SALSA: Semantic Assisted Lifelong SLAM for Indoor Environments

Ayush Jhalani
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
ajhalani@cs.cmu.edu

Heethesh Vhavle
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
heethesh@cmu.edu

Sachit Mahajan
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
sachit@cmu.edu

***Abstract*—We propose a learning augmented lifelong SLAM method for indoor environments. Most of the existing SLAM methods assume a static environment and disregard dynamic objects. Another problem is that most feature and semantic based SLAM methods fail in repetitive environments. The unexpected changes of surroundings corrupts the quality of the tracking and leads to system failure. This project aims to use learning methods to classify landmarks and objects as dynamic and/or repeatable in nature to better handle optimization, achieve robust performance in a changing environment, and to re-localize in a lifelong-setting. We propose using semantic information and assigning scores to object feature points based on their probability to be dynamic and/or repeatable.**

***Index Terms*—semantic, lifelong, SLAM, dynamic, repeatable**

I. Introduction

Indoor scenes such as homes, offices, restaurants, malls, warehouses are the target environments for service robots. Mapping and localizing in such environments is not an easy problem and an ongoing field of research. There are several dynamic objects such as humans and pets and other objects such as furniture including chairs and tables which might move frequently and multiple instances of such objects might appear in the scene. Existing SLAM methods such as [1] work well under the assumption of scene rigidity i.e. all the objects in the environment are static. They rely on features which can be present in abundance in a scene but belong to objects that are repeatable or dynamic in nature. This makes such methods susceptible to drift, missed loop closures, and detecting false loop closures in indoor environments. Re-localizing based on maps generated in such environments is also erroneous as these objects are shifted or repeated. To overcome this, our solution augments the existing feature-based ORB-SLAM2 [1] to better handle such objects in a dynamic environment. We propose using semantic information and classify all objects in the environment as follows:

- Dynamic: Objects such as humans or pets which are moving in the scene and should not be a part of the map.
- Possibly dynamic: Objects whose locations are ephemeral such as furniture and other household items. These are classified differently from dynamic objects as they may remain static from a short duration, but might change in position over a longer period of time.
- Repetitive: Objects which tend to produce similar image features, such as multiple doors in a corridor and chairs in meeting room, potentially leading to false loop closure detections.

We use the term lifelong SLAM to focus on a method that makes maps regardless of the dynamicity in the scene and has capability to re-localize in the environment despite changes that are ephemeral. We plan to build upon a feature-based method which combined with semantic information can be used to identify and use "good features" and improve existing SLAM methods.

The novelty comes from the fact that even though some of these principles have been explored independently in previous literature [2] [3], they have not yet been articulated coherently for a practical robotic system. Some of the related works completely remove dynamic features, but reducing the number of detected features leads to tracking failures. This motivates our work on rather weighing the features based on their reliability. This approach of using a heuristic score for classifying features of specific objects and using them in the both front-end and the back-end of a SLAM pipeline has not yet been pursued at a large-scale. We are building upon the work of [1] and our method will work for monocular, RGB-Ds and stereo systems, however, for the scope of this project we have limited our focus only to monocular systems.

II. Related Work

ORB-SLAM [4] is a complete SLAM system that includes features like map reuse, loop closing and re-localization. It relies on highly efficient and fast ORB features to perform tracking to localize the camera while matching features in the local map and minimizing the re-projection error using motion-only bundle adjustment. Local map is optimized using local bundle adjustment and full bundle adjustment is done on the global map after loop closures. ORB-SLAM does pose-graph optimization for drift correction and loop closures. The system uses DBoW2 [5] and g2o [6] libraries for storing the features and performing optimization using Levenberg-Marquardt method. Our baseline SLAM pipeline uses ORB-SLAM2 [1] and work builds upon their code base.

Our code is open-source and is available on GitHub at https://github.com/heethesh/SLAM-Project.

This paper [7] introduces the concept of lifelong visual maps and assigns a set of rules on which a method can be modeled independent of the sensor suite. It presents a system of visual mapping, using only input from a stereo camera, that continually updates an optimized metric map in large indoor spaces with movable objects: people, furniture, partitions, etc. The system can be stopped and restarted at arbitrary disconnected points, is robust to occlusion and localization failures, and efficiently maintains alternative views of a dynamic environment. It defines three phenomenons that a lifelong system must deal with:

- Incremental Mapping: Maps should have capability to be continuously updated.
- Dynamic Environment: Maps should reflect changes in the environment.
- Localization Failure: Re-localization should be a feature.

