IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

vSLAM using Deep Learning for Semantic Mapping

Vishwanath S. Mahalle[1], Abhishek A. Daberao[2], Mujahid Ahmed Sayyed Salahuddin[3], Shreyas S. Patil[4]
Assistant Professor, Computer Science and Engineering[1]
Students, Computer Science and Engineering[2,3,4]
Shri Sant Gajanan Maharaj College of Engineering, Shegaon, India

Abstract: *As the domains of deep learning and computer vision are developing, we are trying to find new applications as well as utilization in old problems. In this paper we are trying to apply deep learning methods to augment the traditional robotics problem of Simultaneous Localization and Mapping (SLAM). Most traditional SLAM methods build metric maps only, but we are trying to build a semantic map which identifies individual objects in the map meaningfully. We are using basic RGB-D SLAM to start with for the localization part, along with a deep learning-based module for the object detection and recognition. These together provide a semantic map for the environment. For better computational performance and efficiency, we are using OctoMap for the map data structure. To be qualified, our approach has to yield good results in localization, mapping as well as in object detection.*

Keywords: SLAM, vSLAM, Semantic Map, Robotics, Deep Learning, etc.

I. INTRODUCTION

The kind of robots that hold the most potential are the ones which can move through space (mobile), and operate independent of external control (autonomous). SLAM or Simultaneous Localization and Mapping has been a cornerstone robotics problem since we realized that such kinds of robots need a good knowledge of the environment (map) and its location within it.

However, the agent can only fully interact with the environment if it has a notion and perception of the individual objects in it. For instance, to execute a complex command such as "Find apple on the sofa in the living room", a traditional SLAM map won't be much useful. Thus, along with a metric map, one which tells the locations of landmarks in space, we also need to build a semantic map.

II. PREVIOUS WORK

A. Simultaneous Localization and Mapping

The SLAM algorithms have developed over the years such that, now we can get real-time results. However, traditional algorithms only focus on constructing a metric map of the environment built of points and planes. A metric map cannot be used to perform complex tasks that require meaningful interpretation of the environment, such as the task of finding a green cup in the kitchen. Thus, the problem of finding a meaningful interpretation of the environment is quite important. The information acquired about the environment can help to improve location prediction accuracy and the metric predictions can help build better semantic models.

B. Detecting and Segmenting Objects in Environment

We first need to be able to recognize objects in the scene in order to build a semantic map. There are person machine learning based methods in computer vision that can be used to recognize and detect objects. There are also deep learning approaches which can be used to detect objects with varying shape and size but belonging to the same type.

The model of RCNN yields suggestions of objects through selective search. A combined network with AlexNet for feature extraction, and SVM for classification can be used. However, it is quite slow.

The problem can be solved by using Fast RCNN with feature mapping feeding into a dense layer with SoftMax instead of pooling and SVM. This is faster than RCNN but still slower. We can use Faster RCNN, which is based on a region proposal technique for generating suggestions of interesting objects in the scene.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

C. SLAM with Semantics

We know that semantic information can be very useful for robots. The SLAM part of the system is used for localization and mapping while additional object detection and recognition modules are used to create a semantic map.

A SLAM system with semantics can be classified in two types:

1. Using traditional object detection methods. For example, monocular SLAM.
2. Using deep learning methods for object detection and recognition.
3. The goals of the robot are as follows:
4. The agent has to learn the meaning of environment in its operation, so each pixel has information about the object it depicts,

It is very expensive computationally to do semantic classification per pixel.

III. THE SYSTEM

A. Analysis of Traditional SLAM

The metric part of the SLAM problem is pretty developed as compared to the other. But they only work as required in situations of static objects or small dynamic objects. When dealing with small dynamic objects, it is easier than bigger ones as not much deviation is produced.

