2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Exploiting Semantic and Public Prior Information in MonoSLAM

Chenxi Ye[1], Yiduo Wang[2], Ziwen Lu[3], Igor Gilitschenski[4], Martin Parsley[5] and Simon J. Julier[3]

***Abstract*— In this paper, we propose a method to use semantic information to improve the use of map priors in a sparse, feature-based MonoSLAM system. To incorporate the priors, the features in the prior and SLAM maps must be associated with one another. Most existing systems build a map using SLAM and then align it with the prior map. However, this approach assumes that the local map is accurate, and the majority of the features within it can be constrained by the prior. We use the intuition that many prior maps are created to provide semantic information. Therefore, valid associations only exist if the features in the SLAM map arise from the same kind of semantic object as the prior map. Using this intuition, we extend ORB-SLAM2 using an open source pre-trained semantic segmentation network (DeepLabV3+) to incorporate prior information from Open Street Map building footprint data. We show that the amount of drift, before loop closing, is significantly smaller than that for original ORB-SLAM2. Furthermore, we show that when ORB-SLAM2 is used as a prior-aided visual odometry system, the tracking accuracy is equal to or better than the full ORB-SLAM2 system without the need for global mapping or loop closure.**

I. Introduction

Simultaneous Localization And Mapping (SLAM) is an extremely important capability for most autonomous platforms. It gives these systems the freedom to operate in natural, unstructured environments. SLAM is widely implemented on autonomous guided vehicles, self-driving cars, collaborative robots, and mixed reality systems which run on custom headsets and mobile phones. As a result, SLAM directly impacts the lives of billions of people.

However, almost all SLAM systems exhibit drift. Odometry sensors such as Inertial Measurement Units (IMUs) and wheel encoders measure the relative change in a platform's pose. Perception sensors such as cameras and LiDAR measure the relative transformation from a platform-fixed frame to a landmark. As a result, incremental errors are integrated into the map and cause drift. The most common way to reduce this is through *loop closure*: when the platform returns to a visited part of the map, the constraint that the platform has completed a loop can be imposed. This can greatly reduce the errors associated with drift. However, there are two issues with loop closure. First, most loop closure algorithms carry out some kind of search over the entire map to detect loop closure candidates [1]. The greater the drift, the larger this search area needs to be, increasing both the computational cost and the risk of falsely identifying a loop closure event. The second is that the improvement afforded by loop closure declines the further a map feature is from the loop closure point. Therefore, even with loop closure, substantial localization errors can still occur.

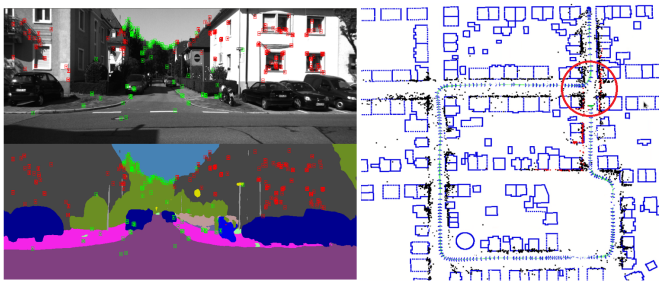


Fig. 1: Demonstration of our method. Left: (top) a keyframe and (bottom) its semantic label. Red points mark keypoints on buildings, which can be reliably associated with available prior information. Right: localization and mapping result of our system without loop closing. The red circle shows the SLAM system as it revisits an earlier part of the map. Because of the prior, almost no drift occurs.

Another way to reduce drift is to use prior maps (P-MAPs) from other sources such as building plans or aerial photographs. If the features in the P-MAP are probabilistically related to those in the SLAM map (S-MAP), they can provide information wherever associations between the P-MAP and S-MAP can be established, and not just at S-MAP loop closure points. The priors can be used in three ways: to seed an S-MAP directly [2, 3], to constrain the platform location [4]–[7], and to constrain the S-MAP feature locations [8, 9]. In the last two cases, a major challenge is to associate the features in the P-MAP with those in the S-MAP. Current techniques include scan matching [4] and Iterative Closest Point (ICP) [5]. These two methods are most reliable when the geometry of the S-MAP is accurate, and the majority of the features are constrained by those in the P-MAP. In our use case — MonoSLAM in urban environments — we find that neither condition holds true.