DS-SLAM [3] is a complete real-time robust semantic SLAM system, which helps reduce the influence of dynamic objects on pose estimation, and meanwhile provide a dense semantic representation of the octo-tree map. Five threads run in parallel in DS-SLAM: tracking, semantic segmentation, local mapping, loop closing, and dense map creation. A real-time semantic segmentation network SegNet is combined with moving consistency checks to filter out dynamic portions of the scene such as people walking. The matched feature points would be removed out of those detected dynamic regions, and thus improve the performance of robustness and accuracy in dynamic scenarios. The paper shows some promising results, however it only targets objects that are dynamic in the scene. The authors use optical flow to detect dynamic objects and classify them as outliers.

This work [8] demonstrates how 3D cuboid object detection and multi-view object SLAM can work in static and dynamic environments and can improve each other's performance. This is done by generating a cuboid proposal from the 2D bounding boxes and using vanishing point sampling. The proposal is scored using a bunch of heuristics, which include semantic labels, histogram of gradients, Harris corner detection, and selected based on the alignment with images edges. Bundle adjustment has been used for jointly optimising poses of cameras, objects and points. Now, the novel thing in this work is that it does not reject the dynamic regions as outliers The 3D object representation and motion model constraints is used to improve camera pose estimation. It has been mentioned that 2D Lucas-Kanade sparse optical flow algorithm and visual object tracking algorithm, and verified that the latter is more reliable in case of large pixel displacements. The latter is then used for 2D bounding box tracking and predicting its next position based on the previous frame. This is then used to optimize the camera pose estimation further.

This paper [9] mainly addresses the problem of loop closure recognition based on low-level features which is often viewpoint-dependent and subject to failure in ambiguous or repetitive environments and handles sets of easily recognizable landmarks belonging to similar classes which would lead to view-independent unambiguous loop closure. Both of these are relevant to key goals of our project statement. They make use of image features and inertial measurement to perform visual odometry based tracking separately but do not recover the geometric structure. The map instead consists of objects (bounding box, class, and confidence). An expectation-maximization based optimization framework is outlined to handle data association (E-step) and pose-graph optimization of the landmarks.

III. Method

In this section, we provide a brief overview of the overall architecture of the system and the key changes made to handle dynamic, possibly-dynamic and repeatable objects in the environment. We have modified several modules in the original ORB-SLAM pipeline as shown in figure 1. We build our system upon ORB-SLAM2 with another thread running in parallel which uses Mask-RCNN [10] to segment object instances and generate score maps. These objects are assigned a heuristic score based on an existing database that contains the associated score for attributes of an object - dynamic, possibly dynamic and/or repeatable. Maintaining a separate database allows the system to be modular for different environments. These heuristic scores are used in both the front-end and the back-end to guide the system behavior. The features lying on dynamic objects such as humans are culled, and not added to the map. The features on the other objects are kept to ensure enough keypoints for tracking. The score assignment and feature point culling is done on a frame-to-frame basis. During pose optimization and bundle adjustment, the information matrix in the cost function is scaled up depending on the heuristic score of features. The heuristics are also used to determine which features are reliable and can be used for generating the bag-of-word representation of a frame. Generating more reliable representation of a frame enables our system to perform loop closures in a more robust fashion.

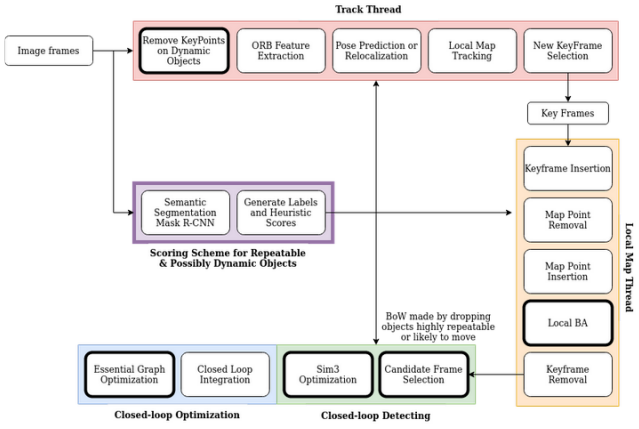

Fig. 1. SALSA pipeline - modified ORB-SLAM pipeline with our changes highlighted.

