in Fig. 3. PFH 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.3, 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 [16] is a Graph-Cut [17] 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 r 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 r 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.

C. Enhanced Random Fern Method for Dynamic Environment SLAM

In EF, to find a global loop, the Fern-based frame encoding takes an input RGB and Depth pair to generate a small code blocks. These blocks are generated on each fern using simple binary feature tests evaluated at randomized, but fixed, image locations [3]. To recover the camera pose, in random fern's harvest mode, the most similar keyframes are retrieved from the saved fern list, and the corresponding camera poses are used for camera relocation. This keyframe encoding strategy doesn't considering the dynamic objects. To adjust random fern encoding to dynamic environments, we constraint the fern encoding:

See Fig.5, after Min-cut, the background and foreground PCDs are extracted. The ferns are generated using the R, G, B and D channels on the random initialized positions. Obviously, if the fern codes are generated from the dynamic object surfaces, the keyframe won't be recognized when the objects move away. Therefore, we prefer to generate ferns when there is no dynamic object current view. This is done by Equation 1, the value of $\boldsymbol{h}$ indicate how many people are there in frame f.

These strategies improve the camera relocation performance after a falling down and enhance the loop findings.

IV. HrpSlam Dynamic SLAM Datasets Development

Existed Dense RGB-D SLAM researches are mostly working for static indoor environments, so as the benchmarks. The famous RGB-D SLAM dataset, such as [4] and [18] include a few slight dynamic situations. Moreover, to the best of our knowledge, there is no public humanoids dynamic SLAM benchmark. Therefore, to evaluate the proposed dynamic SLAM approach, we build up a full dynamic RGB-D datasets for humanoids. "Full dynamic" means this dataset contains both environment dynamics caused by dynamic objects (human motions and obstacles movings) and the robot's dynamic motions (camera sharking and robot falling down). In these datasets, we tele-control a HRP-4 humanoids robot using a joystick, the robot execute the SLAM algorithm using a Asus Xtion PRO LIVE RGB-D sensor. The sensor is connected with a laptop which has a 4-core Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz, 15GiB System memory, and a Nvidia GeForce GTX 765M GPU.

We control the humanoid robot walks into a motion capture room with multiple moving humans and collect two big SLAM dataset: HRPSlam1 (14 min) and HRPSlam2 (13 min). In HRPSlam1, the robot walks around a chair and a table and try to reconstruct these obstacles within one loop. In HrpSlam2, HRP-4 walks along the walls of the room and trying to reconstructed the whole room with one loop closure.

Fig. 1 and Fig. 2 shows the HRPSlam scenes. The cyan and red circles in Fig. 1 indicate the markers and cameras of motion capture system. These five markers are used to estimate the robot head mounted camera's trajectory. the purple sub-screens are the RGB and Depth images in the robot's view.

The ground truths are acquired from motion capture system. The motion capture uses 16 cameras (Raptor series, by Motion analysis), and processed (interpolated missing markers) using Cortex by Motion Analysis. The sampling rate was 200 FPS, as shown in Fig. 6.

Falling down is the biggest problem for biped walking robots, meanwhile, a humanoid SLAM solution should have the ability to deal with falling motions, which brings sudden interruption of camera odometry and enormous pixel distortions. To test and evaluate this ability, the HRP-4 robot falls down several times in each HRPSlam dataset. One falling case is shown in Fig. 2, it occurs at time 10'19" – 10'23" in HRPSlam2. These falling situations are challenging for SLAM algorithms since the camera's sudden motion, feature less position (occurs in front of a flat wall, see the red bound image) and the full occlusion of the moving object, see the green and yellow rectangles which indicate the RGB and Depth input during the occlusions. Furthermore, this 4 second long time discontinues make hard problem for camera re-locating, watch the different between the red and blue bounding images.

These database will be available under the Creative Commons Attribution License (CC-BY 3.0) in the project web page [19].

V. Experimental Results and Evaluations

We compare our method with three state-of-the-art dynamic SLAM methods: Scene Flow (JF) method from [5], ElasticFusion (EF) [2] and Co-Fusion (CF) [4]. All of them are implemented from their open source repositories. 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 performance 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 I reflect the number of removed blurred images in HRPSlam and TUM fr3 datasets. The images in TUM fr3 are acquired by human holding camera, which is stabler than a biped walking humanoid, the removed five images are all resulted from dynamic humans motion distortion. More images are removed as blurred in HRPSlam datasets since the on-board camera is effected by the robot shaking and falling down. In HRPSlam2, the blurred image number and percentage are higher than HRPSlma1, because of that, the HRP-4 falls down five times (it falls twice in HRPSlam1).