But the type of SLAM which uses Features, is vulnerable to bigger dynamic objects. Hence most feature-based systems assume strongly that the objects in the scene will be mostly static. Also most traditional SLAM methods don't allow the mechanism of semantic interpretation, which reduces the ability of the robot to do only simpler tasks and not complex tasks. We need a more powerful system that can accommodate the required object detection and recognition. For instance, if the robot is compressed to fetch a red apple, it first needs to find the red apple in the scene. Only after correctly detecting and identifying the red apple, the robot can plan a navigation strategy to fetch it as compersonded. Also, the system has to map the environment as quickly as possible.

The Octomap structure is totally derived from the OcTree shape useful for finding and building, at the same time as factor clouds most effectively shop every factor with none structures. The Octomap structure allows to integrate both the position and color information with the semantic information. While traditional clouds only store the color and position information. Also, this map relies on various techniques to build a probabilistic map. Hence, we are choosing this particular map structure to facilitate our navigation system.

B. Analysis of ORB-SLAM

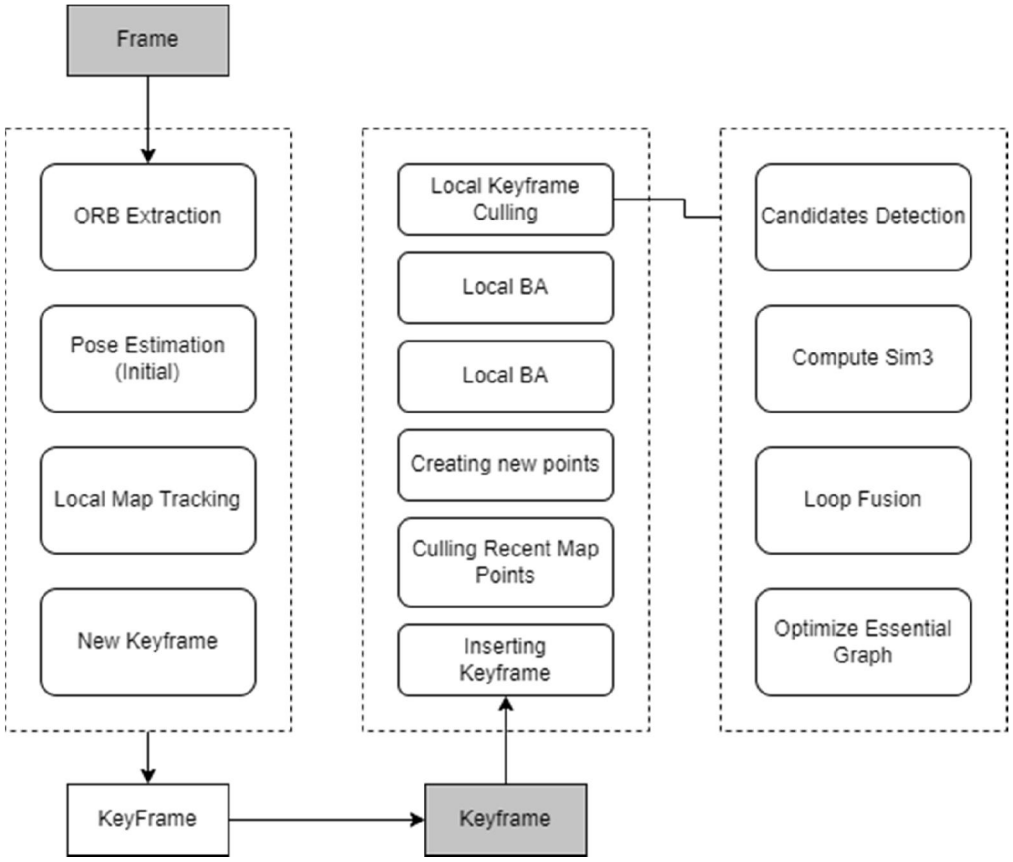
Orb-slam 2 is a widely used and popular SLAM system using a feature-based approach. As shown in Figure (1), it consists of three parallel threads:

1. closing loop,
2. local mapping, and
3. tracking

The thread used for tracking works for determining the position of the camera. with each live frame and decide when to insert the key. In the follow-up sequence, it performs comparison of features with previously obtained features image and optimizes the posture using a beam correction (BA) algorithm.

