Ground Mobile Robot Localization Algorithm Based on Semantic Information from the Urban Environment

Artur Podtikhov(✉) and Anton Saveliev

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 39, 14th Line, St. Petersburg 199178, Russia
apodtikhov@gmail.com

Abstract. This paper presents the SLAM algorithm, which use the semantic information extracted from the urban environment to increase the accuracy of ego-vehicle localization in ORB-SLAM2 system. For this purpose, a semantic segmentation module is added to the standard algorithm to assign an object on each frame to one of a given set of classes. The CARLA Simulator was used as a simulation environment, which generates a photorealistic urban environment with the ability to run an arbitrary number of active elements in it, which usually make localization difficult, causing interference with the system. Based on the environment, a training dataset for semantic segmentation was collected. The training dataset consists of 3,696 pairs of city images and corresponding segmentation masks in which each pixel corresponds to one of 23 semantic labels. Using this dataset, the DeepLabV3+ segmentation model was trained with mean per-class IoU metric equals to 81.48%. By using semantic information to filter potentially dynamic objects and matching key points, we were able to increase the localization accuracy relative to the base algorithm by an average of 23% and build a semantic map of the environment.

Keywords: SLAM · ORB-SLAM2 · DEEPLAB · CARLA · Robot

1 Introduction

Simultaneous Localization and Mapping (SLAM) is a technique used in mobile autonomous vehicles to build a map of an unknown environment or to update a map of a known environment while simultaneously keeping track of agent's location and the traveled path within it. In general terms, the control scheme of a modern mobile robot moving in a known environment can be represented in the following chain of actions: obtaining information about the world around; determining one's own position on a predetermined map; traffic planning with regard to the environment; control over the implementation of planned actions and transmission of control signals to actuators (motors, wheels and other manipulators). However, if the environment is not known in advance, then first you need to build a map of the area. Traditional mapping algorithms require an estimate of the robot's position, while accurate localization requires a previously known map. That is why SLAM methods can be called complex, because they are aimed at solving two mutually dependent tasks: localization and map construction.

A. Ronzhin et al. (Eds.): ICR 2023, LNAI 14214, pp. 164–174, 2023.
https://doi.org/10.1007/978-3-031-43111-1_15

This approach was first proposed at the IEEE Conference on Robotics and Automation in San Francisco in 1985 [1]. Then, and over the next few years, it was solved using various active sensors, such as a laser range finder, lidar or sonar, to determine the position of landmarks in space. SLAM is a cornerstone for autonomous navigation tasks in unknown environment, its applications are found in unmanned vehicles [2, 3], aircraft [4], underwater vehicles [5], virtual reality [6], in space exploration, for example, the surface map of Mars was constructed using SLAM methods [7].

The relevance of solving the problem of simultaneous localization and mapping is due to the fact that maps commonly used for agent navigation mainly reflect the type of space fixed at the time of their construction, and it is not at all necessary that the type of space will be the same at the time the maps are used. At the same time, the complexity of the technical process of determining the current location with the simultaneous construction of an accurate map is due to the low accuracy of the instruments involved in the process of calculating the current location.

Recently, visual SLAM methods, which are based on information from cameras, have become very popular, since cameras are cheaper to purchase and operate, while they can provide more information about the world around the robot. For instance, only cameras can transmit color, therefore, in unmanned vehicles they are used. Although the use of cameras increases the complexity and resource intensity of the algorithms, since they do not allow you to directly calculate the distance to the object of interest.

Most visual SLAM methods rely solely on geometric information, building a map of the unknown terrain in dense/semi-dense (DTAM [8], LSD-SLAM [9]) or keypoint-based (PTAM [10], ORB-SLAM [11]) point clouds. Such maps are homogeneous: the dots on them indicate only the presence or absence of an obstacle and do not carry any additional information. At the same time, visual SLAM works with camera images – a rich source of additional information. Often, when working with images, not the points themselves are used, but the objects they form, for example, various algorithms for analyzing biomedical images are based on this approach. Such enlarged objects are usually obtained using object detection methods, and if more accurate prediction of the boundaries of objects is necessary, using segmentation methods. Simultaneous localization and mapping methods that use this approach to working with images from cameras are grouped under the name Semantic SLAM.