Fig. 5 shows the experiment result of fr3/walking from TUM bench-mark [4]. The first row shows the reconstructed

TABLE I: Removed Blurred Images Numbers

Dynamic DataSet	HRPSlam1	HRPSlam2	fr3/w_xyz	fr3/w_hsphere
Removed Images	781	1231	2	3
Removed Percent	3.06	5.21	0.24	0.25

TABLE II: Translate ATE RMSE (m)

Dynamic DataSet	JF	EF	CF	PFH
HRPSlam1	0.50	2.31	1.38	**0.12**
HRPSlam2	0.31	4.90	1.12	**0.09**
fr3/walk_xyz	0.65	0.67	0.37	**0.03**
fr3/walk_halfsphere	0.80	0.49	0.21	**0.04**

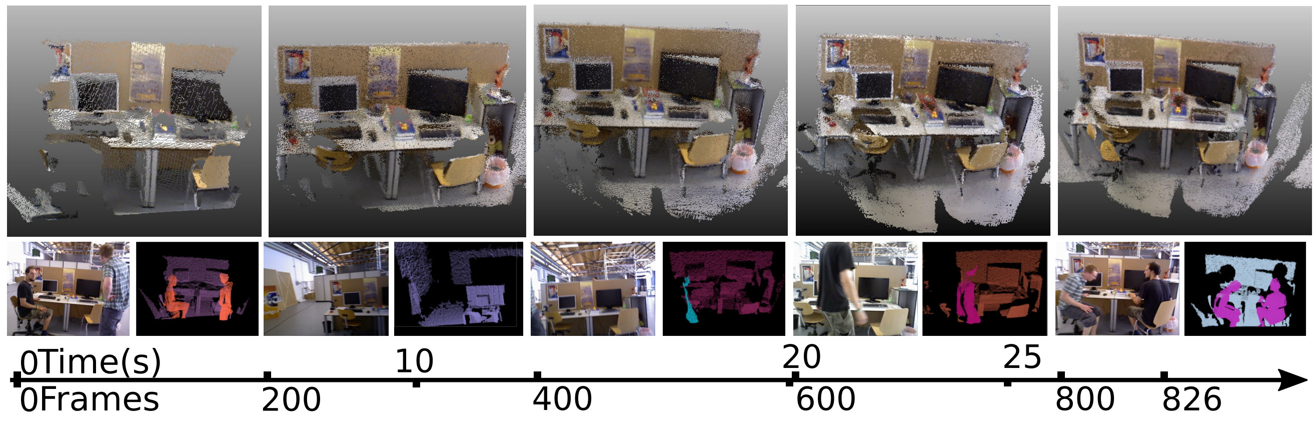
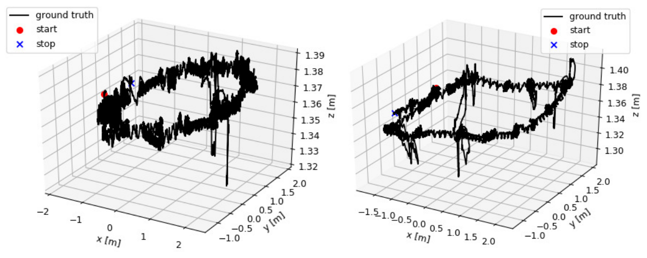


Fig. 5: Experiment result of fr3/walking from TUM benchmark [4]. The first row shows the reconstructed maps, the second row shows the respectively RGB input and foreground-background segmentations. As the frame grows, the first row reconstruction is gradually completed, the RGB viewpoint moves as the 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.



(a) HrpSlam1 Ground Truth (b) HrpSlam2 Ground Truth

Fig. 6: HRPSlam dataset ground truth. Plotted from the motion capture system. The camera trajectories which are calibrated using head and ears markers. The lines sudden drop in z axis at max 8 cm in falling down cases. For visualizing purpose, the start and end coordinates are heavily redrawn in red and blue.

