IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 26, 2020, accepted April 25, 2020, date of publication April 30, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991441

DDL-SLAM: A Robust RGB-D SLAM in Dynamic Environments Combined With Deep Learning

YONGBAO AI[1], TING RUI[1], MING LU[1], LEI FU[1], SHUAI LIU[1], AND SONG WANG[2]
[1]College of Field Engineering, Army Engineering University of PLA, Nanjing 210000, China
[2]People's Liberation Army, Shenyang 110000, China

Corresponding author: Ting Rui (rtinguu@sohu.com)

This work was supported by the National Natural Science Foundation of China under Grant 61671470.

ABSTRACT Visual Simultaneous Localization and Mapping (VSLAM) has developed as the basic ability of robots in past few decades. There are a lot of open-sourced and impressive SLAM systems. However, the majority of the theories and approaches of SLAM systems at present are based on the static scene assumption, which is usually not practical in reality because moving objects are ubiquitous and inevitable under most circumstances. In this paper the DDL-SLAM (Dynamic Deep Learning SLAM) is proposed, a robust RGB-D SLAM system for dynamic scenarios that, based on ORB-SLAM2, adds the abilities of dynamic object segmentation and background inpainting. We are able to detect moving objects utilizing both semantic segmentation and multi-view geometry. Having a static scene map allows inpainting background of the frame which has been obscured by moving objects, therefore the localization accuracy is greatly improved in the dynamic environment. Experiment with a public RGB-D benchmark dataset, the results clarify that DDL-SLAM can significantly enhance the robustness and stability of the RGB-D SLAM system in the highly-dynamic environment.

INDEX TERMS DDL-SLAM, semantic segmentation, multi-view geometry, dynamic environments.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a precondition for some robot applications, such as industrial automation, autonomous vehicles, and collision-less navigation. The SLAM technology was first put forward by Smith *et al.* [1], [2] in 1986.The autonomous robot estimates the pose utilizing data attained by distinct sensors and information of previous locations during it travels around in an uncharted scene, while building incrementally a consistent map of the scene in the meantime. The solution has been seen as a pivotal landmark going after truly autonomous robots over a decade. Nowadays, it is safe to say that the SLAM problem has been solved in many ways, at the very least in theory [3].

Visual SLAM, where the camera is used as the unique exteroceptive sensor, has been extensively investigated over the last years. It uses images as the unique source of external environment information [4], because images contain a large amount of useful information and may be applied to other visual applications, such as semantic segmentation, object detection and tracking. The typical visual SLAM algorithm mainly calculates the camera pose, and rebuilds the 3D map with the multi-view geometry theory. In order to improve the data processing speed, many algorithms extract sparse feature points at first, and achieve inter-frame estimation and loop closing through matching feature points. For instance, SIFT [5] or ORB [6]features are widely applied to visual SLAM, because they have better robustness and superior distinction, as well as fast algorithm processing speed. However, manual sparse image features are limited at present, where there are many challenging difficulties under the following conditions: dynamics, too many or very few feature points, large scale scenarios and so on. In visual SLAM, a hierarchical image feature extraction approach represented by deep learning has emerged over the years, which is applied to visual odometry (e.g. [7]–[10]) and loop closure detection (e.g. [11]–[13]). Deep learning is a representation-learning method with multiple levels of representation, acquired by consisting of simple but non-linear modules that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level [14]. Nowadays, the combination of deep learning and SLAM is mainly in three aspects, namely, inter-frame estimation [7], [9], [10], loop closure detection [15]–[17] and semantic mapping [18], [19].

The associate editor coordinating the review of this manuscript and approving it for publication was Leo Chen.

In the past few decades, some impressive SLAM systems have evolved and achieved good performance in certain cases. Notwithstanding, substantial issues remain unsolved, for instance, how to cope with dynamic objects under the dynamic circumstances, how to make robots fully comprehend the circumstances and complete advanced work. The primary contributions of the paper are:

- A novel RGB-D SLAM framework combined with deep learning is put forward to decrease the impact of moving objects on the camera pose estimation. The combined approach of semantic segmentation and multi-view geometry serves as a preprocessing stage to filter out data which are related to dynamic targets.
- A background inpainting method is utilized to repair the frame background that is covered by moving objects. Then these synthesized frames are used to generate an octree map.

The rest of the paper is organized as follows. Section II briefly presents a review of various SLAM achievements in dynamic scenarios. Subsequently, section III elucidates the architecture of our SLAM system. Whereafter, we show in section IV, the qualitative and quantitative results of performance of DDL-SLAM in the TUM RGB-D dataset [20] revealing the effectiveness, availability and accuracy of the system. In the end, in section V we conclude the paper and a brief discussion is given.