A. Object Classification and Scores

We identified several common objects that appear in our daily life from the COCO dataset [11] and pre-assigned the dynamic, possibly dynamic, and repeatability scores for each of these objects. Figure 2 shows the table of scores assigned for some objects. Object that are completely dynamic such as people and pets are assigned a binary score as feature points on these regions are directly removed. Possibly dynamic and repeatable objects scores are just probability values that range from 0-1. Most objects have been assigned with both a possibly dynamic as well as a repeatable score. For example, a chair might be possible dynamic with a high score and there could be several other similar chairs in the scene that might produce similar feature key-points. We use a segmentation network trained on the COCO [11] dataset to first classify the objects and then encode these three sets of scores in the RGB channels of the segmented images (score maps) as shown in **??**. Dynamic object score are encoded in the red-channel, possibly dynamic in the green-channel, and repeatable scores in the blue-channel of the score map. We used Mask-RCNN [10] from Detectron2 [12] to perform the segmentation here. The masks are slightly padded so that the feature points that lie on the edges of certain objects of interests are guaranteed to fall correctly within the masks. Using this semantic information, the RGB value of a feature point in this score map is looked up and we assign the scores to all extracted ORB features. As discussed above, if a feature point lies in the dynamic region (red-channel), the point is outright culled. Remaining features are given scores looked-up in the green (possibly dynamic) and blue (repeatable) channels of the score maps.

Class	Dynamic	Possibly Dynamic	Repeatable
person	1.00	-	-
bicycle	0.00	0.75	1.00
car	0.00	0.70	0.90
bench	0.00	0.00	0.90
cat	1.00	-	-
backpack	0.00	0.90	0.50
bottle	0.00	0.70	0.80
chair	0.00	0.70	1.00
couch	0.00	0.20	0.70
bed	0.00	0.10	0.80
tv	0.00	0.90	0.90
cell phone	0.00	1.00	1.00
book	0.00	1.00	0.90
clock	0.00	0.20	0.90
vase	0.00	0.30	0.70
...	...	...	...

Fig. 2. All 80 classes from the COCO [11] dataset are assigned scores. This list is only a subset of some interesting objects. Some objects are exclusively labelled as dynamic objects.

B. Optimization

We modify the local and global bundle adjustment, pose optimization, and relative Sim(3) pose optimization in the ORB-SLAM pipeline. First, the possibly dynamic score s_p and repeatable score s_r are averaged using a weighing factor w. This weighted score is then subtracted from 1 so that feature

Fig. 3. Mask-RCNN [10] was used to segment object instances and the scores were mapped according to the scale shown on the right. The left column shows score maps for the OpenLORIS [13] cafe1-2 sequence and the right column shows the score maps for TUM-RGBD [14] walking_static sequence.

points with a higher probability of being dynamic have a lower weight in the optimization. Finally, we scale our score by another factor w_s to control the importance of these scores in the optimization as shown below in equation (1).

$$s_d = (1 - (ws_p + (1 - w)s_r)) * w_s \tag{1}$$

In each of the optimizations mentioned above, we multiply this weighted score s_d with the information matrices in the cost functions. The information matrix in the ORB-SLAM pipeline is just the inverse of the 2x2 covariance matrix of a feature point at a given octave level in the image feature pyramid. The updated equations are shown below and the notations are the same as described in [4]. Equation (2) shows the update bundle adjustment cost function, equations (3) and (4) correspond to the updated cost functions of pose optimization and relative Sim(3) pose optimization respectively.

$$C = \sum_{i,j} \rho_h \left(\mathbf{e}_{i,j}^T s_d \mathbf{\Omega}_{i,j}^{-1} \mathbf{e}_{i,j} \right) \tag{2}$$

$$C = \sum_{i,j} \left(\mathbf{e}_{i,j}^T s_d \mathbf{\Lambda}_{i,j} \mathbf{e}_{i,j} \right) \tag{3}$$

$$C = \sum_{n} \left(\rho_h \left(\mathbf{e}_1^T s_{d1} \mathbf{\Omega}_{1,i}^{-1} \mathbf{e}_1 \right) + \rho_h \left(\mathbf{e}_2^T s_{d2} \mathbf{\Omega}_{2,j}^{-1} \mathbf{e}_2 \right) \right) \tag{4}$$