TABLE III: Translate RPE RMSE (m/s)

Dynamic DataSet	JF	EF	CF	PFH
HRPSlam1	0.17	0.81	0.78	**0.05**
HRPSlam2	0.51	2.90	1.12	0.07
fr3/walk_xyz	0.15	0.21	0.37	**2.11**
fr3/walk_halfsphere	0.37	0.79	0.31	**4.50**

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. One can compare the moving object detection and removal abilities by checking when and how many moving object ghost shadows are integrated into the reconstructed scenes in the first row. One comparison with EF and PFH is given in TABLE. II. The result of EF, 67.78 cm absolute trajectory error makes an obvious wrong reconstruction. PFH achieves only 3.10 cm ATE which is quite close to the result of original PF, since that the adjustments of PFH are mainly on the back-end side which have slight effect on small scale reconstruction datasets. In slightly dynamic cases, is the human stay static for a few seconds, EF will fusion the human into the reconstructed maps, while, the proposed approach can avoid this drawback. See the first scene in Fig.5, the static man sitting on the chair is segmented and removed as foreground segments by PFH as well. The dynamic performance of the proposed is even as good as the EFs static performance.

HrpSlam2 experiment environment is shown in Fig. 7, From top to bottom, camera frame from 0 to 6000 (30FPS). From left to right: real time mapping result, robot RGB input image, depth input image, PFH output dynamic object segmentations and the instant video screen shot. The first column real-time mapping results include no moving objects, and keeps a right camera localization and tracking. These experimental results intuitively prove the dynamic ability of our SLAM methods. One can compare the moving object detection and removal abilities through checking when and how many moving object ghost shadows are integrated into the reconstructed scenes in the first column.

Table II and III show the root-mean-square error (RMSE) of translate ATE (m) and RPE (m/s). All of these datasets are dynamic scenes. From these two Tables, one can obviously find that our PFH method achieved the smallest estimation errors in highly dynamic situations.

For fairly comparison, since the other methods are not considering about humanoids falling down cases, these results are from estimated camera transforms without falling down time periods. Fig. 8 compares JF and our PFH using

TABLE IV: Number of Ferns and Detected Loops

Dynamic DataSet	PF Ferns	Loops	PFH Ferns	Loops
HRPSlam1	87	5	81	**13**
HRPSlam2	98	4	92	**16**