II. RELATED WORK

The SLAM problem in dynamic circumstances has been an active field of research in robot community over the years. Some SLAM systems process dynamic contents as outliers and then filter out observations of them. Subsequently, the observations of static areas in the scene are utilized to implement mapping, localization and navigation. The concept of dynamic environments can be further classified in low-dynamic environments, which consist of static objects and entities that move slowly or seldom like doors, chairs, tables or parked cars, and highly-dynamic environments which are continuously changing their pose and occupy most of the scene like moving people or cars.

In low-dynamic environments, [21] presents an algorithm of occupancy grid mapping for robots running in circumstances where non-stationary objects frequently move, [22] proposes a SLAM method for detecting and tracking moving targets simultaneously using a laser scanner, and in [23]an approach is proposed for adding the time dimension to the process of mapping to make a robot preserve an exact map while running in dynamic scenes, where the Dynamic Pose Graph SLAM was presented. However in these methods the laser scanner is used as a sensor, which is different from our approach. On the other hand, [24] describes the parallel execution of monoSLAM and a 3D object tracker, which allows inferring moving objects and occlusion, and [25] proposes an incremental movement segmentation system that effectively segments numerous dynamic targets and concurrently constructs the map of the outdoor scenes with monocular camera. Multiple clues on the basis of optical flow and two view geometry are combined to implement the segmentation. [26]presents a stereo-based visual SLAMMOT (simultaneous localization, mapping and moving object tracking) approach so as to handle moving objects while performing SLAM in highly-dynamic circumstances. In [27]a method of the combination of stereo-based visual SLAM and dense scene flow is put forward to improve traditional algorithms in highly-dynamic and large-scale environments. Furthermore, some RGB-D SLAM systems deal with moving targets in challenging dynamic scenes in the literatures [28]–[32].Our goal is to enhance the robustness and stability of RGB-D SLAM based on ORB-SLAM2 [33] in highly-dynamic scenarios. We propose some effective improvement measures to achieve better results.

III. SYSTEM DESCRIPTION

We will introduce DDL-SLAM at length in this section. It's consist of five aspects. First, the framework of DDL-SLAM is proposed. Second, we briefly describe the semantic segmentation employed in our system. Then the multi-view geometry algorithm which is utilized to improve the dynamic content segmentation is introduced. Subsequently, the tracking and mapping module is demonstrated, which is based on ORB-SLAM2. Finally, we show the method to inpaint the obscured background and build an octree map.

A. FRAMEWORK OF DDL-SLAM

In real life applications (e.g. autonomous robots, unmanned aerial vehicles), exact pose estimation and dependability in severe circumstances are key factors. To the best of our knowledge, ORB-SLAM2 has a prominent performance in various environments from a handheld camera in indoor scenes, to drones flying in outdoor scenarios and unmanned vehicles driving around in a city. Therefore, in DDL-SLAM, its RGB-D SLAM is adopted to provide an overall SLAM scheme, which allows us to detect moving objects and generate the octree map. Fig.1 shows the overview of DDL-SLAM.

The framework of our DDL-SLAM system is displayed in Fig.2. At first, the raw RGB images are dealt with a CNN (convolutional neural network) that segments out pixel-wise the a previousi dynamic objects, for example human. Then the potentially dynamic objects have been segmented, the camera poses are tracked utilizing the static part of the frame at this phase, where the algorithm of ORB-SLAM2 is easier and the computation load is smaller. Afterwards, the multi-view geometry is used to enhance the dynamic objects segmentation. After all of dynamic content has been detected and the camera localization has been completed, the obscured background of the current frame will be reconstructed using static information born of previous frames. Then the inpainted RGB and depth images are utilized to generate the local point cloud that will be transformed and maintained in an octree map. Finally, ORB features of the static part of the frame are extracted to be used in the tracking and mapping thread.

