2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Real-Time Monocular Object-Model Aware Sparse SLAM

Mehdi Hosseinzadeh, Kejie Li, Yasir Latif, and Ian Reid

Abstract— **Simultaneous Localization And Mapping (SLAM) is a fundamental problem in mobile robotics. While sparse point-based SLAM methods provide accurate camera localization, the generated maps lack semantic information. On the other hand, state of the art object detection methods provide rich information about entities present in the scene from a single image. This work incorporates a real-time deep-learned object detector to the monocular SLAM framework for representing generic objects as quadrics that permit detections to be seamlessly integrated while allowing the real-time performance. Finer reconstruction of an object, learned by a CNN network, is also incorporated and provides a shape prior for the quadric leading further refinement. To capture the dominant structure of the scene, additional planar landmarks are detected by a CNN-based plane detector and modelled as independent landmarks in the map. Extensive experiments support our proposed inclusion of semantic objects and planar structures directly in the bundle-adjustment of SLAM - *Semantic SLAM* - that enriches the reconstructed map semantically, while significantly improving the camera localization.**

I. INTRODUCTION

Simultaneous Localization And Mapping (SLAM) is one of the fundamental problems in mobile robotics [1] that aims to reconstruct a previously unseen environment while localizing a mobile robot with respect to it. The representation of the map is an important design choice as it directly affects its usability and precision. A sparse and efficient representation for Visual SLAM is to consider the map as collection of points in 3D, which carries information about geometry but not about the semantics of the scene. Denser representations [2], [3], [4], [5], [6], remain equivalent to a collection of points in this regard.

Man-made environments contain many objects that can be used as landmarks in a SLAM map, encapsulating a higher level of abstraction than a set of points. Previous object-based SLAM efforts have mostly relied on a database of predefined objects – which must be recognized and a precise 3D model fit to match the observation in the image to establish correspondence [7]. Other work [8] has admitted more general objects (and constraints) but only in a slow, offline structure-from-motion context. In contrast, we are concerned with online (real-time) SLAM, but we seek to represent a wide variety of objects. Like [8] we are not concerned with high-fidelity reconstruction of individual objects, but rather to represent the location, orientation and rough shape of objects, while incorporating fine point-cloud reconstructions on-demand. A suitable representation is therefore a quadric [9], which captures a compact representation of rough extent and pose while allows elegant data-association. In addition to objects, much of the large-scale structure of a general scene (especially indoors) comprises dominant planar surfaces. Planes provide information complimentary to points by representing significant portions of the environment with few parameters, leading to a representation that can be constructed and updated online [10]. In addition to constraining points that lie on them, planes permit the introduction of useful affordance constraints between objects and their supporting surfaces that leads to better estimate of the camera pose.

This work aims to construct a sparse semantic map representation consisting not only of points, but planes and objects as landmarks, all of which are used to localize the camera. We explicitly target real-time performance in a monocular setting which would be impossible with uncritical choices of representation and constraints. To that end, we use the representation for dual quadrics proposed in our previous work [11] to represent and update general objects, however [11] has fundamental limitations from two aspects: **(1)** from front-end perspective such as: **a)** reliance on the *depth* channel for plane segmentation and parameter regression, **b)** precomputation of Faster R-CNN [12] based object detections to permit real-time performance, and **c)** ad-hoc object and plane matching/tracking. **(2)** From the back-end perspective: **a)** conic observations are assumed to be axis-aligned thus limiting the robustness of the quadric reconstruction, **b)** all detected landmarks are maintained in a single global reference frame. This work in addition to addressing the mentioned limitations, proposes new factors amenable for real-time inclusion of plane and object detections while incorporating fine point-cloud reconstructions from a deep-learned CNN, wherever available, to the map and refine the quadric reconstruction according to this object model.

The main contributions of the paper as follows: (1) integration of two different CNN-based modules to segment planes and regress the parameters (2) integrating a real-time deep-learned object detector in a monocular SLAM framework to detect general objects as landmarks along a data-association strategy to track them, (3) proposing a new observation factor for objects to avoid axis-aligned conics, (4) representing landmarks relative to the camera where they are first observed instead of a global reference frame, and (5) wherever available, integrating the reconstructed point-cloud model of the detected object from single image by a CNN to the map and imposing additional prior on the extent of the reconstructed quadric based on the reconstructed point-cloud.

All of the authors are with the Australian Center for Robotic Vision (ACRV) at the School of Computer Science, University of Adelaide {firstname.lastname}@adelaide.edu.au.

II. RELATED WORK

SLAM is well studied problem in mobile robotics and many different solutions have been proposed for solving it. The most recent of these is the graph-based approach that formulates SLAM as a nonlinear least squares problem [13]. SLAM with cameras has also seen advancement in theory and good implementations that have led to many real-time systems from sparse ([14],[2]) to semi-dense ([3], [15]) to fully dense ([4], [6], [5]).