If track is off, it does the global transposition with the Bag of Word, then looks up points on the map by reprojecting and optimizing the state with local coordinates. Later, the tracking sequence can select if another key-frame can be added. After receiving another key, the mapping string will find a new key-frame using triangulation. It then optimizes the relative key-frame pose and maps the points against BA. In the end, excess keys and bad quality map coordinates will be discarded. Closed loop stream responds to loop close with every key-frame. If a loop is detected, the similarity transformation is calculated, representing the drift accumulated in the loop. Then the two sides of the loop are aligned and the duplicate points are merged. Finally, optimization of the graph setting on the same constraints is performed on the order of for global consistency.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252



C. Semantics Based SLAM

In the suggested slam method, orbslam2 is implemented for mapping the environment and localisation of the robot with all rgbd frames. The tracking stream is used for locating the main frame instead of the reference frame to reduce the influence of moving objects. The local mapped stream adds some key to generate the semantic message because the semantic message extraction cannot satisfy the real-time execution requirement. Once the orbslam2 is done with finding the key, the output is given to the YOLO network to find possible semantic information. We are going to implement a smaller YOLO network trained on mscoco dataset, which helps us identify 80 unique object types.

Next, item regularization primarily based totally on crf is used to accurately assess the possibilities of every item computed through yolo. In this system, constraints among gadgets are computed in step with the records of ms-coco dataset, and its miles are then used to optimize the item possibilities computed through yolo detection. When correct labels of every item are captured, a clear out system is used to offer greater solid capabilities and get rid of the risky capabilities which might be constantly found at the transferring gadgets.

At the equal time, the transient gadgets are created, which comprise factor clouds produced through projection. When a transient item is generated, we use records affiliation module to determine both to create a brand-new item or accomplish it with present item withinside the map in step with the matching score. In records affiliation, with a view to discover correspondence among existing gadgets and transient gadgets, we first construct courting among key and gadgets. Then, the kd-tree shape is used to boost up the computation of the matching score. When the prevailing gadgets may be mixed with the transient gadgets, the previous may be up to date with the brand-new detection through a recursive Bayesian system. Finally, map era makes use of factor clouds saved in gadgets to generate maps primarily based totally on octomap, that's expanded through multi-threads awareness and rapid line rasterization algorithm.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

D. Objects and Key-frames

To integrate the semantic concept in the ORB_SLAM2 framework, we construct the relationship between key and objects by referring to the implementation method between key and map coordinates, which already exist. in ORBSLAM2. In the ORB system a key-frame records the map coordinates found in the visual input, and each map point stores the observed key of the map point sequentially. We can thus establish a relationship between key and do some optimizations, such as analyzing if keys are redundant or deciding if a point on the map is high quality or not. Thus, we construct the pair between key and corresponding object as follows. To our knowledge, such object consists of:

1. For each point the coordinates on the image.
2. Class labels from a fixed sized set as well as a score of confidence obtained from the Bayesian method.
3. This object can be observed by key.
4. Kdtree structure is generated through point clouds of objects, used for fast searching.
5. The class label to which this object belongs.
6. Multiple observations.

Each key-frame must store:

1. The corresponding RGB image used for object detection.
2. Corresponding depth images are used to create point clouds.
3. Objects observed in this key-frame.

As such we can pair the keys with the map coordinates.

IV. MAP WITH SEMANTICS

This section is dedicated to explanation of the modules of this system, including Advanced SLAM, Object Detection, Object Modification, Temporary Object Creation, Data Binding, Object Model Update and creating a map.

A. Improvised Version of SLAM

The proposed semantic SLAM is built on the basis of ORBSLAM2. In ORB_SLAM2, the tracking sequence positions the camera at each frame in four steps.