All semantic SLAM methods can be divided into 2 broad categories according to the type of problem being solved: improving map representation [12–17] and improving localization [18–24]. The purpose of using methods that improve the presentation of a map is to add an additional "semantic" layer to it, so that points on it are distinguishable from each other and belong to a certain class. Such map representations can be useful in navigation, often in articles the following example is given: semantic information provides an ideal level of abstraction for a robot to understand and execute human commands (e.g., "bring me a cup of coffee", "leave the house through the red door") and provide people with models of the environment that are easy to understand. In turn, methods aimed at improving localization consider segmentation not as a goal, but as a tool that helps to take into account additional non-geometric information during localization. Such methods, for instance, include filtering moving objects and localization or mapping

solely on the basis of those objects that a priori cannot change their location in the world, thus helping the robot to localize in the so-called "dynamic" environments.

This paper presents a new method of semantic SLAM, which uses one of the most stable and accurate algorithms ORB-SLAM2 [25] as a basic algorithm for localization and map building. But it also considers semantic information using the DeepLabV3+ [26] model for semantic segmentation in order to: (a) build meaningful maps, where each point is associated with the class of the object to which it belongs, and (b) use semantic information to increase localization accuracy (by excluding potentially dynamic scene objects and building associations between points from different frames).

2 Description of the Training Data Collection Methodology for Semantic Segmentation

To collect a dataset for training the segmentation network, a high-quality map "Town10HD" from the CARLA Simulator [15] was used, which is an urban area with various infrastructure facilities. On this map, software developers pre-set a list of locations in which it is recommended to spawn cars in order for them to appear on the road directed towards traffic (Fig. 1).

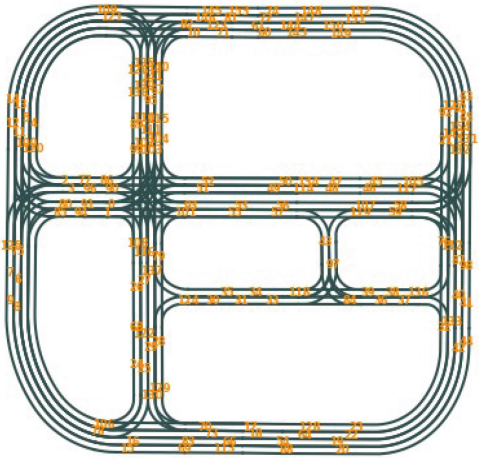

Fig. 1. Schematic image of the Town10HD city map from the CARLA Simulator. The points where the training images from the camera were collected are marked in orange (Color figure online).

The recommended points for car spawn were used to collect a dataset of camera images and corresponding ground-truth segmentation masks according to the following algorithm:

1. The car was spawned at a given point (X_i, Y_i, Z_i), parallel to the ground surface with a rotation angle relative to the perpendicular to the surface equal to 0.

2. On the hood of the car, 2 pseudo-cameras were spawned: standard RGB and segmentation, both with a resolution of 800 × 600 pixels.
3. The car (with the cameras) turned through an angle of 15°.
4. The images received from the cameras were recorded and saved to disk.
5. Steps 3–4 were repeated until the car made a complete turn.
6. The car and both cameras were destroyed.
7. The transition to the next spawn point and, respectively, to point 1 was performed.

Thus, 3,696 pairs of images with segmentation masks were collected from the 154 recommended vehicle spawn points. Figure 2 shows an example of an image obtained with an RGB camera mounted on a car hood (left) and its corresponding ground-truth segmentation mask (right), in which each pixel belongs to one of the given classes. In total, CARLA provides a segmentation map for 23 classes, which are listed in Table 1.

