IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 6, 2020, accepted March 18, 2020, date of publication March 24, 2020, date of current version April 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2983121

VPS-SLAM: Visual Planar Semantic SLAM for Aerial Robotic Systems

HRIDAY BAVLE[1], **PALOMA DE LA PUENTE**[1], **(Member, IEEE), JONATHAN P. HOW**[2], **(Fellow, IEEE), AND PASCUAL CAMPOY**[1], **(Member, IEEE)**

[1]Centre for Automation and Robotics, Computer Vision and Aerial Robotics Group, Universidad Politécnica de Madrid (UPM-CSIC), 28006 Madrid, Spain
[2]Aerospace Controls Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA 02139, USA

Corresponding author: Hriday Bavle (hriday.bavle@upm.es)

This work was supported by the MISTI-Spain for the financial support in the project entitled Drone Autonomy and the Spanish Ministry of Economy and Competitivity for its funding Project (Complex Coordinated Inspection and Security missions by UAVs in cooperation with UGV) under Grant RTI2018-100847-B-C21. The work of Paloma de la Puente was supported in part by the Spanish Ministry of Economics and Competitivity under Grant DPI2017-86915-C3-3-R COGDRIVE.

ABSTRACT Indoor environments have abundant presence of high-level semantic information which can provide a better understanding of the environment for robots to improve the uncertainty in their pose estimate. Although semantic information has proved to be useful, there are several challenges faced by the research community to accurately perceive, extract and utilize such semantic information from the environment. In order to address these challenges, in this paper we present a lightweight and real-time visual semantic SLAM framework running on board aerial robotic platforms. This novel method combines low-level visual/visual-inertial odometry (VO/VIO) along with geometrical information corresponding to planar surfaces extracted from detected semantic objects. Extracting the planar surfaces from selected semantic objects provides enhanced robustness and makes it possible to precisely improve the metric estimates rapidly, simultaneously generalizing to several object instances irrespective of their shape and size. Our graph-based approach can integrate several state of the art VO/VIO algorithms along with the state of the art object detectors in order to estimate the complete 6DoF pose of the robot while simultaneously creating a sparse semantic map of the environment. No prior knowledge of the objects is required, which is a significant advantage over other works. We test our approach on a standard RGB-D dataset comparing its performance with the state of the art SLAM algorithms. We also perform several challenging indoor experiments validating our approach in presence of distinct environmental conditions and furthermore test it on board an aerial robot. **Video:** https://vimeo.com/368217703 **Released Code:** https://bitbucket.org/hridaybavle/semantic_slam.git

INDEX TERMS SLAM, visual SLAM, visual semantic SLAM, autonomous aerial robots, UAVs.

I. INTRODUCTION

Many indoor autonomous missions related to different applications require the usage of small-size aerial robots, able to navigate around narrow constrained spaces. This kind of vehicles cannot carry a lot of weight, so they only can be equipped with light sensors, such as RGB or RGB-D cameras, and processing units with limited computational resources. To operate in a truly autonomous way, accurate localization and meaningful mapping results are needed, which is indeed a challenging problem, especially regarding robustness.

Simultaneous Localization and Mapping (SLAM) using visual sensors may be feature-based (sparse, semi-dense or dense) or intensity-based. Most semi-dense SLAM techniques, like [1]–[3], rely on low level characteristic features of the environment such as points, lines and planes. This kind of approaches typically deteriorate in performance in the presence of illumination changes and repetitive patterns. On the other hand, other state of the art SLAM based techniques, such as [4]–[6], focus on dense 3D mapping of the environment, hence requiring high end CPU and GPU hardware in order to achieve real-time operation, which is a clear limitation on board an aerial robot with low computational capabilities.

Recent improvements in computer vision algorithms have made it possible to achieve object based detectors running real-time on lower end CPUs or GPUs. Combining such detectors with Visual Odometry (VO)/ Visual Inertial

The associate editor coordinating the review of this manuscript and approving it for publication was Charith Abhayaratne.

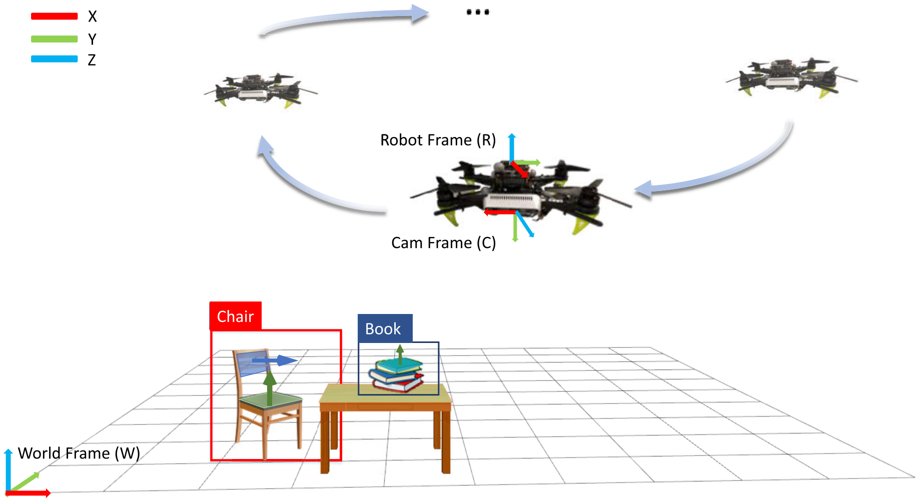


FIGURE 1. The aerial robotic platform used for validating the presented visual semantic slam approach with the frame of references.

Odometry (VIO) systems depending on low-level features can improve the accuracy of the data associations and provide more robust loop closures without high computational requirements, as shown in [7], [8]. Although adding semantic information to SLAM systems undoubtedly provides additional knowledge, extracting the accurate 3D position of the semantic objects is a challenging problem with important implications, since errors in the position estimation can induce errors in the data association and mapping of such semantic objects. The inaccuracies in estimating the 3D positions of the semantic objects are mainly due to two factors; (1) Uneven and complex 3D structures of the different instances of semantic object classes. (2) Errors in the semantic object detections i.e bounding boxes provided by the object detectors do not fit accurately around the detected object.

Several objects in common indoor environments present vertical and/or horizontal planar surfaces which can be extracted to improve the relative position estimation of these objects. Hence, in order to overcome the above mentioned limitations and to achieve a robust and lightweight SLAM algorithm, we propose a semantic SLAM approach using planar objects within the semantic detections.

The proposed algorithm can be divided into two parts. In the first part, the robot state is propagated using a VO/VIO estimate. Low-level features from the environment are used at this stage for the propagation of the robot state. Due to the inaccuracies in low-level feature detection and matching, as well as to errors and biases in the IMU measurements (for VIO systems), the VO/VIO estimations of the robot state often accumulate errors over time. We address this by associating the high-level planar surfaces of the detected semantic objects with the previously mapped semantic planes. To extract the planar surfaces within the detections, the output provided by state of the art object detectors is combined with a carefully applied plane extraction technique. Hence, the second part of the algorithm corrects the estimation and builds a sparse semantic map of the planar surfaces extracted from the semantic detections.

The created semantic map consists of planar surfaces represented by their centroids and normal orientations along with their class labels and the planar surface type (i.e. horizontal or vertical), which may be augmented by new detections of the semantic objects. To summarize, the main contributions of the presented work are:

- Robust and lightweight semantic SLAM algorithm suitable for running on board an aerial robot.
- Incorporation of fast planar extraction inside the semantic detections, for accurate high-level data association and mapping of the semantic landmarks.

The remainder of this document is organized as follows; Sect. II explains the current state of the art in geometric as well as semantic SLAM. Sect. III explains the semantic detection and planar extraction part along with Sect. IV describing the graph creation using the VO/VIO measurements and the extracted semantic information. Sect. V presents the performed experiments and obtained results using a standard dataset as well as using additional field experiments, comparing the accuracy of our approach with several state of the art geometric and semantic SLAM approaches. Finally, Sect. VI discusses the obtained results along with Sect. VII providing the final conclusions.

II. RELATED WORK

The research community has witnessed a great interest in visual SLAM based algorithms applied to robotics, so there is a vast visual SLAM related literature. Recently, SLAM techniques combining both geometric as well as semantic information have gained popularity and significant relevance [9]. It is now widely recognized that the incorporation of object-level information for accurate data associations and loop closures can increase the quality, robustness and interpretability of the solutions [10]–[12].

Salas-Moreno *et al.* [13] presented one of the first works in this direction: a real-time semantic SLAM approach called *SLAM++*. *SLAM++* was developed for an RGB-D sensor and it applies the ICP algorithm for 3D camera pose tracking, adding the estimates to a pose graph. It then integrates the relative 3D poses estimated from semantic objects previously stored in a database, in order to jointly optimize all the poses. Murali *et al.* [14] present an approach which integrates semantic information into a visual SLAM system. Within a gated factor graph framework, the semantic information is used for detecting the inliers/outliers of the system in order to achieve robust performance in the presence of dynamic obstacles. A pre-trained deep learning based object detector provides the semantic information of the objects. Sunderhauf *et al.* [15] propose a semantic mapping method combining ORB-SLAM2 [1] with deep learning based object detectors and 3D unsupervised segmentation of the planar information of the object detections. Our proposed approach is similar to this approach except that the authors only provide a semantic mapping framework -not a complete SLAM framework- and they perform time consuming data associations using euclidean distances between the 3D points of the detected object-landmark pairs, instead of more accurate Mahalonobis distance computation using the extracted landmark covariances.

Parkhiya *et al.* [16] present a monocular semantic SLAM approach. They use a deep network to learn 2D characteristic features from category specific objects, e.g. chair, and match it with a 3D CAD model to estimate the relative 3D pose of the semantic object. These semantic objects are added as landmarks along with the VO estimated pose of the robot into a graph optimization framework to obtain a corrected metric pose of the robot.