Recently, there has been a lot of interest in extending the capability of a point-based representation by either applying the same techniques to other geometric primitives or fusing points with lines or planes to get better accuracy. In that regard, [10] proposed a representation for modelling infinite planes and [16] use Convolutional Neural Network (CNN) to generate plane hypothesis from monocular images which are refined over time using both image planes and points. [17] proposed a method to fuse points and planes from an RGB-D sensor. In the latter works, they try to fuse the information of planar entities to increase the accuracy of depth inference.

Quadrics based representation was first proposed in [18] and later used in a structure from motion setup [9]. [19] reconstructs quadrics based on bounding box detections, however it is not explicitly modelled to remain bounded ellipsoids. Addressing previous drawback, [20] still relies on ground-truth data-association in a non-real-time quadric-only framework. [21] presented a semantic mapping system using object detection coupled with RGB-D SLAM, however object models do not inform localization. [7] presented an object based SLAM system that uses pre-scanned object models as landmarks for SLAM but can not be generalized to unseen objects. [22] presented a system that fused multiple semantic predictions with a dense map reconstruction. SLAM is used as the backbone to establish multiple view correspondences for fusion of semantic labels but the semantic labels do not inform localization.

III. OVERVIEW OF THE LANDMARK REPRESENTATIONS AND FACTORS

For the sake of completeness, this section presents an overview of the representations and factors proposed originally in our previous work [11]. In the next sections, we propose new multi-edge observation and unary prior factors. The SLAM problem can be represented as a bipartite factor graph $\mathcal{G}(\mathcal{V}, \mathcal{F}, \mathcal{E})$ where $\mathcal{V}$ represents the set of *vertices* (variables) that need to be estimated and $\mathcal{F}$ represents the set of *factors* (constraints) that are connected to their associated variables by the set of edges $\mathcal{E}$. We propose our SLAM system in the context of factor graphs. The solution of this problem is the optimum configuration of vertices (MAP estimate), $\mathcal{V}^*$, that minimizes the overall error over the factors in the graph (log-likelihood of the joint probability distribution). The pipeline of our SLAM system is illustrated in Fig 1.

A. Quadric Representation

A quadric surface in 3D space can be represented by a homogeneous quadratic form defined on the 3D projective space $\mathbb{P}^3$ that satisfies $\mathbf{x}^\top \mathbf{Q} \mathbf{x} = \mathbf{0}$, where $\mathbf{x} \in \mathbb{R}^4$ is the homogeneous 3D point and $\mathbf{Q} \in \mathbb{R}^{4\times4}$ is the symmetric matrix representing the quadric surface. However, the relationship between a point-quadric $\mathbf{Q}$ and its projection into an image plane (a conic) is not straightforward [23]. A widely accepted alternative is to make use of the dual space ([18], [9], [19]) which represents a dual quadric $\mathbf{Q}^*$ by the envelope of planes $\boldsymbol{\pi}$ tangent to it, viz: $\boldsymbol{\pi}^\top \mathbf{Q}^* \boldsymbol{\pi} = 0$, which simplifies the relationship between the quadric and its projection to a conic. A dual quadric $\mathbf{Q}^*$ can be decomposed as $\mathbf{Q}^* = \mathbf{T}_Q \mathbf{Q}_c^* \mathbf{T}_Q^\top$ where $\mathbf{T}_Q \in \mathbf{SE}(3)$ transforms an axis-aligned (canonical) quadric at the origin, $\mathbf{Q}_c^*$, to a desired $\mathbf{SE}(3)$ pose. Quadric landmarks need to remain bounded, i.e. ellipsoids, which requires $\mathbf{Q}_c^*$ to have 3 positive and 1 negative eigenvalues. In [11] we proposed a decomposition and incremental update rule for dual quadrics that guarantees this conditions and provides a good approximation for incremental update. More specifically, the dual ellipsoid $\mathbf{Q}^*$ is represented as a tuple $(\mathbf{T}, \mathbf{L})$ where $\mathbf{T} \in \mathbf{SE}(3)$ and $\mathbf{L}$ lives in $\mathbf{D}(3)$ the space of real diagonal 3×3 matrices, i.e. an axis-aligned ellipsoid accompanied by a rigid transformation. The proposed approximate update rule for $\mathbf{Q}^* = (\mathbf{T}, \mathbf{L})$ is:

$$\mathbf{Q}^* \oplus \Delta\mathbf{Q}^* = (\mathbf{T}, \mathbf{L}) \oplus (\Delta\mathbf{T}, \Delta\mathbf{L}) = (\mathbf{T} \cdot \Delta\mathbf{T}, \mathbf{L} + \Delta\mathbf{L}) \quad (1)$$

where $\oplus : \mathbb{E} \times \mathbb{E} \longmapsto \mathbb{E}$ is the mapping for updating ellipsoids, $\Delta\mathbf{L}$ is the update for $\mathbf{L}$ and $\Delta\mathbf{T}$ is the update for $\mathbf{T}$ that are carried out in the corresponding lie-algebra of $\mathfrak{d}(3)$ (isomorphic to $\mathbb{R}^3$) and $\mathfrak{se}(3)$, respectively.

