Monocular Object and Plane SLAM in Structured Environments

Shichao Yang, Sebastian Scherer

***Abstract*— We present a monocular Simultaneous Localization and Mapping (SLAM) using high level object and plane landmarks, in addition to points. The resulting map is denser, more compact and meaningful compared to point only SLAM. We first propose a high order graphical model to jointly infer the 3D object and layout planes from single image considering occlusions and semantic constraints. The extracted cuboid object and layout planes are further optimized in a unified SLAM framework. Objects and planes can provide more semantic constraints such as Manhattan and object supporting relationships compared to points. Experiments on various public and collected datasets including ICL NUIM and TUM mono show that our algorithm can improve camera localization accuracy compared to state-of-the-art SLAM and also generate dense maps in many structured environments.**

I. Introduction

Semantic understanding and SLAM are two fundamental problems in computer vision and robotics. In recent years, there has been great progress in each field. For example, with the popularity of Convolutional Neural Network (CNN), the performance of object detection [1], semantic segmentation [2], 3D understanding [3] has been improved greatly. In SLAM or Structure from Motion (SfM), approaches such as ORB SLAM [4] and DSO [5] are widely used in autonomous robots and Augmented Reality (AR) applications. However, the connections between visual understanding and SLAM are not well explored. Most existing SLAM approach represent the environments as a point cloud, either sparse or semi-dense, which may not satisfy many high level and intelligent tasks. For example in autonomous driving, vehicles need to be detected in 3D space to keep safety and in AR application, 3D objects and layout planes also need to be localized for more realistic physical interactions.

There are typically two categories of approaches to combine visual understanding and SLAM. The decoupled approach first builds the SLAM point cloud then further labels [6] [7] or detects 3D objects [8] and planes [9], while the coupled approach jointly optimizes the camera pose with the object and plane location. In this paper, we develop a coupled approach to demonstrate that high level object and plane landmarks can improve both camera pose estimation and dense mapping. Most existing object SLAM such as SLAM++ [10] [11] requires prior object models to detect and model the object, which limits the application in general environments. Some prior works also utilize architectural planes for dense 3D reconstruction but mostly rely on RGBD [12] sensors or LiDAR scanner [13].

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. {shichaoy, basti}@andrew.cmu.edu

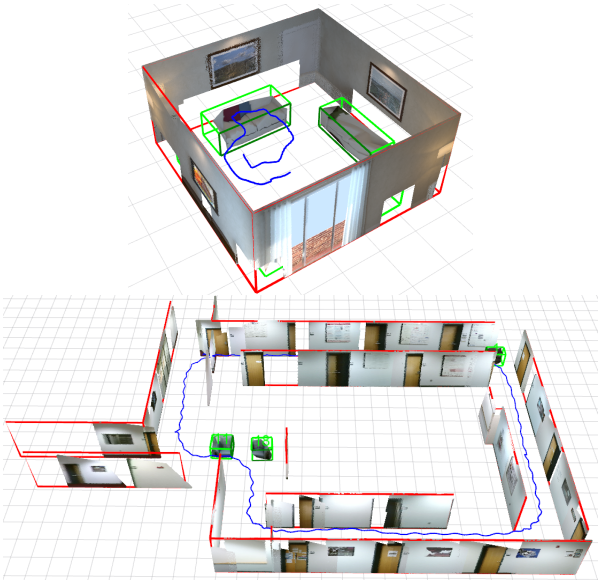


Fig. 1. Example result of dense SLAM map with points, objects (green box), planes (red rectangle) reconstructed using only a monocular camera. (top) ICL living room dataset. (bottom) Collected long corridor dataset.

In this work, we propose a monocular SLAM incorporating objects and planes, without prior object and room models. The approach is divided into two steps. The first step is single image structured 3D understanding. Many layout plane and cuboid object proposals are generated based on semantic image cues then the best subset of them is selected to minimize occlusions and intersections. Then the second step is multi-view SLAM optimization. Planes and objects are further optimized with camera poses and point features in a unified bundle adjustment (BA) framework. They can provide more semantic and geometric constraints compared to points, such as Manhattan world assumption and object supporting relationships, to improve camera pose estimation. The optimized plane and object positions are also beneficial for the final consistent and dense 3D mapping. In summary, our contributions are as follows:

- Propose a high order graphical model with efficient inference for the joint structured reasoning of 3D objects and layout planes.
- Propose the first monocular SLAM method incorporating points, objects and planes, and show improvements on both localization and mapping over state-of-the-art algorithms.