Grinvald *et al.* [17] propose a semantic mapping system based on a pose acquired from a geometric VIO sensor. This method utilizes geometric planar segmentation of a point cloud data and then uses semantic detections for the data association step and to further refine the segmentation. McCormac *et al.* [18] present an object-level SLAM system using RBG-D cameras called *Fusion++*, segmenting Truncated Signed Distance Function (TSDF) representations of the objects using the Mask-RCNN object detector. The objects are used for tracking, re-localization and loop closure, and their extracted pose is optimized over a pose graph. Bowman *et al.* [8] develop an extension to their previous work [7] thereby using semantic objects such as chairs and doors in a semantic SLAM approach,. The authors decompose the joint metric semantic SLAM problem into subcategories, namely (1) Continuous optimization of pose and (2) Discrete optimization of the semantic data association and semantic labels. The framework tightly couples the inertial, geometric and semantic information. Atanasov *et al.* [19] provide an extension to the framework presented in [8], extracting descriptive semantic features by using a convolutional neural network from semantic objects like cars, in order to tightly couple them with the geometric and inertial information.

One of the latest publications on semantic SLAM [10] proposed to track different possible hypotheses for the data association, in a robust framework for the context of urban driving. Also very recently, Yang *et al.* [12] proposed a unified SLAM framework including high level objects and planes based on monocular information. They do not require prior models and incorporate quite a novel and general object-to-plane constraint. Besides the fact that depth information is not considered, one significant difference from our approach is that this work uses 2D bounding boxes to represent the objects. Other innovative approaches have focused on point-wise semantic labeling for 3D lidar data within the SLAM framework itself [11]. This work also highlights urban scenarios for autonomous driving as an important application area.

Our proposed approach aims towards fast and efficient extraction of planar surfaces from objects, which can be used as semantic features, and thus generalizes to several semantic objects with planar surfaces, creating a sparse optimizable map of the environment, requiring minimal computational resources and hence capable of running on board aerial robotic platforms with low computational resources. Fig. 2 presents a global overview of the system with its distinct components explained in the following sections.

III. SEMANTICS BASED PLANAR EXTRACTION

A. SEMANTIC OBJECT DETECTION

The semantic object detection can be performed using any state of the art object based detectors. We select the You Only Look Once (YOLOv2) ([20]) object detector, in order to satisfy the on board computational limitations of the aerial robots. We select the lightweight Tiny-YOLOv2 model trained on the COCO dataset ([21]), providing a real-time performance with an average GPU consumption of 300 mb. The object detector is modified in order to provide detections only of the relevant objects above a certain probability threshold.

Although Tiny-YOLO does not require high computation, it requires an on board computer with a GPU support. Hence, in order to test our approach on board aerial robots without any GPU support, we also utilize a CPU based implementation of a shape and color based object detector, which is capable of detecting in real-time blue as well as red colored cube shaped objects. The detector first processes the color based information in HSV colored space in order to filter out objects based on color. The filtered image is then processed using a shape image processor which takes into account the approximate shape of the corresponding object in order to detect cube shaped objects of the corresponding color. More details regarding the detector can be found in [22].

The object detection is performed on the images received from the RGB camera. The detected bounding boxes from the

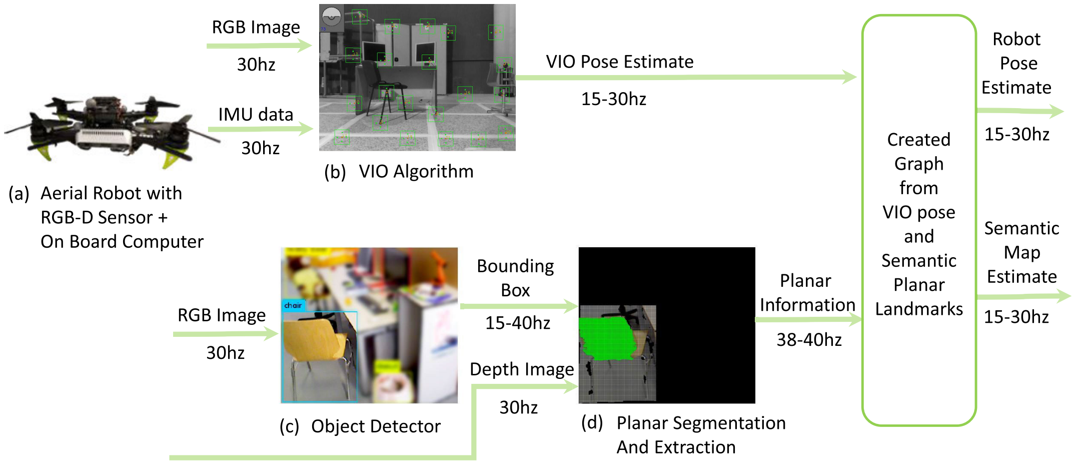


FIGURE 2. An overview of the proposed algorithm with all the connected components. (a) The aerial robot with the on board RGB-D camera. (b) VIO algorithm. (c) Object detector used. (d) Planar segmentation and extraction.

object detectors are then provided to the object segmentation (Sect. III-B) which segments the planar information from the 3D point clouds generated from the depth image registered with the RGB image.

B. SEMANTIC OBJECT SEGMENTATION

The received bounding boxes of the semantic objects from Sect. III-A, which are extracted from the RGB images, are used to extract the relevant 3D point cloud data of the semantic object. This 3D point cloud is generated from the corresponding depth image registered with the RGB image. As shown in Fig. 7c the received bounding boxes from the detector can have errors, as the bounding boxes do not fit the objects perfectly. Taking a median of characteristic 3D points or all 3D points within the detected bounding boxes as presented in [8] can cause errors in the calculation of the relative 3D position of the semantic object inducing errors in the mapping and data association of the semantic data. In order to minimize the errors in the mapping of the semantic objects, inspired from our previous approaches of planar clustering and segmentation ([23], [24]), we segment all the horizontal and vertical planar surfaces present within the detected bounding boxes with their centroids as well as their normal orientations in the following manner:

1) NORMAL EXTRACTION

In order to perform fast and real-time extraction of normals at each 3D point, we use the integral normal estimation technique ([25]). Briefly, this technique first converts the z component of the input 3D point cloud $\boldsymbol{I_i}$ to an integral image $\boldsymbol{I_m}$, since in an integral image, the sum of all values within a particular region is calculated, it makes the computation very fast and efficient. After performing a smoothing, choosing the appropriate smoothing areas using depth change maps, a normal $\boldsymbol{n_p}$ at point $\boldsymbol{p}$ is calculated as: $\boldsymbol{n_p} = \boldsymbol{p_h} \times \boldsymbol{p_v}$. Where $\boldsymbol{p_h}$ is the vector of 3D right and left neighbors of point $\boldsymbol{p}$ and $\boldsymbol{p_v}$ is the vector of the 3D top and bottom neighbors of point $\boldsymbol{p}$. The matrix $\boldsymbol{N_p^c}$ contains all the normal orientations of each 3D point in the point cloud, and is passed to the centroid extraction step explained below.

2) CENTROID EXTRACTION

For fast and robust planar surface and centroid extraction, we use the method proposed in [26]. This approach takes as input the $\boldsymbol{N_p^c}$ computed from the previous step as well as all the corresponding 3D points vector. All planes are represented by the planar equation $ax + by + cz + d = 0$ and each 3D point is thus represented as a vector consisting of the euclidean point, its normal and n_d:

$$p = \{x, y, z, n_x, n_y, n_z, n_d\} \tag{1}$$

where $n_d = \{x, y, z\} \cdot \{n_x, n_y, n_z\}$. Euclidean distances between the normal directions and the distances n_d are computed for the neighboring points in order to find all the connected components. The computed connected components are then checked for their curvatures in order to filter non-planar components. After the extraction of all the planar surfaces with their centroids and normals, we first check whether the planar surfaces are horizontal planes as follows:

$$d_{hor} = ||\boldsymbol{n_p} - \boldsymbol{n_g}|| \tag{2}$$

where $\boldsymbol{n_p}$ is the normal to the extracted planar surface and $\boldsymbol{n_g}$ is the normal of the ground planar surfaces which is known. If the the d_{hor} is less than t_{hor}, then the planar surface $\boldsymbol{n_p}$ is labeled as a horizontal plane. If the d_{hor} is greater than the t_{hor}, the planar surface is checked for vertical threshold as:

$$d_{vert} = \boldsymbol{n_p} \cdot \boldsymbol{n_g} \tag{3}$$

If d_{vert} is less than the t_{vert} the planar segment $\boldsymbol{n_p}$ is labeled as a vertical plane. The extracted planar surfaces thus contain the following information:

- Centroids $\boldsymbol{s}_p = \{p_x, p_y, p_z\}$
- Normals $\boldsymbol{s}_n = \{n_x, n_y, n_z, n_d\}$
- Planar type label $s_o = \begin{cases} 1, & \text{if horizontal} \\ 0, & \text{if vertical} \end{cases}$
- Class type label s_c = corresponding detected class type.

IV. GRAPH SLAM

The pose estimates from the VO/VIO algorithms can accumulate errors in absence of characteristic features in the environment and as seen from III-B semantic detections can have errors due to occlusions, insufficient lighting, resulting in uncertainties in the 3D position estimates as well. Due to the presence these uncertainties the use of filtering techniques like the Extended Kalman Filter (EKF) SLAM could cause divergence in the estimate of the robots as well as the landmarks poses. As opposed to filtering techniques which consider only the most recent previous state of the robot, graph slam based techniques provide the advantage of considering all the previous robots states, as well as account for higher non-linearities. Hence, in order to robustly fuse the measurements from the VO/VIO and the semantic detections, we use graph slam based optimization. The algorithm can be divided into three main stages:

A. VO/VIO ODOMETRY

The advantage of using loosely coupled approach for fusing VO/VIO estimations with semantic data, allows for integrating several state of the art VO/VIO systems, into the framework, using the best approach for a particular environment. In this work we integrate into our framework three different visual odometry algorithms namely: 1. ROVIO ([27]) 2. Snap VIO[1] 3. RTAB-map odometry ([28]).