In this paper, we explore how semantic segmentation can improve the association between the features in the S- and P-MAPs. We use the following intuition: P-MAPs are often created to convey semantic information, so many of the features in a P-MAP have well-defined semantic labels. If a feature in an S-MAP is associated with a feature in the P-MAP, both features must have the same semantic label.

We focus on the problem of monocular SLAM because it is both technically challenging and widely used for low cost SLAM systems such as those on phones and drones.

[1]Carried out while at the Department of Computer Science, University College London, chenxi.ye.18@alumni.ucl.ac.uk
[2]Oxford Robotics Institute, University of Oxford, ywang@robots.ox.ac.uk
[3]Department of Computer Science, University College London, {ziwen.lu, s.julier}@ucl.ac.uk
[4]Computer Science and Artificial Intelligence Lab, MIT, igilitschenski@mit.edu
[5]Mo-Sys Engineering Ltd., martin@mo-sys.com

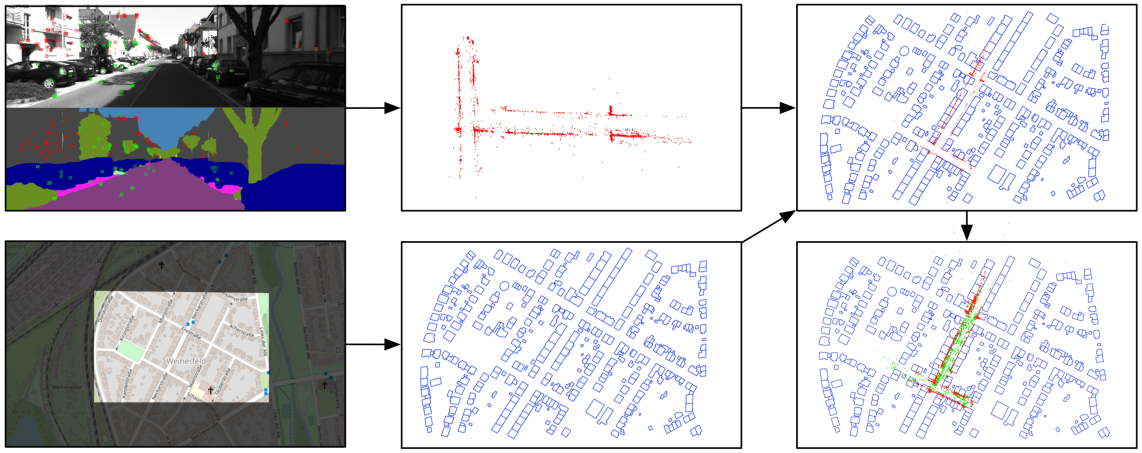


Fig. 2: Schematic illustration of semantically-aided MonoSLAM. Top Left: Semantic segmentation applied to keyframes. Top Middle: These are used to extract building feature points from the drift-corrupted S-MAP. Bottom Left: P-MAP identified. Bottom Middle: Features extracted from P-MAP. Top Right: S-MAP and P-MAP overlaid for data association. Bottom Right: Data association and constraint application between S-MAP and P-MAP.

To implement this approach, we develop a new robust data association technique which relates the S- and P-MAPs together. Our approach, illustrated in Fig. 1, uses semantic segmentation to identify potentially compatible S-MAP and P-MAP features, and multiRANSAC to extract the building geometry. We also develop a novel method for constraining landmark locations rather than platform poses. This method is robust to errors in the P-MAP. We shown the performance of the approach on an extended version of ORB-SLAM2 [1] using the challenging case of a single camera.

The structure of this paper is as follows. The problem statement is introduced in Section II. Section III describes the graphical formulation of SLAM. Previous methods for utilising priors are reviewed in Section IV. Our system is described in Section V, evaluated in Section VI, and summarized and concluded in Section VII.

II. Problem Statement

We are studying the case of a mobile platform operating in an urban environment. The goal is to estimate the full six Degree of Freedom (DoF) pose of the platform. The platform is equipped with a monocular sensor and uses SLAM.

