DynaSLAM2: A V-SLAM Towards Dynamic Circumstances

Li Ao 1*, Ni Yue 2*

University of Science and Technology of China

ABSTRACT

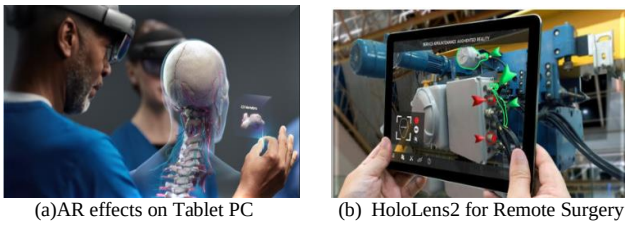
Simultaneous Localization and Mapping (SLAM) technology is proposed to solve the problem of robot self-positioning and environmental cognition. Over the past decades, many impressed SLAM systems have been developed and achieved good performance under certain circumstances. However, SLAM technology still have difficulty in handling all kinds of environments, such as extraordinarily dynamic or rough environments. In this paper, a semantic visual SLAM based on ORB-SLAM2 towards dynamic environments named DynaSLAM2 is proposed. Our proposal is to use Mask R-CNN to pixel-wise segment the priori dynamic objects in the frames(e.g., walking people), so that the SLAM algorithm does not have to extract features on them. DynaSLAM2 outperforms the accuracy of standard visual SLAM baselines and a high robustness in highly dynamic scenarios. And it also estimates a map of the static parts of the scene, which is a must for long-term applications in real-world environments.

Keywords: Dynamic objects , semantic segmentation , Mask R-CNN, ORB-SLAM2.

Index Terms: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

1 INTRODUCTION

AR(augments reality) technology requires the fusion of virtual scenes into real-world scenes for presentation to users. One of the most critical issues is how to relate virtual scenes to real-world scenarios. Online simultaneous positioning and map construction SLAM(simultaneous localization and mapping) is the key to solving this problem.



(a)AR effects on Tablet PC (b) HoloLens2 for Remote Surgery

Fig.1. AR Technology Application

SLAM technology was originally proposed to solve the problem of robot self-positioning and environmental cognition. The robot uses the surrounding environment information acquired by the sensor to perform its own positioning in real time and reconstruct the surrounding three-dimensional information to realize automatic navigation. Therefore, SLAM technology relies heavily on sensor selection. The visual SLAM [2,3] with camera-based sensor is the most common SLAM technology, including: pure vision based method, visual and IMU(inertial measurement unit) combined method, RGB-D camera based method.

However, some issues remain unsolved, for example, most of the existing algorithms are fragile. On the one hand, they have difficulty in handling all kinds of environments, such as extraordinarily dynamic or rough environments. On the other hand, their map models are often based on geometric information, like the landmark-based map and point cloud map, so they do not provide any high-level understanding of the surroundings.

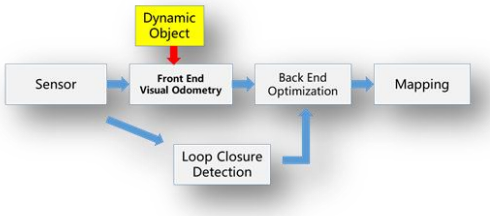


Fig.2.Dynamic factors towards V-SLAM Frame

The typical SLAM methods only provide a map with geometric entities (points, planes, etc.), which do not have semantic attributes distinction between them. However, the semantic information is needed for the robot to understand the scene surrounding them. With the development of deep learning, some networks could achieve good performance in semantic segmentation[4]. Therefore, combining these networks with SLAM could produce the semantic map, and thus improve the perception level of robots [5, 6].

The robustness of the SLAM system in dynamic environments is also a challenge. While modern SLAM system has been successfully demonstrated mostly in specific circumstances, unexpected changes of surroundings would probably corrupt the quality of the state estimation and even lead to system failure. For example, the presence of dynamics in the environment, like walking people, might deceive feature association in vision-based SLAM system. Although some progress has been made to reduce the impact of the changes of the environment in laser-based SLAM, for instance, some approaches combine inertial measurement unit with visual SLAM to improve robustness, like [7], but the problem is still not well solved in pure vision-based SLAM.