In the following, we first introduce the related work and single image 3D understanding in Sec III, then explain multi-view SLAM optimization in Sec IV, followed by experiments

in Sec V.

II. Related Work

A. Single image understanding

There has been much work on the separate object and layout detection. The classic 3D object detection depends on the hand-crafted features such as edge and texture [14]. Deep network is also used to directly predict the cuboid object poses from single images [15]. For layout plane detection, the popular room model based on vanishing points is proposed by Hedau *et al* [16]. Recent learning based approaches such as [17] and RoomNet [3] can achieve impressive results in Manhattan room environments. These approaches can generate roughly correct plane models, but they are not suitable for SLAM landmark optimization because CNN prediction may be inconsistent across frames. In addition, most of them only apply to the restricted four-wall room models.

Our work is more related to the joint and holistic 3D understanding of object and planes. Their positions are optimized based on the spatial and semantic relationships such as occlusion, intersection, and concurrence [18]. Most of them utilize RGBD camera and are not running in real time. More recent works directly predict the 3D occupancy of objects and planes utilizing CNN [19].

B. Object and Plane SLAM

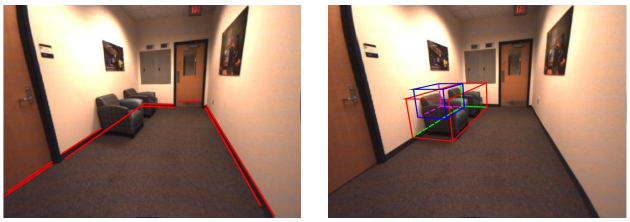
Apart from the commonly used points, there is also research on object and plane based SLAM. One simple way is to first build classic point SLAM then detect objects and planes. It can improve the object detection accuracy due to multi-view and point cloud information [8], but is likely to fail if the point cloud quality is low. We here focus on the SLAM which explicitly uses objects and planes as landmarks. The first well-known system is called Semantic Structure from Motion which jointly optimizes camera poses, objects, points and planes [20]. Several object based SLAM [10] [11] are also proposed but all depend on the prior object models. Without prior models, the recent QuadricSLAM [21] and CubeSLAM [22] propose two different object representation.

There is also some work using plane or superpixel to generate dense map [23] [24]. Lee [12] estimates the layout plane and point cloud registration iteratively to reduce RGB-D mapping drift. Similarly, planes are shown to provide long-range constraints compared to points in indoor building environments [25] [26].

Recently, [27] proposes a similar work to jointly optimize objects, planes, points with camera poses. The difference is that we use monocular camera instead of RGBD camera and also have different object representations.

III. Single Image Understanding

Similar to several single image understanding research [18], we represent the environment as a set of layout planes such as wall, floor, and cuboid objects. The goal is to simultaneously infer their locations from 2D image.



(a) (b)

Fig. 2. Single image generated plane and cuboid proposals. (a) Wall plane proposals are represented as ground edges. (b) Different cuboid proposals for the same object instance.

We first generate a number of object and plane proposals (hypothesis), then select the best subset of them satisfying occlusion constraints via Conditional Random Field (CRF) optimization.

A. Proposal generation

1) Layout Plane Proposal: We project the actual detected ground-wall edges to 3D space to generate plane proposals, which can be directly used as the latter SLAM landmark because edge observation is consistent across frames. Our prior work [26] also adopts this idea and we extend it to work robustly in large environments with objects.

We first detect all image edges then select some edges close to the ground-wall segmentation [2] boundary. For room environments, layout plane prediction score [17] is additionally used to select possible edges. If the edge lies partially inside object regions, we further extend it to intersect with other edges as shown in Fig 2(a), because it may be occluded by foreground objects.

2) Object Cuboid Proposal: We follow CubeSLAM [22] to generate cuboid proposals based on 2D bounding box detection and then score proposals based on image features. For each object instance, we select the best 15 cuboid proposals for latter CRF optimization. More cuboid proposals may improve the final performance but also increase computation a lot. Two of these are shown in Fig 2(b) for illustration.