B. Plane Representation

Following [10], a plane $\boldsymbol{\pi}$ as a structural entity in the map is represented minimally by its normalized homogeneous coordinates $\boldsymbol{\pi} = (a, b, c, d)^\top$ where $\mathbf{n} = (a, b, c)^\top$ is the normal vector and d is the signed distance to origin.

C. Constraints between Landmarks

In addition to the classic point-camera constraint formed by the observation of a 3D point as 2D feature point in the camera, we model constraints between higher level landmarks and their observations in the camera. These constraints also carry semantic information about the structure of the scene, such as Manhattan assumption and affordances. We present a brief overview of these constraints here. In the next sections we present the newly introduced factors regarding plane and object observations and object shape priors, induced by the single-view point-cloud reconstructions.

1) Point-Plane Constraint: For a point $\mathbf{x}$ to lie on its associated plane $\boldsymbol{\pi}$ with the unit normal vector $\mathbf{n}$, we introduce the following factor between them:

$$f_d(\mathbf{x}, \boldsymbol{\pi}) = \| \mathbf{n}^\top (\mathbf{x} - \mathbf{x}_o) \|^2_{\sigma_d} \quad (2)$$

which measures the orthogonal distance of the point and the plane, for an arbitrary point $\mathbf{x}_o$ on the plane. $\|\mathbf{e}\|_{\boldsymbol{\Sigma}}$ notation is the Mahalanobis norm of $\mathbf{e}$ and is defined as $\mathbf{e}^\top \boldsymbol{\Sigma}^{-1} \mathbf{e}$ where $\boldsymbol{\Sigma}$ is the associated covariance matrix.

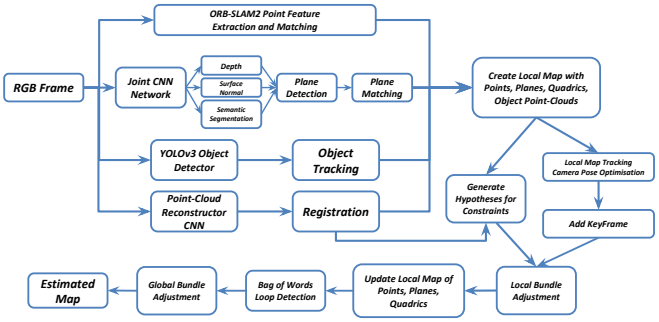


Fig. 1: The pipeline of our proposed SLAM system.

2) Plane-Plane Constraint (Manhattan assumption): Manhattan world assumption where planes are mostly mutually parallel or perpendicular, is modelled as:

$$f_{\parallel}(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \parallel |\mathbf{n}_1^\top \mathbf{n}_2| - 1 \parallel^2_{\sigma_{par}} \quad \textit{for parallel planes} \quad (3)$$

$$f_{\perp}(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) = \parallel \mathbf{n}_1^\top \mathbf{n}_2 \parallel^2_{\sigma_{per}} \quad \textit{for perpendicular planes} \quad (4)$$

where $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$ have unit normal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$.

3) Supporting/Tangency Constraint: In normal situations planar structure of the scene affords stable support for common objects, for instance floors and tables support indoor objects and roads support outdoor objects like cars. To impose a supporting affordance relationship between planar entities of the scene and common objects, we introduce a factor between dual quadric object $\mathbf{Q}^*$ and plane $\boldsymbol{\pi}$ that models the tangency relationship as:

$$f_t(\boldsymbol{\pi}, \mathbf{Q}^*) = \parallel \boldsymbol{\pi}^\top \hat{\mathbf{Q}}^* \boldsymbol{\pi} \parallel^2_{\sigma_t} \quad (5)$$

where $\hat{\mathbf{Q}}^*$ is the normalized dual quadric by its matrix Frobenius norm. Please note that this tangency constraint is the direct consequence of choosing dual space for quadric representation, which is not straight-forward in point space.