The system is provided with a P-MAP. The information is available from Open Street Map [10]. It consists of a series of polygonal building footprints. This P-MAP $\mathbf{P}$ consists of m features, $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$. Each feature is a 2D line segment which is specified by a pair of start and end points in a world-fixed coordinate frame, $\mathbf{p}_j = \{\mathbf{p}^s, \mathbf{p}^e\}_j$.

The goal is to develop a SLAM algorithm which uses the 2D prior information to constrain the 3D position of the landmarks and, in turn, constrain the 3D pose of the platform.

We begin by describing the SLAM framework used.

III. Graph-based SLAM

We use the conventional formulation of a SLAM system with a keyframe-based backend. Using notation from [11] and [12], the state of the set of keyframes is given by the set $\boldsymbol{X}_{0:k} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_k\}$. The S-MAP $\boldsymbol{S}$ consists of n features, $\boldsymbol{S} = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_n\}$. The sequence of observations is $\boldsymbol{Z}_{1:k} = \{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k\}$. Given this system, the SLAM problem is to compute

$$p\left(\boldsymbol{X}_{0:k}, \boldsymbol{S} | \boldsymbol{Z}_{1:k}\right). \tag{1}$$

Using the standard Markov assumptions, this joint probability can be factorized as

$$p\left(\boldsymbol{X}_{0:k}, \boldsymbol{S} | \boldsymbol{Z}_{1:k}\right) = p\left(\boldsymbol{x}_0\right) \prod_{(i,j)\in\mathcal{G}} p\left(\boldsymbol{z}_{ij} | \boldsymbol{x}_i, \boldsymbol{s}_j\right). \tag{2}$$

$\mathcal{G}$ is the set of pairs of indices which link feature observations to platform poses, $p\left(\boldsymbol{x}_0\right)$ is the prior on the initial pose, and $p\left(\boldsymbol{z}_{ij} | \boldsymbol{x}_i, \boldsymbol{s}_j\right)$ is the likelihood of the observation of landmark j at timestep i. Taking negative log likelihoods and assuming Gaussian distributions, the Maximum A Posteriori estimate is given by [12],

$$\boldsymbol{X}^*_{0:k}, \boldsymbol{S}^* = \arg\min_{\boldsymbol{X}_{0:k}, \boldsymbol{S}} \boldsymbol{e}_0^{\mathrm{T}} \boldsymbol{\Omega}_0 \boldsymbol{e}_0 + \sum_{\langle i,j\rangle\in\mathcal{G}} \boldsymbol{e}_{ij}^{\mathrm{T}} \boldsymbol{\Omega}_{ij} \boldsymbol{e}_{ij}. \tag{3}$$

IV. Using Prior Information in SLAM

As explained above, the sensors used in SLAM algorithms only compute relative transformations, hence drift can arise. One way to overcome this is to combine information from the P-MAP into the SLAM process, changing Eq. (1) to

$$p\left(\boldsymbol{X}_{1:k}, \boldsymbol{S} | \boldsymbol{Z}_{1:k}, \mathbf{P}\right). \tag{4}$$

This prior information is used in two main ways: to constrain the platform pose, and to constrain the feature locations.

A. Using Priors to Constrain Platform Pose

By reducing errors in the platform pose, the errors in the underlying map will be reduced as well. One of the earliest examples was the work by Kümmerle et. al. [4]. Line features were extracted from an aerial map to create building boundaries. The robot, equipped with a laser scanner, used scan matching and Monte Carlo localization to estimate the absolute platform pose over time. However, it is not always the case that a single scan is sufficient for a

match. The works by Vysotska et al. [5] and Floros et al. [13], for example, attempted to overcome this limitation by using observations collected over several timesteps to build local S-MAPs. These maps were aligned with the prior information (derived from Open Street Map) using ICP. However, both of these approaches use sensing systems which provide measurements of depth (either using depth sensors or stereo cameras) to produce dense, geometrically accurate local maps which mostly contain features derived from the prior map. Real-time monocular SLAM systems, however, typically only produce sparse features points. The only work we are aware of which uses localization with a single camera is the work by Caselitz et al. [7]. They used MonoSLAM to build an accurate local map of the environment and matched it with a dense LiDAR scan of the environment, which they created themselves. However, they did not compare their method with Open Street Map or other open sources of priors.