B. CRF Model definition

Given all the proposals, we want to select the best subset from them. We start by defining a binary random variable $x_i \in \{0, 1\}$ for each plane and cuboid proposal, indicating whether it will be selected. This multi-variable label optimization problem is also called CRF. We want to optimize the labels to minimize the following different energy functions or called potentials:

$$E(\mathbf{x}|\mathbf{I}) = \sum_i \psi_i^U(x_i) + \sum_{i<j} \psi_{ij}^P(x_i, x_j) + \sum_{\mathbf{x_c}<\mathcal{C}} \psi_{\mathbf{c}}^{HO}(\mathbf{x_c}) \tag{1}$$

where ψ_i^U and ψ_{ij}^P are the unary and pairwise potential energy. $\psi_{\mathbf{c}}^{HO}$ is the high order term of clique $\mathbf{x_c}$. These potentials will be explained in more details in the following.

1) Unary potential: It represents the proposal quality itself. For planes, the energy depends on the edge's distance to the ground-wall segmentation contour and layout edge prediction score if in room environments. Long edges are also preferred compared to short ones which are likely to be outliers due to detection errors. Proper weighting and normalization is needed to combine them together. For objects, we can directly use the cuboid fitting error explained in [22], based on the vanishing point and edge alignment.

2) Pairwise Potential: There are different forms of pairwise relationships between objects and planes for example the semantic co-occurrence [18]. Here we only utilize the geometric relationship to minimize the 3D occlusion and intersection. For object-object, ψ_{ij}^{P} is defined as the 3D intersection of union. For object-plane, it represents the truncation ratio of object volume by plane. For plane-plane, ψ_{ij}^{P} indicates the angle overlapping ratio between each other. Note that there is no pairwise potential between cuboid proposals belonging to the same object.

3) High order potential: As explained in Section III-A, for each 2D object instance, many 3D cuboid proposals are generated from it but at most one of them can be selected. Thus high order potential becomes:

$$\varphi^{HO}(\mathbf{x_c}) = \begin{cases} 0 & \text{if } \sum_{x_i \in \mathbf{x_c}} x_i \leq 1 \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

C. Efficient CRF inference

There has been extensive research on high order discrete CRFs [28] but efficient inference is still challenging in many cases. However, our high order term in Eq 2 is very sparse because at most one variable can be 1 in one clique $\mathbf{x_c}$, therefore, we can design efficient inference for it. Max-product loopy belief propagation [29] is adopted. The most computationally expensive part is the message from variable node i to clique c:

$$m_{c\to i}^{t}(x_i) = \min_{\mathbf{x}_c^{-i}} \left(f_c(\mathbf{x}_c) + \sum_{j \in c \setminus \{i\}} m_{j\to c}^{t-1}(x_j) \right) \tag{3}$$

where $\mathbf{x}_c^{-i}$ denotes all the variables in clique c except variable i. $m_{j\to c}^{t-1}(x_j)$ denotes the message from node to clique. For a clique with N binary nodes, there are totally 2^N clique states of $\mathbf{x}_c$. However there are only $N+1$ valid states in our problem $\{1,0,...0\},...\{0,0,...1\},\{0,0,...0\}$ denoted as $\{\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_{N+1}\}$. Therefore, we only need to check $N+1$ states and find the minimum in Eq 3. We can further observe that every adjacent $\mathbf{y}_i$ only has two different variables, therefore $\sum_{j \in c \setminus \{i\}} m_{j\to c}^{t-1}(x_j)$ for each $\mathbf{y}_i$ can be computed iteratively. Therefore, the average time complexity of computing $m_{c\to i}^{t}(x_i)$ is $O(1)$ instead of the naive $O(2^N)$. More details can be found at the appendix.

IV. SLAM Optimization

The selected object and plane proposals are treated as SLAM landmarks and further optimized through multi-view BA. Similar to the common point landmarks, we need to define the new parameterization and different measurement functions between them.

A. Parameterization

There are four different components in the map: camera, point, object and plane. Camera pose T_c and point P follow standard forms $T_c \in SE(3)$ and $P \in \mathbb{R}^3$.

The cuboid pose is similarly defined in [22] by 9 DoF parameters: $O = (T_o, D)$, where $T_o \in SE(3)$ is 6 DoF pose, and $D \in \mathbb{R}^3$ is dimensions. Other shapes can also be used for example ellipsoids [21].