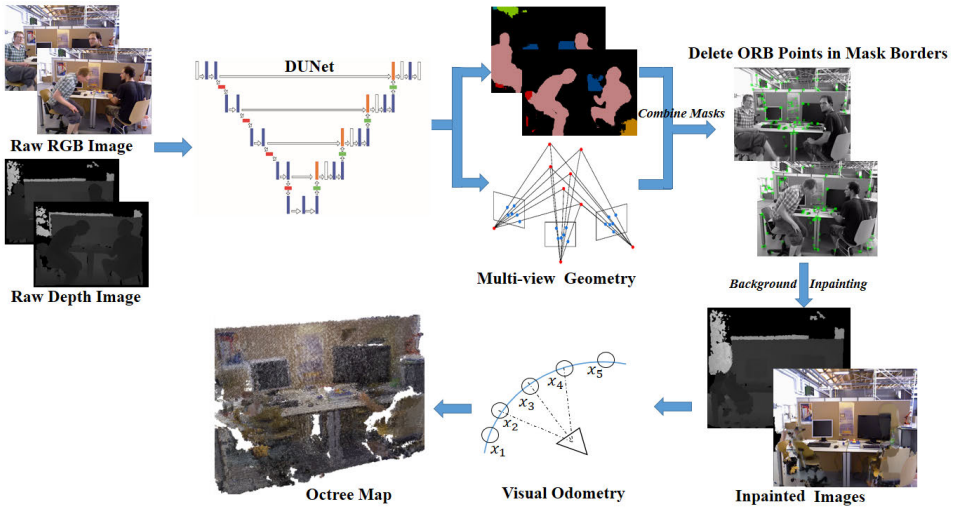


FIGURE 1. The overview of DDL-SLAM. The raw RGB image is used to semantic segmentation through DUNet. Both semantic segmentation and multi-view geometry are utilized to combine masks in order to filter dynamic objects out thoroughly. Then delete ORB points in mask borders. An octree map is constructed in a separate thread on the basis of the keyframe poses and inpainted images.

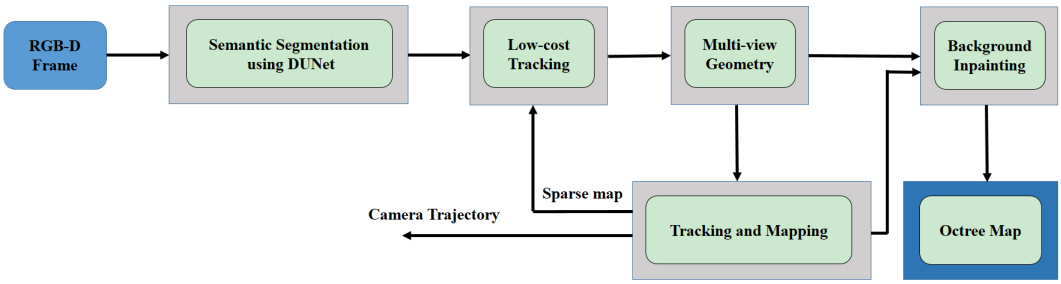


FIGURE 2. The framework of DDL-SLAM. The images pass through a CNN (DUNet) for computing the pixel-wise semantic segmentation of the a priori dynamic content. The multi-view geometry is utilized to segment more accurately, for which a low-cost tracking algorithm is required. The tracking and mapping thread is the same as ORB-SLAM2. An octree map is built based on inpainting the background obscured by dynamic objects.

B. SEMANTIC SEGMENTATION

In order to detect dynamic objects, DDL-SLAM adopts DUNet [34](deformable U-Net [35]) to implement pixel-wise semantic segmentation on the basis of the PyTorch implementation by Tramac. [1] DUNet is an FCN-based [36] network, it greatly enhances deep neural networks' capability of segmentation.

The DUNet trained on PASCAL VOC dataset [37] could segment these classes that are potentially movable (bicycle, person, boat, bird, horse, sheep, cat, cow, dog, aeroplane, bus, car, motorbike, train). In real applications, the moving objects likely to occur are inclusive of this list. The network could be also trained on MS COCO [38], if other potentially dynamic classes came out.

The input of DUNet is an original RGB image of size $h \times w \times 3$, and the output of the network is a matrix of size $h \times w \times n$, where n is the number of dynamic objects in the image. For each of output channel $i \in n$ a binary mask is acquired. By the means of merging all the channels into one, the segmentation of all dynamic objects that appear in the image of a scene is acquired.

[1] https://github.com/Tramac/awesome-semantic-segmentation-pytorch

C. SEGMENTATION OF DYNAMIC CONTENT USING DUNET AND MULTI-VIEW GEOMETRY

Although the majority of dynamic objects can be segmented with DUNet, there are a handful of objects which cannot be detected just by this means. The reason is that they are not transcendentally dynamic, but movable. For example, the cup, telephone and book keep still in the Fig.3 (a), then they become movable some time separately in the Fig. 3, (c), (d). The multi-view geometry is added to the system so as to improve the dynamic objects segmentation. The segmentation of the dynamic content formerly acquired through the DUNet is refined, what's more, new dynamic objects instances which are static most of the time and not set to be moving in the network stage are detected.The algorithm of multi-view geometry is shown in Algorithm 1.