Generally, dynamic objects in visual SLAM reveals several challenges for both mapping and tracking, including:

1) Prevent the tracking algorithm from using matches that belong to dynamic objects.

2) Prevent the mapping algorithm from including moving objects as part of the 3D map.

Many applications would greatly benefit from progress along these lines. Among others, augmented reality, autonomous vehicles, and medical imaging. All of them could for instance safely reuse maps from previous runs. Detecting and dealing with

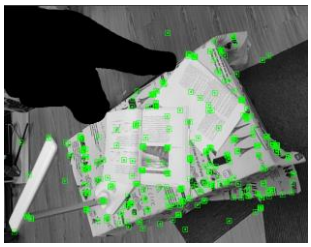
* la961028@mail.ustc.edu.cn

dynamic objects is a requisite to estimate stable maps, useful for long-term applications. If the dynamic content is not detected, it becomes part of the 3D map, complicating its usability for tracking or relocation purposes.

In this work we focus on an on-line algorithm to deal with dynamic objects in monocular and stereo SLAM. This is done by adding a front-end stage to the state-of-the-art ORB-SLAM2 system[1],with the purpose of having a more accurate tracking and a reusable map of the scene.

In the monocular and stereo cases , our proposal is to use Mask R-CNN to pixel-wise segment the priori dynamic objects in the frames(e.g., walking people), so that the SLAM algorithm does not extract features on them. On the other hand , the frame is delivered to tracking thread just as ORB-SLAM2.

(a) Input monocular frame of D3_test



(b) Frames with dynamic content after semantic segmentation in Tracking and Mapping

Fig.3. Overview of DynaSLAM2 results

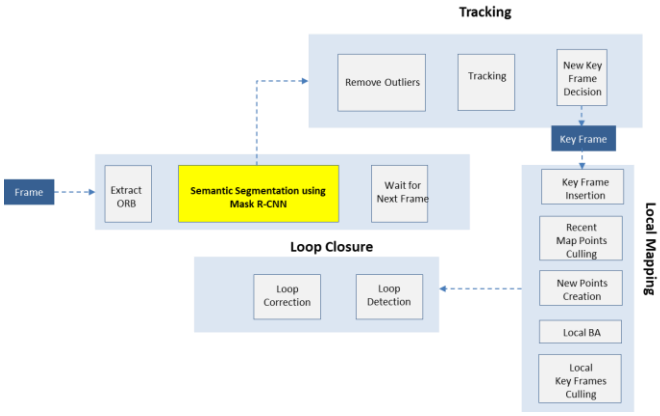


Fig.4. Block diagram of our proposal. During the SLAM process, our work use Mask R-CNN to pixel-wise segment the priori dynamic objects in the frames(e.g., walking people). Then the features on the dynamic objects segments are not extracted. the frame is directly delivered to tracking thread directly just as ORB-SLAM2.

2 THEORY

In this work we present DynaSLAM2, a visual SLAM system that, building on ORB-SLAM2, adds the capabilities of dynamic object detection.

2.1 ORB-SLAM2

ORB-SLAM2[1] is a complete SLAM solution based on monocular, stereo and RGB-D cameras. It enables map reuse, loopback detection and relocation. Whether it's small indoor handheld devices, drones in factory environments, and cars driving in the city, the ORB-SLAM2 can work in real time on a standard CPU. The ORB-SLAM2 is bundle adjustment (BA) method with monocular and stereo observations allows for accurate trajectory estimation with metric scale. This method allows the trajectory accuracy of the metric scale to be evaluated. In addition, the ORB-SLAM2 includes a lightweight positioning mode that uses a visual odometer to track unmapped areas and match feature points while allowing zero drift.

What's more, ORB_SLAM2 consists of 3+1 parallel threads, including tracking, local mapping, loopback detection, and global BA optimization after loopback detection. The reason is 3+1, because the fourth thread is only executed after loopback detection and confirmation.