We adopt the infinite plane in [30] which represents plane as a quaternion $\boldsymbol{\pi} = (\mathbf{n}^\top, d)^\top$ st. $\|\boldsymbol{\pi}\| = 1$, suitable for graph optimization. n is the plane normal and d is plane distance to the origin. In some environments, we can use the Manhattan assumptions, namely the plane normal is fixed and parallel to one of the world frame axes. Therefore only one number d is needed to represent it.

B. Measurements

Different constraint functions between the map elements are proposed to formulate a factor graph optimization. Camera-point observation model is standard reprojection error [4].

1) Camera-plane: Different from RGBD based plane SLAM which can directly get plane measurements from point cloud plane fitting [27] [30], we need to pop up the plane to get local plane measurement which depends on the camera pose. [26] updates the measurement after graph optimization which is not an optimal solution. Therefore we update it in each iteration during the optimization. Denote the wall-ground edge as l, then plane error is defined as:

$$e_{cp} = \| \log \left(\boldsymbol{\pi}_{obs}(l, T_c), \boldsymbol{\pi} \right) \| \tag{4}$$

Where $\boldsymbol{\pi}_{obs}$ is the process of projecting ground edge l onto 3D ground shown as blue plane in Fig .3(a). It also depends on camera pose T_c. Then the generated plane is compared with plane landmarks $\boldsymbol{\pi}$ using log quaternion error defined in [30].

2) Camera-object: We follow the cuboid observation functions defined in the prior work [22]. The cuboid landmark is projected onto the image plane to get the 2D bounding box shown as the red rectangles in Fig. 3(b). Then it is compared with the blue detected 2D bounding box:

$$e_{2D} = \| \ (c,d) - (c_m, d_m) \ \|_2 \tag{5}$$

where (c, d) is the center and dimension of the 2D box. This 2D measurement error has much less uncertainty compared to 3D cuboid error in [22]. To make the optimization robust, different weights are assigned to different objects' error. More weight is given to semantic confident and geometric close objects.

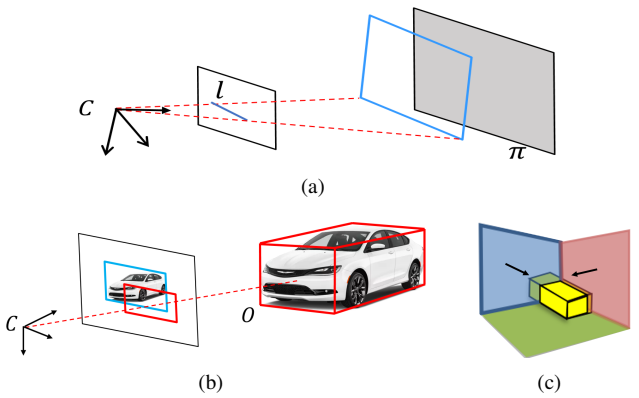


Fig. 3. SLAM observation functions. (a) Camera-plane observations. The detected ground edge is projected to 3D space to compare with landmark plane. (a) Camera-object observations. 3D cuboid landmark is projected onto images and compared with the detected 2D box. (c) Plane-object observation error depends on the object volume occluded by planes.

3) Object-plane: There are different forms of object-plane constraints depending on the environment assumptions for example objects are supported by planes [27] or object orientation matches the nearby plane normal. We here design a weaker but more general constraint that objects should not be occluded by nearby planes in the camera view shown in Fig 3(c). If the plane normal is defined to point to the camera, then the object-plane occlusion error is defined as:

$$e_{op} = \sum_{i=1:8} \max(0, -\boldsymbol{\pi} P_{oi}) \tag{6}$$

Where P_{oi} is one of the eight cuboid corners. If the cuboid lies on the front side of plane (towards camera), $e_{op} = 0$.

4) Point-plane: It is usually difficult to accurately detect if points belong to a plane from 2D images as layout planes are usually background and points may belong to foreground objects. To improve the robustness, we first select points in the 2D wall plane polygon then filter out points that are farther away from the 3D plane than a threshold. The point-plane error is defined as:

$$e_{pp} = \|\boldsymbol{\pi} P\| \tag{7}$$

Note that to be robust to outliers, huber loss is applied to all above error functions.