C. Loop Closures

ORB-SLAM uses a bag-of-words representation for storing frame information in a memory-efficient manner. A loop closure is detected by comparing bag-of-words representation and the geometric consistency of those words (it is also known that the frames captured close in time are not taken into the consideration for loop closure). The presence of possibly dynamic objects and the repetitive objects introduces similar feature descriptors consequently, words that can lead to erroneous loop closures. Our approach uses the heuristics to determine the reliability of a keypoint that can be chosen to create a feature descriptor. We use Russian Roulette culling to determine if the point is required to be kept or dropped. In

the Russian Roulette approach, we generate a random number from a uniform distribution $x \in U[0, 1]$ and compare it against a predefined threshold for each set of scores. Feature points that have a higher dynamic score, therefore, have a higher chance of being dropped. Figure 4 shows the features on the rolling chair (marked red in the left image) affect the bag-of-words representation of the frame. This bag-of-words representation can be much different when the rolling chair is absent as it does not contain any of the feature marked in red. Removal of such features classifies the frames to be similar and potential site for loop closure.

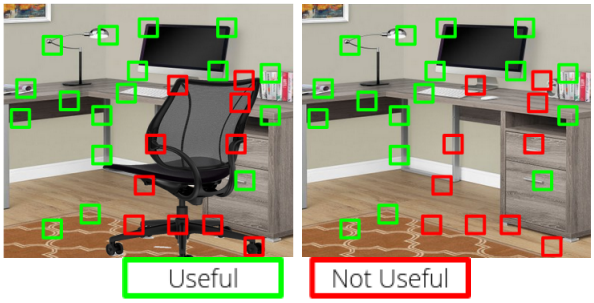

Fig. 4. The image on the left represents the features on static objects marked in green and features on possibly-dynamic objects marked in red. The image in the right shows how the BOW representation changes due to absence of the possibly-dynamic objects.

IV. Experimental Results

To test out our hypothesis, we require scenarios where there are indoor objects such as furniture as well as dynamic objects such as humans in the scene. The first dataset we used was TUM-RGBD [14] dataset's dynamic objects category. We found that the ORB-SLAM2 algorithm acquired a lot of drift in these datasets due to the presence of dynamic objects. We tested on two sequences set in an office environment - 'freiburg3_walking_static' where the camera remains static and 'freiburg3_walking_xyz' where the camera moves. These sequences are intended to evaluate the robustness of visual SLAM and odometry algorithms to quickly moving dynamic objects in large parts of the visible scene.

To test our other parameters we needed a dataset which had a lot of possibly dynamic and repeatable objects where we could map and localize and then re-localize after certain changes were made to the scene. For this task, we used OpenLORIS [13] dataset's cafe sequences which consists of a person traversing a cafeteria at different times with change in the placement of objects such as chairs, bottles, and several dynamic objects. Both the datasets provided us the RGB camera feed as well as the ground-truth. We restricted our scope to monocular SLAM.

Segmentation: We could accurately classify objects in the scenes using Mask-RCNN [10] to generate the semantic scores maps as shown in figure 3.

Score Assignment: We manually assigned scores for all 80 classes of objects in the COCO [11] dataset as shown in figure 2. The scores assigned were based on heuristics

Fig. 5. Results of feature point culling on OpenLORIS [13] cafe1-2 sequence.

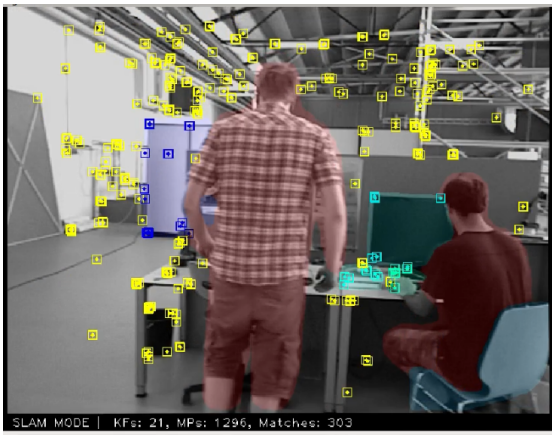

Fig. 6. Results of feature point culling on TUM-RGBD [14] walking_static sequence.

and a knob that can be tuned to improve performance of the system. Another parameter to tune is the weight given to each of dynamic and repeatable scores which was used in cost functions of the optimizers. Since our environments roughly had an equal number of both types of objects and there was a huge overlap in them, i.e. object being both possibly dynamic and repeatable, we weighed them equally by setting $w = 0.5$ in equation (1). The scaling factor w_s was to 1.0. The results of feature point culling and score assignment is shown in figures 5 and 6. Dynamic feature points in the red mask region are removed. Possibly dynamic and repeatable feature points have a shade a green/blue and rest of the static object features points are labelled in yellow.