Extraction of ORB features from RGB image.

Using the features from ORB to match with the frame of reference and compute the camera orientation and calculate the number of matching map coordinates.

Third, the camera orientation is again optimized with the corresponding local map coordinates searched in the relative key-frame.

Finally, the tracking thread decides whether a new key-frame should be inserted based on certain guidelines.

We will modify the thread tracker in order to integrate the SLAM and machine learning based detection system.

To reduce the impact of Smart Objects, the second stage of the tracking feed is changed to tracking with a key-frame instead of a reference frame. If SLAM follows the frame of reference, large dynamic subjects allow us to easily calculate the camera state. The error is induced due to the fact that SLAM will start tracking the dynamic object affecting the results. However, if the SLAM tracks work on the key, it can still calculate the correct camera pose before inserting a new key-frame. Using the new key-frame will help us to avoid this as it does not contain the dynamic object. We will use the Levenberg Marquardt method from the G2O library for the optimization of the robot state from the image. However, for good results this method will need pretty accurate initial estimates. Thus, as beginning G2O values we will use a continuous model for motion.

The third step of the tracking process is modified to compare the number of matches with the results of the second step to determine if the current tracked frame is lost. In ORBSLAM2, the paired internals are compared to a constant value in the third step. It is easy to lose tracking when the camera is moving fast, as the ORB feature points of the current frame can only match the map coordinates observed by the last frame. Here the total number of detected map coordinates will be lower than a fixed value. Therefore, the third step should compare the number of matched elements with the number of matched elements calculated using the final image. In the second step, we will use the function to estimate the number of coordinates the match among the latest and previous key.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

Modified track wire to create more key. In ORBSLAM2, the tracking module mainly tracks by key-frame, less and less features can be seen as the camera moves, resulting in easy loss of tracking. Therefore, more key are needed to generate more map coordinates and this is useful for proper tracking.

B. Detecting Objects

In our system of SLAM, the objects are useful entities that can provide information for the map and also improve location accuracy. Thus, we need a powerful and fast detection method. There are some deep learning-based methods like FCN and DeepLab that can yield segmentation at pixel level, but even object bounding boxes provided by RCNN, Fast RCNN and Faster RCNN are enough. Also, they suggest more than one region for object detection resulting in even suggesting same regions more than once. This is therefore a drawback as it limits the results from being used in real time for SLAM. For instance, the rate of 5 fps handled by Faster RCNN is still very less for the likes of SLAM. But the YOLO network (You Look Only Once) divides input images into 'nn' grid then each such grid is detected only once. This change imparts the YOLO algorithm at a fast rate of upto 45 fps, thus we can use this algorithm for our case.

We are going to use YOLO network and use COCO dataset for training it. There is another option named PASCAL VOC, but it only provides 20 different objects as opposed to the 80 classes of COCO dataset.

Using a powerful GPU like Titan X can yield upto 4090 fps using YOLO network. The GT730 comes with 1 GB graphics memory, inadequate for the functioning. Adjusting the tiny YOLO weights can help with reducing the memory requirements.

C. Regularizing Objects in the Scene

YOLO network does not consider context among objects, while it does easily find the semantic labels for objects. It has been studied that the information of the context can be useful for semantically segmenting, but it does require the availability of referential meaning of any object. In simpler words, the other objects in the scene should affect the selection of label for any particular object. Here is an example to demonstrate this, suppose we are in a scene of kitchen, we label some items as knife, orange, lemon and red ball. According to the context, we can say that the red ball might more probably be an apple. This shows that when implemented correctly in case of multiple object detection, context information can be very useful.

If we were to not use a system for categorizing these items without the use of context information, the labels would be kept same and would not be changed. If we were to not use the classifying system with a content information system, the labels above would have been the ultimate ones. But using context gives us additional information to correctly classify the object correctly. Forcing the additional information, the label of red ball changes to apple, which is more probable in the given context.