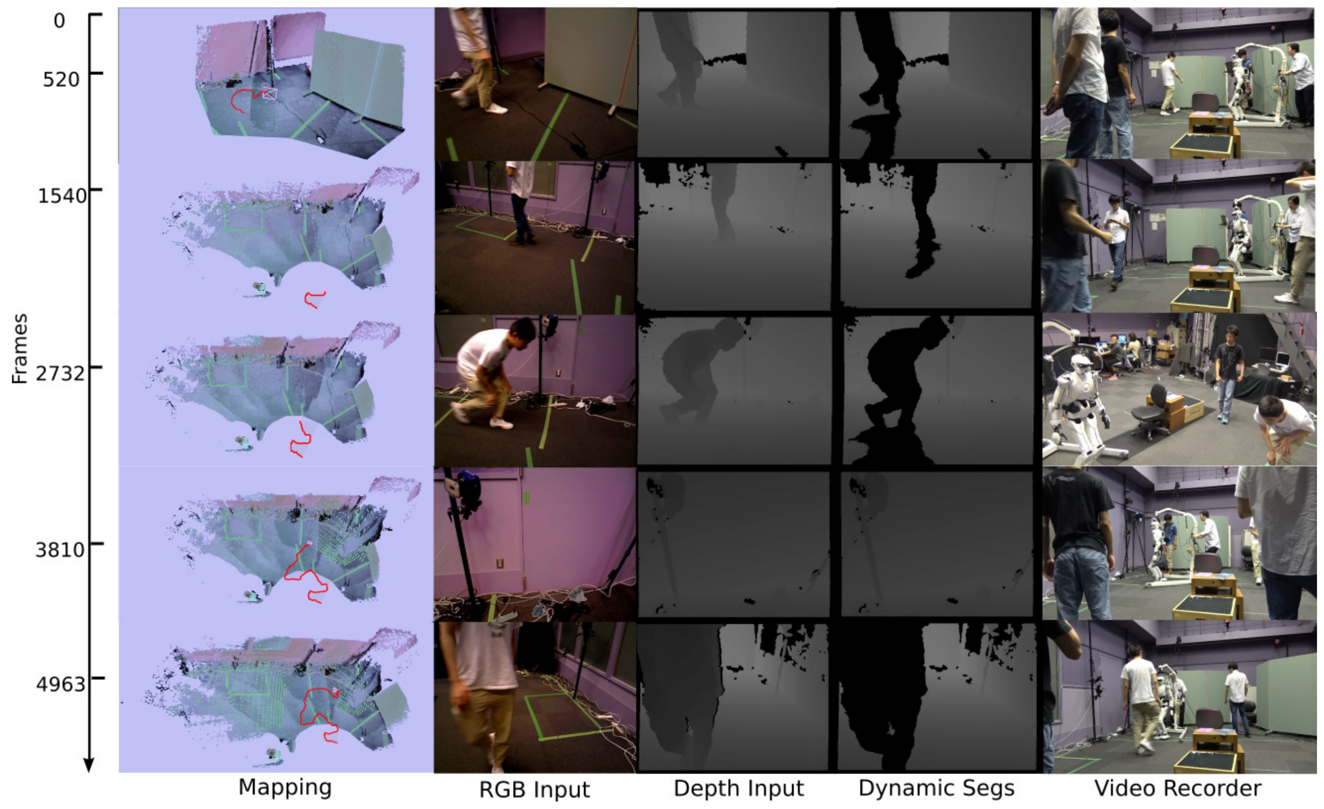
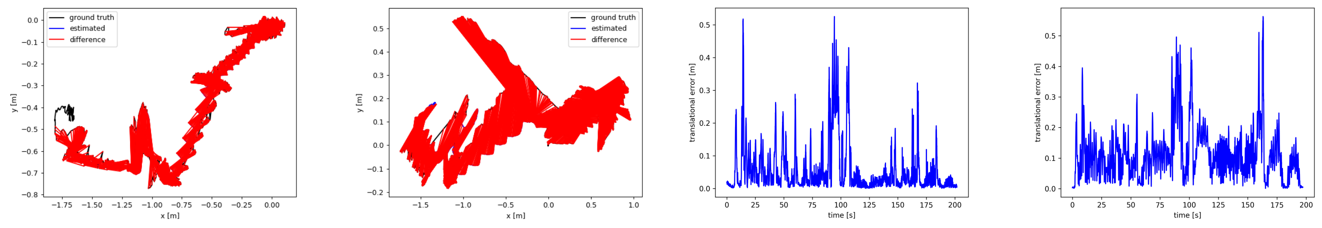


Fig. 7: The proposed method working on HrpSlam2. Top to bottom, camera frame from 0 to 6000 (30 FPS). From left to right: real time mapping result, robot RGB input image, depth input image, dynamic object segmentation (colored in pure black) outputs, and the instant video screen shot. The Leftest reconstructed maps which include no moving objects, and keeps a right camera localization and tracking



(a) PFH Reconstruction ATE (b) JF Reconstruction ATE (c) PFH Reconstruction RPE (d) JF Reconstruction RPE

Fig. 8: Evaluation of the Proposed PFH compare to JF, both are the result of first 6000 frames on HrpSlam2, as shown in Fig7. (a) and (c) are absolute trajectory error (ATE) and the relative pose error (RPE) of PFH, while (b), (d) are ATE and RPE of JF. PFH achieves very small trajectory error, average 12 cm (short red line segments in (a)), while JF gets big ATE, long red line segments in (b). (c) and (d) indicate PFH achieves about 3 times smaller average RPE than JF.

first 6000 frames on HrpSlam2 (before HRP-4 falls down at frame 6200), as shown in Fig. 7. (a) and (c) are absolute trajectory error (ATE) and the relative pose error (RPE) of PFH, while (b), (d) are ATE and RPE of JF. PFH achieves very small trajectory error, average 12 cm (short red line segments in (a)), while JF gets big ATE, long red line segments in (b). (c) and (d) indicate PFH achieves about 3 times smaller average RPE than JF.

TABLE IV indicates the loop detection difference between PFH and the original PF. PF doesn't consider the dynamic motion from the robot-self, PFH removes the blurred images caused by the execution of biped walking and enhanced the random ferns strategy to relocate the camera pose after falling down. From this table, PFH generated less ferns than PF since we force it to encode ferns only in static views. Though the number of ferns decreases, PFH find more loops than PF. That is because, PF cannot relocate the camera and find more loops after falling down, while PFH could.