IV. MONOCULAR PLANE DETECTION

Man-made environments contain planar structures, such as table, floor, wall, road, etc. If modelled correctly, they can provide information about large feature-deprived regions providing more map coverage. In addition, these landmarks act as a regularizers for other landmarks when constraints are introduced between them. The dominant approach for plane detection is to extract them from RGB-D input [11] which provides reliable detection and estimation of plane parameters. In a monocular setting, planes need to be detected using a single RGB image and their parameters estimated, which is an ill-posed problem. However, recent breakthroughs enable us to detect and estimate planes. Recently, PlaneNet [24] presented a deeply learned network to predict plane parameters and corresponding segmentation masks. While planar segmentation masks are highly reliable, the regressed parameters are not accurate enough for small planar regions in indoor scenes (see Section VI). To address this shortcoming, we use a network that predicts depth, surface normals, and semantic segmentations. Depth and surface normal contain complementary information about the orientation and distance of the planes, while semantic segmentation allows reasoning about identity of the region such as wall, floor, etc.

A. Planes from predicted depth, surface normals, and semantic segmentation

We utilize the state-of-the-art joint network [25] to estimate depth, normals, and segmentation for each RGB frame in real-time. We exploit the redundancy in the three separate predictions to boost the robustness of the plane detection by generating plane hypotheses in two ways: **1)** for each planar region in the semantic segmentation (regions such as floor, wall, etc.) we fit 3D planes using surface normals and depth for orientation and distance of the plane respectively, and **2)** depth and surface normals predictions are utilized in the connected component segmentation of the reconstructed point-cloud in a parallel thread ([26], [11]). Plane detection $\boldsymbol{\pi} = (a, b, c, d)^\top$ is considered to be valid if the cosine distance of normal vectors $\mathbf{n} = (a, b, c)^\top$ and also the distance between the d value of the two planes from two estimations are within a certain threshold. The corresponding plane segmentation is taken to be the intersection of the plane masks of the two hypotheses.

Note that the association between 3D point landmarks and planes, useful for the factor described in III-C, is extracted from the resulting mask. The 3D point is considered as an inlier if the corresponding 2D keypoint inside the mask also satisfies the certain geometric distance threshold.

B. Plane Data Association

Once initialized and added to the map, the landmark planes need to be associated with the detected planes in the incoming frames. Matching planes is more robust than feature point matching due to the inherent geometrical nature of planes [11]. To make data association more robust in cluttered scenes, when available, we additionally use the detected keypoints that lie inside the segmented plane in the image to match the observations. A plane in the map and a plane in the current frame are deemed to be a match if the number of common keypoints is higher than a threshold th_H and the unit normal vector and distance of them are within certain threshold. If the number of common keypoints is less than another threshold th_L (or zero for feature-deprived regions) meaning that there is no corresponding map plane for the detected plane, the observed plane is added to the map as a new landmark. The map can now contain two or more planar regions that might belong to the same infinite plane such as two tables with same height in the office. However, additional constraints on parallel planes are also introduced according to evidence (Section III-C).

C. Multi-Edge Factor for Plane Observation

After successful data association, we can introduce the observation factor between the plane and the camera (keyframe). We use a relative key-frame formulation (instead of the global frame) for each plane landmark $\boldsymbol{\pi}_r$ which is expressed relative to the first key-frame ($\mathbf{T}_r^w$) that observes it. For an observation $\boldsymbol{\pi}_{obs}$ from a camera pose $\mathbf{T}_c^w$, the

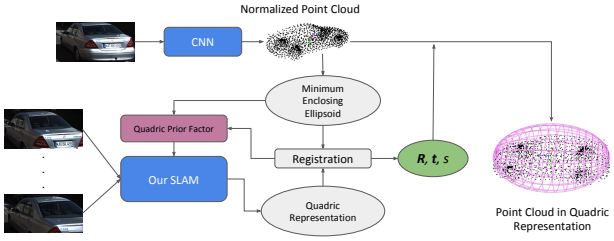


Fig. 2: Single-view point-cloud reconstruction imposes a shape prior constraint on a multi-view reconstructed quadric in our system (See Section V-B)

multi-edge factor (connected to more than two nodes) for measuring the plane observation is given by:

$$f_\pi(\boldsymbol{\pi}_r, \mathbf{T}_r^w, \mathbf{T}_c^w) = \| \, d(\mathbf{T}_c^{r\,-\top}\boldsymbol{\pi}_r, \boldsymbol{\pi}_{obs}) \, \|^2_{\boldsymbol{\Sigma}_\pi} \tag{6}$$

where $\mathbf{T}_c^{r\,-\top}\boldsymbol{\pi}_r$ is the transformed plane from its reference frame to the camera coordinate frame and d is the geodesic distance of the $\mathbf{SO}(3)$ [10] and $\mathbf{T}_c^w$ is the pose of the camera which takes a point in the current camera frame ($\mathbf{x}_c$) to a point in the world frame $\mathbf{x}_w = \mathbf{T}_c^w \mathbf{x}_c$.