ROVIO is a monocular VIO algorithm based on an Extended Kalman Filter (EKF). ROVIO uses direct image intensity errors of image patches in order to achieve robust tracking. These image intensity errors are used as innovation terms during the update stage of the EKF. These image intensity measurements are tightly coupled with an IMU to accurately estimate the pose of the robot with true metric scale. But due to the noise in the image intensity calculations as well as the IMU measurements, pose estimates provided by ROVIO can accumulate huge drift over time and in many scenarios the pose estimate can diverge without recovery.

During very high angular motions of the aerial robots and less characteristic features in the environment, the pose estimated by ROVIO can tend to diverge completely from the true robot pose. Due to this limitation, for high speed flights on board the aerial robots, we use the snap VIO algorithm. The snap VIO algorithm is also a monocular approach which is able to estimate the pose of the robot, where the robot pose estimates drift with time but the algorithm does not tend to diverge completely from its true value as in case of ROVIO.

In order to compare our proposed approach with a standard RGB-D dataset ([29]) which does not provide synchronized IMU data, we use the RTAB-map RGB-D based visual odometry module. The odometry algorithm uses Feature to map (F2M) approach which registers a new keyframe with local map of features created from the previous keyframe. It uses the Good Features to Track (GFTT) ([30]) for extracting the keypoint features and uses the BRIEF descriptors of the extracted features to match against those of the local map created. The motion estimation is performed using Perspective-n-Point algorithm (PnP) as presented in the OpenCV library ([31]). A local bundle adjustment is performed in order to refine the obtained odometry estimate.

B. GRAPH CONSTRUCTION

The robot state vector $\boldsymbol{x} = [\boldsymbol{x_r}, \boldsymbol{R_r}]$ is propagated over keyframes k. Where $\boldsymbol{x}_r = [x, y, z]$, x, y and z are the robots position estimates along the x, y and z axis with respect to the world frame of reference W respectively (Fig. 1) and $\boldsymbol{R}_r$ is the rotation matrix of the robot with respect to the world frame W. We assume that the initial state of the robot is known.

Each landmark consists of it state and covariance with the labels for the planar surface type as well as the class type represented as $\boldsymbol{L} = \boldsymbol{L}_1, .., \boldsymbol{L}_n$. Where $\boldsymbol{L}_i = (\boldsymbol{l}_{z_i}, \boldsymbol{l}_{\sigma_i}, l_{o_i}, l_{c_i})$, $\boldsymbol{l}_{z_i}$ being the 3D position of the i-th landmark, $\boldsymbol{l}_{\sigma_i}$ is the covariance of the i-th landmark and l_{o_i} and l_{c_i} being the planar as well as the class type of the i-th landmark.

The front end of the algorithm comprising of VO/VIO provides the 3D pose estimates in the world frame of reference W. The estimate of the robot state $\boldsymbol{x}_r$ at time t is added to factor graph as a keyframe node K_t. The constraint between adjacent keyframes K_{t-1} and K_t is added in the form of an edge using the pose increment between them $\boldsymbol{u}_r(k)$. The pose increment obtained from the the VO/VIO poses at time $t-1$ and time t can be derived as:

$$\boldsymbol{u}_{\boldsymbol{r}_t} = \ominus \boldsymbol{x}_r(k-1) \oplus \boldsymbol{x}_r(k) \tag{4}$$

$\boldsymbol{x}_r(k-1)$ and $\boldsymbol{x}_r(k)$ are the pose measurements received at time $t-1$ and t respectively. Each keyframe K_i is added to factor graph depending on time as well as motion constraints of the robot.

Each detected semantic object $\boldsymbol{S}_i$ after undergoing the process of data association (Sect. IV-C), is either added to the factor graph as a landmark node augmenting the map of semantic landmarks $\boldsymbol{L}$ or associated with the currently mapped semantic landmarks. The relative pose of landmark observed from the keyframe K_i is added as the constraint between the landmark and the respective keyframe. Fig. 3 shows the graph constructed using n keyframes and three semantic landmarks detected with their extracted planar surfaces.

C. DATA ASSOCIATION

The semantic planar surfaces extracted from section III-B are received first by the data association stage in the following manner: $\boldsymbol{S}_i = \{\boldsymbol{s}_{z_i}, \boldsymbol{s}_{n_i}, s_{o_i}, s_{c_i}\}$. $\boldsymbol{S}_i$ is the first detected and extracted semantic planar surface i, containing $\boldsymbol{s}_{z_i}$ and $\boldsymbol{s}_{n_i}$

[1] https://github.com/ATLFlight/ros-examples

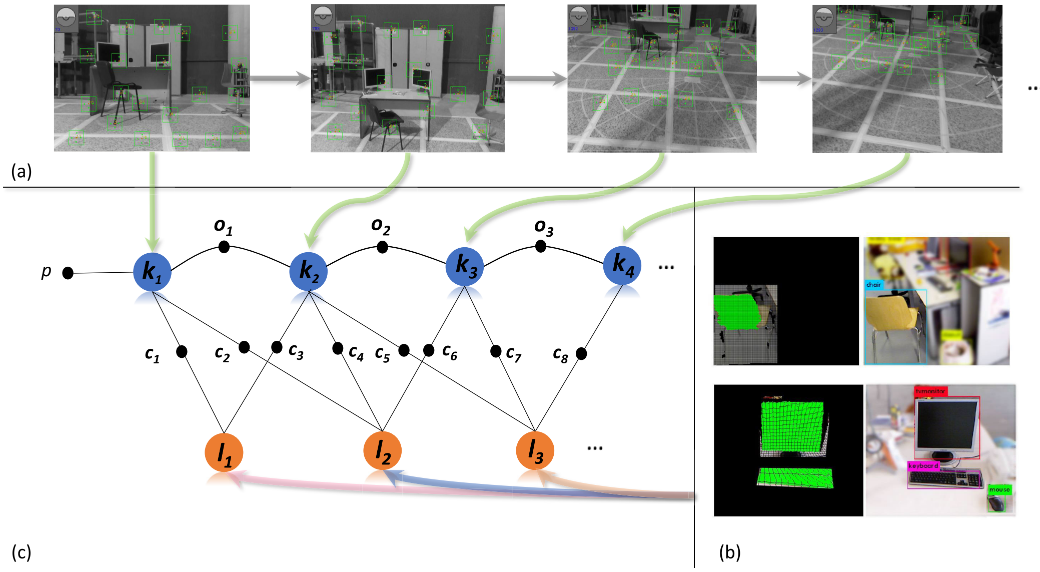


FIGURE 3. The structure of the graph created, using the VIO and the semantic landmarks. (a) VIO feature extraction and pose estimation. (b) The semantic detections in the RGB image with the inliers of the planar surfaces in green extracted from the registered point cloud. (c) The constructed graph, where K_i is the keyframe created for a VIO pose connected with the relative poses between them.

as the centroids of the detected planar surface and its corresponding normal orientations in the camera frame C (see Fig. 1). s_{o_i} and s_{c_i} the planar surface type and the class labels respectively. The first received semantic object does not undergo the data association and is directly mapped as the first semantic landmark which can be represented as:

$$\boldsymbol{L}_i = \{\boldsymbol{l}_{z_i}, \boldsymbol{l}_{n_i}, \boldsymbol{l}_{\sigma_i}, l_{o_i}, l_{c_i}\} \tag{5}$$

where the $\boldsymbol{l}_{z_i}$ and $\boldsymbol{l}_{n_i}$ are the semantic landmark centroids and normals in the world frame of reference obtained as:

$$\boldsymbol{l}_{z_i} = \boldsymbol{x}_r \oplus {}^{w}\boldsymbol{R}_c \cdot \boldsymbol{s}_{z_i} \tag{6}$$

$$\boldsymbol{l}_{n_i} = {}^{w}\boldsymbol{R}_c \cdot \boldsymbol{s}_{n_i} \tag{7}$$

where ${}^{w}\boldsymbol{R}_c$ is the rotation matrix from the camera frame to the world frame. $\boldsymbol{l}_{\sigma_i}$ is the uncertainty of the estimated position of the semantic landmark. The initial value of $\boldsymbol{l}_{\sigma_i}$ is decided based on the number of 3D points contained in planar surface. Semantic object with lower number of 3D points corresponds to high certainty of error in it estimated centroids and normal orientations and will have a higher $\boldsymbol{l}_{\sigma}$. After the mapping of the first semantic landmark, each detected semantic planar surface $\boldsymbol{S}_k$ undergoes the data association process in the following three steps:

- First, the received semantic planar surface is checked for whether its class label matches the class label of the semantic landmarks, and whether its planar type is equal to the planar type of the semantic landmark. It further undergoes a check for whether the number of 3D points representing the planar surface are greater than a certain threshold t_p and if the area of the semantic planar surface is greater than a threshold t_a. This step ensures exclusion of erroneous centroids of the planar surfaces extracted due to the detected bounding boxes fitting incompletely over a semantic object.
- In the second step the normal orientation of the planar surface in the camera frame is converted to the world frame using Eq. 7 and is represented as $\boldsymbol{l}_{n_k}$. The difference $\boldsymbol{l}_{n_k}$ between $\boldsymbol{l}_{n_i}$ has to be lower than a predefined threshold t_n, for the planar surface to pass to the next step.
- If the semantic planar surface passes the first step and the second step, the relative 3D measurements of the centroids are converted from the camera frame to the world frame using the Eq. 6. We can then compute the Mahalonobis distance for the detected semantic object with mapped landmarks. If the computed Mahalonobis distance is greater than a given threshold, the detected semantic object is mapped as a new landmark $\boldsymbol{L}_j$, else the semantic object $\boldsymbol{S}_k$ is matched with the current semantic landmark $\boldsymbol{L}_i$.