VI. Discussions

We chose EF SLAM framework for humanoids dynamic SLAM since that, firstly, EF, CF, SF, and the other "Fusion family" RGB-D methods apply a non-rigid environment modeling scheme, which improves the approach robustness in changing environments. For instance, In EF, when the robot revisit to a desk, and there is a book on the desk which was moved, EF can discard the book shape in the elder desk model and fusion the new book model to the new position.

Secondly, EF's frame-to-model reconstruction provides a dense environment map. Compare with the sparse mapping approaches, the dense point clouds map is more convenient for the other robot system applications, such as plane areas extraction, footstep planning, etc.

However, the loop closure performance of EF is passable in humanoid dynamic SLAM applications. The random ferns loop detection method cannot generate enough reliable ferns in dynamic environments, thus it's hard to find a global loop closing using the original random ferns method in HRPSlam benchmark. Deformation graph based surfaces fusion plus random ferns based global loop testing make EF family methods are good for one room-size environment reconstruction, but not for large scale environment reconstruction. In dynamic environment, we enforce random ferns by constraint fern generation in static scenes, but this strategy turns weaker when there are long time continuous dynamic objects within the scenes. To this end, a graph optimization back-end with sufficient semantic information (for example, the objects in the environment, which are moving? which are static but movable?) could be helpful.

VII. Conclusions

In this paper, we propose a human motion detection method to detect and remove human motions in dynamic environments. we apply our previous PoseFusion dynamic human SLAM method into hard humanoids dynamic SLAM scenes. We involve front-end camera distortion removal strategy, and enhanced global loop finding strategy to deal with the biped walking dynamic motions and the falling down cases. Experimental results on both typical benchmark and the real humanoid robot dynamic cases indicate our work achieved excellent performances not only in the cases dealing with dynamic human objects, but also in falling down relocations situations. By combining dynamic motion recognition method with module modification method, we show that a dense dynamic RGB-D SLAM for a humanoid robot can really be achieved.

VIII. Acknowledgements

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

References

[1] T. Zhang and Y. Nakamura, "Posefusion: RGB-D SLAM in dynamic human environment," 2018, international Symposium on Experimental Robotics, accepted for presentation. 1, 2

[2] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*. IEEE Computer Society, 2017, pp. 1302–1310. 1

[3] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 3, pp. 448–461, 2010. 1, 2, 3

[4] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IROS*. IEEE, 2012, pp. 573–580. 1, 4, 5

[5] O. Stasse, A. J. Davison, R. Sellaouti, and K. Yokoi, "Real-time 3d slam for humanoid robot considering pattern generator information," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 348–355. 1

[6] R. Scona, S. Nobili, Y. R. Petillot, and M. Fallon, "Direct visual SLAM fusing proprioception for a humanoid robot," in *IROS*. IEEE, 2017, pp. 1419–1426. 1

[7] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016. 1, 2

[8] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," 2012. 2

[9] 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), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136. 2

[10] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," *arXiv preprint arXiv:1706.06629*, 2017. 2

[11] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from RGB-D cameras based on geometric clustering," in *ICRA*. IEEE, 2017, pp. 3992–3999. 2

[12] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: Background reconstruction for dense rgb-d slam in dynamic environments." Institute of Electrical and Electronics Engineers, 2018. 2

[13] T. Zhang and Y. Nakamura, "Moving humans removal for dynamic environment reconstruction from slow-scanning lidar data," in *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE, 2018, pp. 449–454. 2

[14] S. Pertuz, D. Puig, and M. A. Garcia, "Analysis of focus measure operators for shape-from-focus," *Pattern Recognition*, vol. 46, no. 5, pp. 1415–1432, 2013. 2

[15] J. L. Pech-Pacheco, G. Cristóbal, J. Chamorro-Martinez, and J. Fernández-Valdivia, "Diatom autofocusing in brightfield microscopy: a comparative study," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 3. IEEE, 2000, pp. 314–317. 2

[16] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 39–46. 3

[17] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *International journal of computer vision*, vol. 70, no. 2, pp. 109–131, 2006. 3

[18] A. Handa, T. Whelan, J. McDonald, and A. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014. 4

[19] T. Zhang. (2018, Nov.) Hrpslam: A benchmark for rgb-d dynamic slam and humanoid vision. [Online]. Available: http://www.ynl.t.u-tokyo.ac.jp/wp/dataset/HRPSlam 4

[20] C. Santacruz and Y. Nakamura, "Reactive stepping strategies for bipedal walking based on neutral point and boundary condition optimization," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3110–3115.