We use a special algorithm called Conditional Random Filed or CRF, to incorporate the local information of the object in the image with multiple classifying models. This can be modeled as a situation of posterior maximization. Unary potentials can be made to imitate how a pixel or region belongs to a class, and along with that a pair potential between two patches or pixels. The model of CRF that we are using, consist of all unary and pairwise potentials as well as weights. If we limit the range of local region by keeping the pairwise potentials in 4-8 neighbors, we won't capture much context. Considering the problem, person CRF approaches were designed lately. Allowing longer ranges imparts the ability to create more detailed labeling as compared to the sparse alternative. The approach of using such dense CRF along with Gaussian kernel potentials is becoming popular. We create such a probability based dense CRF.

D. Generation of Objects

The information about the label of any object in the scene also helps us to eliminate the problems caused by dynamic objects in the scene, as we can only allow particular classes of objects to be considered as features and points. The objects are first put in one of the classes, either dynamic (cat, car) or static (apple, plant). While building map we ignore the objects in the dynamic class for selection of map points.

Once the objects are labeled and features and map points are filtered, we can proceed with generation of intermediate / temporary objects with additional information such as the size, type and confidence score of the object as well as the points in the cloud encompassed by the object. The RGB-D generated clouds do have some problems with noise though. We use simple statistical techniques to remove noise which deviates from the mean by large value.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

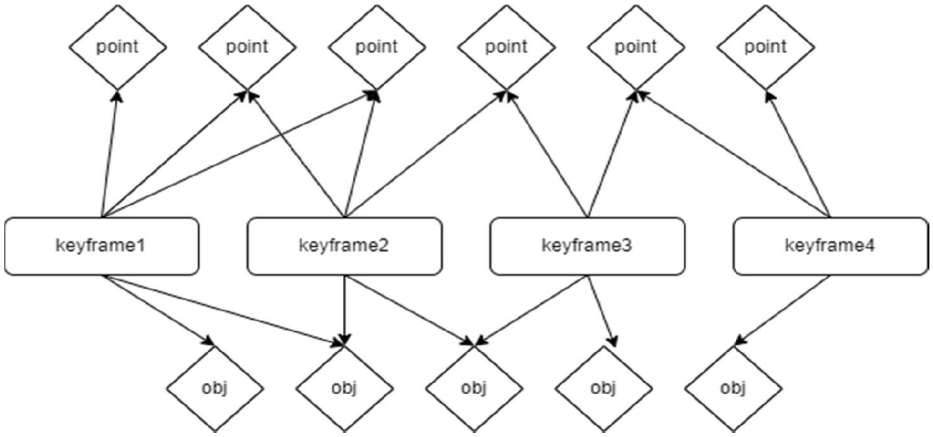
We are going to sample the points in clouds down to 5mm in our version. After all that is done, we find out if the objects are already in the map or are new observations.

E. Associating Data

Associating data is a very crucial step in the process of general SLAM as well as the version we are building. This process involves creating an object for the temporary object in the map. We start by choosing a matching object for each temporary object. We find the similarity between map points and key frames, which allows us to associate an object in map to the temporary object. Such frames are more probable to contain an object of interest similar to the one in map.

We can avoid some undesirable circumstances produced by moving items by using the key-object relationship. A person goes past his desk in an office, for example, and three photos are picked as important, as illustrated in Figure (2). We detect the television screen from the first key-frame, which depicts a person starting to block a white television screen. The second key-frame illustrates that a person is blocking the television screen, and the television screen is not visible in this key-frame. The third key-frame, on the other hand, shows the person stepping over the television screen, and it returns. It generates a temporary object labeled the television screen in the third key-frame. According to the key-frame and object relationship, we do observe that the first key-television frame's screen is a potential object for the temporal object seen in the second key-frame. The map coordinates of the first key-frame and the third key-frame are nearly identical, indicating that the first key-frame is one of the relative keys for the third key-frame. Because the last key-frame does not contain the white television screen, the transient object (the television screen in the third key-frame) will be handled as a new object according to the data link if we utilize the object that the last key-frame observed (it is the second key-frame in this situation).