B. Using Priors to Constrain the Positions of Map Features

Priors can be used to constrain the map feature spatial distribution. One way to achieve this is to simply use a map created from an earlier SLAM run as the P-MAP for a subsequent run. It is widely used in distributed SLAM systems such as CCM-SLAM [3]. However, this can only be used in the special case that the P-MAP and S-MAP use identical features. Therefore, a second approach is to transform the P-MAP features into S-MAP ones. Georgiou et al., for example, convert a floor plan into an occupancy grid [2]. Once features have been converted into an S-MAP, the regular SLAM data association mechanisms can be used.

In general, the features in the P-MAP and S-MAP are sufficiently different from one another that there is no one-to-one mapping. Rather, the P-MAP features act as a constraint on the S-MAP features. Parsley et al. [9] developed a framework to address this. They argued that the real world contains latent structures, such as physical objects. These latent structures induce different features in different mapping systems. Features are associated between two maps when the features in each map arise from the same latent structure. The framework has two main elements to it: a data association technique to identify the relationship between features in the S- and P-MAPs, and a constraint mechanism.

We use this approach to design a SLAM system.

V. Semantically-Assisted MonoSLAM

Our proposed scheme is illustrated in Fig. 2. We developed our approach using ORB-SLAM2 [1], with the KITTI dataset [14]. Because it is free and publicly available, we used Open Street Map as the source of prior information. For semantic segmentation, we used DeepLabV3 [15], an open source model and trained it on the CityScapes dataset [16]. We note that although we have designed our algorithm specifically for the MonoSLAM case, it can be readily applied in systems with depth sensors or stereo.

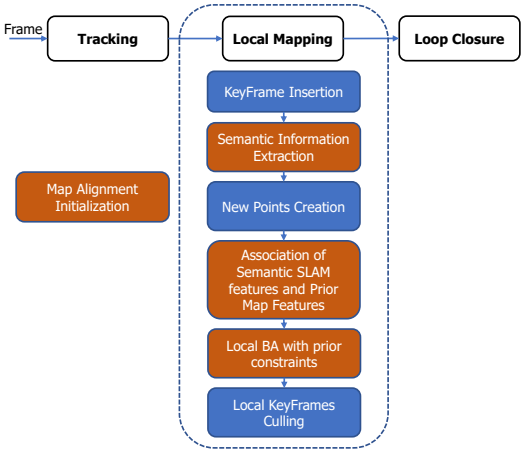


Fig. 3: The pipeline of modified ORB-SLAM2 system. Red blocks show modified steps and blue blocks are original steps in ORB-SLAM2

The pipeline of our modified variant of ORB-SLAM2 is shown in Fig. 3. We discuss the blocks which differ from the original implementation of ORB-SLAM2 below.

A. Initial Map Alignment

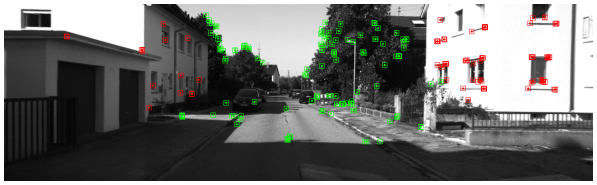
The first step is to align the P-MAP and S-MAP at the start of the run. For this initial development, we use GPS information. To achieve this, we take the first 15 GPS measurements, identify which keyframes these are associated with (using log time), and apply edges to the graph which constrain the 2D position assuming identity covariance matrix. Used Huber kernel.