During the graph optimization step (Sect. IV-D), both the positions $\boldsymbol{l}_z$ as well as the covariance $\boldsymbol{l}_{\sigma}$ of all the mapped semantic landmarks are optimized.

D. GRAPH OPTIMIZATION

After the construction of the graph (Sect. IV-B), the graph optimization step consists of finding a configuration of the nodes that best fits the given VO/VIO measurements and the semantic landmarks. $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)^T$ is the vector of state of the robot, where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are the poses of nodes i and j connected using the edge $\hat{z}_{ij}$, which is the relative pose between them obtained from the VO/VIO estimates. $\boldsymbol{\Omega}_{ij}$ is the information matrix between nodes i and j. z_{ij} is the semantic landmark measurement observed by the nodes i and j. The log likelihood of the measurement therefore can be given as:

$$l_{ij} = [z_{ij} - \hat{z}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)]^T \cdot \boldsymbol{\Omega}_{ij} \cdot [z_{ij} - \hat{z}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j)] \tag{8}$$

where,

$$\boldsymbol{e}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) = z_{ij} - \hat{z}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{9}$$

is the difference between the expected observations from the VO/VIO odometry and the received measurements from the semantic landmarks. For a pair of observations C, the least square estimation problem thus seeks to find $\boldsymbol{x}^{\star}$ that best fits all the previous observations, given as:

$$\boldsymbol{x}^{\star} = \underset{x}{\text{argmin}} \sum_{i,j \in C} \boldsymbol{e}_{ij}^T \boldsymbol{\Omega}_{ij} \boldsymbol{e}_{ij} \tag{10}$$

In order to increase the robustness to possible outliers present in the semantic detections, Psuedo-Huber cost function is added to all the semantic measurement constraints. The solution to Eq. 10 can be found by linearizing around the initial guess $\tilde{\boldsymbol{x}}$, which leads to iteratively solving for the a linear system with matrix $\boldsymbol{H}$ and right hand vector $\boldsymbol{b}$ as:

$$\boldsymbol{H} = \sum_{i,j \in C} \boldsymbol{J}_{ij}(\tilde{\boldsymbol{x}})^T \boldsymbol{\Omega}_{ij} \boldsymbol{J}_{ij}(\tilde{\boldsymbol{x}}) \tag{11}$$

$$\boldsymbol{b}^T = \sum_{i,j \in C} \boldsymbol{e}_{ij}^T \boldsymbol{\Omega}_{ij} \boldsymbol{J}_{ij}(\tilde{\boldsymbol{x}}) \tag{12}$$

where, $\boldsymbol{J}_{ij}$ is the Jacobian of the error function computed in $\tilde{\boldsymbol{x}}$.

In this optimization process, the robot pose as well the semantic landmarks positions are optimized. After the factor graph optimization we can also recover the updated covariances of the mapped landmarks, in order to compute the Mahalonobis distance required for data association (Sect. IV-C). Re-observing the semantic object and correctly associating it the the mapped landmark results in the loop closure after the optimization step. For fast and efficient computation of the non-linear optimization problem, G2O ([32]) framework is utilized with Lavenberg-Marquardt solver. Algorithm. 1 explains the complete working the proposed algorithm.

Algorithm 1 Visual Semantic SLAM

```
Input: VO/VIO estimated ouput of the robots pose $\boldsymbol{x}$,
       along with the detected and segmented semantic
       data $\boldsymbol{S}_j$
Output: Corrected pose of the robot $\boldsymbol{x}_r$ and the semantic
        landmark map $\boldsymbol{L}$
VO/VIO
if $time > time_{thres}$ & $dist > dist_{thres}$ then
    Add estimated pose of the robot from the VO/VIO as a keyframe node $K_i$
    Add an edge between the $K_{i-1}$ and $K_i$ using the relative pose
Data Association
if $first\_semantic\_data$ then
    $\boldsymbol{l}_i = \{\boldsymbol{l}_{z_i}, \boldsymbol{l}_{\sigma_i}, l_{o_i}, l_{c_i}\}$
    Add a new landmark node with its relative pose
    Add an edge between the landmark node and current keyframe $K_i$
else
    for $i = 1$ to $num_{detections}$ do
        for $j = 1$ to $num_{landmarks}$ do
            if $s_{o_i} = l_{o_k}$ and $s_{c_i} = l_{c_k}$ then
                $\boldsymbol{l}_{\sigma_i}$ ← Get the optimized landmark covariances
                $\boldsymbol{l}_{\sigma_j}$ ← based on the number of 3D points in the semantic planar surface
                Calculate the min-Mahalonobis distance
                $\boldsymbol{v}_i = \boldsymbol{l}_{z_i} - \boldsymbol{l}_{z_j}$
                $\boldsymbol{Q} = H \cdot \boldsymbol{l}_{\sigma_i} \cdot (H)^T + \boldsymbol{l}_{\sigma_j}$
                $d_i = \boldsymbol{v}_i^T \cdot \boldsymbol{Q} \cdot \boldsymbol{v}_i$
            else
                Map a new semantic landmark
                $\boldsymbol{l}_{\sigma_k} = \boldsymbol{l}_{\sigma_j}$ & $\boldsymbol{L}_k = [\boldsymbol{l}_{z_k}, \boldsymbol{l}_{n_i}, \boldsymbol{l}_{\sigma_k}, l_{o_k}, l_{c_k}]$
                Add a new landmark node with its relative pose
                Add an edge between the landmark node and current keyframe $K_i$
            end
        end
        if $min\text{-}Mahalonobis\ distance \leq thres_{dist}$ then
            Add an edge between the mapped landmark with the current keyframe $K_i$
        else
            Map a new semantic landmark
            $\boldsymbol{l}_{\sigma_k} = \boldsymbol{l}_{\sigma_j}$ & $\boldsymbol{L}_k = [\boldsymbol{l}_{z_k}, \boldsymbol{l}_{n_i}, \boldsymbol{l}_{\sigma_k}, l_{o_k}, l_{c_k}]$
            Add a new landmark node with its relative pose
            Add an edge between the landmark node and current keyframe $K_i$
        end
    end
end
Back-End
Sparse map of the semantic landmarks
Optimize the robot pose $\boldsymbol{x}_r$
Optimize the landmark poses and covariances $\boldsymbol{l}_{z_i}$ and $\boldsymbol{l}_{\sigma_i}$
```

V. EXPERIMENTS AND RESULTS

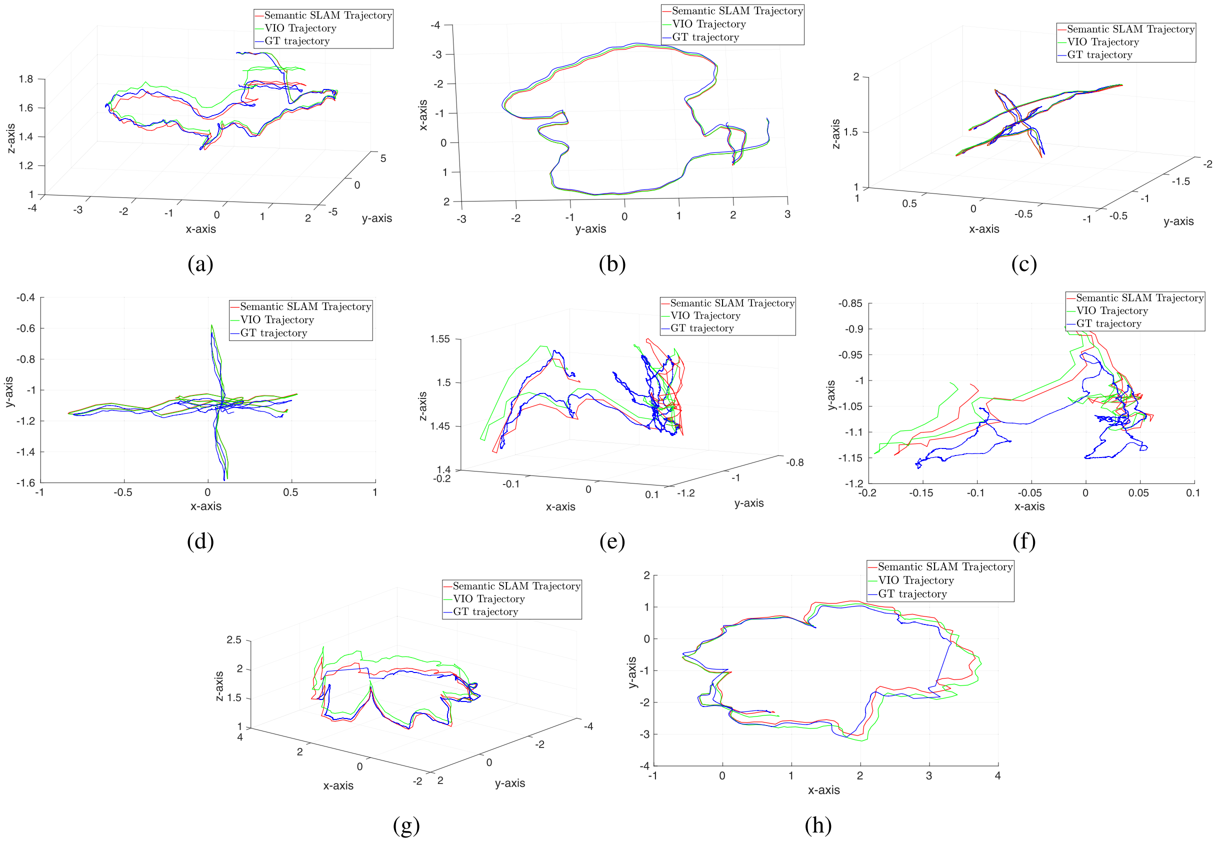
A. STANDARD DATASET

To validate our approach, we test on standard dataset and compare it with state of the art approaches based on geometric as well as object based SLAM approaches.