The first three main parallel threads:

a. Tracking: By searching for local map features, using pure motion BA to minimize re-projection errors to locate each frame of the camera;

b. Local mapping: Manage local maps by optimizing local BA and optimize;

c. Loop closure: Detect large loops and correct drift errors by performing pose map optimization. This thread triggers the fourth thread;

d. Global BA: After the pose map optimization, calculate the optimal structure and motion results of the whole system.

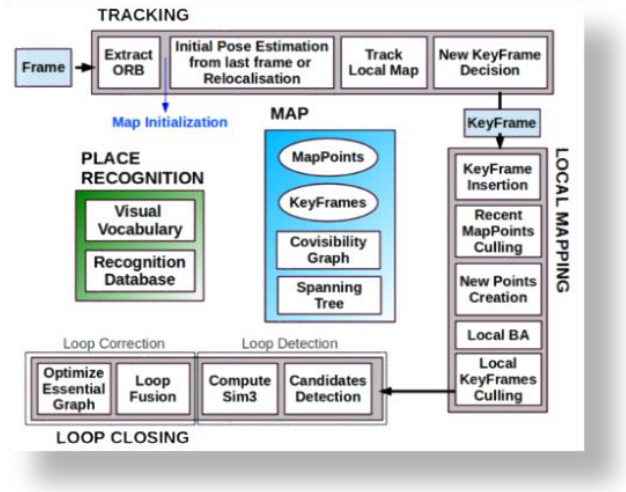


Fig.5. Block diagram of ORB-SLAM2

The system uses the DBOW2 embedded location recognition model for relocation to prevent tracking lost, or known map scene reinitialization, loop closure and so on. The system uses ORB features for tracking, mapping, and location recognition tasks. The advantage of ORB feature is rotational invariance and scale invariance. Feature extraction and feature matching can be achieved quickly with this method, which can meet the demand of real-time operation. In addition, ORB features display high precision in the position recognition process based on word bag.

2.2 Semantic Segmentation

To deal with dynamic objects , we focus on DynaSLAM2 algorithm by applying Mask R-CNN to ORB-SLAM2 system.

Mask R-CNN, extends Faster R-CNN [9] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression (Figure 3). The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to pixel manner. Mask R-CNN is simple to implement and train given the Faster R-CNN framework, which facilitates a wide range of flexible architecture designs. Additionally, the mask branch only adds a small computational overhead, enabling a fast system and rapid experimentation.

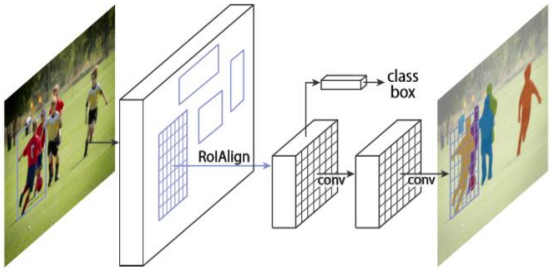


Fig.6. The Mask R-CNN framework for instance segmentation.

Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN). In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI. This is in contrast to most recent systems, where classification depends on mask predictions. The approach follows the spirit of Fast R-CNN that applies bounding-box classification and regression in parallel (which turned out to largely simplify the multi-stage pipeline of original R-CNN [10]).

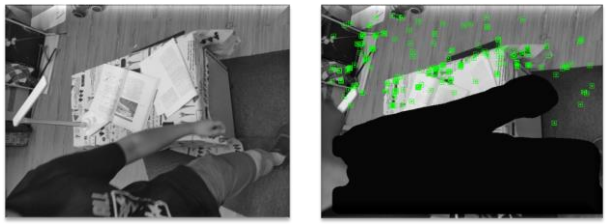
Formally, during training, we define a multi-task loss on each sampled RoI as $L=L_{\text{cls}}+L_{box}+L_{mask}$. The classification loss L_{cls} and bounding-box loss L_{box} are identical as those defined in [8]. The mask branch has a Km2dimensional output for each RoI, which encodes K binary masks of resolution m×m, one for each of the K classes. To this we apply a per-pixel sigmoid, and define L_{mask} as the average binary cross-entropy loss. Foran RoI associated with ground-truth class k, L_{mask} is only defined on the k-th mask (other mask outputs do not contribute to the loss). Since the fully connected layer is not used and RoIAlign is used, the output can be in one-to-one correspondence with the input pixels.