D. TRACKING AND MAPPING

Based on ORB-SLAM2 this module is mainly constituted of three parallel threads: tracking, local mapping and loop closure. The RGB and depth images, as well as their segmentation mask are input to this stage of the DDL-SLAM. The ORB features belonging to the image segmentation classified as static are extracted in the tracking thread. Then the camera poses are estimated with the previous frames by

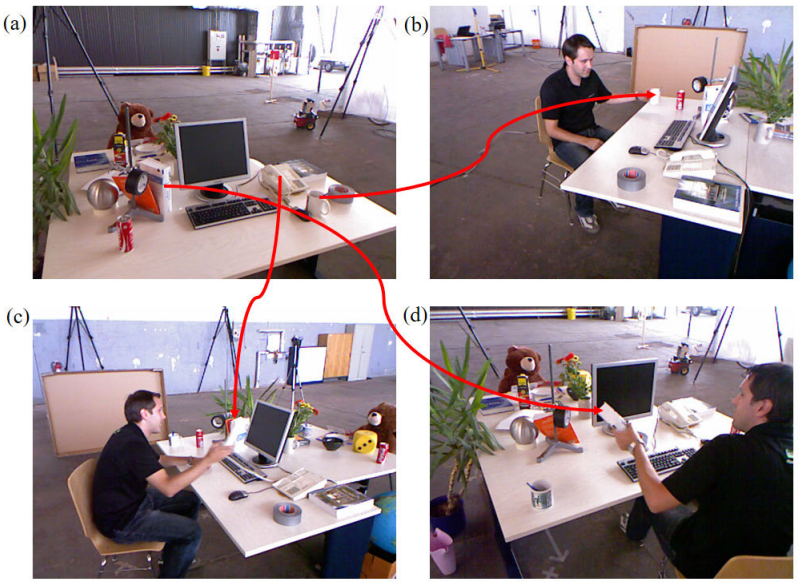


FIGURE 3. RGB images of freiburg2_desk_with_person from the TUM RGB-D dataset [20].

Algorithm 1 Multi-view Geometry

Data: Current frame F_1, depth image md
Result: MasksList
//Find the previous keyframes that have the highest overlaps;
vRefFrames $F' \leftarrow GetRefFrames(F_1)$;
//Find dynamic keypoints;
for *each keypoint x of* F' **do**
    Compute the projected point x' and projected depth $d_p in F_1$;
    Compute corresponding 3D point X ;
    if $\alpha \leftarrow \angle XXx' > \tau_\alpha$ **then**
        mask == dynamic;
        MasksList $\leftarrow$ added;
    end
    if $\Delta d \leftarrow d_p - d_{x'} > \tau_d$ **then**
        mask == dynamic;
        MasksList $\leftarrow$ added;
    end
end
MasksList $\leftarrow$ CombineMasks(F_1,mask);

finding features matching in the local map and minimizing the re-projection error employing motion-only bundle adjustment (BA). The algorithm manages the local map and optimizes it, and performs local BA at the same time in the local mapping thread. It detects large loops and corrects the accumulated drift using a pose-graph optimization in the loop closing thread. Then the thread starts the next thread to execute full BA after the pose-graph optimization, to calculate the optimal structure and motion solution.

E. BACKGROUND INPAINTING AND OCTREE MAP BUILDING

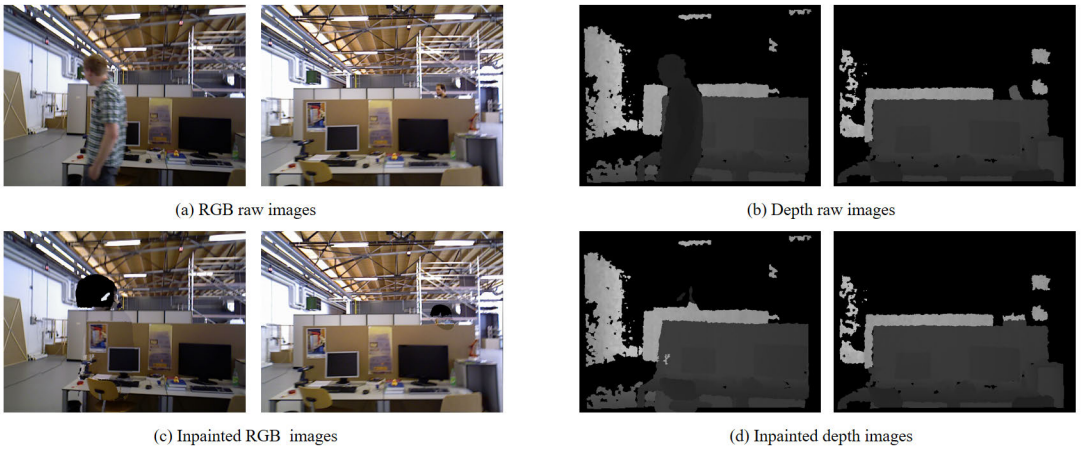
To inpaint the obscured background utilizing static information born of previous views, the last 15 previous keyframes are selected to project into the dynamic parts of the current frame. The synthetic images from input frames of some sequences in the TUM RGB-D dataset are displayed in Fig.4. It can be seen how all the dynamic objects have been successfully detected and removed. Moreover, a majority of the segmented areas have been correctly inpainted using the information of static background. However, a few blocks are not inpainted completely on account of their missing parts of the scene have not come up heretofore in the keyframes, or, they do not have valid depth information though they have appeared. These gaps cannot be rebuilt just with geometric approaches and a more elaborate inpainting technique will be required in the future research work.