C. Data association

Data association for different landmarks across multiple views is important for SLAM. For point association, we use the point feature matching in ORB SLAM [4]. Object association follows the work of CubeSLAM [22]. Each object contains a set of feature points belonging to it then we can find object matching which has the most number of shared map points exceeding a threshold (10 in our implementation). This approach is easy to implement and can also easily detect dynamic objects.

Plane association is based on the two following criteria. One is the geometry information such as the plane normal angle difference and plane distance to each other. The other is the shared feature points matching similar to objects. The point plane belonging relation is also used when computing point-plane error in Eq 7.

V. Experiments

A. Implementation details

For object detection, we use similar settings as object SLAM in [22]. For plane proposals, we first detect and merge line segments then remove lines shorter than 50 pixels and more than 50 pixels away from the wall-ground segmentation boundary.

For the SLAM part, our system is built on the feature point based ORB SLAM, augmented with objects and planes. We compute the jacobians of the newly created observation functions then perform BA using g2o library. The explicit image recognition based loop closure in ORB SLAM is disabled to better show the improvements by objects and planes. Since the number of objects and planes is far less compared to point features, the overall BA optimization is still efficient enough to run in real time. Meanwhile, outlier objects or planes usually have more severe effects compared to outlier points, thus strict outlier rejections need to be used. The object and plane landmark will be deleted if it has not been observed 3 times in recent 15 frames after creation or if there are less than 10 stable feature points associated with it, except the white wall surfaces with few 2D features initially. In most of the experiments, we use the Manhattan plane representation with a fixed surface normal in Section IV-A to improve the performance. If the initial generated wall surface normal difference with Manhattan direction exceeds 30 degrees, it will also be treated as outliers.

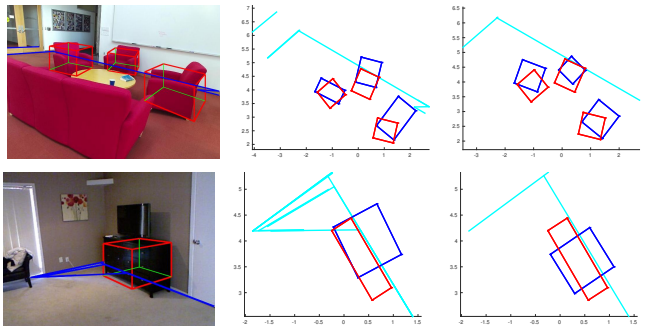
In addition to being used as SLAM landmarks, objects and planes can also provide depth initialization for feature points. When the inlier feature point ratio (features matched to the map divided by total features) is below 0.3, we create some new map points directly using depth from objects and planes. This can improve monocular SLAM performance in low texture environments and large rotation scenarios.

Different from the prior monocular plane SLAM [26], ground plane is not used in this work because there is no actual edge measurement corresponding to the ground plane.

Objects and planes can also benefit the final dense mapping. We directly back-project plane region pixels, excluding object areas, onto the optimized plane landmark. For feature points belonging to objects, we create triangular meshes in 3D space to get dense mapping. Note that in the SLAM optimization, planes are represented as infinite planes but for visualization purposes, we need to keep track of the plane boundary polygon.

B. Single Image Result

We first show the single image layout plane and cuboid object detection result. Some examples of proposal generation and CRF optimization are shown in Fig 4. The middle and right columns show the top view of object proposals before and after CRF optimization. We can easily see from it that



(a) (b) (c)

Fig. 4. Single image raw proposal generation and CRF optimization illustrations. (a) Raw plane and object proposals. (only draw one cuboid for brevity) (b) Top view of raw proposals. Red rectangle is ground truth object and blue is estimated. Cyan line is wall plane edges. (c) Top view of CRF selected proposals. Object pose is more accurate after optimization. Plane and object occlusion is also minimized.

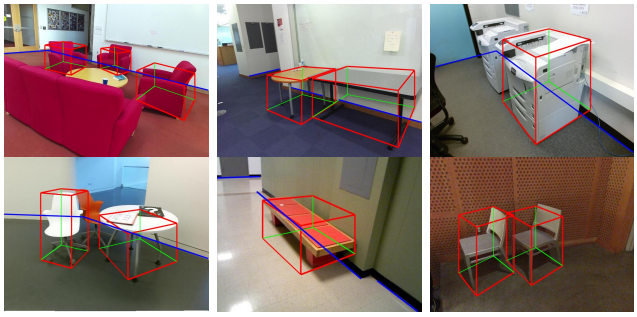


Fig. 5. More single image CRF optimized object and plane proposals.