B. Semantic Information Extraction

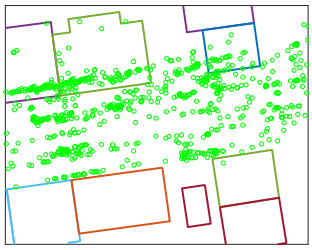
Since the P-MAP only contains building footprint information, it only provides information which can constrain S-MAP features on the exterior of a building. Using our intuition of semantic compatibility, we applied DeepLabv3+ [15] to label each keyframe. For the ith feature $\boldsymbol{m}_i$ we extract the local corresponding semantic label $\boldsymbol{s}_i$. We form a set of semantically labelled features $\boldsymbol{S} = \{\boldsymbol{s}_1, ..., \boldsymbol{s}_m\}$. Of DeepLab's 20 classes, only features labelled as building in the construction subclass are used for finding prior constraints in the P-MAP. An example of a keyframe with semantically labelled features is shown in Fig. 4(a).

Fig. 4(b) and 4(c) shows how semantic labels are used to significantly reduce the number of inapplicable (non-building) points which need to be considered with geometric data association.

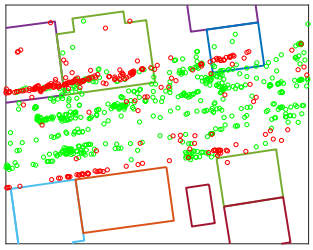
However, the figure illustrates several challenges. Firstly, the local geometry is distorted. Secondly, the segmentation can be noisy due wall-like structures present in the world but not in the P-MAP (such as gate posts) and misclassifications. Although DeepLabv3+ has extremely good benchmark performance (82.1% IoU class performance, 92.0% IoU category [17]), misclassifications still occur relatively frequently. For example, in Fig. 4(c) several points on the road are classified as walls, and several points on buildings are classified as non-building. Therefore, while



(a) Semantically labelled features in a keyframe: red (building), green (other).



(b) Original features.



(c) Semantically labelled features: red (building), green (other).

Fig. 4: Overlay of S-MAP features on the Open Street Map polygons (P-MAP).

semantic segmentation greatly reduces the number of points to consider, it is insufficient on its own.

C. S-MAP / P-MAP Data Association

As shown in Fig. 4(c), features on the building surfaces (marked in red) resemble the flat geometry of building facades. To associate S-MAP point features with P-MAP lines features, we use a multi-step data association procedure:

1) **Grouping semantically related points.** For each feature point labelled as building, the closest wall to it in the P-MAP is found using nearest neighbour based on the previous localization result; points that relate to the same building block are grouped together. However, since the feature points are not of high geometric accuracy, they cannot be directly associated with the P-MAP.
2) **Line Extraction.** Although the feature points suffer from global geometric distortion due to drift, their relative geometry is locally accurate. We use multiRANSAC [18] to extract line features from each cluster. This process is illustrated in Fig. 5. MultiRANSAC preserves local topological features inherent to buildings and urban environments. Those line segments that are nearly orthogonal to one another represent building corner structures.
3) **Line / Prior Map Association.** These line segments are associated with specific building walls in the P-MAP. At building corners, the fitted wall segments are almost orthogonal to one another (Fig. 5), and help to resolve the association ambiguity regarding which of the orthogonal walls a point associates with. A greater challenge lies in the parallel wall segments, such as the front and rear of a building. Features wrongly associated to the rear side of a building can lead to localization failure. We rely on the expected depth measurement of each wall in the P-MAP to the camera to define observable facades of each building. In our experiments, we have found that this method is essential during vehicle turning. A small drift in vehicle heading can lead to large map point displacement, which in turn can result in significant misassociations. Our solution greatly reduces the drift in orientation estimation and improves the robustness of the system.

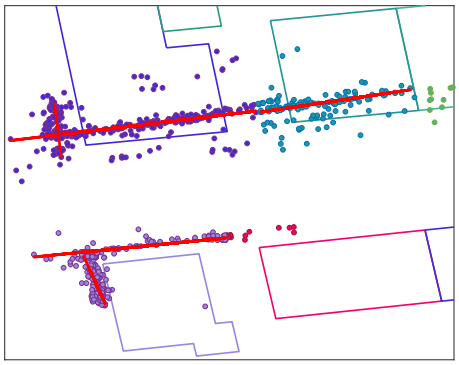


Fig. 5: Lines fit to the clustered points using multiRANSAC.