1) RGB-D SLAM TUM DATASET

This dataset[2] ([29]) consists of point cloud data provided from a kinect sensor and a motion capture system for the ground truth data. The odometry data for the dataset is obtained using the RTAB map RGB-D visual odometry algorithm ([28]) explained in Sect. IV-A and several semantic objects such as chairs, tv-monitors, books and keyboard are detected and mapped as semantic landmarks and used for loop closure

[2] https://vision.in.tum.de/data/datasets/rgbd-dataset/download



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

FIGURE 4. The trajectory of the camera estimated by the proposed algorithm when comparing it to the ground truth estimated trajectory for the RGB-D TUM dataset. (a) and (b) present the 3D and 2D trajectories for Freiburg3 Long Office Household sequence. (c) and (d) present the 3D and 2D trajectories for Freiburg2 XYZ, (e) and (f) show the 3D and 2D trajectories for sequence freigburg2 RPY and (g) and (h) shpw the 3D and the 2D trajectories for freigburg2 desk.

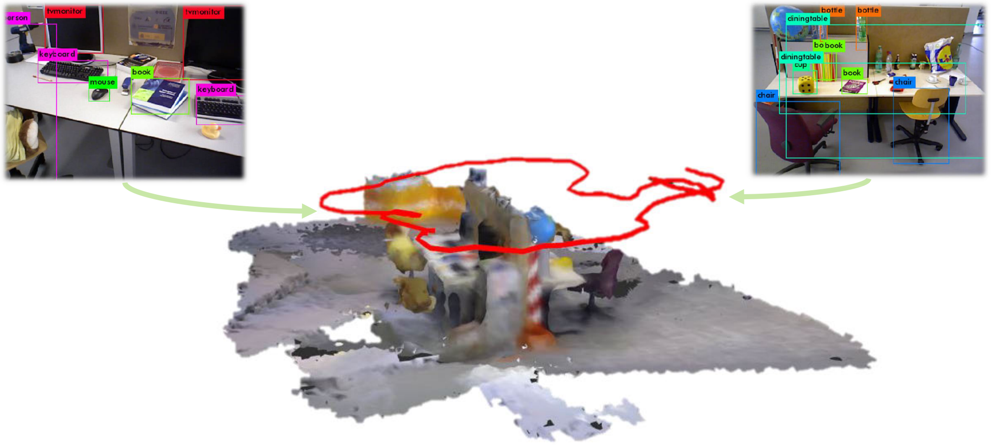


FIGURE 5. The freigburg3 long office household sequence presenting the generated 3D point cloud map of the environment with the estimated trajectory of the camera shown in red, along with detections received from the yolo detector.

detection. We describe below the experiments performed on several sequences of the dataset with the obtained results and discuss their results in Sect. VI.

a: Freiburg3 LONG OFFICE HOUSEHOLD (fr3/office)

In this sequence the kinect RBG-D camera is moved along an environment consisting of several office materials consisting of chairs, tables, books, tv-monitors etc. The camera is moved in different translational motions as well as rotational motions in order to validate the robustness of the proposed approaches. Fig. 5 shows the 3D point cloud map generated using the pose estimated by the semantic SLAM algorithm. Figures 4a and 4b present the trajectory of the camera estimated by our proposed algorithm and the ground truth trajectory.

During the execution of this sequence, the average trajectory error (ATE) ([29]) estimated by our proposed algorithm is 0.033 m, whereas the ATE of the VIO algorithm is 0.043 m.

b: Freiburg2 XYZ (fr2/xyz)

In this sequence the kinect camera was moved in translational motion along with x, y and z axis without any rotational motion. Only two semantic objects, a TV-monitor and a keyboard were used as semantic landmarks. Figures 4c and 4d present the estimated trajectory by our proposed algorithm when comparing it to the ground truth trajectory. The ATE of our proposed algorithm for this sequence is 0.0181 m and that of the VIO is 0.0182 m.

c: Freiburg2 RPY (fr2/rpy)

This sequence is similar to the Freiburg2 XYZ experiment, with additional rotational motion of the camera along with with the translational motion. Only two semantic objects, a TV-monitor and a keyboard were used as semantic landmarks. Figures 4e and 4f present the estimated trajectory by our proposed algorithm when comparing it to the ground truth trajectory. The ATE of our proposed algorithm for this sequence is 0.019 m and that of the VIO is 0.0202 m.

d: Freiburg2 DESK (fr2/desk)

In this sequence the kinect camera is moved in translational as well as rotational motion along a desk containing semantic object such as tv-monitors, chairs, keyboard and books. Figures 4g and 4h compare the trajectory estimated by our proposed algorithm with the ground truth trajectory. In this sequence as can be seen from the figure, during several time instances there is absence of ground truth trajectory estimation. The ATE for the robot pose and the VIO pose are 0.076 m and 0.103 m respectively.

B. FIELD EXPERIMENTS

We evaluate our algorithm on several experiments using different system setups and validate it in different indoor scenarios. All the field experiments are compared to the state of the art ORB-SLAM2 ([1]) as its one of the widely used open source SLAM frameworks available. Although our approach uses VIO, which incorporates additional IMU information, the aim of this comparison is to evaluate the performance of our proposed framework which uses higher level semantic information, against a SLAM framework using only low level features.

1) SYSTEM SETUP

For the experiments two different system setups were tested as described below:

a: HAND-HELD SETUP

In this experimental setup, we use a single Intel RealSense D435i camera.[3] This version of the RealSense consists of an RGB camera, two infrared cameras providing depth information and an IMU sensor. The ROVIO is used for the odometry information and is executed using the RGB camera and the IMU. The RGB camera is used for semantic detections along the depth information for extracting the 3D point cloud data required for extracting the planar surfaces from the detected semantic objects.

b: AERIAL ROBOTIC SETUP

In this experimental setup, we use the same Intel RealSense camera except that, we use the odometry obtained from Snapdragon VIO[4] sensor setup. We use the Snapdragon VIO as it is optimized for working on board the aerial robots, hence can achieve high speed flights without the problem of complete divergence of the VIO algorithm.

2) RESULTS

In this section, we present the results obtained using the different system setups as explained in the previous section in several challenging indoor scenarios in order to validate the accuracy of our proposed approach. Table. 1 presents the average runtime of each component of the algorithm on board an Nvidia TX2 computer and Table. 3 presents the ATE with respect to the ground truth data obtained during the execution of field experiments described in Sect. V-B2c, Sect. V-B2d and Sect. V-B2e respectively.

TABLE 1. Average frequencies in hz of each component of the proposed algorithm on board an Nvidia TX2.

	Average Frquency
Planar Segmentation	29.22
Object Detection (Shape and Color)	38.81
Object Detection (Tiny-YOLO)	15
Graph Optimization	14.30
VIO (ROVIO [27])	13.8
VIO (SNAP)	30

TABLE 2. Absolute Trajectory Error (ATE) m of the compared algorithms for the TUM dataset.

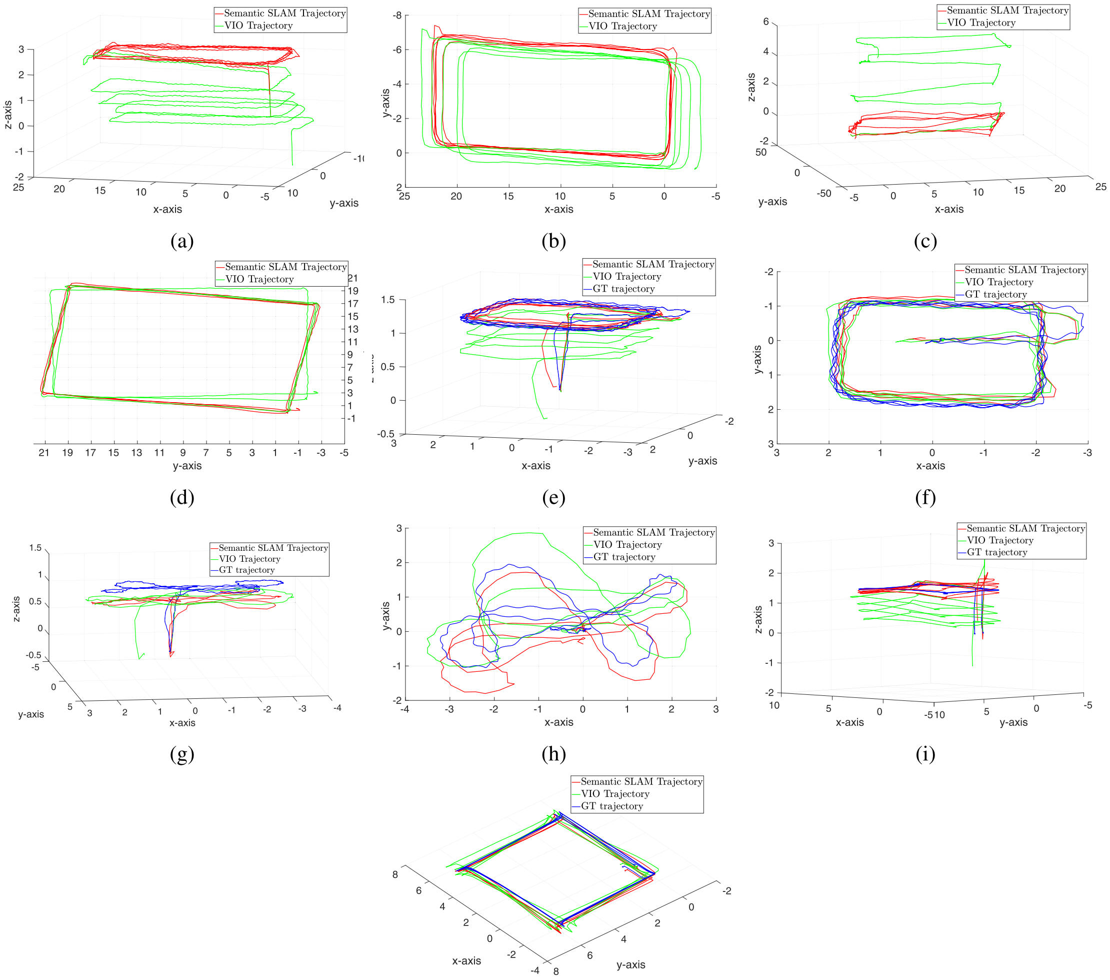
	fr2/xyz	fr2/rpy	fr2/desk	fr3/office
RTAB-VO [28]	0.0182	0.020	0.102	0.043
Our approach	0.018	0.019	0.076	0.033
Object based SLAM				
MaskFusion [33]	0.041	0.076	0.108	0.102
Fusion++ [18]	0.020	–	0.114	0.108
Geometric SLAM				
ORB-SLAM2 [1]	**0.004**	–	**0.009**	**0.010**
Elastic Fusion [5]	0.011	**0.015**	0.071	0.017
Kintinuous [34]	0.029	0.018	0.034	0.030
DVO SLAM [35]	0.018	0.018	0.017	0.035
RGB-D SLAM [29]	0.08	–	0.057	0.032