A random subset of images of 80% of the original data set was used for training, with the remaining 20% exclusively for validating the results.

Fig. 2. An example of an RGB camera image (left) and a ground-truth semantic segmentation mask (left) from the training dataset.

3 Segmentation Model

DeepLabV3+ was used as the segmentation model, with the resnext50_32x4d encoder [27] pre-trained on the ImageNet dataset [28]. A small number of augmentations were used: random cropping the image to a size of 512 × 512 pixels, horizontal flipping (with probability 0.5), adding normally distributed noise (with probability 0.2), and performing a random four-point perspective (with probability 0.5). The loss function chosen was FocalLoss [29] since the class distribution in the dataset is highly irregular. Optimization was performed using AdamW optimizer [30]. The training batch size was set to 6 and the learning rate was set to 1e-4.

The table shows that for large objects the segmentation accuracy is quite high, while for objects with a small area (such as traffic lights, road signs and poles) it is less. However, the obtained distribution of accuracy for different classes is consistent with the distribution of accuracy of the best segmentation models of the CityScapes benchmark, so this distribution can be associated with the limitations of modern semantic segmentation architectures.

Table 1. Metrics reflecting the quality of segmentation on the validation dataset. The MISSING label marks classes that do not exist in the Town10HD map.

ID	Class label	Per-class IoU	Per-class Accuracy	ID	Class label	Per-class IoU	Per-class Accuracy
0	Unlabeled	SKIP	SKIP	12	TrafficSign	69.65%	76.47%
1	Building	95.05%	97.66%	13	Sky	93.70%	96.20%
2	Fence	32.83%	42.83%	14	Ground	94.17%	97.62%
3	Other	84.67%	89.02%	15	Bridge	MISSING	MISSING
4	Pedestrian	MISSING	MISSING	16	RailTrack	98.10%	98.97%
5	Pole	58.49%	66.19%	17	GuardRail	MISSING	MISSING
6	RoadLine	84.98%	89.98%	18	TrafficLight	78.44%	87.59%
7	Road	98.65%	99.47%	19	Static	81.89%	90.01%
8	SideWalk	97.01%	98.43%	20	Dynamic	79.31%	89.68%
9	Vegetation	84.76%	93.74%	21	Water	63.54%	74.03%
10	Vehicles	90.62%	95.70%	22	Terrain	79.56%	85.09%
11	Wall	82.62%	87.96%				

4 The Algorithm Developed

As previously mentioned, the ORB-SLAM2 algorithm was chosen as the base algorithm for simultaneous localization and mapping. Interaction with the CARLA simulation environment was performed using the ros_bridge package which allows to receive sensor and odometry information from the simulator and publish them to ROS topics.

Figure 3 shows a generalized architecture of the proposed algorithm. The architecture is almost the same as that of ORB-SLAM2, except for the new block responsible for semantic segmentation included in the Tracking thread. The rest of the changes are internal and adjust some functions, which will be described below.

Semantic Segmentation Block. In order to integrate the image segmentation model into the system, it was converted from the PyTorch format to TorchScript, after which it became possible to use it in scripts written in C++. The resulting model is initialized by the GPU in the Tracking module and applied after each new frame is received, thus, at the start of the algorithm, there is not only the image itself, but also a segmentation mask that matches each pixel of the image with a semantic class.

Extract ORB Block. Since storing a full segmentation mask for each frame requires a significant amount of RAM, the corresponding semantic information is stored only for selected key points. For this purpose, at the moment of extracting key points and ORB descriptors, semantic information is added to these key points, indicating that the point belongs to one of the 23 classes, after which the rest of the mask is removed.

New Points Creation Block. When a map point is created, semantic information is also added to it, with each point storing a list of all predicted semantic classes, when it is

seen from different angles, at the current moment its class is the class it takes most often. This approach reduces the segmentation error and eliminates outliers. Also, a map point is considered inactive and does not participate in further calculations if it belongs to one of the potentially dynamic (or low-informative) classes (Unlabeled, Other, Pedestrians, Vehicles, Sky, Dynamic).