Despite these steps, false positives can still arise. There are three main sources: outdated prior information, poor initial estimation from SLAM, and wrongly associated prior information. We investigated a number of different approaches and found that Dynamic Covariance Scaling [19] was extremely effective, improving the stability of the system against bad initialization and outliers. This is particularly useful when the P-MAP contains substantial errors.

D. Local BA with Prior Constraints

Once the associations between the P-MAP and S-MAP have been proposed, constraints between the two sets of features can be applied, and local bundle adjustment can be carried out. Because the P-MAP is only an approximation of building footprints, we use soft constraints. Furthermore, because the prior information is in 2D, we can only provide 2D constraints on the full target motion.

The constraint on each feature minimizes the Euclidean distance between an S-MAP feature point and a line segment in the P-MAP. Suppose the S-MAP 3D feature point $\boldsymbol{s}_i = (m_x, m_y, m_z)_i$ is to be constrained by the 2D P-MAP feature $\mathbf{p}_j = \{\mathbf{p}^s, \mathbf{p}^e\}_j$. We first define $\mathbf{m}_i^* = (m_x, m_y)_i$ to be the projection of the feature point on the 2D map plane. We then find the closest point in $\mathbf{p}_j$ to $\mathbf{m}_i^*$. This is given by

$$\mathbf{m}_{ij}^* = \mathbf{p}_j^s + r(\mathbf{p}_j^e - \mathbf{p}_j^s), \tag{5}$$

where r is

$$r = \text{clamp}\left(\frac{(\mathbf{m}_i^* - \mathbf{p}_j^s)\cdot(\mathbf{p}_j^e - \mathbf{p}_j^s)}{|\mathbf{p}_j^e - \mathbf{p}_j^s|^2}, 0, 1\right). \tag{6}$$

$\text{clamp}(x, a, b)$ constrains $a \leq x \leq b$. This makes sure that the nearest point in the P-MAP must lie on the line segment.

Once the closest point is determined we compute the normal and parallel error with respect to the wall $e_{ij} = \mathbf{m}_i^* - \mathbf{m}_{ij}^*$. This vector penalizes both the normal distance from the wall and whether the point falls outside the wall

segment. Empirically, we found a suitable covariance matrix for this error term to be $\mathbf{\Sigma} = \text{diag}\left(0.1^2, 0.3^2\right)$.

It should be noted that this only provides constraints on the projection of 3D feature points to the 2D ground plane. Although this only constrains a subset of the features and platform poses, it can still have a significant impact on performance. For example, by correcting scale drift in the ground plane, this information directly reduces drift normal to the ground plane.

We found it necessary to modify the local bundle adjustment component in the original ORB-SLAM2 implementation to include more feature points and keyframes. In the original implementation, for a newly-observed keyframe f, the co-visibility graph defines a set of keyframes F_1 that share observation of features with f, and another set F_2 that share features with F_1 but not f. The set $\{f \cup F_1\}$ was then optimized while F_2 was held fixed. However, we found that the performance improvements from the prior were limited with such a design. F_2 created overly strong constraints on the local map. Therefore, we included F_2 in the local map too, and defined F_3 as a set of keyframes that are connected to F_2 but not included in $\{f \cup F_1 \cup F_2\}$. We carried out local optimization over $\{f \cup F_1 \cup F_2\}$ with F_3 as fixed constraints.

VI. Evaluation

To test the performance of our algorithm, we used the KITTI dataset because it contains trajectories in urban environments together with ground truth. Furthermore, because of the severe drift from MonoSLAM, we needed a relatively high density of buildings to ensure that constraints are readily available. As a result, we used subsets of two KITTI sequences: 00 (timesteps 435–1319) and 05 without loop closure (timesteps 432–1319) and with loop closure (timesteps 432–1502). Fig. 7 overlays the ground truth trajectory, the trajectory computed by the original ORB-SLAM2 MonoSLAM implementation and our implementation using semantics and prior information.

For sequence 00, 39% of S-MAP features were labelled as building. 46% of these building features were successfully associated with the P-MAP. The additional constraints caused the bundle adjustment time to rise from $302\,\text{ms}$ to $416\,\text{ms}$. Similar values were found for the other runs.