Then these synthesized frames are used to generate the local point cloud, which will be transformed and maintained in a global octree map. The octree map expression [39] is flexible, compact and updatable. What's more, it is stored efficiently and employed easily for navigation.

IV. EXPERIMENTAL RESULTS

The DDL-SLAM system has been evaluated in the public datasets TUM RGB-D in this section. It provides many sequences in dynamic environments with ground truth acquired using a highly accurate motion capture system, for example walking, sitting and desk. There are two youngsters walking from the foreground to background, then they sit down at the desk in the sequences named walking. These sequences are highly dynamic and hence difficult for general SLAM systems. In the sitting sequences, two youngsters sit at a desk while speaking and gesticulating. These sequences are considered as low-dynamic because the people seldom move. All of the experiments are carried out on a computer with Intel i7 CPU, NVIDIA TITAN GPU, and 12GB memory.

DDL-SLAM adopts ORB-SLAM2 generally accepted as the state-of-art algorithm at present as a global SLAM solution. So we make a comparison against RGB-D ORB-SLAM2. The metric of absolute trajectory error (ATE)



(a) RGB raw images (b) Depth raw images

(c) Inpainted RGB images (d) Inpainted depth images

FIGURE 4. The synthetic images of background inpainting. Two RGB raw images are shown in Figure 4 (a), the output of our system is shown in Figure 4 (c), in which dynamic content has been segmented and the background has been reconstructed. Figure 4 (b) and (d) show the depth images input and output respectively, which have also been processed.

TABLE 1. Results of metrics absolute trajectory error (ATE [m]).

Sequences	ORB-SLAM2 (RGB-D)				DDL-SLAM				Improvements			
	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.
w_halfsphere	0.495	0.402	0.275	0.289	0.029	0.025	0.02	0.015	94.10%	93.80%	92.70%	94.80%
w_xyz	0.696	0.591	0.514	0.368	0.014	0.012	0.011	0.007	98.00%	98.00%	97.90%	98.10%
w_rpy	0.486	0.424	0.343	0.238	0.031	0.027	0.021	0.017	93.60%	93.60%	93.90%	92.90%
w_static	0.416	0.379	0.312	0.171	0.006	0.006	0.005	0.003	98.60%	98.40%	98.40%	98.20%
s_halfsphere	0.082	0.077	0.074	0.028	0.019	0.017	0.015	0.009	76.80%	77.90%	79.70%	67.90%
s_static	0.008	0.007	0.006	0.004	0.006	0.005	0.005	0.003	25.00%	28.60%	16.70%	25.00%
desk_person	0.07	0.068	0.067	0.018	0.072	0.071	0.071	0.016	-	-	-	-

TABLE 2. Results of metrics translational drift (RPE [m/s]).

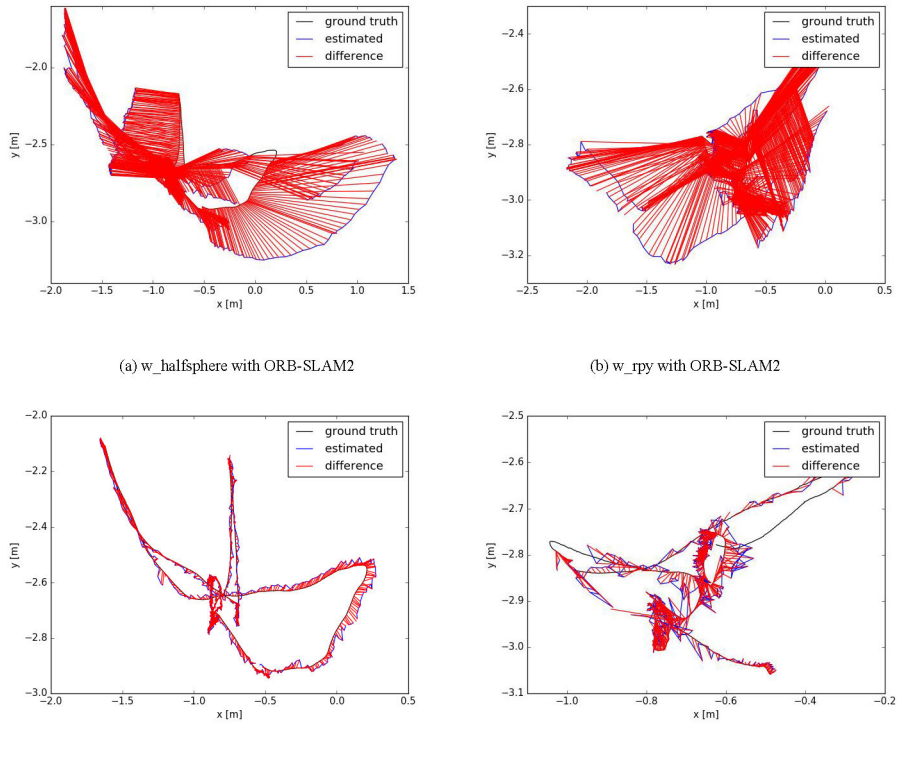
Sequences	ORB-SLAM2 (RGB-D)				DDL-SLAM				Improvements			
	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.
w_halfsphere	0.36	0.219	0.065	0.285	0.028	0.024	0.022	0.013	92.20%	89.10%	66.20%	95.40%
w_xyz	0.399	0.313	0.286	0.247	0.019	0.016	0.014	0.009	95.20%	94.90%	95.10%	96.40%
w_rpy	0.359	0.243	0.097	0.264	0.041	0.034	0.029	0.022	88.60%	86.00%	70.10%	91.70%
w_static	0.232	0.093	0.015	0.212	0.009	0.008	0.007	0.004	96.10%	91.40%	53.30%	98.10%
s_halfphere	0.051	0.035	0.019	0.038	0.024	0.021	0.018	0.011	52.90%	40.00%	5.30%	71.10%
s_static	0.009	0.008	0.007	0.004	0.008	0.007	0.006	0.004	11.10%	12.50%	14.30%	0
desk_person	0.01	0.009	0.008	0.005	0.011	0.009	0.008	0.006	-	-	-	-