CRF can select non-overlapped wall edges and better cuboid proposals to minimize occlusions and object intersection.

More results of CRF selected object and plane proposals are shown in Fig 5. The algorithm is able to work in different environments from rooms to corridors but it may still fail to detect all the wall planes and objects when there is severe object occlusion and unclear edges for example in the right column of Fig 5.

We also evaluate quantitatively the CRF optimization performance on SUN RGBD dataset. Compared to [22] which selects the best cuboid proposal without considering planes, our CRF joint reasoning of object and plane improves the 3D object intersection over union (IoU) by 5% shown in Table I. Note that to emphasize the optimization effect, we only evaluate on images where CRF generates different results, no matter good or bad, compared to the single image detection [22]. This is because many images have no visible ground edges or they are far from objects and have no actual constraints on object positions thus CRF optimization will have no effects on those object detections.

C. SLAM Result

We then evaluate the SLAM tracking and mapping performance on both public datasets including ICL-NUIM [31], TAMU Indoor [32], TUM mono [33], and our collected datasets by KinectV2 sensor.

TABLE I

3D OBJECT IOU ON SUN RGBD SUBSET DATA

Method	Before [22]	After CRF
3D IoU	0.35	0.40

(a) (b) (c)

Fig. 6. (a) Layout prediction score map [17] (b) Semantic segmentation by [2] (c) Our CRF optimized cuboid object and wall planes. It cannot detect the occluded wall surface while multi-view SLAM can build a complete map in Fig. 1.

1) Qualitative Results: A sample frame of ICL sequence is shown in Fig 6. The left and middle images show the raw image overlayed by layout prediction and the semantic segmentation. Both of them have noise and CRF optimization in Fig .6(c) shows a roughly correct 3D model but it cannot fully detect the occluded wall segments. After the multi-view SLAM optimization, the algorithm is able to build a more consistent and complete map shown in Fig. 1.

More 3D mapping and camera pose estimation in different datasets and environment configurations are shown in Fig 7. After BA, objects and planes' locations are more accurate compared to the single view detection and most objects lie inside the room. Note that not all objects are mapped because the 2D object detector might miss some and SLAM might also treat some of them as outliers due to inconsistent observations. In some scenarios such as the top left, our algorithm cannot detect the full wall plane due to severe object occlusions. To improve the visualization robustness, if there is not enough map point observed in some region of a plane polygon, no dense pixels will be projected, shown as the void segments on the wall surface in the middle image.

2) Quantitative Results: We then show the quantitative camera pose comparison with ORB SLAM and DSO. For datasets in Table II, the initial map of both ORB and ours is scaled by the truth initial camera height. Then we can directly evaluate the absolute translation error without aligning the pose in scale, to show that object and planes can improve the pose estimation and reduce monocular drift. Each algorithm runs in each sequence for 5 times and the mean error is reported here. From the table, we can see that in most of the scenarios, the added objects and planes landmark constraints improve the camera pose estimation. There are two main reasons for this. One is that even though we disable explicit loop closure, due to object and plane's long-range visibility properties, the algorithm may still associate with the old plane landmark to reduce the final drift. The second reason is the depth initialization of features especially in large camera rotations. Due to the strict outlier rejection and robust BA

TABLE II

ABSOLUTE CAMERA TRANSLATION ERROR ON VARIOUS DATASETS

Method	ORB SLAM [4]	Ours
ICL living 0	3.08	**0.8**
ICL living 2	3.25	**2.06**
ICL living 3	**5.36**	5.38
ICL office 0	6.23	**5.93**
ICL office 2	5.00	**2.63**
Tamu corridor	3.87	**0.97**
Our room 1	0.15	**0.05**
Our corridor 1	2.25	**0.30**
Our corridor 2	2.93	**0.24**
Our corridor 3	1.84	**0.49**

TABLE III

POSE ALIGNMENT ERROR ON TUM-MONO DATASET

Method	ORB [4]	DSO [5]	Ours
Corridor 36	1.81	4.01	**0.94**
Room 37	0.60	0.55	**0.35**
Corridor 38	23.9	**0.55**	7.65

optimization, even if it doesn't improve the result, it won't seriously damage the system.