a: LONG HALL EXPERIMENT

We performed several experiments in an indoor environment consisting of a long passageway of approximately 22 m in length and 6 m in width. The experiments are performed using

[3] https://www.intelrealsense.com/depth-camera-d435i/

[4] https://github.com/ATLFlight/ros-examples



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

FIGURE 6. The results obtained of the field experiments. where (a) and (b) represent the 3D and the 2D plot for the long hall experiment. (c) and (d) demonstrate te 3D and the 2D plot for the long corridor experiment. (e) and (f) are the 3D and 2D plots obtained for the repetitive trajectory with several semantic objects experiment. (g) and (h) are the 3D and 2D plots obtained for the experiment of random trajectory with several semantic experiment. (j) and (k) demonstrate the 3D and 2D plots for the experiment on board the aerial robot.

TABLE 3. Absolute Trajectory Error (ATE) m obtained during the field tests.

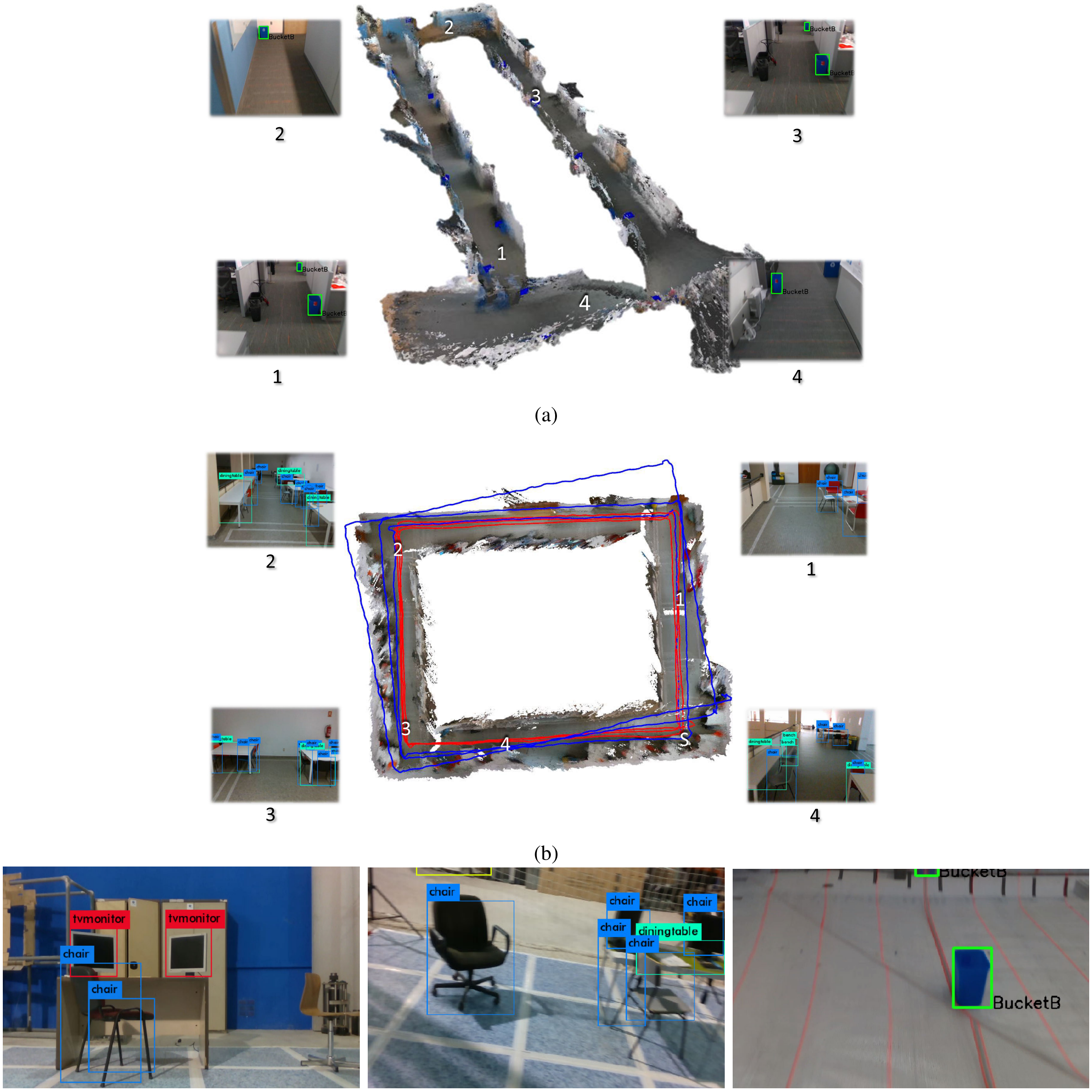
	Repetitive Trajectory	Random Trajectory	On-board Aerial Robot
Our Approach	**0.225**	0.393	**0.280**
VIO	0.312	0.550	0.651
ORB-SLAM2 [1]	0.267	**0.317**	–

the on board aerial robotic setup explained in Sect. V-B1. Due to the restrictions of flying in the area, the aerial robotic setup is moved in a hand held fashion. During this experiment, a total of 5 rounds are performed, covering an approximate total trajectory length of 250 m. The idea of this experiment is to test the robustness of the algorithm in presence of long trajectories and compare the results with VIO estimates of the robot. Since no ground truth measurements are present during this experiment, the 5 rounds are performed in a repetitive pattern, in order to demonstrate the drift accumulated by the VIO after each round and the accurate drift free pose estimate provided by our approach. We present the accuracy of the algorithm through the 3D map of the environment constructed using the estimated pose estimated by our algorithm.

Planar surfaces extracted from the recycle bins, which were commonly present in the environment, are used as semantic objects. A total of only 15 of these randomly placed semantic objects are used for mapping and improving the drift of the VIO. Fig. 6 shows the 3D and the 2D plots obtained when performing the experiments and Fig. 7a shows the 3D point cloud map generated during this experiment along with the detections of the recycle bins at distinct time intervals.

b: LONG CORRIDOR EXPERIMENT

The aim of this experiment is to validate the robustness of our algorithm in presence of large errors in estimations of the VIO/VO algorithms as well as in presence of large clutter of semantic objects. The experiment is performed in a long corridor of length 14 m and width 20 m, with challenging



(a)

(b)

(c) (d) (e)

FIGURE 7. The generated 3D map of the environments along with the detections of the semantic objects during the execution of the field experiments.

illumination conditions adversely effecting the performance of the VIO algorithm. This experiment is performed using the hand held camera setup as explained in Sect. V-B1. The environment consists of a clutter of several semantic objects in the form of chairs and tables. Only the planar surfaces of chairs are used for the semantic mapping, as the horizontal planes of the tables do not fit entirely into the bounding boxes, which cause errors in its relative position estimation. A total of three repetitive trajectory loops are performed around the corridor, covering an approximate total trajectory length of 204 m. Fig. 6c and Fig. 6d present the 3D and the 2D plots obtained during this experiment, comparing the pose estimated using our approach and the VIO pose estimate. In order to demonstrate the accuracy of our algorithm in absence of ground truth data, we present the 3D map of the corridor (Fig. 7b), generated using the 3D pose estimated by our approach, also showing the trajectories estimated by our approach and the VIO, along with the detections at distinct time instances.

c: REPETITIVE TRAJECTORY WITH SEVERAL SEMANTIC OBJECTS

In order to validate the approach in presence of several semantic objects of different shapes and sizes, and in presence of ground truth information, we perform an experiment with the hand held camera setup explained in Sect. V-B1. Several semantic objects such as chairs, tv-monitors, keyboards and books are used, placing them in random positions. A total of 4 rounds are performed in a repetitive manner during this experiment with a total approximate trajectory length of 69 m. Fig. 6 shows the 2D and the 3D plots obtained in this experiment, comparing the results of output of the algorithm with the ROVIO, ORB-SLAM2 and ground truth data. Our proposed approach has an ATE of 0.225 m, whereas ATE of the VIO algorithm used is 0.312 m. The ORB-SLAM2 algorithm has an ATE of 0.267 m.

d: RANDOM TRAJECTORY WITH SEVERAL SEMANTIC OBJECTS

The aim of this experiment is to test the robustness of the proposed approach in presence of random translational as well as rotational motions of the camera with a random clutter of several semantic objects. The experiment is performed using the hand held camera setup explained in Sect. V-B1. The semantic objects present are chairs, tv-monitors, books and keyboards. The clutter of the semantic objects induces errors in the detections of the yolo detector as seen in Fig. 7d. Fig. 6g and Fig. 6h show the 3D and the 2D plots obtained during the execution of this experiment. The results obtained from our approach are compared with the ROVIO, and the ORB-SLAM2. Our approach has an ATE of 0.393 m, whereas the ATE's of ROVIO and ORB-SLAM2 are 0.550 m and 0.317 m respectively.

e: ON-BOARD AERIAL ROBOT

In order to test the robustness of the algorithm in presence of high speed motions of an aerial robot, we run it on board an aerial robot flying at maximum speeds of 2 m/s. The on board aerial robotic setup explained in Sect. V-B1 is used with the aerial robot flown manually, performing a squared trajectory with several loops. The semantic objects used in this scenario are the blue colored recycle bins. A total of only 6 recycle bins are used, which proved sufficient along with the VIO data in order to correctly estimate the trajectory of the aerial robot. During the take-off phase of the aerial robot, due to its sudden motion, it can induce huge drift in the z-axis estimate of the VIO pose which cannot be corrected as no semantic landmark is mapped during this initial time instant. In order to compensate for this initial drift, we introduce the approximate location of the first landmark, which is optimized during the optimization process along with the other semantic landmarks and the robot poses. Fig. 6 represents the 3D and 2D plots obtained during this experiment. During this experiment, our proposed approach has a ATE of 0.280 m and the VIO algorithm used has an ATE of 0.651 m. ORB-SLAM2 losses feature tracking during this experiment due to high speed motions as well as insufficient features in the environment.