V. Incorporating Object with Point-Cloud Reconstruction

As noted earlier, incorporating general objects in the map as quadrics leads to a compact representation of the rough 3D extent and pose (location and orientation) of the object while facilitating elegant data association. State-of-the-art object detector such as YOLOv3 [27] can provide object labels and bounding boxes in real-time for general objects. The goal of introducing objects in SLAM is both to increase the accuracy of the localization and to yield a richer semantic map of the scene. While our SLAM proposes a sparse and coarse realization of the objects, wherever the fine model reconstruction of each object is available it can be seamlessly incorporated on top of the corresponding quadric and even refines the quadric reconstruction as discussed in V-B.

A. Object Detection and Matching

For real-time detection of objects, we use YOLOv3 [27] trained on COCO dataset [28] that provides axis detections as aligned bounding boxes for common objects. For reliability we consider detections with 85% or more confidence.

Object Matching: To rely solely on the geometry of the reconstructed quadrics (by comparing re-projection errors) to track the object detections against the map is not robust enough particularly for high-number of overlapping or partially-occluded detections. Therefore to find optimum matches for all the detected objects in current frame, we solve the classic optimum assignment problem with Hungarin/Munkres [29] algorithm. The challenge of using this classic algorithm is how to define the appropriate cost matrix. We establish the cost matrix of this algorithm based on the idea of maximizing the number of common robustly matched keypoints (2D ORB features) inside the detected bounding boxes. Since we want to solve the minimization problem, the cost matrix is defined as:

$$\mathbf{C} = [c_{ij}]_{N \times M} \tag{7}$$

$$c_{ij} = K - p(b_i, q_j) \tag{8}$$

where $p(b_i, q_j)$ gives the number of projected keypoints associated with candidate quadric q_j inside the bounding box b_i, and $K = \max_{i,j} p(b_i, q_j)$ is the maximum number of all of these projected keypoints. N and M are the total number of bounding box detections in current frame and candidate quadrics of the map for matching, respectively. Candidate quadrics for matching are considered to be the quadrics of the map that are currently in front of the camera.

To reduce the number of mismatches furthermore, after solving the assignment problem with the proposed cost matrix, the solved assignment of b_i^* to q_j^* is considered successful if the number of common keypoints satisfies a certain high threshold $p(b_i^*, q_j^*) \geq th_{high}$ and the new quadric will be initialized in the map if $p(b_i^*, q_j^*) \leq th_{low}$. Assignments with $p(b_i^*, q_j^*)$ values between these thresholds will be ignored.

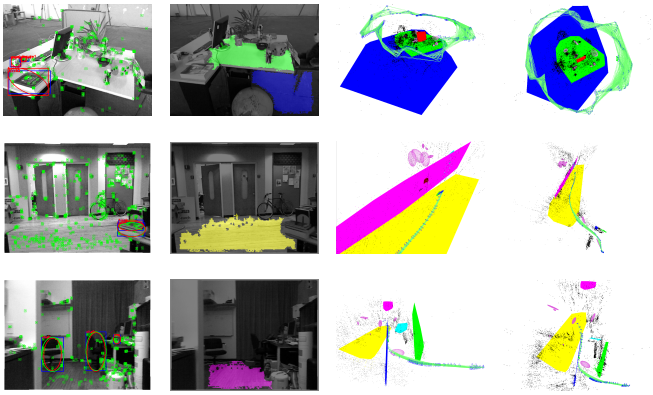
B. Point-Cloud Reconstruction and Shape Priors

In this section, we present a method of estimating fine geometric model of available objects established on top of quadrics to enrich their inherent coarse representation. It is difficult to estimate the full 3D shape of objects from sparse views using purely classic geometric methods. To bypass this limitation, we train a CNN adapted from Point Set Generation Net [30] to predict (or hallucinate) the accurate 3D shape of objects as point clouds from single RGB images.

The CNN is trained on a CAD model repository ShapeNet [31]. We render 2D images of CAD models from random viewpoints and, to simulate the background in real images, we overlay random scene backgrounds from the SUN dataset [32] on the rendered images. We demonstrate the efficacy of this approach for outdoor scenes, particularly for general car objects in KITTI [33] benchmark in section VI-B. Running alongside with the SLAM system, the CNN takes an amodal detected bounding box of an object as input and generates a point cloud to represent the 3D shape of the object. However, to ease the training of the CNN, the reconstructed point cloud is in a normalized scale and canonical pose. To incorporate the point cloud into the SLAM system, we need to estimate seven parameters to scale, rotate and translate this point cloud. First we compute the minimum enclosing ellipsoid of the normalized point cloud, and then estimate the parameters by aligning it to the object ellipsoid from SLAM.