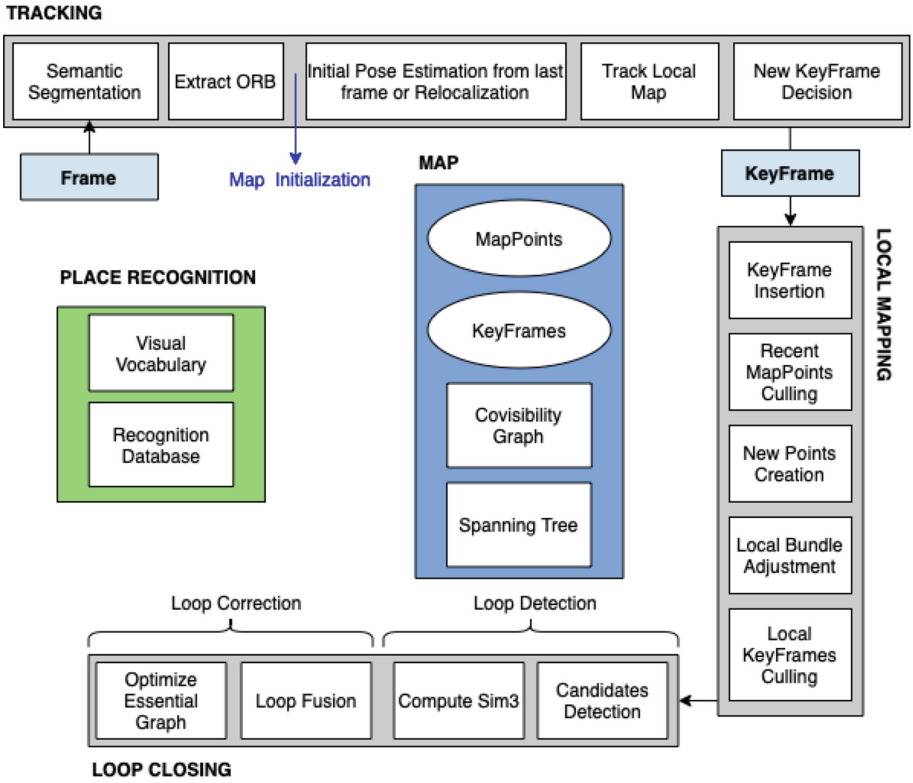

Fig. 3. Generalized architecture of ORB-SLAM2, to which a semantic segmentation block has been added, which is triggered on receipt of each new frame.

Key Point Association. The keypoint (or map points and keypoints) association algorithm is one of the central algorithms of ORB-SLAM2, since it is used in almost all submodules. In the original algorithm, the association is performed solely based on the calculation of the distance between the two ORB descriptors. To account for semantic information, a penalty factor equal to 0.5 * current distance is added to this distance.

5 Results Analysis

To evaluate the quality of localization, 3 experiments were conducted in a simulation environment, lasting from 1 to 3 min, sensor information and ground-truth odometry were stored at a frequency of 20 frames per second. Simulations were run from various recommended vehicle spawn points, in addition, except ego-vehicle, 20 cars with a built-in autopilot and 10 pedestrians were also generated in the environment. After that, ORB-SLAM2 model and the developed modification were launched separately. Having ground-truth and predicted odometry, it is impossible to compare them directly, because when using a monocular camera, it is possible to restore the world coordinates of a map point with an accuracy to scale constant. The Horn algorithm [31] was used to estimate the scale constant and alignment of coordinate systems. For each simulation, it shows the ground-truth trajectory of the car (blue), the trajectory obtained using the ORB-SLAM2 algorithm (orange) and the trajectory obtained using the developed algorithm (green). From the motion trajectories it is difficult to draw conclusions about the increase in localization accuracy, therefore, plots of localization errors are also attached (Fig. 4). The reconstructed trajectories for all simulations are shown in Fig. 5. The horizontal axis denotes the frame number, the vertical axis denotes the distance between the ground-truth position of the vehicle at a given time and the predicted position. Comparison of localization accuracy over the entire route was performed using the metric of the mean percentage absolute error in the Cartesian coordinate system, calculated by the formula:

$$MAPE([x, y], [\hat{x} + \hat{y}]) = MEAN\left(\frac{100\%}{n_{samples}} \sum \frac{\left[|x_i - \hat{x}_i|, [y_i - \hat{y}_i]\right]}{[|x_i|, |x_i|]}\right). \quad (1)$$

The results of comparing the quality of localization are in Table 2. It can be seen that the proposed algorithm performs slightly better than the basic algorithm in determining the location of the vehicle, while from the reconstructed trajectories (Fig. 5) it can be concluded that the predictions change slightly, mainly due to reducing the probability of wrong key points matching. At the same time, if it is strictly forbidden to assign points corresponding to different semantic classes, it fails to initialize the map (due to the fact that the number of matches falls below the threshold value), therefore, to further improve the approach, it is necessary to improve the quality of semantic segmentation.

A side effect of our work is the construction of a semantic map of the environment; after a complete route around of the city, the map of the area looks like Fig. 6. It can be concluded that the algorithm for determining key points basically extracts points from buildings (white color on the map), road markings (purple) and trees (green).

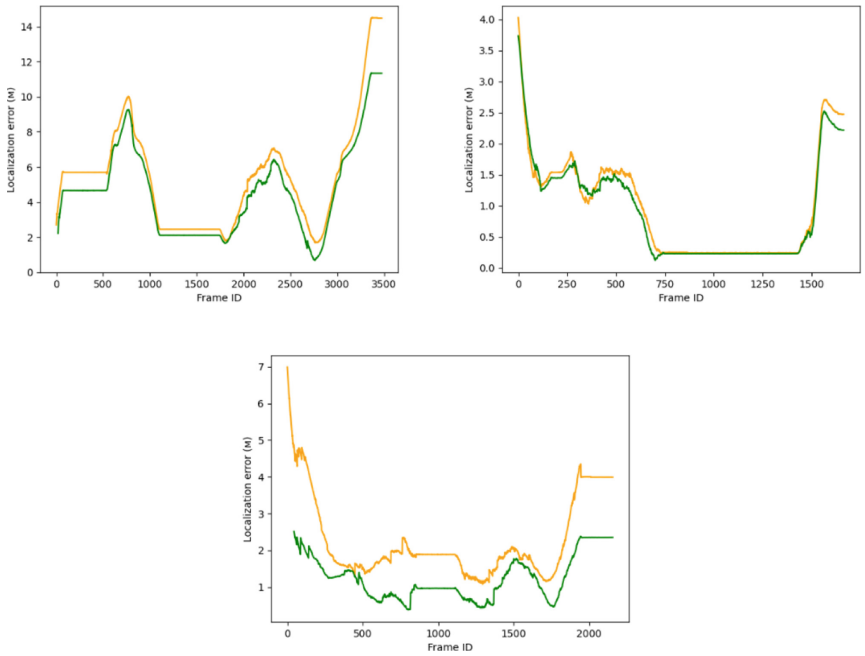

Fig. 4. Plots for estimating position errors for each frame (in meters). Green colors indicate the errors of the implemented algorithm, orange is ORB-SLAM2 (Color figure online).

Table 2. Localization accuracy in different simulations. The table shows the duration of the route, it's length (in simulator units) and the mean absolute percentage error of estimating the ego vehicle location by the base and developed implementation. The last column displays the % change in the localization error of the developed algorithm relative to the base one.