Of all the objects available at disposal when selecting the object, we choose the one with most similarity to the intermediate object. We use Euclidean distance based nearest neighbor method to choose the most similar object, along with k-D tree for better performance. The matching pairs allow us to estimate scores between temporary objects and the selected candidate. IF no similar object is found, the object in the current frame is considered to be a newly observed object.



F. Generating Map

In this implementation, every key-frame contains points from the 3D cloud as well as the segmented point clouds. If we project the 3D points as per the poses, we can generate the map. But such point cloud-based map is of very less use for complex activity like planning and grasp point selection, as they do not incorporate any structures for storing points, and this turns out bad for searching. Also, the point-based clouds will not help us to discern among empty, unknown area and won't help us to remove any noise. The structure of Octomap that we are using in this implementation allows us to store the point clouds and also enables us to distinguish between unknown and empty areas and removes noise.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

V. EVALUATION

A. Evaluation of Tracking

Because it delivers correct truth to the footage, the TUM dataset is a useful dataset for testing camera localization accuracy. It features seven sequence types shot at 30 frames per second and 640 x 480 resolution by the RGBD camera. Handheld SLAM sequencing, Robot SLAM sequencing, Structure vs. Textures and sequences were the only methods we employed. Among the seven series, this is the most dynamic object because these four reflect the majority of everyday scenes. These strings also contain a variety of objects, which can provide more semantic information than other sorts of strings. The mobile SLAM sequence was recorded by a human, resulting in a complex and unstable trajectory, but the Robot SLAM sequence recorded by a real robot should be stable and straightforward. The Structure vs Texture sequence, on the other hand, is primarily concerned with the environment's textures and structure. However, they are all experienced in a scene that is devoid of moving objects. Furthermore, the trajectory of moving items is manually recorded, but when the recording contains dynamic objects, the trajectory becomes unstable in such settings. In the benchmark, we employ different SLAM RGBD for comparison. Each string is analysed five times, and the accuracy of its localisation is determined using the root mean square error (RMSE).

B. Evaluation of Localization Accuracy

First, we put enhanced SLAM to the test to see how accurate our localization is. In the benchmark, we used an RGB-D camera to compare ORBSLAM2. The benchmark's mean RMSE error is shown in Table I. In a static setting, the RMSE error of the Advanced SLAM is slightly higher than that of ORBSLAM2, indicating that our SLAM's accuracy is comparable to that of ORBSLAM2. The most severe RMSE error occurs when the number of common features between the previous reference frame and the current key-frame is more than the number of common features between the last key-frame and the current key-frame when tracking the reference frames. In static situations, tracking SLAM with a reference frame can enable more precise localization due to the quantity of common features.

Our SLAM, on the other hand, outperforms ORBSLAM2 in scenes with dynamic objects. Because our SLAM tracks by key rather than frame of reference, if additional systems track the SLAM, the trajectory will simply follow the moving item when a large moving object passes the camera. reference. Because a huge moving object has a large number of items and SLAM is based on the premise that the number of objects in the moving object is considerably smaller than the number of objects in the static object, the original ORBSLAM2 is used in this scenario. Large moving objects are considered static by the brain. This effect can be considerably reduced if SLAM tracks with key. SLAM only tracks the last key-frame containing only static objects before adding a new key-frame containing moving objects to the local mapping stream.

C. Evaluation of Detection Capabilities

We record stone RGB-D sequences in an interior environment to demonstrate the possibilities of our object-oriented semantic mapping system, then compare the number of items recorded in map for each class. Our system can recognise the majority of objects in the scenes; however, a few are missed. Furthermore, because YOLO's output is in the form of a bounding box, each object contains some of its background.