TABLE 3. Results of metrics rotational drift (RPE [deg/s]).

Sequences	ORB-SLAM2 (RGB-D)				DDL-SLAM				Improvements			
	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.	RMSE	Mean	Median	S.D.
w_halfsphere	7.388	4.61	1.58	5.773	0.782	0.687	0.622	0.375	89.40%	85.10%	60.60%	93.50%
w_xyz	7.714	6.044	5.301	4.794	0.622	0.491	0.406	0.382	91.90%	91.90%	92.30%	92.00%
w_rpy	6.853	4.673	1.965	5.012	0.91	0.762	0.663	0.497	86.70%	83.70%	66.30%	90.10%
w_static	4.174	1.715	0.369	3.805	0.242	0.217	0.203	0.106	94.20%	87.30%	45.00%	97.20%
s_halfphere	1.118	0.886	0.655	0.682	0.714	0.615	0.529	0.363	36.10%	30.60%	19.20%	46.80%
s_static	0.283	0.255	0.245	0.122	0.275	0.246	0.226	0.122	2.80%	3.50%	7.80%	0
desk_person	0.426	0.355	0.314	0.236	0.438	0.366	0.319	0.241	-	-	-	-

is very suitable for measuring performance of the visual system. And the metric of relative pose error (RPE) is utilized to measure the drift of the visual odometry. So we compute the metrics ATE and RPE for the quantitative evaluation.

Tab.1 shows the quantitative comparison results, where halfsphere, xyz, static and rpy in the first column stand for four categories of camera ego-motions [20]: (1) halfsphere: a camera moves according to the trajectory of a 1-meter diameter hemisphere, (2) xyz: a camera