We can see that, through the constraints, the prior information has significantly reduced the scale drift error as compared with the original monocular ORB-SLAM2. Even after the loop closure, the system with the prior information still shows a better consistency as illustrated in Fig. 7(c).

Although the proposed model has achieved a significant reduction in drift, we found it to suffer from problems in several scenarios. One such scenario, as noted above, is that environments with few buildings provided insufficient features. The other major failure scenario is demonstrated in Fig. 6. In this case, the vehicle drove along a straight road, where the building facades form a straight line. As a result, the prior information is insufficient to constrain drift along the road. Thus when the vehicle turns the corner, the trajectory is sufficiently far off that the line segments no longer associate with the prior map.

TABLE S1: Absolute KeyFrame Trajectory RMSE(m)

Sequence	monocular ORB-SLAM2		With semantic prior	
	t_{abs}	t^*_{abs}	t_{abs}	t^*_{abs}
$00^{\dagger}$	6.34	15.61	1.67	1.97
$05^{\dagger,a}$	23.34	47.61	5.92	7.02
$05^{\dagger,b}$	4.45	6.28	2.38	3.16

† indicates the sequence is cut.
^{a},^{b} means the sequence with and without loop closure respectively.
* denotes the alignment of trajectory is made via the first 20 keyframes.

Fig. 6: Example of failed case. Accumulated drift in previous long straight road exceeds the range that the prior constraint could correct and hence wrong association is made leading to low tracking accuracy.

VII. Discussion and Conclusions

In this paper, we have presented a method which uses semantic information to incorporate prior information into a MonoSLAM system. We have shown that our approach is capable of enhancing localization and mapping accuracy. The results are extremely close to the ground truth. Even when loop closures occur, our method is able to produce significantly smaller errors.

There are two main areas of further work. First, the method we implemented is still fragile. It requires a sufficient density of prior map features to constrain drift. As illustrated in Fig. 6, if the drift becomes too large, the proposed system is unable to associate the labelled features with the map. We are exploring several approaches for improving robustness, including recent developments in multi-modal hypothesis representation in SLAM [20]. Second, the method only exploits semantic labels associated with buildings. However, other parts of the scene, such as the road surface, are also commonly labelled in semantic segmentation algorithms and datasets. We are exploring how these could be used to provide additional cues for data association, and to provide more robust operation in dynamic environments.

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, Oct. 2017.

[2] C. Georgiou, S. Anderson, and T. Dodd, "Constructing Informative Bayesian Map Priors: A Multi-Objective Optimisation Approach Applied to Indoor Occupancy Grid Mapping," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 274–291, 2017.