D. Mapping

The discovered objects are transformed to Octomap, as seen. The TUM dataset is used to assess the accuracy of the localization experiments. The RGB and depth images are included in the TUM dataset. We mix RGB and Depth pictures to build point clouds in the proposed semantic SLAM, and then assign a position to each point cloud. As a result, Octomap can be used to construct a map using point clouds and their placements.

To compare the improved Octomap to the original Octomap, we built a map with multi-threads at five different resolutions using a sequence from the TUM dataset. Both of them use a sequence of 30 frames to generate a map five times. Displays the time it took to build a map using upgraded Octomap against the original Octomap. The new Octomap is noticeably faster than the original Octomap. Computing empty voxels is a key stage in Octomap generation, as shown above. The upgraded Octomap's speed of empty voxel computation is roughly double that of the original Octomap.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 1, June 2022
Impact Factor: 6.252

The main reason behind this is that we employ a more efficient approach to compute empty voxels and we use thread pools to reduce thread creation, whereas Octomap uses OpenMP to launch individual threads to compute empty voxels. Inserting new voxels is another crucial stage in the Octomap building process. The new Octomap is nearly four times faster than the old one after adding multi-threads to boost speed. The main reason for this is that the original Octoamp only had one thread for voxel updates.

VI. Result and Conclusion

In this research, we propose a semantic SLAM system that integrates ORB-SLAM and produces semantic maps using object-level elements. We modify ORB-SLAM2 in three ways to integrate object detection in the proposed system: we use YOLO to detect and recognise objects to generate semantic information, we build the relationship between key and objects to generate semantic maps, and we filter feature points to improve the accuracy of SLAM with semantic information. In addition, we use a Fast Line Rasterization Algorithm and multi-threads to boost Octomap's efficiency and speed up the mapping process. Finally, we assess our semantic SLAM using three criteria. We show that our semantic SLAM is more accurate in dynamic situations than ORB-SLAM2 and RGB-D SLAM, and that it builds maps faster with the enhanced Octomap. Furthermore, we illustrate the effectiveness of object detection via a quantitative evaluation of a real-world dataset captured in an office. We can improve our semantic SLAM system in the future in the following ways. (1) YOLO is used to extract semantic signals from photos in the form of Bounding Boxes.

The models created by Bounding Box have an excessive amount of backdrop. As a result, we require a Deep-Learning Method capable of producing more accurate profiles. (2) To filter features extracted from photos, we use semantic messages. To get the most out of these messages, we can use semantic messages as landmarks and use graph-based optimization to increase localization accuracy. (3) In Data Association, we use a closest neighbour search to determine how similar candidate objects and temporary objects are, although this method is easily influenced by camera localization accuracy. To lessen the effect of camera location for data association, we can extract FPFH or other features from temporary and candidate items, and compute the number of match features between temporary and candidate objects.

REFERENCES

[1]. H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," in IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006, doi: 10.1109/MRA.2006.1638022.

[2]. T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," in IEEE Robotics & Automation Magazine, vol. 13, no. 3, pp. 108-117, Sept. 2006, doi: 10.1109/MRA.2006.1678144.

[3]. N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian and M. E. Munich, "The vSLAM Algorithm for Robust Localization and Mapping," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005, pp. 24-29, doi: 10.1109/ROBOT.2005.1570091.

[4]. Guclu, O., Can, A.B. Fast and Effective Loop Closure Detection to Improve SLAM Performance. J Intell Robot Syst 93, 495–517 (2019). https://doi.org/10.1007/s10846-017-0718-z

[5]. "Fully Convolutional Networks for Semantic Segmentation", Jonathon Long, Evan Sehlhamer, Trevor Darrell.

[6]. "SVO: Fast Semi-direct monocular SLAM", Pizzoli M, Scaramuzza, Forster C.

[7]. "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera", Izadi S.

[8]. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", Darrell T, Donahue J, Girshick R