Loop Closures: We were able to accurately re-localize the camera in the the cafe sequence of the OpenLORIS dataset which can be seen in the videos. On average, we dropped 40-90 keypoints using the thresholds and sampling methods which have been described above. The tunable parameters include the the thresholds for accepting keypoints possibly dynamic which was kept 0.7 and repetitive objects which was kept 0.7, and the Russian Roulette threshold of 0.6 that

allowed us to randomly decide if we want to keep a given keypoint for generating ORB-features and thus, our bag-of-words representation. The parameters were assigned based on the assumption that possibly-dynamic and repetitive objects are equally likely to spoil our loop closures.

Performance Metric: We chose Absolute Trajectory Error (ATE) as the error metric. The absolute trajectory error directly measures the difference between points of the true and the estimated trajectory. ATE is well-suited for measuring the performance of visual SLAM systems, and we have computed the RMSE for evaluating our results. In contrast, the RPE is well-suited for measuring the drift of a visual odometry system, for example the drift per second.

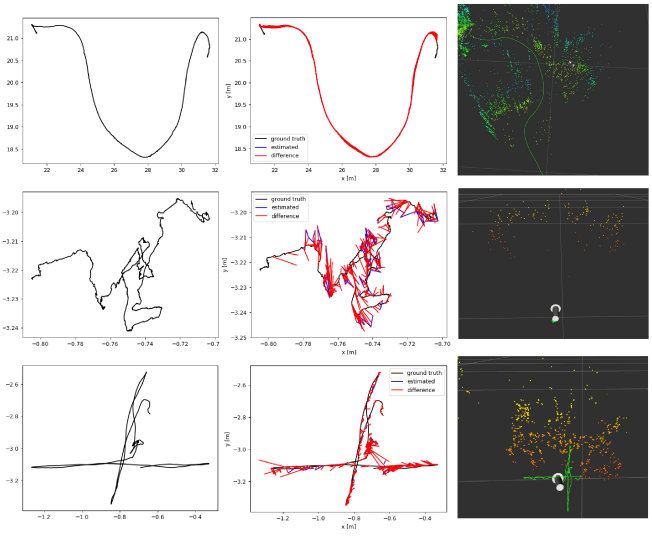

Fig. 7. Results of tracking on OpenLORIS [13] cafe1-1 sequence, TUM-RGBD [14] walking_static and walking_xyz sequences. The columns depict the ground truth trajectory, estimated trajectory results, and tracking and mapping results visualized in Rviz.

A. Evaluation using TUM RGB-D Dataset

ORB-SLAM2 performs poorly in both freiburg3_walking_static and freiburg3_walking_xyz due to the two men walking around and moving chairs. We outperformed ORB-SLAM2 in both sequences. This also validated our assumption that the features of dynamic objects cause erroneous estimation and should be removed.

TABLE I
RMSE Absolute Trajectory Error (ATE) on TUM-RGBD Dataset

Sequences	ORB-SLAM2	DS-SLAM	SALSA (Ours)
walking_static	0.4030m	0.0081m	**0.0059m**
walking_xyz	0.1780m	**0.0247m**	0.0521m

The trajectories and ATE plots are shown in figure 7. It appears that the static scene had small movements in the chair position, here our methods outperformed both DS-SLAM [3] and ORB-SLAM2.

TABLE II
RMSE Relative Pose Error (RPE) on TUM-RGBD Dataset

Sequences	ORB-SLAM2	DS-SLAM	SALSA (Ours)
walking_static	0.2162m	0.0102	**0.00829m**
walking_xyz	0.4124m	0.0333	**0.02951m**

B. Evaluation using OpenLORIS-Scene Dataset

We chose two sequences of a cafeteria where we aimed to initialize in the first sequence and re-localize in the second one. We did lose tracking in first sequence due to a pure rotation in a scene with reduced features, however ORB-SLAM2 lost tracking about one second before our method does. We successfully re-localized in the second sequence and this can be seen in the project video. These results strengthened our belief that unlike dynamic objects whose features should be removed, the features of possibly dynamic and repeatable objects should not be removed outright rather using a weighing method, we should bias our system to use better features as fewer features will cause tracking failures in feature-based methods such as ORB-SLAM.

TABLE III
RMSE Absolute Trajectory Error (ATE) on OpenLORIS Dataset

Sequences	ORB-SLAM2	SALSA (Ours)
cafe1_1	0.0777m	**0.0464m**
cafe1_2	0.0813m	**0.0588m**

It was clear that removing dynamic objects increased the performance of our method, to further evaluate we tested out the same sequence again while disabling our optimizer and DBoW2 changes i.e. just removing dynamic features against removing dynamic features and biasing optimizer based on heuristic scores.