No.	Duration (s)	Length (m)	ORB-SLAM2	Developed Algorithm	Relative change in localization error
1	172.8	610	25.85%	21.7%	–16.05%
2	83.2	241.7	1.10%	1.02%	–7.27%
3	105.6	456.36	7.83%	4.22%	–46.1%

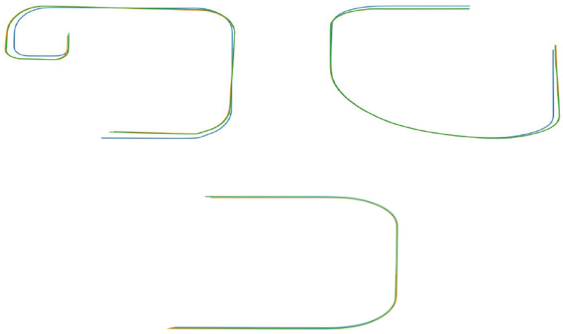

Fig. 5. The trajectory of the car. The ground-truth trajectory is shown in blue, the trajectory predicted by the ORB-SLAM2 algorithm in orange, and the trajectory predicted by developed algorithm in green. The order of the images corresponds to the sequence numbers of the simulations in Table 2.

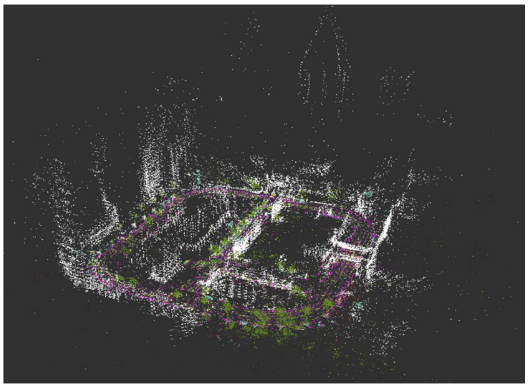

Fig. 6. Semantic map of the city Town10HD in the form of a point cloud, obtained after simulating the movement of a car throughout the city. Different colors indicate urban infrastructure objects belonging to different classes (Color figure online).

6 Conclusion

In this paper, we proposed an algorithm for simultaneous localization and mapping, taking into account semantic information about the objects of the urban environment. The proposed approach excludes potentially dynamic objects from the consideration of the algorithm and improves the matching of key points. The developed algorithm has demonstrated a 23% increase in localization accuracy on mean absolute percentage error relative to the basic algorithm, at the same time, it requires more computing resources to apply the segmentation model. The quality of the current segmentation model does not allow to completely eliminate the comparison of key points assigned to different semantic classes, so further development of the algorithm should be aimed at increasing the segmentation accuracy and increasing the number of semantic classes under consideration.