Shape Prior on Quadrics: After registering the reconstructed point-cloud and the quadric from SLAM, we impose a further constraint only on the shape (extent) of the quadric, Fig 2, feasible due to the decomposition of quadric representation. This prior affects the ratio of major axes of the quadric $\mathbf{Q}^*$ by computing the intersection over union of the registered enclosing normalized cuboid of the point-cloud $\mathcal{M}$ and enclosing normalized cuboid of the quadric:

$$f_{prior}(\mathbf{Q}^*) = \|1 - IoU_{cu}(cuboid(\mathbf{Q}^*), cuboid(\mathcal{M}))\|^2_{\sigma_p} \tag{9}$$



(a) ORB Features and Detected Objects (b) Segmented Planes (c) Generated Map (Side) (d) Generated Map (Top)

Fig. 3: Qualitative results for different TUM and NYUv2 datasets. The sequences vary from rich planar structures to multi-object cluttered office scenes

where *cuboid* is a function that gives the normalized enclosing cuboid of an ellipsoid.

As an expedient approach, we currently pick a single high-quality detected bounding box as the input to the CNN, however, it is not complicated to extend to multiple bounding boxes by using a Recurrent Neural Net to fuse information from different bounding boxes, as done in 3D-R2N2 [34].

C. Multi-Edge Factor for Non-Aligned Object Observation

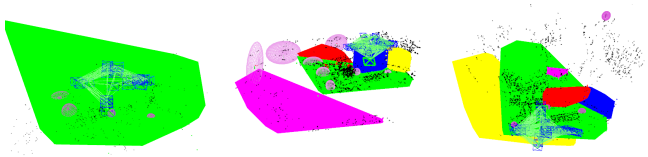
We propose an observation factor for the quadric without enforcing that to be observed as an axis-aligned inscribed conic (ellipse). Unlike [19] that uses the Mahalanobis distance of detected and projected bounding boxes, which is not robust and penalizes more for large errors and outliers, we use the error function based on Intersection-over-Union (IoU) of these bounding boxes that is also weighted according to the *confidence score* s of the object detector. This factor provides an inherent capped error, however it implicitly emphasizes on the significance of the good initialization of quadrics to have a successful optimization. Similar to plane landmarks, we use the relative reference key-frame $\mathbf{T}_r^w$ to represent the coordinates of the objects, we introduce the multi-edge factor, for object observation error, between dual quadric $\mathbf{Q}_r^*$ and camera pose $\mathbf{T_c^w}$ as:

$$f_Q(\mathbf{Q}_r^*, \mathbf{T}_r^w, \mathbf{T}_c^w) = \| \ 1 - IoU_{bb}(B^*, B_{obs}) \ \|_{s^{-1}}^2 \qquad (10)$$

where B_{obs} is the detected bounding box and B^* is the enclosing bounding box of the projected conic $\mathbf{C}^* \sim \mathbf{P}\mathbf{Q}_r^*\mathbf{P}^\top$ with the projection matrix $\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{I}_{3\times3} & \mathbf{0}_{3\times3}\end{bmatrix}\mathbf{T}_c^r$ of the camera with calibration matrix $\mathbf{K}$,[23], and $\mathbf{T}_c^r = \mathbf{T}_c^w(\mathbf{T}_r^w)^{-1}$ is the relative pose of the camera from the reference key-frame of the quadric.

VI. EXPERIMENTS

The proposed system is built in C++ on top of the state-of-the-art ORB-SLAM2 [14] and utilizes its front-end for tracking ORB features, while the back-end for the proposed system is implemented in C++ using g2o [35]. Evaluation is performed on a commodity machine with Intel Core i7-4790 in near 20 fps and carried out on publicly available TUM [36], NYUv2 [37], and KITTI [33] datasets that contain rich planar low-texture scenes to multi-object offices and outdoor scenes. Qualitative and quantitative evaluations are carried out using different mixture of landmarks and comparisons are presented against point-based monocular ORB-SLAM2 [14].



(a) PlaneNet detector (b) Proposed detector (c) Baseline detector

Fig. 4: Qualitative comparison of using different plane detectors in our monocular SLAM system for `fr1/xyz`.

A. TUM and NYUv2

Qualitative evaluation on TUM and NYUv2 for sequences `fr2/desk`, `nyu/office_1b`, and `nyu/nyu_office_1` is illustrated in Fig. 3 for different scenes and landmarks. Columns (a)-(d) show the image frame with tracked features and possible detected objects, detected and segmented planes, and the reconstructed map from two different viewpoints, respectively. For some low or no texture sequences in TUM and NYUv2 datasets point-based SLAM system fail to track the camera, however the present rich planar structure is exploited by our system along with the Manhattan constraints to yield more accurate trajectories and semantically meaningful maps.

The reconstructed maps are semantically rich and consistent with the ground truth 3D scene, for instance in `fr2/desk`, with presence of all landmarks and constraints, the map consists of planar monitor orthogonal to the desk, and quadrics corresponding to objects are tangent to the supporting desk, congruous with the real scene. Red ellipses in Fig. 3 column (a) are the projection of their corresponding quadric objects in the map. Further evaluations can be found in the supplemental video.