VI. DISCUSSIONS

A. STANDARD DATASETS

Sect. V-A1 presents the evaluations for the RGBD TUM dataset and Table. 2 compares the results obtained using our proposed algorithm with the VO algorithm as well as other state of the art techniques. It can be observed from Freiburg3 Long Office Household (Fig. 4a and Fig. 4b) and Freiburg2 desk (Fig. 4g and Fig. 4h) experiments that, the front end VO algorithm using the low level features, accumulates errors as the camera navigates around the environment and our method combing the VO and a sparse map of the semantic landmarks is able ot correct this accumulated error (see Table. 2). In the Freiburg2 XYZ (Fig. 4c and Fig. 4d) and Freiburg2 RPY (Fig. 4e and Fig. 4f) experiments, even though our method provides better results than the VO algorithm, the pose estimate is improved only by a short margin. This is due to the fact that, in these experiments the camera performs very short trajectories hence the VO has less drift.

As it can be seen from Fig. 5, even with errors in the predicted bounding boxes around the detected semantic objects, our algorithm is able to estimate the camera pose (see Fig. 4), accurately extracting, mapping and associating the planar surfaces of the detected objects. The effect of the noise of the detections is minimized in the data associations due to the fact that the extracted planar surfaces are only considered valid if their area is beyond a certain threshold, which is this case was set to 15 cm^2. Hence the extracted planar surfaces from the incomplete bounding boxes around the semantic objects, or due to incomplete semantic objects present in the image during certain time instances, are rejected as their computed area is smaller than the threshold.

It can be observed from the Table. 2, that in these sequences ORB-SLAM2 outperforms in most of them, this is partially due to the fact that ORB-SLAM2 compensates offline the scale bias in the depth maps for some of the sequences. As our framework is intended to work online it uses RTAB-map odometry as the VO and although in the presence of errors in the VO estimates as well as the detections, achieves better results as compared to the object based SLAM techniques i.e *MaskFusion* and *Fusion++* and comparable results to the state of the art geometric SLAM techniques based on low level features of the environment.

B. FIELD EXPERIMENTS

Sect. V-B presents the system setup and the results obtained from the experiments performed using the our own field tests. We discuss the obtained results during all these performed experiments below:

1) LONG HALL EXPERIMENT

Fig. 6 compares the results of our algorithm obtained during this experiment with the VIO algorithm. During this experiment the aerial robotic setup repeated the same 5 rounds around the long hallway. As seen from the 3D plot Fig. 6a and the 2D plot Fig. 6b the VIO estimates accumulate an error of over 3 m in the z-axis and x-axis, along with errors in the orientation. Whereas our proposed approach which uses these noisy estimates from the VIO, along with the planar surfaces of the recycle bins is able to correctly estimate the pose of the robot without any significant drift. It can also be appreciated from this experiment that even with presence of huge errors in the pose estimates of the VIO, our algorithm requires very few semantic features to accurately estimate the pose of the robot. In order to check the accuracy of the pose estimate of the robot, Fig. 7a presents the 3D map of the environment constructed using the pose estimated from our algorithm. Visually, it can be observed that using the pose estimated by our approach, the long hall is accurately constructed. ORB-SLAM2 was also tested on this experiment, but since ORB-SLAM2 uses the front RealSense camera, it looses tracking when moving close to the walls, as the walls are featureless, whereas since our method can utilize odometry information generated from different sources, it is not limited to only the odometry generated from the front camera and can work successfully in such scenarios.

2) LONG CORRIDOR EXPERIMENT

Fig. 6 presents the execution of the proposed long corridor experiment. This proposed experiment is performed in very challenging indoor scenario, which consists of changing illumination conditions affecting to a great extend the pose estimate of the VIO algorithm. This changing illumination can be observed from Fig. 7b, where the detection images numbered 3 and 4, present very low illumination as compared to the detection images numbered 1 and 2. This low illumination conditions, effect the accuracy of the pose estimated by the VIO algorithm, accumulating large errors in its x and y positions and a huge error of around 6 m in its z direction (see VIO pose estimate in Fig. 6c).

Due to the clutter of semantic objects (chairs), the yolo object detector, also estimates several inaccurate bounding boxes around the detected objects (see Fig. 7b). In our approach this noise in the detections, does not adversely effect the relative pose estimation of the semantic objects, due to several safety checks performed before adding it as a semantic landmark, including the planar surface thresholding, which in this experiment is empirically set to 10 cm^2. Using these high noise VIO pose estimates and the semantic detections in the corridor, as seen from Fig. 6 our approach clearly is able to correct the drift present in the VIO estimations, performing accurate loop closures, even in such challenging indoor environment. We also evaluate the performance of ORB-SLAM2 during this experiment. As seen from the detection images (Fig. 7b), the environment consists of several featureless white colored walls, because of which the ORB-SLAM2 which only uses low-level feature information looses its tracking when passing close to the surfaces. Whereas, since our proposed framework can easily integrate both VO or VIO algorithms, we choose the estimates from the ROVIO algorithm, which accumulates large drift but does not loose feature tracking due to the additional inertial information.

3) REPETITIVE TRAJECTORY WITH SEVERAL SEMANTIC OBJECTS

Fig. 6 presents the experiment performed using several semantic objects with different shapes and sizes such as chairs, tv-monitors, keyboards and books. Even with semantic objects like chairs, which have different complex 3D structure, the algorithm is able to accurately estimate the 3D position of such semantic objects and map them, hence is able to estimate a drift free trajectory of the robot, when comparing it with the VIO pose estimates and the ground truth pose. As seen in Fig. 7c the yolo detector estimates the bounding boxes around the object with errors. Many times the bounding boxes do not completely fit the object or overfit the object. Due to these detection errors, the estimated 3D pose of the semantic object using a median of the 3D points inside the bounding boxes can deviate from the true position. Since we extract the planar surface information from these semantic objects and allow only for planar surfaces above a certain area which in this experiment is set to 15 cm^2, we can accurately map and localize using the semantic landmarks, irrespective of the errors in the detections.

We also compare the results obtained from our experiments with the with ORB-SLAM2 based on only geometric features. Since in this experiment, the camera is tilted towards the ground around 25° and since the floor contains several repetitive patterns, the ORB-SLAM2 degrades in tracking performance having a higher ATE of 0.267 m as compared to our approach which has an ATE of 0.226 m.

4) RANDOM TRAJECTORY WITH SEVERAL SEMANTIC OBJECTS

Fig. 6 demonstrates the experiment performed in a random trajectory fashion with a clutter of semantic objects. Due to insufficient lighting conditions and several rotational as well as translation motions of the camera, during this experiment the trajectory estimated by the ROVIO algorithm degrades in its performance and accumulates huge drifts in its estimations (see Fig 6g and Fig. 6h). It can also be observed from Fig. 7d during this experiment the detections received from the yolo detector have significant errors, for example due to the clutter of the semantic objects several erroneous bounding boxes are estimated around a chair. Planar extraction is not effected to a great extent even with these detection errors as the planar surfaces are extracted only if they satisfy the 3D points threshold and the planar area threshold which is this case is 0.15 cm^2. Thus, even with the huge drift present in the VIO estimations as well as errors in the detections, our approach is able to correct the estimated trajectory of the camera to the ground truth trajectory, with the ATE estimated by our algorithm being 0.393 m, whereas as the ATE of ROVIO accumulating large drift in its estimates, being 0.550 m.

The ATE of the trajectory estimated by the ORB-SLAM2 during this experiment is 0.317 m. The ORB-SLAM2 also degrades in its performance during this experiment, but does not diverge as much as the ROVIO. During this experiment the odometry estimated by the ROVIO diverges as much as 1.5 m from the ground truth trajectory, which is corrected to a great extent by our algorithm using the semantic landmarks (see Fig. 6h). But as the front end of our algorithm does depend on the estimated VIO trajectory, these huge errors in the VIO estimates results in the performance of our algorithm in this experiment, being a bit inferior when comparing it with the ORB-SLAM2.

5) ON-BOARD AERIAL ROBOT

Fig. 6 represents the plots for the experiment performed on board the aerial robot. The robot performs several loops flying at a maximum speeds of 2m/s. Due to high motions of the aerial robot, the VIO accumulates large drift in its position as well as orientation. Whereas our approach, as can be seen from 3D and 2D plots (Fig. 6i and Fig. 6j respectively), even with such high speeds of the aerial robot and using only a total of 6 recycle bins does not diverge from the ground truth pose of the robot. This experiment proves that our proposed approach is able to correct the position as well as orientation error present in the VIO algorithm using few semantic features and at high speed motions of the aerial robot.