References

1. Chatila, R., Laumond, J.: Position referencing and consistent world modeling formobile robots. In: Proceedings. International Conference on Robotics and Automation, vol. 2, pp. 138–145. IEEE (1985)
2. Henning, L., Andreas, G., Bernd, K.: Visual slam for autonomous ground vehicles. In: IEEE International Conference on Robotics and Automation, Shanghai, pp. 1732–1737, China (2011)
3. Qin, T., Chen, T., Chen, Y., Su, Q.: Avp-slam: semantic visual mapping and localization for autonomous vehicles in the parking lot. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5939– 5945. IEEE (2020)
4. Milford, M.J., Schill, F., Corke, P., Mahony, R., Wyeth, G.: Aerial slam with a single camera using visual expectation. In: 2011 IEEE international conference on robotics and automation, pp. 2506–2512. IEEE (2011)
5. Ribas, D., Ridao, P., Tardo´s, J.D., Neira, J.: Underwater slam in man-made structured environments. J. Field Robot. **25**(11–12), 898–921 (2008)
6. Jinyu, L., Bangbang, Y., Danpeng, C., Nan, W., Guofeng, Z., Hujun, B.: Survey and evaluation of monocular visual-inertial slam algorithms for augmented reality. Virtual Reality Intell. Hardware **1**(4), 386–410 (2019)
7. Zheng, B., Zhang, Z.: An improved EKF-SLAM for mars surface exploration. Int. J. Aerosp. Eng. **2019**, 1–9 (2019)
8. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: Dtam: Dense tracking and mapping in real-time. In: 2011 International Conference on Computer Vision, pp. 2320–2327. IEEE (2011)
9. Engel, J., Scho¨ps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 Sep 2014, Proceedings, Part II 13, pp. 834–849. Springer (2014)
10. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225–234. IEEE (2007)
11. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Rob. **31**(5), 1147–1163 (2015)
12. Dub´e, R., Cramariuc, A., Dugas, D., Nieto, J., Siegwart, R., Cadena, C.: Segmap: 3d segment mapping using data-driven descriptors, arXiv preprint arXiv:1804.09557 (2018)
13. Hermans, A., Floros, G., Leibe, B.: Dense 3d semantic mapping of indoor scenes from rgb-d images. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2631–2638. IEEE (2014)
14. McCormac, J., Handa, A., Davison, A., Leutenegger, S.: Semantic fusion: dense 3dsemantic mapping with convolutional neural networks. In: 2017 IEEE International Conference on Robotics and automation (ICRA), pp. 4628–4635. IEEE (2017)
15. Rosinol, A., Abate, M., Chang, Y., Carlone, L.: Kimera: an open-source library for real-time metric-semantic localization and mapping. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 1689–1696. IEEE (2020)
16. Salas-Moreno, R.F., Newcombe, R.A., Strasdat, H., Kelly, P.H., Davison, A.J.: Slam++: simultaneous localisation and mapping at the level of objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1352–1359 (2013)
17. Stückler, J., Behnke, S.: Multi-resolution surfel maps for efficient dense 3d modeling and tracking. J. Vis. Commun. Image Represent. **25**(1), 137–147 (2014)
18. Bowman, S.L., Atanasov, N., Daniilidis, K., Pappas, G.J.: Probabilistic data association for semantic slam. In: International Conference on Robotics and Automation (ICRA), pp. 1722–1729. IEEE (2017)

19. Fuentes, O., Savage, J., Contreras, L.: A SLAM system based on Hidden Markov Models. Inform. Autom. **21**(1), 181–212 (2022). https://doi.org/10.15622/ia.2022.21.7
20. Mahamudul Hashan, A., Md Rakib Ul Islam, R., Avinash, K.: Apple leaf disease classification using image dataset: a multilayer convolutional neural network approach. Inform. Autom. **21**(4), 710–728 (2022). https://doi.org/10.15622/ia.21.4.3
21. Ganti, P., Waslander, S.L.: Network uncertainty informed semantic feature selection for visual slam. In: 2019 16th Conference on Computer and Robot Vision (CRV), pp. 121–128. IEEE (2019)
22. Gawel, A., Del Don, C., Siegwart, R., Nieto, J., Cadena, C.: X-view: graph-basedsemantic multi-view localization. IEEE Robot. Autom. Lett. **3**(3), 1687–1694 (2018)
23. Lianos, K.N., Schonberger, J.L., Pollefeys, M., Sattler, T.: Vso: Visual semantic odometry. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 234–250 (2018)
24. Stenborg, E., Toft, C., Hammarstrand, L.: Long-term visual localization using semantically segmented images. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 6484–6490. IEEE (2018)
25. Mur-Artal, R., Tardo´s, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans. Robot. **33**(5), 1255–1262 (2017)
26. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
27. Xie, S., Girshick, R., Doll´ar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1492–1500 (2017)
28. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
29. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dolla´r, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
30. Reddi, S.J., Kale, S., Kumar, S.: On the convergence of a dam and beyond, arXivpreprint arXiv:1904.09237 (2019)
31. Horn, B.K.: Closed-form solution of absolute orientation using unit quaternions. Josa a **4**(4), 629–642 (1987)