One of the main reasons for the improved accuracy of camera trajectory and consistency of the global map is the addressing of subtle but extremely important problem of scale drift. In a monocular setting, the estimated scale of the map can change gradually over time. In our system, the consistent metric scale of the planes (from CNN) and the presence of point-plane constraints allow observation of the absolute scale, which can further be improved by adding priors about the extent of the objects represented as quadrics.

One of the important factors that can affect the system performance is the quality of estimated plane parameters. Reconstructed maps are shown in Fig. 4 for two different monocular plane detectors incorporated in our system: **a)** PlaneNet [24], **b)** our proposed plane detector (See Section IV). Baseline comparison is made against a depth based

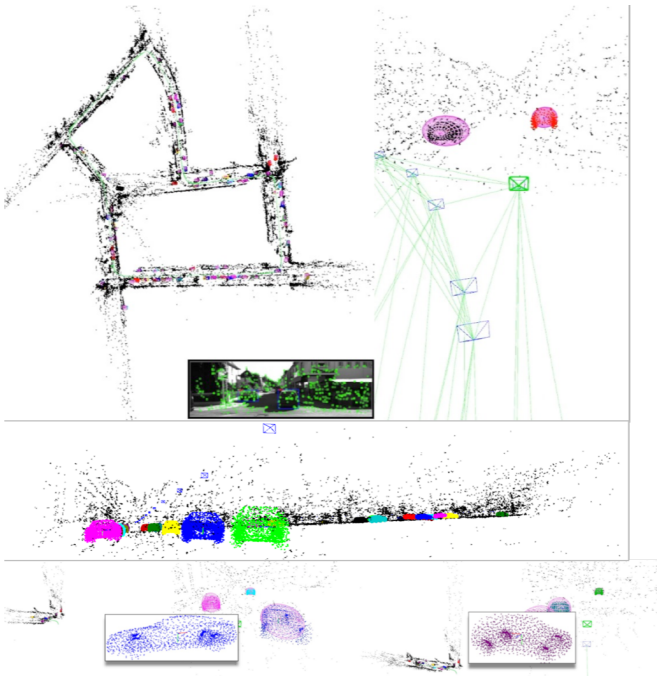


Fig. 5: Reconstructed map and camera trajectories for KITTI-7 with our SLAM. Proposed object observation and shape prior factors are effective in this reconstruction

plane detector that uses connected component segmentation of the point cloud ([26], [11]). The detected planes are then used in the monocular system for refinement. As seen in Fig. 4(a) PlaneNet only captures the planar table region successfully and fails for the other regions. The proposed detector captures the monitors on the table shown in column (b), however it misses the monitor behind and also reconstructs the two same height tables with a slight vertical distance. As shown in Fig. 4(c) the baseline plane detector captures the smaller planar regions more accurately and same height tables as one plane, as expected because of using additional *depth* information. Table II reports the comparison of these three approaches for plane detection in different sequences of TUM datasets. It can be seen that the depth based detector is the most informative, however the proposed method is better than PlaneNet in most cases.

We perform an ablation study to demonstrate the efficacy of introducing various combinations of the proposed landmarks and constraints. The RMSE of Absolute Trajectory Error (ATE) is reported in Table I. Estimated trajectories and ground-truth are aligned using a similarity transformation [38]. In the first case, points are augmented with planes (PP) and constraint for points and corresponding planes is

TABLE I: RMSE (cm) of ATE for our monocular SLAM against monocular ORB-SLAM2. Percentage of improvement over ORB-SLAM2 is represented in [ ]. See VI-A

Dataset	# KF	ORB-SLAM2	PP	PP+M	PO	PPO+MS
fr1/floor	125	1.7971	1.6923	**1.6704** [7.05%]	—	—
fr1/xyz	30	1.0929	1.0291	0.9802	1.0081	**0.9680** [11.43%]
fr1/desk	71	1.3940	1.2961	1.2181	1.2612	**1.2126** [13.01%]
fr2/xyz	28	0.2414	0.2213	0.2189	0.2243	**0.2179** [9.72%]
fr2/rpy	12	0.3728	0.3356	0.3354	0.3473	**0.3288** [11.79%]
fr2/desk	111	0.8019	0.7317	0.7021	0.7098	**0.6677** [16.74%]
fr3/long_office	193	1.0697	0.9605	0.9276	0.9234	**0.8721** [18.47%]

TABLE II: RMSE for ATE (cm) using different plane detection methods in our monocular SLAM. See VI-A

Dataset	PlaneNet [24]	Proposed Detector	Baseline
fr1/xyz	0.9701	**0.9680**	0.8601
fr1/desk	1.2191	**1.2126**	1.0397
fr2/xyz	0.2186	**0.2179**	0.2061
fr1/floor	**1.6562**	1.6704	1.4074

included. This already improves the accuracy over baseline and imposing additional Manhattan constraint in the second case (PP+M) improves ATE even further. In these two cases the error is significantly reduced by first exploiting the structure of the scene and second by reducing the scale-drift, as discussed earlier, using metric information about planes.