For TUM mono data in Table III, no truth camera height is available thus we evaluate the monocular scale alignment error [5]. Results of DSO and ORB are taken from the supplementary material of DSO. Our semantic SLAM can work robustly in these challenging datasets even though there is large camera rotation and sometimes the camera may be upside down. In the cluttered dataset such as Room 37, there are only a few planes with a few observed frames thus our algorithm almost reduces to point SLAM and achieves similar results. In Corridor 38, our algorithm and ORB SLAM are much worse compared to DSO because there are many areas of white walls with few feature points which are difficult for feature based SLAM.

VI. CONCLUSION

In this work, we propose the first monocular SLAM and dense mapping algorithm combining points with high level object and plane landmarks through unified BA optimization. We show that semantic scene understanding and traditional SLAM optimization can improve each other.

For the single image, we propose a fast 3D object and layout joint understanding for general indoor environments. Cuboid and plane proposals are initially generated from 2D object and edge detection. Then a fast sparse high order CRF inference is proposed to select the best proposals. In the SLAM part, several new measurement functions are designed for planes and objects. Compared to points, objects and planes can provide long-range geometric and semantic constraints such as intersection and supporting relationships to improve the pose estimation. Strict outlier rejection and robust optimization are proposed to improve the robustness.

We evaluate the SLAM algorithm in various public indoor datasets including rooms and corridors. Our approach can improve the camera pose estimation and dense mapping in most environments compared to the state-of-the-art.

In the future, more general planes in addition to wall planes need to be considered to produce a denser and more complete map. Dynamic objects and object surface mapping can also be addressed to improve the robustness and visualization.

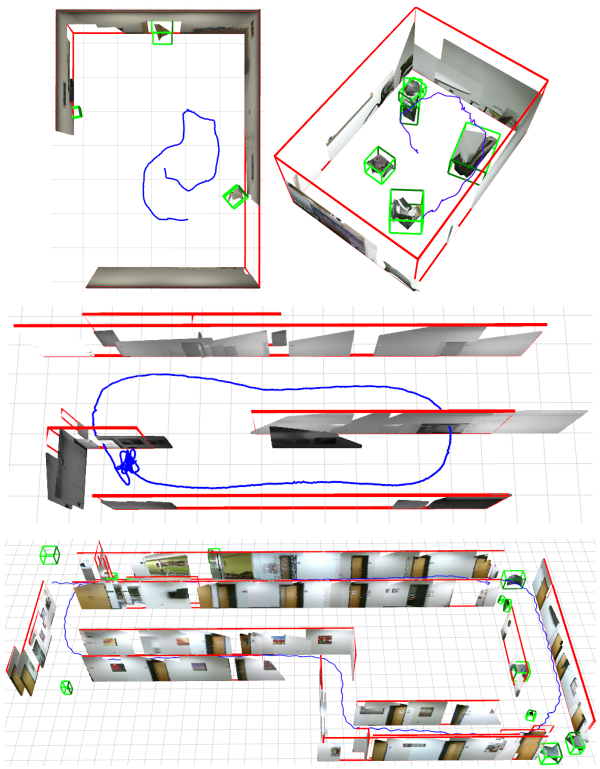


Fig. 7. More dense mapping results with objects and planes. (top) ICL-NUIM office 2, our room. (middle) TUM-mono 36. (bottom) our collected long corridor.

APPENDIX

We here explain the CRF inference of Section III-C in more detail. If there are N variable $x_1, x_2, ..., x_n$ in a clique $\mathbf{x}_c$. As mentioned before, there are $N+1$ special states. For each state $\mathbf{y}_k$, we define $\mathbf{s}_k = \sum_{j \in \mathbf{y}_k} m_{j \to c}^{t-1}(x_j)$. Note that all $\mathbf{s}_k$ can be computed iteratively in $O(N)$ as adjacent $\mathbf{y}_k$ is almost the same. The min and second min of $\mathbf{s}_k$ is be recorded. Then we can compute clique to variable i message by:

$$m_{c \to i}^{t}(x_i) = \begin{cases} \mathbf{s}_i - m_{i \to c}^{t-1}(x_i) & \text{if } x_i = 1 \\ \min\limits_{k=1:N+1, k \neq i} \mathbf{s}_k - m_{i \to c}^{t-1}(x_i) & \text{if } x_i = 0 \end{cases} \quad (8)$$

When $x_i = 1$, only one state $\mathbf{y}_i$ is feasible. Otherwise, we need to evaluate all $N+1$ states to find the minimum. As we already record the min and second min, evaluating Eq 8 only takes O(1) computation.