The purpose of RoIPool is to derive a small feature map (eg 7x7) from the ROI determined by the RPN network. The ROIs vary in size, but RoIPool becomes 7x7. The RPN network will propose a number of RoI coordinates expressed as [x, y, w, h], then input RoI Pooling, and output a 7x7 size map for classification and positioning. The problem is that the output size of RoI Pooling is 7x7. If the RoI size of the RON network is 8*8, there is no guarantee that the input pixel and the output pixel are one-to-one. First, they contain different amounts of information (some are 1 pair). 1, some are 1 to 2), and secondly their coordinates can not be associated with the input (1 to 2 of which RoI output pixel corresponding to which input pixel coordinates?). This has no effect on the classification, but it has a great impact on the segmentation. The output coordinates of RoIAlign are obtained using an interpolation algorithm, which is no longer quantized; the value in each grid is no longer using max, and the difference algorithm is also used.

For detecting dynamic objects we intend to use a CNN that obtains a pixel-wise semantic segmentation of the images. In our experiments we use Mask R-CNN[8],which is the state of the art for object instance segmentation. Mask R-CNN can obtain both pixel-wise semantic segmentation and the instance labels. For this work we use the pixel-wise semantic segmentation information, but the instance labels could be useful in future work for the tracking of the different moving objects.

The input of Mask R-CNN is the RGB original image. The idea is to segment those classes that are potentially dynamic or movable (person, bicycle, car, motorcycle, airplane, bus, train, truck, boat, bird, cat, dog, horse, sheep, cow).We consider that for most environments, the dynamic objects likely to appear are included within this list. If other classes were needed, the network, trained on MS COCO [11], could be fine-tuned with new training data.

The output of the network, assuming that the input is an RGB image of size m×n×3, is a matrix of size m×n×l, where l is the number of objects in the image. For each output channel i ∈ l a binary mask is obtained. By combining all the channels into one, we can obtain the segmentation of all dynamic objects appearing in one image of the scene.



(a) Input RGB Frame (b) Frame after Semantic Segmentation

Fig.7. Input and Output Frames of Semantic Segmentation.

3 Experiment Result

We evaluate the overall performance of a SLAM system considering tracking accuracy, initialization quality, tracking robustness, relocalization time and the computation efficiency. The criteria are as follows:

ε_{APE} / ε_{ARE} – *absolute* position / rotation error

ε_{RPE} / ε_{RRE} – *relative* position / rotation error

ε_{bad} – *the ratio of bad poses* (100% – *completeness*)

ε_{init} – *initialization quality*

ε_{RO} – *trackingrobustness*

t_{RL} – *relocalization*

We convert each criteria error ε_{i} into a normalized score by $s_i=\frac{\sigma_i^2}{\sigma_i^2+\varepsilon_i^2}\times 100\%$, where σ_i^2 is the variance controlling the normalization function shape. The complete score is a weighted sum of all the individual scores as:

$$S=\omega_{APE}s_{APE}+\omega_{ARE}s_{ARE}+\omega_{RPE}s_{RPE}+\omega_{RRE}s_{RRE}+\omega_{bad}s_{bad}+\omega_{init}s_{init}+\omega_{RO}s_{RO}+\omega_{RL}s_{RL}$$

TABLE.I. APE, RPE, ARE, RRE, Badness, Initialization Quality for Sequence C1-C11 and D8-D10