The indoor environment during this experiment consists of a monochromatic white colored surface with few geometric features and the aerial robot also performs high speed angular and translational motions. Due to these reasons, the ORB-SLAM2 looses the feature tracking and is unable to estimate the pose of the robot. Since our framework loosely couples the VIO estimates, it can integrate several VIO algorithms in a loosely coupled fashion, using the VIO algorithms best suited for the particular application. The Snap VIO being optimized to run on board an aerial robot is hence used, which degrades in its performance along with time but does not loose featuring tracking.

VII. CONCLUSION

In this paper we present a fast, robust and lightweight visual semantic slam algorithm using commonly available planar surfaces as high-level semantic information. It is capable of running on board the aerial robot in order to estimate its drift free trajectory. We test the algorithm with a standard dataset available in the literature, showing that the algorithm is able to provide better results compared to the state of the art VO/VIO algorithms and object based SLAM techniques as well as comparable results to the geometric SLAM techniques. We also perform several experiments in different challenging indoor scenarios, with hand-held camera setup as well as with an on board aerial robotic setup, demonstrating the capability of the algorithm to work in these challenging indoor environments as well as on board aerial robots flying at velocities of 2 m/s. As the framework loosely couples VO/VIO estimates and the semantic mapping approach, we are able to integrate and test our algorithm with several state of the art VO and VIO approaches, selecting the best performing algorithm in a given scenario. Video[5] demonstrates the working of the algorithm in the proposed field tests and the source code[6] of the algorithm is publicly available in order for the scientific community to take advantage of the presented work and improve it for adding further enhancements.

[5] https://vimeo.com/368217703

[6] https://bitbucket.org/hridaybavle/semantic_slam.git

ACKNOWLEDGMENT

The authors would like to thank Dr. K. Khosoussi for his useful ideas during the execution of this work and J. Tordesillas for his help in the experiments.

REFERENCES

[1] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," 2016, *arXiv:1610.06475*. [Online]. Available: http://arxiv.org/abs/1610.06475

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Computer Vision—ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 834–849.

[3] A. J. B. Trevor, J. G. Rogers, and H. I. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3041–3048.

[4] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, and S. Hodges, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.

[5] T. Whelan, S. Leutenegger, R. S. Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *Proc. Robot., Sci. Syst.*, Rome, Italy, Jul. 2015, doi: 10.15607/RSS.2015.XI.001.

[6] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, "Real-time 3D reconstruction in dynamic scenes using point-based fusion," in *Proc. Int. Conf. 3D Vis.*, Jun. 2013, pp. 1–8.

[7] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, "Semantic localization via the matrix permanent," *Robot., Sci. Syst.*, vol. 2, nos. 1–3, Jan. 2016.

[8] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1722–1729.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[10] L. Bernreiter, A. R. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. C. Lerma, "Multiple hypothesis semantic mapping for robust data association," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3255–3262, Oct. 2019.

[11] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537.

[12] S. Yang and S. Scherer, "Monocular object and plane SLAM in structured environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3145–3152, Oct. 2019.

[13] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1352–1359.

[14] V. Murali, H.-P. Chiu, S. Samarasekera, and R. T. Kumar, "Utilizing semantic visual landmarks for precise vehicle navigation," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[15] N. Sunderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 5079–5085.

[16] P. Parkhiya, R. Khawad, J. Krishna Murthy, B. Bhowmick, and K. M. Krishna, "Constructing category-specific models for monocular object-SLAM," 2018, *arXiv:1802.09292*. [Online]. Available: http://arxiv.org/abs/1802.09292

[17] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric instance-aware semantic mapping and 3D object discovery," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 3037–3044, Jul. 2019.

[18] J. McCormac, R. Clark, M. Bloesch, A. J. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level SLAM," *CoRR*, vol. abs/1808.08378, Sep. 2018. [Online]. Available: http://arxiv.org/abs/1808.08378

[19] N. Atanasov, S. L. Bowman, K. Daniilidis, and G. J. Pappas, "A unifying view of geometry, semantics, and data association in SLAM," in *Proc. IJCAI*, 2018, pp. 5204–5208.

[20] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *CoRR*, vol. abs/1612.08242, Aug. 2018.

[21] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," *CoRR*, vol. abs/1405.0312, Aug. 2018. [Online]. Available: http://arxiv.org/abs/1405.0312

[22] C. Sampedro, H. Bavle, A. Rodriguez-Ramos, A. Carrio, R. A. S. Fernandez, J. L. Sanchez-Lopez, and P. Campoy, "A fully-autonomous aerial robotic solution for the 2016 international micro air vehicle competition," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017.

[23] H. Bavle, S. Manthe, P. de la Puente, A. Rodriguez-Ramos, C. Sampedro, and P. Campoy, "Stereo visual odometry and semantics based localization of aerial robots in indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1018–1023.

[24] H. Bavle, J. Sanchez-Lopez, P. Puente, A. Rodriguez-Ramos, C. Sampedro, and P. Campoy, "Fast and robust flight altitude estimation of multirotor UAVs in dynamic unstructured environments using 3D point cloud sensors," *Aerospace*, vol. 5, no. 3, p. 94, 2018.

[25] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab, "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2684–2689.

[26] A. J. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proc. Semantic Perception Mapping Explor.*, 2013. [Online]. Available: https://www2.gmu.edu/

[27] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, Sep. 2017.

[28] M. Labbé and F. Michaud, "RTAB-map as an open-source LiDAR and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, Mar. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21831

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[30] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[31] A. Bradski, *Learning OpenCV—Computer Vision With the OpenCV Library: Software That Sees*, G. Bradski and A. Kaehler, Eds., 1st ed. Newton, MA, USA: O'Reilly Media, 2008.

[32] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.

[33] M. Runz, M. Buffier, and L. Agapito, "MaskFusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2018, pp. 10–20.

[34] T. Whelan, M. Kaess, M. F. Fallon, H. Johannsson, J. J. Leonard, and J. J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *Proc. AAAI*, 2012.

[35] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 2100–2106.

HRIDAY BAVLE received the Ph.D. degree *(cum laude)* in automatic control and robotics from the Universidad Politécnica de Madrid. He is currently a Postdoctoral Researcher with the Centre for Automation and Robotics (CAR), Computer Vision and Aerial Robotics (CVAR) Group. Aerial Robotics is his core research field with his main focus being localization and mapping for aerial robots using on board perception. He has executed several challenging industrial research projects using autonomous aerial robots, such as Inspection of Thermal Power Plants and Inspection of Airplanes. He has also participated and received several awards in international aerial robotics competitions, such as the International Micro-Aerial Vehicles (IMAV) Competition and the Mohamed Bin Zayed International Robotics Challenge (MBZIRC).

PALOMA DE LA PUENTE (Member, IEEE) received the engineering degree in automatic control and electronics, and the M.Sc. and Ph.D. degrees in robotics and automation from the Universidad Politécnica de Madrid (UPM), in 2007, 2008, and 2012, respectively. She was a Predoctoral Visitor with Caltech for more than two and three months. She was a Postdoctoral Researcher with DISAM-UPM for six months and with the ACIN Institute of Automation and Control-Vienna University of Technology (TUW), for two years. She also had professional experience with Ixion Industry and Aerospace. She is currently an Assistant Professor with UPM. She has participated in several national and European projects and also in international robotics competitions. Her current research interests include mobile robots navigation, mapping, SLAM, spatial cognition, sensor data processing, human–robot interaction for service robotics, and systems engineering.

JONATHAN P. HOW (Fellow, IEEE) is currently the Richard C. Maclaurin Professor of aeronautics and astronautics, was honored for contributions to guidance and control of air and space vehicles. He is also a full time Professor with the Aerospace Controls Lab, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology. He serves as the Head of the Information sector within the Department of Aeronautics and Astronautics, is the Director of the Ford-MIT Alliance, and was a member of the USAF Scientific Advisory Board (SAB), from 2014 to 2017. His researches focus on planning and learning under uncertainty, and he was the control lead for the MIT DARPA Urban Challenge team. Other research interests include the design and implementation of distributed robust planning algorithms to coordinate multiple autonomous vehicles in dynamic uncertain environments, robust and adaptive control to enable autonomous agile flight and aerobatics, and reinforcement learning for real-time mechanical and aerospace applications. Prof. How is also an AIAA Fellow. His work has been recognized with multiple awards, including the Institute of Navigation Burka Award, the IFAC Automatica award for best applications paper, the AeroLion Technologies Outstanding Paper Award for the Journal Unmanned Systems, the IEEE Control Systems Society Video Clip Contest, and numerous AIAA Best Paper in Conference Awards. He was awarded the Air Force Commander Public Service Award, in 2017, for his contributions to the SAB. He is the Editor-in-Chief of the IEEE CONTROL SYSTEMS MAGAZINE, an Associate Editor of the AIAA *Journal of Aerospace Information Systems*, and an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS.

PASCUAL CAMPOY (Member, IEEE) is currently a Full Professor in automation and robotics with the Universidad Politécnica de Madrid (UPM), Madrid, Spain, and a Visiting Professor with TUDelft, The Netherlands. He has also been Visiting Professor with Tongji University, Shanghai, China, and QUT, Australia. He is also a Lecturer in control, machine learning, and computer vision. He is leading the Centre for Automation and Robotics (CAR), Computer Vision and Aerial Robotics (CVAR) Research Group. He has also been the Head Director of over 40 research and development projects, including research and development of European projects, national research and development projects, and over 25 technological transfer projects directly contracted with the industry. He is the author of over 200 international scientific publications. He holds nine patents, three of them registered internationally. He received several international prizes in UAV competitions, such as IMAV 2012, IMAV 2013, IARC 2014, IMAV 2016, and IMAV 2017.