TABLE IV
RMSE Absolute Trajectory Error (ATE) on OpenLORIS Dataset

Sequence	Partial SALSA	SALSA
cafe1_1	0.0596m	**0.0464m**

Note: Partial SALSA performs only dynamic feature removal using semantic information.

Carrying out all these experiments, this validated our hypothesis that modifying the optimization and loop closure threads helped improve the performance. The project did pan out like we expected and our assumptions were correct and the results are our proof-of-concept. The videos of our results are uploaded on YouTube here: TUM-RGBD sequences and OpenLORIS sequences.

V. Challenges

One of the major challenges was finding a suitable dataset that has RGB data for indoor environments along with dynamic objects and ground-truth semantic segmentation maps. To evaluate our work, we needed abundance of repeated and dynamic objects, we had initially planned to collect our own

data, however, due to campus being shut down we could not do that. Since we are defining the scores for dynamic and repeatable objects based on heuristics, there is no guarantee that they will generalize well. They will have to be tuned for each environment and we wanted to do parameter tuning and further testing on custom data which would have accurately evaluated the impact of our work. Defining a suitable score weighting function for the optimizer was also laborious, we ended up scaling up the uncertainties using the heuristics while assigning equal weight to both of the scores. Since our approach required changes in various threads and functions, navigating and working with the ORB-SLAM2 code base and g2o, and DBoW2 libraries, was overwhelming as we never used them before. We had initially planned on using RGB-D data and had acquired a camera for the same, however we switched to the monocular approach as one the datasets we were using had unreliable depth calibration information. Furthermore, integrating the semantic segmentation thread in real-time caused a huge drop in our frame rate, hence we decided to pre-compute the semantic score maps for our experiments. These challenges did push us off the planned timeline, but we were able to recover and achieve all of out target goals given the circumstances.

VI. Conclusion

In this project, a a lifelong SLAM method for indoor environments which exploits semantic understanding of the scene is proposed. Semantic information is used to score key points based on their possibly-dynamic and/or repeatable nature and this is used to weigh the features. We observed that both tracking and mapping are improved by ignoring dynamic objects. Our method managed to perform tracking in case of a static camera where ORB-SLAM2 fails due to the presence of dynamic objects. We observed that map points are robust when dynamic key points are removed and this helps in faster re-localization. We tackled the scene rigidity assumption of existing SLAM systems as well as feature deficiency caused by culling too many features by assigned scores rather than removing them completely, hence, performing better at loop closures and re-localization. This knowledge helps us move one step closer towards building lifelong SLAM systems.

VII. Future work

Currently, we are classifying only "things" in the environment and this can be extended to classify "stuff" as well which includes categories such as terrain, pavements, and buildings that might help extend our method to outdoor environments as well. In order to generate lifelong maps, we can use the heuristic scores to weigh map points and hence change maps if any shift in objects is detected and use the semantic information to make maps more meaningful such as dense semantic octomaps. Semantic segmentation is expensive and not perfect, therefore, robust outlier rejection must be employed to overcome these limitations and its better that this expensive task is done intermittently instead of every frame and interpolate the results using motion model for immediate frames. Visual SLAM is vulnerable to pure rotational motion and it is better to utilize additional sensors such as inertial sensors to overcome this limitation. Our assumption that objects of certain classes will be dynamic can lead to loss of features points, we need to employ methods such as optical flow to determine whether the object was actually dynamic thereby retaining more features. As mentioned previously, we would like to collect our own data and tune the knobs to get more in-depth understanding on how these changes impact the performance.

Acknowledgment

This work was carried out as the final project for 16-833 Robot Localization and Mapping (Spring 2020) at Carnegie Mellon University. We thank Prof. Michael Kaess and our TA, Sudharshan Suresh for their advise.

References

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[3] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1168–1174.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[6] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3607–3613.

[7] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1156–1163.

[8] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[9] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[11] 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 *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[12] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[13] X. Shi, D. Li, P. Zhao, Q. Tian, Y. Tian, Q. Long, C. Zhu, J. Song, F. Qiao, L. Song, Y. Guo, Z. Wang, Y. Zhang, B. Qin, W. Yang, F. Wang, R. H. M. Chan, and Q. She, "Are we ready for service robots? the openloris-scene datasets for lifelong slam," 2019.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.