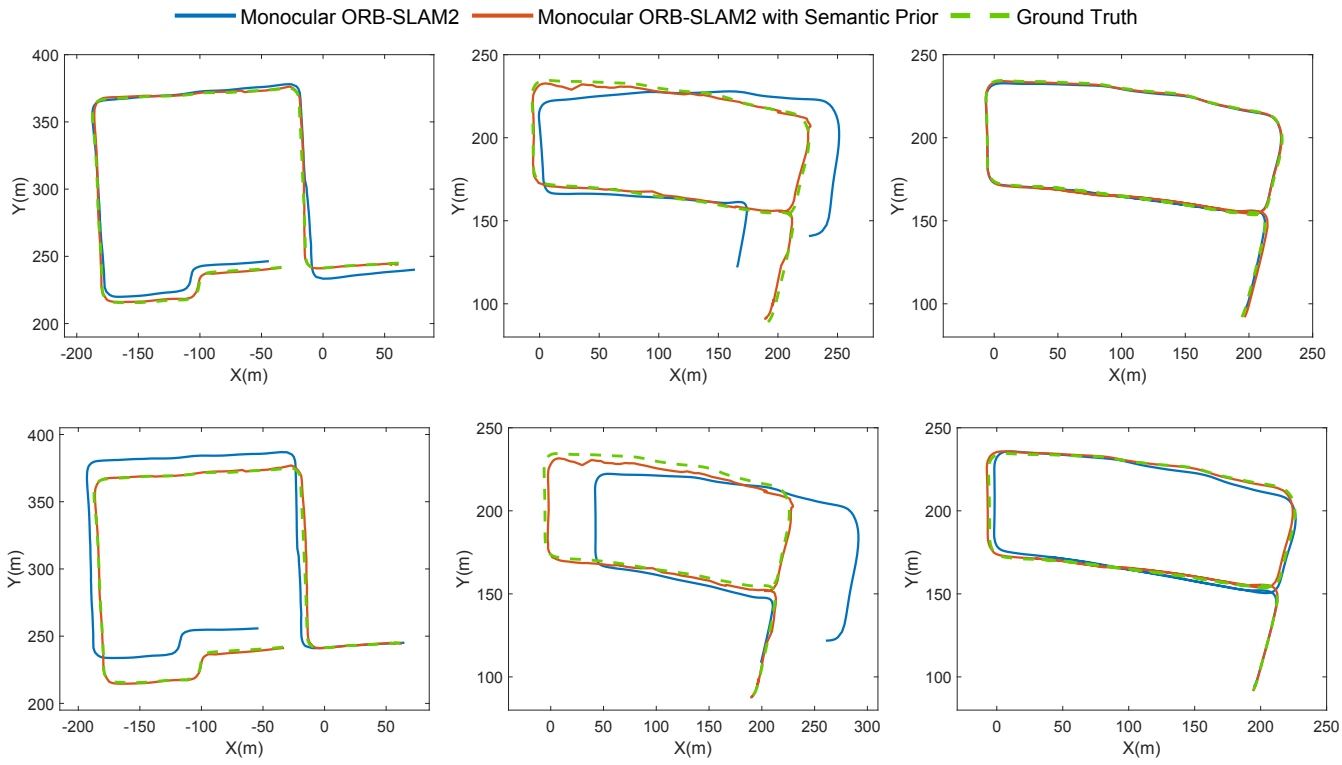


Fig. 7: Trajectory comparison from KITTI on (a) part of sequence 00, (b) part of sequence 05 before loop closure, and (c) part of sequence 05 after loop closure. First row: 7-DoF alignment is applied to whole trajectory. Second row: 6-DoF alignment is applied to first 20 keyframes with the scale from whole trajectory alignment.

[3] P. Schmuck and M. Chli, "CCM-SLAM: Robust and Efficient Centralized Collaborative Monocular Simultaneous Localization and Mapping for Robotic Teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[4] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, "Large Scale Graph-Based SLAM Using Aerial Images as Prior Information," *Autonomous Robots*, vol. 30, no. 1, pp. 25–39, Jan. 2011.

[5] O. Vysotska and C. Stachniss, "Exploiting Building Information from Publicly Available Maps in Graph-Based SLAM," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[6] M. Mielle, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "SLAM Auto-Complete: Completing a Robot Map Using an Emergency Map," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Oct. 2017, pp. 35–40.

[7] T. Caselitz, B. Steder, M. Ruhnke, and W. Burgard, "Monocular Camera Localization in 3d Lidar Maps," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 1926–1931.

[8] M. P. Parsley and S. J. Julier, "Towards the Exploitation of Prior Information in SLAM," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 2991–2996.

[9] ——, "Exploiting Prior Information in GraphSLAM," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 2638–2643.

[10] OpenStreetMap Contributors, "Planet Dump Retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[11] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, Jun. 2006.

[12] 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*, May 2011, pp. 3607–3613.

[13] G. Floros, B. van der Zander, and B. Leibe, "OpenStreetSLAM: Global Vehicle Localization Using OpenStreetMaps," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1054–1059.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation," in *ECCV*, 2018.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[17] Cityscapes, "Benchmark suite," https://www.cityscapes-dataset.com/benchmarks, 2019.

[18] M. Zuliani, C. S. Kenney, and B. Manjunath, "The Multiransac Algorithm and its Application to Detect Planar Homographies," in *IEEE International Conference on Image Processing 2005*, vol. 3. IEEE, 2005, pp. III–153.

[19] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust Map Optimization Using Dynamic Covariance Scaling," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 62–69.

[20] K. Doherty, D. Fourie, and J. Leonard, "Multimodal Semantic SLAM with Probabilistic Data Association," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2419–2425.