REFERENCES

[1] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[3] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *IEEE International Conference on Computer Vision*, 2017.

[4] Raul Mur-Artal, JMM Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[5] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[6] Abhijit Kundu, Yin Li, Frank Dellaert, Fuxin Li, and James M Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *European Conference on Computer Vision (ECCV)*, pages 703–718. Springer, 2014.

[7] Shichao Yang, Yulan Huang, and Sebastian Scherer. Semantic 3d occupancy mapping through efficient high order crfs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.

[8] Sudeep Pillai and John Leonard. Monocular slam supported object recognition. *Robotics: Science and systems*, 2015.

[9] Sid Yingze Bao, Axel Furlan, Li Fei-Fei, and Silvio Savarese. Understanding the 3d layout of a cluttered room from multiple images. In *IEEE Winter Conference on Applications of Computer Vision*, pages 690–697. IEEE, 2014.

[10] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[11] Dorian Gálvez-López, Marta Salas, Juan D Tardós, and JMM Montiel. Real-time monocular object slam. *Robotics and Autonomous Systems*, 75:435–449, 2016.

[12] Jeong-Kyun Lee, Jaewon Yea, Min-Gyu Park, and Kuk-Jin Yoon. Joint layout estimation and global multi-view registration for indoor reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[13] Jianxiong Xiao and Yasutaka Furukawa. Reconstructing the worlds museums. *International journal of computer vision*, 110(3):243–258, 2014.

[14] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *IEEE International Conference on Computer Vision*, pages 2992–2999, 2013.

[15] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *International Conference on Computer Vision*, pages 1849–1856. IEEE, 2009.

[17] Yuzhuo Ren, Shangwen Li, Chen Chen, and C-C Jay Kuo. A coarse-to-fine indoor layout estimation (cfile) method. In *Asian Conference on Computer Vision*, pages 36–51. Springer, 2016.

[18] Dahua Lin, Sanja Fidler, and Raquel Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *IEEE International Conference on Computer Vision*, pages 1417–1424, 2013.

[19] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[20] Sid Yingze Bao, Mohit Bagra, Yu-Wei Chao, and Silvio Savarese. Semantic structure from motion with points, regions, and objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2703–2710. IEEE, 2012.

[21] Lachlan James Nicholson, Michael J Milford, and Niko Sunderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 2018.

[22] Shichao Yang and Sebastian Scherer. CubeSLAM: Monocular 3d object detection and slam without prior models. *arXiv preprint arXiv:1806.00557*, 2018.

[23] Alejo Concha and Javier Civera. DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 5686–5693. IEEE, 2015.

[24] Lingni Ma, Christian Kerl, Jörg Stückler, and Daniel Cremers. Cpa-slam: Consistent plane-model alignment for direct rgb-d slam. 2016.

[25] Ming Hsiao, Eric Westman, Guofeng Zhang, and Michael Kaess. Keyframe-based dense planar slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5110–5117. IEEE, 2017.

[26] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer. Pop-up SLAM: a semantic monocular plane slam for low-texture environments. In *International conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[27] Mehdi Hosseinzadeh, Yasir Latif, Trung Pham, Niko Suenderhauf, and Ian Reid. Towards semantic slam: Points, planes and objects. *arXiv preprint arXiv:1804.09111*, 2018.

[28] Chaohui Wang, Nikos Komodakis, and Nikos Paragios. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 117(11):1610–1627, 2013.

[29] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[30] Michael Kaess. Simultaneous localization and mapping with infinite planes. In *International Conference on Robotics and Automation (ICRA)*, pages 4605–4611. IEEE, 2015.

[31] A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[32] Yan Lu and Dezhen Song. Robustness to lighting variations: An rgb-d indoor visual odometry using line segments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 688–694. IEEE, 2015.

[33] J. Engel, V. Usenko, and D. Cremers. A photometrically calibrated benchmark for monocular visual odometry. In *arXiv:1607.02555*, July 2016.