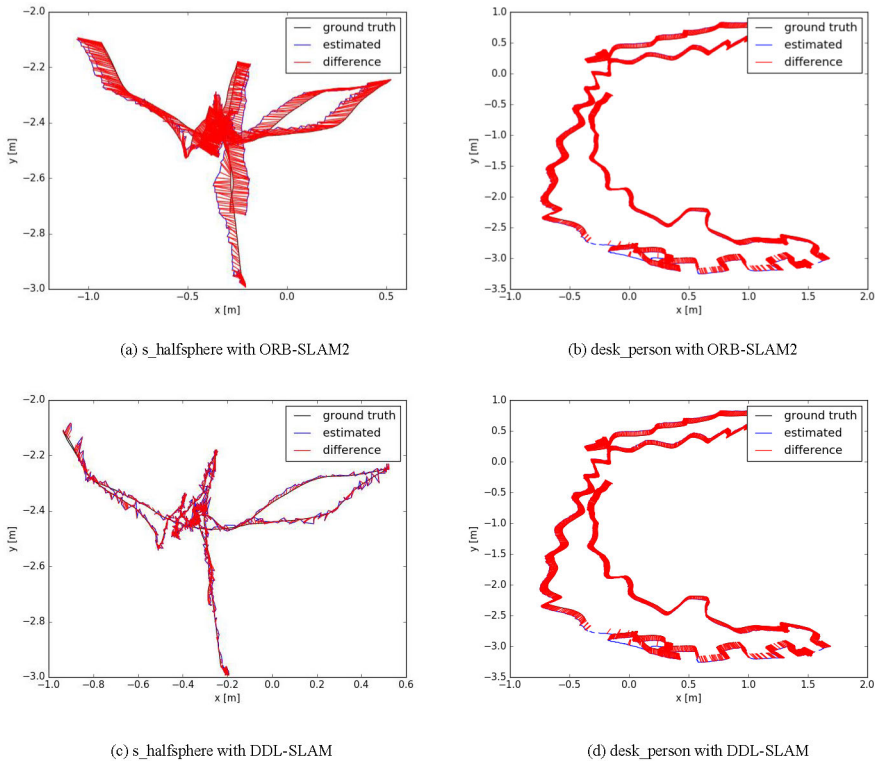
(a) w_halfsphere with ORB-SLAM2

(b) w_rpy with ORB-SLAM2

(c) w_halfsphere with DDL-SLAM

(d) w_rpy with DDL-SLAM

FIGURE 5. Plots of ATE for the highly-dynamic sequences freiburg3/w_halfsphere, w_rpy, (a) and (b) are drawn with RGB-D ORB-SLAM2, (c) and (d) are drawn with DDL-SLAM.



(a) s_halfsphere with ORB-SLAM2

(b) desk_person with ORB-SLAM2

(c) s_halfsphere with DDL-SLAM

(d) desk_person with DDL-SLAM

FIGURE 6. Plots of ATE for the low-dynamic sequences freiburg3/s_halfsphere, freiburg2/desk_person, (a) and (b) are drawn with RGB-D ORB-SLAM2, (c) and (d) are drawn with DDL-SLAM.

respectively moves along the x-y-z axes, (3) static: a camera is kept static manually, and (4) rpy: a camera revolves over roll, pitch and yaw axes. The values of Root-mean-square Error (RMSE), Mean Error, Median Error and Standard

Deviation (S.D.) are presented in the research, while RMSE and S.D. are more focused on account of they can preferably demonstrate the robustness and stability of the system. It is obvious from Tab.1, our method makes the property in most highly-dynamic sequences attain an order of magnitude enhancement. As far as ATE is concerned, the improvement values of RMSE and S.D. can respectively come up to 98.6% and 98.2%. The results show that DDL-SLAM can significantly enhance the robustness and stability of SLAM system in highly-dynamic scenarios. And in low-dynamic scenes, the error is similar to the original RGB-D ORB-SLAM2 system. The primary cause is that original ORB-SLAM2 is adept in the low-dynamic environments and achieves good performance, therefore the upside potential of the performance is restricted. Tab.2 and Tab.3 display the performance of visual odometry. It can be seen that the results coincide with the above ATE analysis.

Fig.5 displays the selected ATE curve graphs for the highly-dynamic sequences. It is obvious that the errors are significantly decreased with our method. The selected ATE curve graphs of the low-dynamic sequences are shown in Fig.6. It can be seen that the original ORB-SLAM2 expresses good performance in these cases. With our method combined into the SLAM system, the ATE values are greatly decreased. However, in the freiburg2_desk_with_person sequences, we found that our means could not improve the original capacity. We think the primary cause is that there are not moving objects in the early stage of the sequences, as a matter of fact the scenes during this period are static. What's more, the low-dynamic movements are generally not successive in the sequences and moving objects always turn out to be motionless in some frames.

V. CONCLUSION

In this research, a robust and stable RGB-D SLAM (DDL-SLAM) system in highly-dynamic environments using deep learning is proposed. A pixel-wise semantic segmentation convolutional neural network named DUNet is integrated with the multi-view algorithm to filter out all dynamic content of the scenario. Afterwards, the matched ORB feature points will be deleted from those detected dynamic areas, and the synthetic RGB frames without dynamic objects and with the background inpainting, as well as their matching synthesized depth images are acquired. Quantitative evaluations were put into effect utilizing the challenging dynamic sequences of TUM RGB-D dataset. Experimental results elucidate that DDL-SLAM exceeds ORB-SLAM2 obviously due to its the accuracy and robustness in highly-dynamic scenarios. Nevertheless, our approach still possesses a few limitations to be improved. For example, the real-time performance of the algorithm requires to be improved, a more elaborate inpainting background method needs to be put forward, or the octree map attained by our system would be endowed with semantic information to be employed for navigation in future work.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Proc. 2nd Annu. Conf. Uncertainty Artif. Intell.* New York, NY, USA: ACM, 1986 pp. 435–461.

[2] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, Dec. 1986.

[3] J. Boal, Á. Sánchez-Miralles, and Á. Arranz, "Topological simultaneous localization and mapping: A survey," *Robotica*, vol. 32, no. 5, pp. 803–821, Aug. 2014.

[4] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.*, vol. 43, pp. 55–81, Nov. 2015.

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, Nov. 2004.

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[7] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in *Proc. 10th Int. Conf. Comput. Vis. Theory Appl.*, Lisbon, Portugal: SCITCC Press, 2015, pp. 486–490.

[8] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Piscataway, NJ, USA, Dec. 2015, pp. 2758–2766.