For the sequences containing common COCO [28] objects, the presence of objects represented by quadric landmarks along with points is explored in the third case (PO). This case demonstrates the effectiveness of integrating objects in the SLAM map. Finally, the performance of our full monocular system (PPO+MS) is detailed in the last right column of Table I with the presence of all landmarks points, planes, and objects and also Manhattan and supporting/tangency constraints. This case shows an improvement against the baseline in all of the evaluated sequences, in particular for fr3/long_office we have seen a significant decline in ATE (18.47%) as a result of the presence of a large loop in this sequence, where our proposed multiple-edges for observations of planes and quadric objects in key-frames have shown their effectiveness in the global loop closure.

B. KITTI benchmark

To demonstrate the efficacy of our proposed object detection factor, object tracking, and also shape prior factor induced from incorporated point-cloud (reconstructed by CNN from single-view) in our SLAM system, we evaluate our system on KITTI benchmark. For reliable frame-to-frame tracking, we use the stereo variant of ORB-SLAM2, however object detection and plane estimation are still carried out in a monocular fashion. The reconstructed map with quadric objects and incorporated point-clouds (See Section V-B) is illustrated for **KITTI-7** in Fig. 5. The instances of different cars are rendered in different colors.

VII. CONCLUSIONS

This work introduced a monocular SLAM system that can incorporate learned priors in terms of plane and object models in an online real-time capable system. We show that introducing these quantities in a SLAM framework allows for more accurate camera tracking and a richer map representation without huge computational cost. This work also makes a case for using deep-learning to improve the performance of traditional SLAM techniques by introducing higher level learned structural entities and priors in terms of planes and objects.

VIII. ACKNOWLEDGMENT

This work was supported by ARC Laureate Fellowship FL130100102 to IR and the ARC Centre of Excellence for Robotic Vision CE140100016.

REFERENCES

[1] 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 Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2320–2327.

[5] V. A. Prisacariu, O. Kähler, S. Golodetz, M. Sapienza, T. Cavallari, P. H. Torr, and D. W. Murray, "Infinitam v3: A framework for large-scale 3d reconstruction with loop closure," *arXiv preprint arXiv:1708.00783*, 2017.

[6] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136.

[7] 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 *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 2013, pp. 1352–1359. [Online]. Available: https://doi.org/10.1109/CVPR.2013.178

[8] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese, "Semantic structure from motion with points, regions, and objects," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2012.

[9] P. Gay, V. Bansal, C. Rubino, and A. D. Bue, "Probabilistic structure from motion with objects (psfmo)," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3094–3103.

[10] M. Kaess, "Simultaneous localization and mapping with infinite planes," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4605–4611.

[11] M. Hosseinzadeh, Y. Latif, T. Pham, N. Sünderhauf, and I. Reid, "Structure aware SLAM using quadrics and planes," *arXiv preprint arXiv:1804.09111*, 2018.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[13] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

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

[15] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 15–22.

[16] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 1222–1229.

[17] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane slam for hand-held 3d sensors," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5182–5189.

[18] G. Cross and A. Zisserman, "Quadric reconstruction from dual-space geometry," in *Computer Vision, 1998. Sixth International Conference on*. IEEE, 1998, pp. 25–31.

[21] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid, "Meaningful maps with object-oriented semantic mapping," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017, pp. 5079–5085.

[19] N. Sünderhauf and M. Milford, "Dual Quadrics from Object Detection BoundingBoxes as Landmark Representations in SLAM," *arXiv preprints arXiv:1708.00965*, Aug. 2017.

[20] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2019.

[22] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4628–4635.

[23] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[24] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. Furukawa, "PlaneNet: Piece-wise Planar Reconstruction from a Single RGB Image," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, "Real-time joint semantic segmentation and depth estimation using asymmetric annotations," *arXiv preprint arXiv:1809.04766*, 2018.

[26] A. Trevor, S. Gedikli, R. Rusu, and H. Christensen, "Efficient organized point cloud segmentation with connected components," in *3rd Workshop on Semantic Perception Mapping and Exploration (SPME), Karlsruhe, Germany*, 2013.

[27] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[29] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 12, pp. 83–97.

[30] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image." in *CVPR*, vol. 2, no. 4, 2017, p. 6.

[31] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[32] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3485–3492.

[33] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[34] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *European conference on computer vision*. Springer, 2016, pp. 628–644.

[35] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

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

[37] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *ECCV*, 2012.

[38] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Am. A*, vol. 4, no. 4, pp. 629–642, Apr 1987.