Sequence	APE/Score	RPE/Score	ARE/Score	RRE/Score	Badness	InitQuality
C1_test	56.031/0.6258	12.557/0.2226	3.675/0.8026	0.803/0.0949	44.575/0.1771	5.128/0.2284
C3_test	12.554/0.9709	5.980/0.5581	1.202/0.9744	0.603/0.1568	0.000/1.0000	0.426/0.9722
C5_test	13.858/0.9647	9.961/0.3128	1.326/0.9690	0.803/0.0949	0.000/1.0000	4.855/0.2483
C7_test	9.542/0.9830	4.975/0.6460	0.616/0.9931	0.261/0.4981	0.000/1.0000	0.674/0.9449
C9_test	44.686/0.7245	9.642/0.3269	2.378/0.9066	0.626/0.1471	2.992/0.9795	4.336/0.2928
C11_test	44.931/0.7223	12.305/0.2297	3.349/0.8304	0.880/0.0803	5.910/0.9245	3.408/0.4012
D8_test	10.722/0.9786	3.693/0.7680	2.409/0.9044	0.459/0.2429	43.525/0.1842	0.621/0.9527
D10_test	23.784/0.9027	7.792/0.4265	2.635/0.8877	0.582/0.1664	14.945/0.6569	0.333/0.9859


(a)D0_test (Rotating the phone rapidly at some time)


(b) D3_test (A person walks in and out)


(c) D4_test (An object occasionally occluding the camera)

Fig.7.Sequence D0, D3, D4 represent some scenes under tough circumstances(rotating camera, dynamic objects, occlusion)

TABLE.II. Robustness for Sequence D0, D3, D4.

Sequence	Description	Robustness	Score
D0_test	Rotating the phone rapidly at some time.	0.140	0.9962
D3_test	A person walks in and out.	0.050	0.9995
D4_test	An object occasionally occluding the camera.	0.035	0.9998

TABLE.III. Summary Result of All Test Sequence

Sequence	APE	RPE	ARE	RRE	Badness	InitQuality	Robustness	Relocalization Time	Final Score
Summary	0.8590	0.4363	0.9085	0.1852	0.7403	0.6289	0.9985	0.9974	0.9072

It should be noted that not all sequences are evaluated for all the criteria. We evaluated APE, RPE, ARE, RRE, Badness, Initialization Quality for Sequence C1-C11 and D8-D10 in TABLE I.

Most images in Sequence C1-C11 and D8-D10 are collected from indoor texture-rich areas without dynamic objects.

We evaluated Robustness for Sequence D0, D3, D4.Sequence D0, D3, D4 represent some scenes under tough circumstances(rotating camera, dynamic objects, occlusion) in TABLE II.

From the robustness result and the summary result in TABLE III we could see that our method can achieve high accuracy and robustness in tough dynamic scenarios. In our method, only the static parts of the scene is modelled, which is necessary for long-term applications in real-world environments.

4 Conclusion

We have presented a visual SLAM system that, building on ORB-SLAM2, adds a motion segmentation approach that makes it robust in dynamic environments for monocular, stereo and RGB-D cameras. Our system accurately tracks the camera and creates a static and therefore reusable map of the scene.

In the monocular case, our accuracy is slightly better than the results of ORB-SLAM2 obtained however the robustness has been improved and a static map of the scene with an earlier initialization.

By using Mask R-CNN, most of the dynamic objects can be segmented and not used for tracking and mapping. However, there are objects that cannot be detected by this approach because they are not *a priori* dynamic, but movable(e.g., a book carried by someone), we may consider combining a geometric test with

semantic segmentation to solve this problem with monocular cameras.

From the summary result we could see that our method can achieve high accuracy and robustness in tough dynamic scenarios. In our method, only the static parts of the scene is modelled, which is necessary for long-term applications in real-world environments.

REFERENCES

[1] R. Mur-Artal and J. D. Tardos, “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.

[2] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*,2007, pp. 225–234.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robot.*,vol.31,no. 5, pp. 1147–1163, Oct. 2015.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561,* 2015

[5] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A.J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352-1359.

[6] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots—an object based approach," *Robotics and Autonomous Systems,* vol. 55, no. 5, pp. 359-371, 2007

[7] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," (in English), *International Journal Of Robotics Research,* vol. 34, no. 3, pp. 314-334, Mar 2015.

[8] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in ´ *Proc.IEEE Int. Conf. Comput. Vision,* 2017, pp. 2980–2988

[9] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[11] J. Hosang, R. Benenson, P. Doll´ar, and B. Schiele. What makes for effective detection proposals? PAMI, 2015