[9] A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac, and A. Davison, "GVNN: Neural network library for geometric computer vision," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), vol. 9915. Berlin, Germany: Springer-Verlag, 2016, pp. 67–82.

[10] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with CNNs for Frame-to-Frame ego-motion estimation," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 18–25, Jan. 2016.

[11] D. Bai, C. Wang, B. Zhang, X. Yi, and Y. Tang, "Matching-range-constrained real-time loop closure detection with CNNs features," *Robot. Biomimetics*, vol. 3, no. 1, pp. 70–75, Dec. 2016.

[12] X. Gao and T. Zhang, "Loop closure detection for visual SLAM systems using deep neural networks," in *Proc. 34th Chin. Control Conf. (CCC)*, Piscataway, NJ, USA, Jul. 2015, pp. 5851–5856.

[13] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," *Auto. Robots*, vol. 41, no. 1, pp. 1–18, Jan. 2017.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[15] Z. T. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," 2014, *arXiv:1411.1509*. [Online]. Available: https://arxiv.org/abs/1411.1509

[16] J. Wu, L. Ma, and X. Hu, "Delving deeper into convolutional neural networks for camera relocalization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 5644–5651.

[17] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2938–2946.

[18] N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5079–5085.

[19] X. Li and R. Belaroussi, "Semi-dense 3D semantic mapping from monocular SLAM," 2016, *arXiv:1611.04144*. [Online]. Available: http://arxiv.org/abs/1611.04144

[20] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[21] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, Mar. 2002, pp. 1014–1019.

[22] H. Zhao, M. Chiba, R. Shibasaki, X. Shao, J. Cui, and H. Zha, "SLAM in a dynamic large outdoor environment using a laser scanner," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1455–1462.

[23] A. Walcott-Bryant, M. Kaess, H. Johannsson, and J. J. Leonard, "Dynamic pose graph SLAM: Long-term mapping in low dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 1871–1878.

[24] S. Wangsiripitak and D. W. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 375–380.
[25] R. K. Namdev, A. Kundu, K. M. Krishna, and C. V. Jawahar, "Motion segmentation of multiple objects from a freely moving monocular camera," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 4092–4099.
[26] K.-H. Lin and C.-C. Wang, "Stereo-based simultaneous localization, mapping and moving object tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 3975–3980.
[27] P. F. Alcantarilla, J. J. Yebes, J. Almazan, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1290–1297.
[28] Y. Wang and S. Huang, "Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 1841–1846.
[29] L. Riazuelo, L. Montano, and J. M. M. Montiel, "Semantic visual SLAM in populated environments," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2017, pp. 1–7.
[30] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.
[31] D.-H. Kim and J.-H. Kim, "Effective background model-based RGB-D dense visual odometry in a dynamic environment," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1565–1573, Dec. 2016.
[32] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.
[33] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[34] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, "DUNet: A deformable network for retinal vessel segmentation," 2018, *arXiv:1811.01206*. [Online]. Available: https://arxiv.org/abs/1811.01206
[35] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.
[36] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.
[39] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

YONGBAO AI received the B.E. degree in mechanical engineering and automation from Beihang University, Beijing, China, in 2014, and the M.S. degree in mechanical engineering from the Army Engineering University of PLA, Nanjing, China, in 2017. He is currently pursuing the Ph.D. degree with the school. His research interests include computer vision, simultaneous localization and mapping, and robotics.

TING RUI received the M.S. and Ph.D. degrees from the PLA University of Science and Technology, Nanjing, China, in 1998 and 2001, respectively. He is currently a Professor with the Army Engineering University of PLA. He mainly applies computer vision, machine learning, multimedia, and video surveillance. He has authored and coauthored more than 80 scientific articles.

MING LU received the M.S. and Ph.D. degrees from the PLA University of Science and Technology, Nanjing, China, in 1997 and 2001, respectively. He is currently a Professor with the Army Engineering University of PLA. He mainly engaged in teaching and research work in the field of mechanical manufacturing and automation.

LEI FU received the B.S. and M.S. degrees with the College of Instrumentation and Electrical Engineering, in 2008 and 2013, respectively. He is currently pursuing the Ph.D. degree with the College of Field Engineering, Army Engineering University of PLA, Nanjing, China. His research interests include object detection and tracking on unmanned aerial vehicles.

SHUAI LIU received the B.E. degree from the Department of Mechanical Design and Manufacture and Automation, Henan University of Science and Technology, Luoyang, China, in 2016. He is currently pursuing the M.S. degree with the Army Engineering University of PLA, Nanjing, China, in 2018. His research interests include computer vision, simultaneous localization and mapping, and robotics.

SONG WANG received the B.S. and M.S. degrees from Air Force Engineering University, Xi' an, China, in 2008 and 2012, respectively. He is currently with the People's Liberation Army of China. His research interests include visual object tracking and 3D visualization.